Информатика и её применения

Том 6 Выпуск 1 Год 2012

СОДЕРЖАНИЕ

Информатика и её применения

Том 6 Выпуск 1 Год 2012

СОДЕРЖАНИЕ

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 3–11

СКОШЕННЫЕ РАСПРЕДЕЛЕНИЯ СТЬЮДЕНТА, ДИСПЕРСИОННЫЕ ГАММА-РАСПРЕДЕЛЕНИЯ И ИХ ОБОБЩЕНИЯ КАК АСИМПТОТИЧЕСКИЕ АППРОКСИМАЦИИ*

В. Ю. Королев[1], И. А. Соколов[2]

Аннотация: Показано, что скошенные распределения Стьюдента и (несимметричные) дисперсионные гамма-распределения могут выступать в качестве предельных в довольно простых предельных теоремах для регулярных статистик, в частности в схеме случайного суммирования случайных величин, и, следовательно, могут считаться асимптотическими аппроксимациями для распределений многих процессов, связанных с эволюцией сложных систем.

Ключевые слова: скошенное распределение Стьюдента; дисперсионное гамма-распределение; предельная теорема; случайная сумма; теорема переноса

1 Введение

Скошенные распределения Стьюдента и дисперсионные гамма-распределения часто служат математическими моделями статистических закономерностей, хорошо описывающими эффект наличия так называемых тяжелых или полутяжелых хвостов. Такие модели очень важны для адекватного описания статистических закономерностей поведения различных характеристик сложных систем, эволюция которых в значительной мере зависит от информационных потоков, к примеру телекоммуникационных сетей или финансовых рынков. В частности, в финансовой математике хорошо известны так называемые *обобщенные гиперболические процессы* (GH-processes) и *дисперсионные гамма-процессы* (VG-processes).

Обобщенные гиперболические процессы — это процессы Леви (процессы с независимыми стационарными приращениями), одномерные распределения которых имеют обобщенные гиперболические распределения.

Класс обобщенных гиперболических распределений был описан О. Барндорфф-Нильсеном [1]. Плотность обобщенного гиперболического распределения имеет вид:

$$p_{\mathrm{GH}}(x;\lambda,\alpha,\beta,\delta,\mu) = \\ = \frac{\left(\alpha^2-\beta^2\right)^{\lambda/2}}{\sqrt{2\pi}\alpha^{\lambda-1/2}\delta^{\lambda}K_{\lambda}(\delta\sqrt{\alpha^2-\beta^2})}\left(\delta^2+(x-\mu)^2\right)^{(\lambda-1/2)/2} K_{\lambda-1/2}\left(\alpha\sqrt{\delta^2+(x-\mu)^2}\right)\times \\ \times \exp\left(\beta(x-\mu)\right)\,,\ \ x\in\mathbb{R}\,,$$

где $\mu\in\mathbb{R}$;

$$\delta\geq 0\,,\ |\beta|<\alpha\,,\ \text{если}\ \lambda>0\,;$$
$$\delta>0\,,\ |\beta|<\alpha\,,\ \text{если}\ \lambda=0\,;$$
$$\delta>0\,,\ |\beta|\leq\alpha\,,\ \text{если}\ \lambda<0\,;$$

$K_{\lambda}(x)$ — функция Бесселя третьего рода порядка λ. Обобщенное гиперболическое распределение имеет «полутяжелые» хвосты в том смысле, что его плотность удовлетворяет асимптотическому соотношению

$$p_{\mathrm{GH}}(x;\lambda,\alpha,\beta,\delta,\mu)\sim|x|^{\lambda-1}\exp\{(\pm\alpha+\beta)x\}$$

с точностью до постоянного множителя при $x\to\pm\infty$.

В работах [2–7] установлено хорошее согласие обобщенного гиперболического распределения, *гиперболического* ($\lambda=1$) и *нормального обратного гауссовского* ($\lambda=-1/2$) распределений, входящих в семейство обобщенных гиперболических распределений вероятностей, с данными о ценах на датских и немецких биржах, финансовыми индексами NYSE и DAX, обменными курсами валют и т. д.

*Работа выполнена при поддержке РФФИ (проекты 11-01-12026-офи-м, 11-07-00112 и 11-01-00515), а также Министерства образования и науки РФ в рамках ФЦП «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы».

[1]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики; Институт проблем информатики Российской академии наук, vkorolev@comtv.ru

[2]Институт проблем информатики Российской академии наук, isokolov@ipiran.ru

С точки зрения поведения «хвостов» эти распределения занимают как бы промежуточное положение между устойчивыми распределениями с индексом $\alpha < 2$ и нормальными (гауссовскими) распределениями $\alpha = 2$: их «хвосты» убывают быстрее, чем у устойчивых распределений ($\alpha < 2$), но медленнее нормальных.

Класс обобщенных гиперболических распределений весьма широк (см., например, [8]). В частности, заметим, что для $\nu > 0$ при $\lambda = -\nu/2$, $\alpha = \beta = \mu = 0$, $\delta = \sqrt{\nu}$ обобщенное гиперболическое распределение совпадает с классическим распределением Стьюдента с ν степенями свободы.

Известно несколько попыток описать несимметричные обобщения распределения Стьюдента. Пожалуй, наиболее успешная из них — это несимметричное (скошенное, skew) распределение Стьюдента, описанное в работе [9] как некий частный случай обобщенного гиперболического распределения.

В статье [10] распределению, предложенному в [9], дано другое более удобно интерпретируемое определение как специальной сдвиг-масштабной смеси нормальных законов. Согласно [10], скошенным распределением Стьюдента называется распределение с плотностью

$$p_{\mathrm{SS}}(x; a, \sigma, \mu, \lambda) = \\ = \frac{1}{\sqrt{2\pi}\sigma} \int\limits_0^\infty \exp\left\{-\frac{1}{2}\left(\frac{x-au}{\sigma\sqrt{u}}\right)^2\right\} \frac{h(u;\mu,\lambda)}{\sqrt{u}}\, du. \quad (1)$$

Здесь $a \in \mathbb{R}$; $\sigma > 0$; $\mu > 0$; $\lambda > 0$; $h(x;\mu,\lambda)$ — плотность обратного гамма-распределения, т. е. распределения случайной величины U^{-1}, где U — случайная величина с гамма-распределением с параметром формы μ и параметром масштаба λ:

$$h(x;\mu,\lambda) = \frac{\lambda^\mu}{\Gamma(\mu)}\, x^{-\mu-1} \exp\left\{-\frac{\lambda}{x}\right\}. \quad (2)$$

Напомним, что плотность распределения само́й случайной величины U имеет вид:

$$g(x;\mu,\lambda) = \frac{\lambda^\mu}{\Gamma(\mu)} x^{\mu-1} e^{-\lambda x}, \;\; x \geq 0\,. \quad (3)$$

Здесь и далее $\Gamma(\,\cdot\,)$ — эйлерова гамма-функция:

$$\Gamma(z) = \int\limits_0^\infty e^{-y} y^{z-1}\, dy\,, \;\; z > 0\,.$$

Дисперсионные гамма-процессы, предложенные в работах [11, 12], — это процессы Леви, одномерные распределения которых являются дисперсионными гамма-распределениями. Плотность дисперсионного гамма-распределения имеет вид:

$$p_{\mathrm{VG}}(x; a, \sigma, \mu, \lambda) = \\ = \frac{1}{\sqrt{2\pi}\sigma} \int\limits_0^\infty \exp\left\{-\frac{1}{2}\left(\frac{x-au}{\sigma\sqrt{u}}\right)^2\right\} \frac{g(u;\mu,\lambda)}{\sqrt{u}}\, du, \quad (4)$$

где $a \in \mathbb{R}$, $\sigma > 0$, $\mu > 0$, $\lambda > 0$, а $g(x;\mu,\lambda)$ — плотность гамма-распределения с параметрами μ и λ (см. (3)). Как отмечено в упомянутых работах, подобные модели также демонстрируют высокую адекватность при описании динамики цен финансовых активов.

Вместе с тем в прикладной теории вероятностей хорошо известен принцип, согласно которому та или иная модель может считаться в достаточной мере обоснованной только тогда, когда она является *асимптотической аппроксимацией*, т. е. когда существует довольно простая предельная теорема, в которой рассматриваемая модель выступает в качестве предельного распределения [13]. В книге [14] прослежена глубокая связь этого принципа с универсальным принципом неубывания энтропии в замкнутых системах. Обе рассматриваемые в данной статье модели имеют вид сдвиг-масштабных смесей нормальных законов. Как известно, нормальное распределение обладает максимальной энтропией среди всех распределений, носителем которых является вся числовая прямая и имеющих конечный второй момент. Если бы моделируемая сложная система была информационно изолирована от окружающей среды, то в соответствии с принципом неубывания энтропии, который в теории вероятностей проявляется в виде предельных теорем [14], наблюдаемые статистические распределения ее характеристик были бы неотличимы от нормального. Но поскольку любая математическая модель по своему определению не может учесть все факторы, влияющие на состояние или эволюцию моделируемой системы, то параметры этого нормального закона изменяются в зависимости от состояния среды, внешней по отношению к моделируемой системе. Другими словами, эти параметры являются случайными и изменяются под влиянием информационных потоков между системой и внешней средой. Таким образом, во многих ситуациях разумные модели статистических закономерностей изменения параметров сложных систем должны иметь вид сдвиг-масштабных смесей нормальных законов, частными случаями которых являются (1) и (4).

К сожалению, в первоисточниках упомянутые выше модели вводились чисто умозрительно как распределения процесса броуновского движения со

случайным временем, в каждый момент имеющим гамма-распределение (дисперсионное гамма-распределение) или обратное гамма-распределение (скошенное распределение Стьюдента). «Асимптотического» обоснования этих моделей пока дано не было.

Частному случаю дисперсионных гамма-распределений — несимметричному распределению Лапласа — и его практическому применению посвящена работа [15] (см. также [8, 16].

В данной работе будет показано, что скошенные распределения Стьюдента и дисперсионные гамма-распределения могут выступать в качестве предельных в довольно простых предельных теоремах для регулярных статистик, в частности в схеме случайного суммирования случайных величин, и, следовательно, могут считаться асимптотическими аппроксимациями для распределений многих процессов, например, сходных с неоднородными случайными блужданиями.

2 Симметричный случай

Сначала убедимся в том, что симметричные дисперсионные гамма-распределения и распределения Стьюдента можно считать асимптотическими аппроксимациями в довольно простых предельных схемах. Символ $\Longrightarrow$ будет обозначать сходимость по распределению. Рассмотрим традиционную для математической статистики постановку задачи. Пусть случайные величины $N_1, N_2, \ldots, X_1, X_2, \ldots$ определены на одном и том же измеримом пространстве $(\Omega, \mathcal{A})$. Пусть на $\mathcal{A}$ задана вероятностная мера P. Предположим, что при каждом $n \geq 1$ случайная величина N_n принимает только натуральные значения и независима от последовательности $X_1, X_2, \ldots$ Пусть $T_n = T_n(X_1, \ldots, X_n)$ — некоторая статистика, т. е. измеримая функция от случайных величин $X_1, \ldots, X_n$. Для каждого $n \geq 1$ определим случайную величину T_{N_n}, положив $T_{N_n}(\omega) = T_{N_n(\omega)}\big(X_1(\omega), \ldots, X_{N_n(\omega)}(\omega)\big)$ для каждого элементарного исхода $\omega \in \Omega$.

Стандартную нормальную функцию распределения будем обозначать $\Phi(x)$:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{x} e^{-z^2/2}\, dz\,, \quad x \in \mathbb{R}\,.$$

Будем говорить, что статистика T_n асимптотически нормальна, если существуют $\delta > 0$ и $t \in \mathbb{R}$ такие, что

$$\mathsf{P}\left(\delta\sqrt{n}\,(T_n - t) < x\right) \Longrightarrow \Phi(x) \ \ (n \to \infty)\,. \qquad (5)$$

Примеры асимптотически нормальных статистик хорошо известны. Свойством асимптотической нормальности обладают, например, выборочное среднее (при условии существования дисперсий), центральные порядковые статистики или оценки максимального правдоподобия (при достаточно общих условиях регулярности) и многие другие статистики.

Наши дальнейшие рассуждения будут основаны на следующей лемме.

Лемма 1. *Пусть $\{d_n\}_{n\geq 1}$ — некоторая неограниченно возрастающая последовательность положительных чисел. Предположим, что $N_n \to \infty$ по вероятности. Пусть статистика T_n асимптотически нормальна в смысле* (5). *Для того чтобы существовала такая функция распределения $F(x)$, что*

$$\mathsf{P}\left(\delta\sqrt{d_n}\big(T_{N_n} - t\big) < x\right) \Longrightarrow F(x) \ \ (n \to \infty),$$

необходимо и достаточно, чтобы существовала функция распределения $H(x)$, удовлетворяющая условиям:

$$H(0) = 0\,, \ \ F(x) = \int\limits_0^\infty \Phi\big(x\sqrt{y}\big)\, dH(y)\,, \ \ x \in \mathbb{R}\,;$$

$$\mathsf{P}(N_n < d_n x) \Longrightarrow H(x) \ \ (n \to \infty).$$

Доказательство. Данная лемма по сути является частным случаем теоремы 3 из [17], доказательство которой, в свою очередь, основано на общих теоремах о сходимости суперпозиций независимых случайных последовательностей [18, 19].

Пусть $F_{\mathrm{VG}}(x; a, \sigma, \mu, \lambda)$ — функция распределения, соответствующая плотности $p_{\mathrm{VG}}(x; a, \sigma, \mu, \lambda)$ (см. (4)). Непосредственным следствием леммы 1 является следующее утверждение.

Теорема 1. *Пусть $\{d_n\}_{n\geq 1}$ — некоторая неограниченно возрастающая последовательность положительных чисел. Предположим, что $N_n \to \infty$ по вероятности. Пусть статистика T_n асимптотически нормальна в смысле* (5). *Для того чтобы*

$$\mathsf{P}\left(\delta\sqrt{d_n}\,(T_{N_n} - t) < x\right) \Longrightarrow F_{\mathrm{VG}}(x; 0, \delta, \mu, \lambda)$$

при $n \to \infty$, необходимо и достаточно, чтобы

$$\mathsf{P}(N_n < d_n x) \Longrightarrow H(x; \mu, \lambda) \quad (n \to \infty)\,, \qquad (6)$$

где $H(x; \mu, \lambda)$ — функция обратного гамма-распределения, соответствующая плотности (2).

Пусть $\gamma > 0$, $P(x; \gamma)$ — функция распределения Стьюдента, соответствующая плотности:

$$p(x; \gamma) = \frac{\Gamma\left((\gamma+1)/2\right)}{\sqrt{\pi\gamma}\,\Gamma(\gamma/2)} \left(1 + \frac{x^2}{\gamma}\right)^{-(\gamma+1)/2}, \quad -\infty < x < \infty\,. \qquad (7)$$

Здесь $\gamma > 0$ — параметр, часто называемый *числом степеней свободы*. В частности, при $\gamma = 1$ плотность (7) имеет вид:

$$p(x;1) = \frac{1}{\pi(1+x^2)}\,, \quad -\infty < x < \infty\,,$$

что соответствует распределению Коши. Несложно убедиться, что

$$p(x;\gamma) = p_{\mathrm{SS}}\left(x; 0, 1, \frac{\gamma}{2}, \frac{\gamma}{2}\right).$$

У распределения Стьюдента с параметром γ отсутствуют моменты порядка $\delta \geq \gamma$. Пусть $G(x;\mu,\lambda)$ — функция гамма-распределения, соответствующая плотности $g(x;\mu,\lambda)$ (см. (3)).

С помощью леммы 1 в работе [20] доказано следующее утверждение (см. также [8, 16]).

Теорема 2. *Пусть $\gamma > 0$ произвольно и $\{d_n\}_{n\geq 1}$ — некоторая неограниченно возрастающая последовательность положительных чисел. Предположим, что $N_n \to \infty$ по вероятности. Пусть статистика T_n асимптотически нормальна в смысле* (5). *Для того чтобы при $n \to \infty$*

$$\mathsf{P}\left(\delta\sqrt{d_n}\,(T_{N_n} - t) < x\right) \Longrightarrow P(x;\gamma)\,,$$

необходимо и достаточно, чтобы

$$\mathsf{P}(N_n < d_n x) \Longrightarrow G\left(x; \frac{\gamma}{2}, \frac{\gamma}{2}\right) \quad (n \to \infty)\,. \qquad (8)$$

Также можно убедиться, что если условие асимптотической нормальности (5), в котором используется «растягивающая» нормировка, заменить аналогичным условием со «сжимающей» нормировкой, характерной для предельных теорем для сумм случайных величин, то в соответствующих предельных теоремах условия (6) и (8) поменяются местами. Пусть в дополнение к приведенным выше условиям случайные величины $X_1, X_2, \ldots$ одинаково распределены, причем $\mathsf{E}X_1 = 0$, $0 < \mathsf{D}X_1 = \sigma^2 < \infty$. Для $n \geq 1$ обозначим $S_n = X_1 + \cdots + X_n$. Два следующих утверждения являются частными случаями общей теоремы о критериях сходимости случайных сумм [18].

Теорема 3. *Предположим, что $N_n \to \infty$ по вероятности. Для того чтобы*

$$\mathsf{P}\left(\frac{S_{N_n}}{\sigma\sqrt{n}} < x\right) \Longrightarrow F_{\mathrm{VG}}(x;0,\delta,\mu,\lambda)$$

при $n \to \infty$, необходимо и достаточно, чтобы

$$\mathsf{P}(N_n < nx) \Longrightarrow G(x;\mu,\lambda) \quad (n \to \infty)\,,$$

где $G(x;\mu,\lambda)$ — функция гамма-распределения, соответствующая плотности (3).

Теорема 4. *Предположим, что $N_n \to \infty$ по вероятности. Для того чтобы*

$$\mathsf{P}\left(\frac{S_{N_n}}{\sigma\sqrt{n}} < x\right) \Longrightarrow P(x;\gamma)$$

при $n \to \infty$, необходимо и достаточно, чтобы

$$\mathsf{P}(N_n < nx) \Longrightarrow H\left(x; \frac{\gamma}{2}, \frac{\gamma}{2}\right) \quad (n \to \infty)\,, \qquad (9)$$

где $H(x;\mu,\lambda)$ — функция обратного гамма-распределения, соответствующая плотности (2).

Приведем пример ситуации, в которой выполнены условия (6) с $\mu = 1$ и (9) с $\gamma = 2$. В таком случае обратное гамма-распределение становится обратным показательным распределением — распределением случайной величины

$$V = \frac{1}{U}\,,$$

где случайная величина U имеет стандартную показательную функцию распределения $E(x) = 1 - e^{-x}$, $x \geq 0$. При этом

$$Q(x) \equiv \mathsf{P}(V < x) = \mathsf{P}\left(\frac{1}{U} < x\right) = \\ = \mathsf{P}\left(U > \frac{1}{x}\right) = e^{-1/x}\,, \quad x \geq 0\,.$$

Обратное показательное распределение $Q(x)$ является частным случаем распределения Фреше, хорошо известного в асимптотической теории экстремальных порядковых статистик как предельное распределение II типа (см., например, [21]).

Приведем пример ситуации, в которой случайный объем выборки имеет предельное распределение вида $Q(x)$. Пусть $Y_1, Y_2, \ldots$ — независимые одинаково распределенные случайные величины с одной и той же непрерывной функцией распределения. Пусть m — произвольное натуральное число. Обозначим

$$N(m) = \min\left\{n \geq 1 : \max_{1\leq j\leq m} Y_j < \max_{m+1\leq k\leq m+n} Y_k\right\}.$$

Случайная величина $N(m)$ имеет смысл числа дополнительных наблюдений, которые надо произвести, чтобы текущий (по m наблюдениям) максимум был перекрыт. Распределение случайной величины $N(m)$ было найдено С. Уилксом, который в работе [22] показал, что распределение величины $N(m)$ является дискретным распределением Парето

$$\mathsf{P}\,(N(m) \geq n) = \frac{m}{m+n}\,, \quad n \geq 1 \qquad (10)$$

(см. также [23, с. 85]).

Пусть теперь $N^{(1)}(m), N^{(2)}(m), \ldots$ — независимые случайные величины с одним и тем же распределением (10). Целая часть числа a будет обозначаться $[a]$. Так как при любом фиксированном $x > 0$

$$1 - \frac{m}{kx(1+(m-1)/(kx))} \le 1 - \frac{m}{m-1+kx} \le \le 1 - \frac{m}{m+[kx]} \le 1 - \frac{m}{m+kx} \le \le 1 - \frac{m}{kx(1+m/(kx))},$$

то для любого $x > 0$

$$\lim_{k\to\infty} \mathsf{P}\left(\frac{1}{k}\max_{1\le j\le k} N^{(j)}(m) < x\right) = = \lim_{k\to\infty} \mathsf{P}\left(\max_{1\le j\le k} N^{(j)}(m) < kx\right) = = \lim_{k\to\infty}\left(1 - \frac{m}{m+[kx]}\right)^k = \lim_{k\to\infty}\left(1-\frac{m}{kx}\right)^k = = e^{-m/x}.$$

Поэтому, если положить

$$N_n = \max_{1\le j\le n} N^{(j)}(m)$$

при $m = 1$, то теоремы 1 и 4 с $d_n = n$ дают иллюстрацию того, как вместо ожидаемого в соответствии с утверждениями классической асимптотической статистики нормального распределения при замене объема выборки случайной величиной в качестве предельного распределения регулярных статистик возникают соответственно распределение Лапласа или распределение Стьюдента. При этом изменение значения параметра m влечет изменение параметра масштаба (дисперсии) итогового распределения Лапласа в теореме 1.

3 Несимметричный случай

В этом разделе будут приведены просто формулируемые предельные теоремы для сумм со случайным числом слагаемых, в которых в качестве предельных возникают несимметричные дисперсионные гамма-распределения или скошенные распределения Стьюдента.

Пусть $\{X_{n,j}\}_{j\ge1}$, $n = 1, 2, \ldots,$ — последовательность серий независимых и одинаково в каждой серии распределенных случайных величин, а N_n, $n = 1, 2, \ldots,$ — положительные целочисленные случайные величины такие, что при каждом n случайная величина N_n независима от последовательности $\{X_{n,j}\}_{j\ge1}$. Для натуральных k обозначим

$$S_{n,k} = X_{n,1} + \cdots + X_{n,k}.$$

Все теоремы, формулируемые ниже, по сути являются частными случаями следующего утверждения, известного как *теорема переноса* [24].

Лемма 2. *Предположим, что существуют неограниченно возрастающая последовательность натуральных чисел $\{m_n\}_{n\ge1}$ и функции распределения $H(x)$ и $A(x)$ такие, что*

$$\mathsf{P}(S_{n,m_n} < x) \Longrightarrow H(x) \qquad (n\to\infty);$$
$$\mathsf{P}(N_n < m_n x) \Longrightarrow A(x) \qquad (n\to\infty).$$

Тогда существует функция распределения $F(x)$ такая, что

$$\mathsf{P}(S_{n,N_n} < x) \Longrightarrow F(x) \quad (n\to\infty).$$

При этом функция распределения $F(x)$ соответствует характеристической функции

$$f(t) = \int_0^\infty h^u(t)\, dA(u), \quad t\in\mathbb{R},$$

где $h(t)$ — характеристическая функция, соответствующая функции распределения $H(x)$.

Доказательство леммы 2 можно найти, например, в книге [16] (см. теорему 2.9.1 там).

Пусть $\Phi(x)$ — стандартная нормальная функция распределения.

Следствие 1. *Если существуют числа $a\in\mathbb{R}$, $\sigma^2 > 0$, $m_n \ge 1$ и функция распределения $A(x)$ такие, что при $n\to\infty$*

$$\mathsf{P}(S_{n,m_n} < x) \longrightarrow \Phi\left(\frac{x-a}{\sigma}\right); \qquad (11)$$
$$\mathsf{P}(N_n < m_n x) \Longrightarrow A(x), \qquad (12)$$

то

$$\mathsf{P}(S_{n,N_n} < x) \longrightarrow \int_0^\infty \Phi\left(\frac{x-au}{\sigma\sqrt{u}}\right) dG(u).$$

Теорема 5. *Предположим, что существуют числа $a\in\mathbb{R}$, $\sigma^2\in(0,\infty)$, $\mu\in(0,\infty)$, $\lambda\in(0,\infty)$ и последовательность натуральных чисел $\{m_n\}_{n\ge1}$ такие, что выполнены условия* (11) *и* (12) *с $A(x) = G(x;\mu,\lambda)$. Тогда*

$$\mathsf{P}(S_{n,N_n} < x) \Longrightarrow F_{\mathrm{VG}}(x;a,\sigma,\mu,\lambda) \ (n\to\infty), \qquad (13)$$

причем предельная случайная величина Z с дисперсионным гамма-распределением $F_{\mathrm{VG}}(x;a,\sigma,\mu,\lambda)$ может

быть представлена в виде разности независимых случайных величин, имеющих гамма-плотности с одинаковыми параметрами формы и разными масштабными параметрами.

Доказательство. По следствию 1 и определению дисперсионного гамма-распределения (4) из условий (11) и (12) вытекает (13). Остается убедиться, что предельная случайная величина Z может быть представлена в виде разности независимых случайных величин, имеющих гамма-распределения с одинаковыми параметрами формы и разными масштабными параметрами.

По лемме 2 функции распределения $F_{\mathrm{VG}}(x;a,\sigma,\mu,\lambda)$ случайной величины Z соответствует характеристическая функция

$$\begin{aligned}
&\mathsf{E}e^{itZ} = \\
&= \int\limits_0^\infty \exp\left\{z\left(ita - \frac{\sigma^2t^2}{2}\right)\right\}\frac{\lambda^\mu}{\Gamma(\mu)}e^{-\lambda z}z^{\mu-1}\,dz = \\
&= \frac{\lambda^\mu}{\Gamma(\mu)}\int\limits_0^\infty \exp\left\{z\left(ita - \frac{\sigma^2t^2}{2} - \lambda\right)\right\}z^{\mu-1}\,dz = \\
&\qquad = \left(\frac{\lambda}{\lambda - ita + \sigma^2t^2/2}\right)^\mu, \quad t\in\mathbb{R}. \qquad (14)
\end{aligned}$$

Введем перепараметризацию

$$\begin{cases} w - v = \dfrac{a}{\lambda}; \\ vw = \dfrac{\sigma^2}{2\lambda}. \end{cases} \qquad (15)$$

Из первого уравнения (15) получим

$$w = v + \frac{a}{\lambda}.$$

Из второго получим

$$v\left(v + \frac{a}{\lambda}\right) = \frac{\sigma^2}{2\lambda},$$

или

$$v^2 + \frac{a}{\lambda}v - \frac{\sigma^2}{2\lambda} = 0.$$

Система (15) имеет два решения относительно v:

$$v_1 = -\frac{a}{2\lambda} + \frac{1}{2}\sqrt{\frac{a^2}{\lambda^2} + \frac{2\sigma^2}{\lambda}};$$

$$v_2 = -\frac{a}{2\lambda} - \frac{1}{2}\sqrt{\frac{a^2}{\lambda^2} + \frac{2\sigma^2}{\lambda}}.$$

Одно из них — v_1 — положительно. При этом также положительно значение

$$w_1 = v_1 + \frac{a}{\lambda}.$$

В дальнейшем будем использовать параметры

$$\lambda_1 = \frac{1}{v_1} = \left(\frac{1}{2}\sqrt{\frac{a^2}{\lambda^2} + \frac{2\sigma^2}{\lambda}} - \frac{a}{2\lambda}\right)^{-1};$$

$$\lambda_2 = \frac{1}{w_1} = \left(\frac{1}{2}\sqrt{\frac{a^2}{\lambda^2} + \frac{2\sigma^2}{\lambda}} + \frac{a}{2\lambda}\right)^{-1}.$$

В этих обозначениях характеристическая функция (14) принимает вид:

$$\begin{aligned}
\mathsf{E}e^{itZ} &= \left(\frac{\lambda}{\lambda - ita + \sigma^2t^2/2}\right)^\mu = \\
&= \left(\frac{\lambda_1\lambda_2}{\lambda_1\lambda_2 - (\lambda_2-\lambda_1)it + t^2/2}\right)^\mu = \\
&= \left(\frac{\lambda_1\lambda_2}{(\lambda_1 - it)(\lambda_2 + it)}\right)^\mu = \\
&= \left(\frac{\lambda_1}{\lambda_1 - it}\right)^\mu\left(\frac{\lambda_2}{\lambda_2 + it}\right)^\mu = \\
&= \mathsf{E}\exp\left\{it\left[U(\mu,\lambda_1) - U(\mu,\lambda_2)\right]\right\},
\end{aligned}$$

где $U(\mu,\lambda_i)$, $i=1,2$, — независимые случайные величины, имеющие соответственно гамма-распределения с параметром формы μ и параметрами масштаба λ_i, $i=1,2$.

Таким образом, характеристическая функция (8) случайной величины Z является характеристической функцией разности независимых случайных величин, имеющих дисперсионное гамма-распределение с одинаковыми параметрами масштаба и различными параметрами формы. Теорема доказана.

Теорема 5 обобщает теорему 12.7.3 из [16], устанавливающую сходимость распределений случайных сумм к несимметричному распределению Лапласа, являющемуся дисперсионным гамма-распределением с параметром $\mu = 1$.

Из следствия 1 и определения скошенного распределения Стьюдента (1) вытекает следующий результат.

Теорема 6. *Предположим, что существуют числа $a\in\mathbb{R}$, $\sigma^2\in(0,\infty)$, $\mu\in(0,\infty)$, $\lambda\in(0,\infty)$ и последовательность натуральных чисел $\{m_n\}_{n\geqslant 1}$ такие, что выполнены условия* (11) *и* (12) *с $A(x) = H(x;\mu,\lambda)$, где $H(x;\mu,\lambda)$ — функция обратного гамма-распределения с параметрами μ, λ, соответствующая плотности* (2). *Тогда*

$$\mathsf{P}(S_{n,N_n} < x) \Longrightarrow P_{\mathrm{SS}}(x;a,\sigma,\mu,\lambda) \quad (n\to\infty),$$

где $P_{\mathrm{SS}}(x;a,\sigma,\mu,\lambda)$ *— функция скошенного распределения Стьюдента, соответствующая плотности* $p_{\mathrm{SS}}(x;a,\sigma,\mu,\lambda)$ (*см.* (1)).

4 Заключение

Гамма-распределение и обратное гамма-распределение являются частными представителями класса обобщенных гамма-распределений, важная роль которых в моделировании и анализе стохастической структуры информационных потоков описана в книге [25]. Обобщенные гамма-распределения (ОГ-распределения) были впервые описаны как единое семейство в 1962 г. в работе [26] в качестве семейства вероятностных моделей, включающего в себя одновременно гамма-распределения и распределения Вейбулла.

Обобщенным гамма-распределением называется распределение, определяемое плотностью вероятностей вида

$$f(x;\nu,\kappa,\delta) = \begin{cases} \dfrac{|\nu|}{\delta\Gamma(\kappa)}\left(\dfrac{x}{\delta}\right)^{\kappa\nu-1}\exp\left\{-\left(\dfrac{x}{\delta}\right)^{\nu}\right\}, & x\geq 0\,; \\ 0\,, & x<0\,, \end{cases} \tag{16}$$

с параметрами $\nu\in\mathbb{R}$, $\kappa,\,\delta\in\mathbb{R}^{+}$, отвечающими соответственно за *степень, форму и масштаб* (здесь $\Gamma(\kappa)$ — эйлерова гамма-функция: $\Gamma(\kappa)=\int\limits_0^\infty x^{\kappa-1}e^{-x}\,dx$).

Семейство ОГ-распределений включает в себя практически все наиболее популярные абсолютно непрерывные распределения. В частности, семейство ОГ-распределений содержит следующие распределения.

1. Гамма-распределение ($\nu=1$):

$$f(x;\kappa,\theta)=\frac{1}{\Gamma(\kappa)}\theta^{\kappa}x^{\kappa-1}e^{-\theta x}\,,\qquad x\geq 0\,,\ \kappa>0\,,\ \theta>0\,.$$

1.1 Показательное (экспоненциальное) распределение ($\nu=1$, $\kappa=1$):

$$f(x;\theta)=\theta e^{-\theta x}\,,\quad x\geq 0\,,\ \theta>0\,.$$

1.2 Распределение Эрланга ($\nu=1$, $\kappa\in\mathbb{N}$):

$$f(x;\kappa,\theta)=\frac{1}{\Gamma(\kappa)}\theta^{\kappa}x^{\kappa-1}e^{-\theta x}\,,\qquad x\geq 0\,,\ \kappa>0\,,\ \theta>0\,.$$

1.3 Распределение хи-квадрат ($\nu=1$, $\delta=2$):

$$f(x;n)=\frac{1}{2\Gamma(n/2)}\left(\frac{x}{2}\right)^{n/2-1}e^{-x/2}\,,\qquad x\geq 0\,,\ n\in\mathbb{N}\,.$$

2. Распределение Накагами ($\nu=2$):

$$f(x;\mu,\lambda)=\frac{2(\lambda\mu)^{\mu}}{\Gamma(\mu)}x^{2\mu-1}e^{-\lambda\mu x^2}\,,\qquad x\geq 0\,,\ \mu>0\,,\ \lambda>0\,.$$

2.1 Полунормальное распределение (распределение максимума винеровского процесса на отрезке $[0,1]$) ($\nu=2$, $\kappa=1/2$):

$$f(x;\delta)=\sqrt{\frac{2}{\pi\delta}}\exp\left\{-\frac{x^2}{2\delta^2}\right\},\ x\geq 0,\ \delta>0.$$

2.2 Распределение Рэлея ($\nu=2$, $\kappa=1$):

$$f(x;\delta)=\frac{x}{\delta^2}\exp\left\{-\frac{x^2}{2\delta^2}\right\}\,,\ \ x\geq 0\,,\ \delta>0\,.$$

2.3 Хи-распределение ($\nu=2$, $\delta=\sqrt{2}$):

$$f(x;n)=\frac{1}{2^{n/2-1}\Gamma(n/2)}\,x^{n-1}\exp\left\{-\frac{x^2}{2}\right\}\,,\qquad x\geq 0\,,\ n\in\mathbb{N}\,.$$

2.4 Распределение Максвелла — распределение модуля скорости движения молекул в разреженном газе ($\nu=2$, $\kappa=3/2$):

$$f(x;\delta)=\sqrt{\frac{2}{\pi}}\,\frac{x^2}{\delta^3}\,\exp\left\{-\frac{x^2}{2\delta^2}\right\}\,,\qquad x\geq 0\,,\ \delta>0\,.$$

3. Распределение Вейбулла–Гнеденко ($\kappa=1$):

$$f(x;\eta,\mu)=\frac{\eta x^{\eta-1}}{\mu^{\eta}}\,\exp\left\{-\left(\frac{x}{\mu}\right)^{\eta}\right\}\,,\qquad x\geq 0\,,\ \eta>0\,,\ \mu>0\,.$$

4. Обратное гамма-распределение ($\nu=-1$):

$$f(x;\mu,\lambda)=\frac{1}{\mu\lambda\Gamma(\lambda)}\left(\frac{\mu\lambda}{x}\right)^{\lambda+1}\exp\left\{-\frac{\mu\lambda}{x}\right\}\,,\qquad x\geq 0\,,\ \lambda>0\,,\ \mu>0\,.$$

4.1 Распределение Леви ($\nu = -1$, $\kappa = 1/2$):

$$f(x;\mu) = \sqrt{\frac{\mu}{2\pi}}\,\frac{1}{x^{3/2}}\exp\left\{-\frac{\mu}{2x}\right\}, \quad x \geq 0\,,\ \mu > 0\,.$$

5. Логнормальное распределение ($\kappa \to \infty$):

$$f(x;\mu,\delta) = \frac{1}{\delta x\sqrt{2\pi}}\exp\left\{-\frac{(\log x - \mu)^2}{2\delta^2}\right\}, \quad x \geq 0\,,\ \mu \in \mathbb{R}\,,\ \delta > 0\,.$$

Широкая применимость ОГ-распределений обусловлена возможностью их использования в качестве адекватных асимптотических аппроксимаций, поскольку практически все они выступают в качестве предельных в различных предельных теоремах теории вероятностей, а именно:

- показательное распределение выступает в качестве предельного как в схеме максимума (минимума) (см., например, [21]), так и в схеме геометрического суммирования, описывая распределение времени восстановления в прореженных процессах восстановления, выступающих моделями потоков редких событий (см., например, [27]);
- гамма-распределение является безгранично делимым и потому выступает в качестве предельного для распределений сумм независимых равномерно предельно малых случайных величин; при этом распределение Эрланга возникает как допредельное распределение суммы независимых экспоненциально распределенных случайных величин, что в терминах случайной интенсивности может означать, что если случайная интенсивность потока поступления запросов имеет гамма-распределение со значимым параметром формы, то при обработке этих запросов в основном задействованы механизмы последовательной обработки информации;
- распределение Вейбулла–Гнеденко принадлежит к так называемому первому типу предельных распределений экстремальных порядковых статистик (минимума или максимума) (см., например, [21]), что в терминах случайной интенсивности может означать, что если случайная интенсивность потока поступления запросов имеет распределение Вейбулла–Гнеденко со значимым параметром степени, то при обработке этих запросов в основном задействованы механизмы параллельной обработки информации;
- полунормальное распределение (распределение модуля стандартной нормальной случайной величины) возникает как предельное для максимальных частичных сумм независимых случайных величин (см., например, [28]);
- распределение Леви принадлежит к классу устойчивых законов и потому является предельным для сумм независимых одинаково распределенных случайных величин; оно также является распределением времени достижения стандартным винеровским процессом (процессом броуновского движения) фиксированного уровня;
- логнормальное распределение выступает в качестве предельного для распределения размера частиц при дроблении (см., например, [29]).

Эти свойства ОГ-распределений обосновывают, в частности, целесообразность моделирования с их помощью распределения случайной интенсивности потока запросов в информационных системах. Это семейство также широко используется в других прикладных задачах в самых разных областях (см., например, [25]).

Рассмотренные выше четырехпараметрические семейства скошенных распределений Стьюдента и дисперсионных гамма-распределений являются подклассами пятипараметрического семейства распределений

$$W(x;a,\sigma,\nu,\kappa,\delta) = \int\limits_0^\infty \Phi\left(\frac{x-au}{\sigma\sqrt{u}}\right) f(u;\nu,\kappa,\delta)\,du\,, \quad (17)$$

где $f(u;\nu,\kappa,\delta)$ — плотность ОГ-распределения (16). Распределения вида (17) назовем *обобщенными дисперсионными гамма-распределениями*.

Задача поиска универсальной модели статистических закономерностей во многих областях, в частности в финансовой математике или в физике плазмы, подобна задаче отыскания «философского камня» в алхимии и поэтому не имеет точного решения. Однако, основываясь на вышеперечисленных аналитических и асимптотических свойствах представителей семейства ОГ-распределений и следствии 1 как теоретико-вероятностной формализации принципа неубывания неопределенности в сложных системах, можно утверждать, что семейство обобщенных дисперсионных гамма-распределений является *практически* универсальным для многих задач.

Литература

1. *Barndorff-Nielsen O. E.* Exponentially decreasing distributions for the logarithm of particle size // Proc. R. Soc. A, 1977. Vol. 353. P. 401–419.
2. *Eberlein E., Keller U.* Hyperbolic distributions in finance // Bernoulli, 1995. Vol. 1. No. 3. P. 281–299.
3. *Prause K.* Modeling financial data using generalized hyperbolic distributions. — Freiburg: Universität Freiburg, Institut für Mathematische Stochastic, 1997. Preprint No. 48.
4. *Eberlein E., Keller U., Prause K.* New insights into smile, mispricing and value at risk: The hyperbolic model // J. Business, 1998. Vol. 71. P. 371–405.
5. *Barndorff-Nielsen O. E.* Processes of normal inverse Gaussian type // Finance Stochastics, 1998. Vol. 2. P. 41–18.
6. *Eberlein E., Prause K.* The generalized hyperbolic model: Financial derivatives and risk measures. — Freiburg: Universität Freiburg, Institut für Mathematische Stochastic, 1998. Preprint No. 56.
7. *Eberlein E.* Application of generalized hyperbolic Lèvy motions to finance. — Freiburg: Universität Freiburg, Institut für Mathematische Stochastic, 1999. Preprint No. 64.
8. *Королев В. Ю.* Вероятностно-статистические методы декомпозиции волатильности хаотических процессов. — М.: Изд-во МГУ, 2011. 510 с.
9. *Aas K., Haff I. H.* The generalized hyperbolic skew Student's t-distribution // J. Financial Econometrics, 2006. Vol. 4. No. 2. P. 275–309.
10. *Kim Y., McCulloch J. H.* The skew-student distribution with application to U.S. stock market returns and the equity premium. — Columbus: Department of Economics, Ohio State University, 2007. Preprint.
11. *Madan D. B., Seneta E.* The variance gamma (V.G.) model for share market return // J. Business, 1990. Vol. 63. P. 511–524.
12. *Carr P. P., Madan D. B., Chang E. C.* The Variance Gamma process and option pricing // European Finance Rev., 1998. Vol. 2. P. 79–105.
13. *Гнеденко Б. В., Колмогоров А. Н.* Предельные распределения для сумм независимых случайных величин. — М.–Л.: ГИТТЛ, 1949.
14. *Gnedenko B. V., Korolev V. Yu.* Random summation: Limit theorems and applications. — Boca Raton: CRC Press, 1996.
15. *Kotz S., Kozubowski T. J., Podgorski K.* The Laplace distribution and generalizations: A revisit with applications to communications, economics, engineering and finance. — Boston: Birkhauser, 2001.
16. *Королев В. Ю., Бенинг В. Е., Шоргин С. Я.* Математические основы теории риска. — 2-е изд., перераб. и доп. — М.: Физматлит, 2011. 620 с.
17. *Королев В. Ю.* Сходимость случайных последовательностей с независимыми случайными индексами. II // Теория вероятностей и ее применения, 1995. Т. 40. Вып. 4. С. 907–910.
18. *Королев В. Ю.* Сходимость случайных последовательностей с независимыми случайными индексами. I // Теория вероятностей и ее применения, 1994. Т. 39. Вып. 2. С. 313–333.
19. *Korolev V. Yu.* A general theorem on the limit behavior of superpositions of independent random processes with applications to Cox processes // J. Math. Sci., 1996. Vol. 81. No. 5. P. 2951–2956.
20. *Бенинг В. Е., Королев В. Ю.* Об использовании распределения Стьюдента в задачах теории вероятностей и математической статистики // Теория вероятностей и ее применения, 2004. Т. 49. Вып. 3. С. 417–435.
21. *Гумбель Э.* Статистика экстремальных значений. — М.: Мир, 1965.
22. *Wilks S. S.* Recurrence of extreme observations // J. Amer. Math. Soc., 1959. Vol. 1. No. 1. P. 106–112.
23. *Невзоров В. Б.* Рекорды. Математическая теория. — М.: Фазис, 2000.
24. *Гнеденко Б. В., Фахим Х.* Об одной теореме переноса // Докл. АН СССР, 1969. Т. 187. № 1. С. 15–17.
25. *Королев В. Ю., Шоргин С. Я.* Математические методы анализа стохастической структуры информационных потоков. — М.: ИПИ РАН, 2011. 130 с.
26. *Stacy E. W.* A generalization of the gamma distribution // Annals Math. Statistics, 1962. Vol. 33. P. 1187–1192.
27. *Kalashnikov V. V.* Geometric sums: Bounds for rare events with applications. — Dordrecht: Kluwer Academic Publs., 1997.
28. *Королев В. Ю., Соколов И. А.* Математические модели неоднородных потоков экстремальных событий. — М.: ТОРУС ПРЕСС, 2008.
29. *Королев В. Ю.* О распределении размеров частиц при дроблении // Информатика и её применения, 2009. Т. 3. Вып. 3. С. 60–68.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 12–18

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АНАЛИЗА НЕЛИНЕЙНЫХ МНОГОКАНАЛЬНЫХ КРУГОВЫХ СТОХАСТИЧЕСКИХ СИСТЕМ, ОСНОВАННОЕ НА ПАРАМЕТРИЗАЦИИ РАСПРЕДЕЛЕНИЙ*

И. Н. Синицын[1]

Аннотация: Статья посвящена теории и математическому обеспечению для анализа одно- и многомерных распределений процессов в многоканальных нелинейных круговых стохастических системах (КСтС) на базе методов параметризации распределений. Рассматриваются круговые ортогональные разложения (КОР) плотностей круговых случайных величин (КСВ) и процессов, стохастические уравнения многоканальных нелинейных КСтС, интегродифференциальные уравнения для одно- и многомерных плотностей, общий метод КОР, а также методы «намотанной» нормальной аппроксимации (МННА), начальных и центральных моментов. Описаны модули инструментального программного обеспечения «CStS-ANALYSIS» в среде MATLAB. Результаты проиллюстрированы примером.

Ключевые слова: аналитическое моделирование; коэффициенты кругового ортогонального разложения; круговая случайная величина; круговой стохастических процесс; «намотанная» нормальная плотность; нелинейная многоканальная стохастическая система; одно- и многомерные плотности распределения; ортогональное разложение плотности; эталонная плотность; MATLAB; «CStS-ANALYSIS»

1 Введение

В [1–4] рассмотрено математическое обеспечение для спектрально-корреляционного анализа процессов в нелинейных КСтС, основанное на методе круговой статистической линеаризации посредством «намотанного» (*wrapped*) нормального распределения.

Для анализа одно- и многомерных распределений в нелинейных КСтС невысокой размерности прямые методы решения интегродифференциальных уравнений Пугачёва для характеристических функций приводят к алгоритмам, требующим применения только средств суперкомпьютерной техники [5].

Как известно [6–8], для анализа стохастических процессов в многомерных нелинейных КСтС в евклидовом пространстве широкое применение получили методы параметризации распределений (нормальной аппроксимации, начальных и центральных моментов, квазимоментов, а также ортогональных разложений и их модификаций). В настоящее время создано инструментальное программное обеспечение в среде MATLAB [3, 7, 8]. Применительно к нелинейным КСтС это математическое обеспечение требует развития.

Статья включает девять разделов. В разд. 2 и 3 рассматриваются ортогональные разложения плотностей КСВ и стохастических процессов. В разд. 4 приводятся стохастические уравнения нелинейной многоканальной стохастической системы, а также интегродифференциальные уравнения для одно- и многомерных плотностей. Общее математические обеспечение «CStS-ANALYSIS» описано в разд. 5. В разд. 6 рассмотрено математическое обеспечение для метода круговой «намотанной» нормальной аппроксимации. В разд. 7 приведены уравнения методов круговых начальных и центральных моментов. В разд. 8 дан иллюстративный пример. Заключение содержит основные выводы.

2 Ортогональное разложение плотности круговой случайной величины

В задачах стохастической информатики широкое распространение получили модели распределений КСВ, основанные на следующих основных подходах:

– «наматывание» (*wrapping*) распределений линейных СВ X на круг единичного радиуса $\Theta = X(\mathrm{mod}\, 2\pi)$;

*Работа выполнена при финансовой поддержке РФФИ (проект № 10-07-00021).

[1]Институт проблем информатики Российской академии наук, sinitsin@dol.ru

– преобразования к полярным координатам двумерного линейного распределения (так называемые *offset distributions*) или использование стереографической проекции;
– характеризации (параметризации) кругового распределения круговыми моментами, семиинвариантами и другими характеристиками, имеющими важное предметное значение, на основе ортогонального разложения плотности по некоторой биортонормальной системе функций с весом, задаваемым некоторой эталонной плотностью распределения.

Рассмотрим ортогональное разложение плотности $f_\theta(\theta)$ для КСВ Θ, обладающей конечными круговыми моментами, по некоторой известной биортогональной системе функций $\{p_\nu(\theta), q_\nu(\theta)\}$ с весом $w_\theta(\theta)$ таким, что

$$\int_{-\infty}^{\infty} w_\theta(\theta) p_\nu(\theta) q_\nu(\theta)\, d\theta = \delta_{\nu\mu} = \begin{cases} 0 \text{ при} & \nu \neq \mu\,; \\ 1 \text{ при} & \nu = \mu\,, \end{cases}$$

следующего вида [6–8]:

$$f_\theta(\theta) = w_\theta(\theta) \sum_\nu c_\nu p_\nu(\theta)\,, \qquad (1)$$

где

$$c_\nu = \int_{-\infty}^{\infty} f_\theta(\theta) q_\nu(\theta)\, d\theta = \left[q_\nu\left(\frac{\partial}{i\partial\lambda}\right) g(\lambda) \right]_{\lambda=0} \quad (i = \sqrt{-1})\,. \qquad (2)$$

Здесь величины c_ν называются коэффициентами КОР; $g_\theta(\theta)$ — характеристическая функция, соответствующая плотности $f_\theta(\theta)$; $w_\theta(\theta)$ — плотность эталонного распределения.

Если в (1) потребовать совпадения круговых моментов первого и второго порядка плотностей $f_\theta(\theta)$ и $w_\theta(\theta)$, то КОР примет следующий вид:

$$f_\theta(\theta) = w_\theta(\theta) \left[1 + \sum_{\nu=3}^{\infty} c_\nu p_\nu(\theta) \right]. \qquad (3)$$

Замечание 1. Функции $p_\nu(\theta)$ и $q_\nu(\theta)$ необязательно должны быть полиномами. Они могут быть любыми функциями, удовлетворяющими условию биортонормальности и условиям существования интегралов (2). Все сказанное о разложении (3) справедливо и в более общем случае. Однако, если функции $q_\nu(\theta)$ не являются полиномами, то, несмотря на совпадение моментов первого и второго порядка распределений $f_\theta(\theta)$ и $w_\theta(\theta)$, круговые коэффициенты c_ν не будут равны нулю при $\nu = 1, 2$, вследствие чего суммирование по ν будет начинаться с $\nu = 1$.

Замечание 2. Иногда применяют разложение по производным некоторой плотности $w_\theta(\theta)$, имеющей производные и моменты всех порядков [6–8]:

$$f_\theta(\theta) = \sum_{\nu=0}^{\infty} c_\nu w_\theta^{(\nu)}(\theta)\,.$$

В этом случае $p_\nu(\theta) = w_\theta^{(\nu)}(\theta)/w_\theta(\theta)$, а функции $q_\nu(\theta)$ представляют собой полиномы.

В общем случае в зависимости от того, какие величины приняты за параметры конечного отрезка КОР, различают моментные КОР, семиинвариантные КОР, квазимоментные КОР и др. Поэтому, обозначая эти параметры через u, будем записывать КОР (3) в виде:

$$f_\theta(\theta; u) = w_\theta(\theta; u) \left[1 + \sum_{\nu=3}^{\infty} c_\nu p_\nu(\theta) \right];$$

$$c_\nu = \left[q_\nu\left(\frac{\partial}{i\partial\lambda}\right) g(\lambda) \right]_{\lambda=0},$$

где $c_\nu = c_\nu(u)$; $p_\nu(\theta) = p_\nu(\theta, u)$; $q_\nu = q_\nu(\theta, u)$.

3 Ортогональные разложения одно- и многомерных плотностей круговых стохастических процессов

Для действительного КСтП одномерная плотность для момента времени t в силу (2) и (3) определяется следующим КОР:

$$f_1(\theta_t; t, u) = w_1(\theta_t, t, u) \left[1 + \sum_{\nu=3}^{\infty} c_{\nu t} p_\nu(\theta_t, u) \right], \quad (4)$$

где

$$c_{\nu t} = c_{\nu t}(t, u) = \int_{-\infty}^{\infty} f_1(\theta_t; t, u) q_\nu(\theta_t) = \left[q_\nu\left(\frac{\partial}{i\partial\lambda}\right) g_\theta(\lambda; t, u) \right].$$

Для действительных КСтП n-мерные плотности для моментов времени $t_1, \ldots, t_n$ определяются как совокупность согласованных КОР плотностей КСВ $\theta_{t_1}, \ldots, \theta_{t_n}$:

$$f_n(\theta_{t_1}, \ldots, \theta_{t_n}; t_1, \ldots, t_n, u) = w_n(\theta_{t_1}, \ldots, \theta_{t_n}; t_1, \ldots, t_n, u) \times$$

$$\times \left[1+\sum_{k=3}^{\infty}\sum_{\nu_1+\dots+\nu_n=k} c_{\nu_1,\dots,\nu_n}(t_1,\dots,t_n,u)\times \right.$$
$$\left. \times\, p_{\nu_1,\dots,\nu_n}(\theta_{t_1},\dots,\theta_{t_n};t_1,\dots,t_n,u)\right], \quad (5)$$

где

$$c_{\nu_1,\dots,\nu_n}(t_1,\dots,t_n,u) =$$
$$= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_n(\theta_{t_1},\dots,\theta_{t_n};t_1,\dots,t_n,u)\times$$
$$\times\, q_{\nu_1,\dots,\nu_n}(\theta_{t_1},\dots,\theta_{t_n};t_1,\dots,t_n,u)\times$$
$$\times\, d\theta_{t_1}\cdots d\theta_{t_n} =$$
$$= \left[q_{\nu_1,\dots,\nu_n}\left(\frac{\partial}{i\partial\lambda_1},\dots,\frac{\partial}{i\partial\lambda_n};t_1,\dots,t_n,u\right)\times\right.$$
$$\left.\times\, g_n(\lambda_1,\dots,\lambda_n;t_1,\dots,t_n,u)\right]_{\lambda_1=\dots=\lambda_n=0}.$$

Здесь $\{p_{\nu_1,\dots,\nu_n}(\theta_{t_1},\dots,\theta_{t_n},u),\ q_{\nu_1,\dots,\nu_n}(\theta_{t_1},\cdots\cdots,\theta_{t_n},u)\}$ — согласованные биортонормальные системы полиномов, соответствующие согласованным многомерным плотностям $w_n(\theta_{t_1},\dots,\theta_{t_n},u)$, имеющим те же круговые моменты первого и второго порядка, что и КСтП $\Theta(t)=\Theta_t$.

Ограничиваясь в (4) и (5) полиномами не выше N-й степени, получим согласованное приближенное представление всех многомерных плотностей КСтП Θ_t. Этим приближенным представлением можно практически пользоваться, если КСтП имеет конечные круговые моменты до N-го порядка включительно, независимо от того, существуют или не существуют его моменты высших порядков.

При рассмотрении круговых марковских СтП соответствующие КОР используют для двух плотностей: одномерной и переходной.

В рамках спектрально-корреляционной теории принимают $n=1,2$ и ограничиваются рассмотрением круговых дисперсий и ковариационных функций [2].

4 Многоканальные нелинейные круговые стохастические системы

Введем векторный КСтП $Y(t)=Y_t$, составленный из КСтП $\Theta_j(t)=\Theta_{jt}$ $(j=1,\dots,r)$, $Y_t=[\Theta_{1t}\dots\Theta_{rt}]$, где $r_y=r$ — число каналов в КСтС. Пусть векторное нелинейное стохастическое дифференциальное уравнение (понимаемое в смысле Ито), описывающее эволюцию Y_t, имеет вид [6–8]:

$$\dot{Y}_t=\varphi(Y_t,t)+\psi(Y_t,t)V_t\,;\quad Y(t_0)=Y_0\,, \quad (6)$$

где $\varphi(Y_t,t)$ и $\psi(Y_t,t)$ — $(r_y\times 1)$- и $(r_y\times r_V)$-мерные в общем случае известные нелинейные функции; V_t — $(r_V\times 1)$-мерный круговой белый шум в строгом смысле, представляющий собой среднюю квадратическую производную по времени от кругового процесса с независимыми приращениями $W_t, V_t=\dot{W}_t$. Начальное значение Y_0 будем считать независимым от приращений W_t.

Обозначим через $\chi(\mu;t)$ логарифмическую производную по времени от одномерной характеристической функции $h_1(\mu,t)$ кругового белого шума V_t, $\chi(\mu;t)=(\partial/\partial t)h_1(\mu;t)$. Тогда при известных условиях [6, 7] одно- и n-мерные плотности будут определяться следующей системой интегродифференциальных уравнений:

$$\frac{\partial}{\partial t_n} f_n(y_{t_1},\dots,y_{t_n}) =$$
$$= \frac{1}{(2\pi)^{nr}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\left[i\lambda_n^{\mathrm{T}}\varphi(\eta_n,t_n)+\right.$$
$$\left.+\,\chi(\psi(\eta_n,t_n)^{\mathrm{T}}\lambda_n;t_n)\right]\exp\left\{i\sum_{k=1}^{n}\lambda_k^{\mathrm{T}}(\eta_k-y_{t_k})\right\}\times$$
$$\times\, f_n(\eta_1,\dots,\eta_n;t_1,\dots,t_n)\times$$
$$\times\, d\eta_1\cdots d\eta_n d\lambda_1\dots d\lambda_n$$

$(n=1,2,\dots)$ при условиях:

$$f_n(y_{t_1},\dots,y_{t_{n-1}};t_1,\dots,t_{n-1},t_{n-1}) =$$
$$= f_{n-1}(y_{t_1},\dots,y_{t_{n-1}};t_1,\dots,t_{n-1})\delta(y_{t_n}-y_{t_{n-1}});$$
$$f_1(y_t;t_0)=f_0(y_t)\,.$$

Замечание 3. Для дискретных КСтС соответствующие интегроразностные уравнения получаются путем замены оператора дифференцирования по времени на оператор сдвига по времени [6–8].

5 Математическое обеспечение, основанное на ортогональных разложениях

При аппроксимации круговой одномерной плотности конечным отрезком КОР естественно за параметры u принять: m_t — вектор круговых средних; K_t — круговую ковариационную матрицу (от которых зависит эталонная плотность, а следовательно, и полиномы $\{p_\nu(y_t),q_\nu(y_t)\}$) и коэффициенты $c_{\kappa,t}$. Поэтому из (4) для отрезка КОР имеем:

$$\tilde{f}_1(y_t,u) = \\ = w_1(y_t,u)\left[1+\sum_{l=3}^{N}\sum_{|\kappa|=l} c_{\kappa t}p_\kappa(y_t,u)\right]. \quad (7)$$

При этом, согласно [6, 7], m_t, K_t и $c_{\nu t}$ будут определяться следующей системой обыкновенных дифференциальных уравнений:

$$\left.\begin{aligned}&\dot{m}_t = A_0(m_t,K_t,t)+\sum_{l=3}^{N}\sum_{|\nu|=l}A_{1\nu}(m_t,K_t,t)c_{\nu t};\\ &m_0 = m(t_0);\end{aligned}\right\} \quad (8)$$

$$\left.\begin{aligned}&\dot{K}_t = B_0(m_t,K_t,t)+\sum_{l=3}^{N}\sum_{|\nu|=l}B_{1\nu}(m_t,K_t,t)c_{\nu t};\\ &K_0 = K(t_0);\end{aligned}\right\} \quad (9)$$

$$\left.\begin{aligned}&\dot{c}_{\kappa t} = C_{\kappa 0}(m_t,K_t,t)+A_0(m_t,K_t,t)^{\mathrm{T}}q_\kappa^m(\alpha)+\\ &\quad+\mathrm{tr}\left[B_0(m_t,K_t,t)q_\kappa^K(\alpha)\right]+\\ &\quad+\sum_{l=3}^{N}\sum_{|\nu|=l}\left\{C_{\kappa\nu}(m_t,K_t,t)+\right.\\ &\quad\left.+A_{1\nu}(m_t,K_t,t)^{\mathrm{T}}q_\kappa^m(\alpha)+\right.\\ &\quad\left.+\mathrm{tr}\left[B_{1\nu}(m_t,K_t,t)q_\kappa^K(\alpha)\right]\right\}c_{\nu t};\\ &c_{\kappa 0} = c_\kappa(t_0).\end{aligned}\right\} \quad (10)$$

Здесь введены следующие обозначения:

$$\left.\begin{aligned}&A_0(m_t,K_t,t) = \int_{-\infty}^{\infty}\varphi(y_t,t)w_1(y_t,u)\,dy_t;\\ &A_{1\nu}(m_t,K_t,t) = \int_{-\infty}^{\infty}\varphi(y_t,t)p_\nu(y_t,u)\,dy_t);\end{aligned}\right\} \quad (11)$$

$$\left.\begin{aligned}&B_0(m_t,K_t,t) = B_{01}(m_t,K_t,t)+\\ &\quad+B_{01}(m_t,K_t,t)^{\mathrm{T}}+B_{02}(m_t,K_t,t) =\\ &\quad=\int_{-\infty}^{\infty}\left[\varphi(y_t,t)(y_t-m_t)^{\mathrm{T}}+\right.\\ &\quad+(y_t-m_t)\varphi(y_t,t)^{\mathrm{T}}+\\ &\quad\left.+\psi(y_t,t)G_t\psi_(y_t,t)^{\mathrm{T}}\right]w_1(y_t,u)\,dy_t;\\ &B_{1\nu}(m_t,K_t,t) = \int_{-\infty}^{\infty}\left[\varphi(y_t,t)(y_t-m_t)^{\mathrm{T}}+\right.\\ &\quad\left.+(y_t-m_t)\varphi(y_t,t)^{\mathrm{T}}\psi(y_t,t)G_t\psi(y_t,t)^{\mathrm{T}}\right]\times\\ &\quad\times w_1(y_t,u)\,dy_t;\end{aligned}\right\} \quad (12)$$

$$\left.\begin{aligned}&C_{\kappa 0}(m_t,K_t,t) =\\ &\quad=\int_{-\infty}^{\infty}\left\{q_\kappa\left(\frac{\partial}{i\partial\lambda}\right)\left[i\lambda^{\mathrm{T}}\varphi(y_t,t)+\right.\right.\\ &\quad\left.\left.+\chi(\psi(y_t,t)^{\mathrm{T}}\lambda;t)\right]\exp\left[i\lambda^{\mathrm{T}}y_t\right]\right\}_{\lambda=0}\times\\ &\quad\times w_1(y_t,u)\,dy_t;\\ &C_{\kappa\nu}(m_t,K_t,t) =\\ &\quad=\int_{-\infty}^{\infty}\left\{q_\kappa\left(\frac{\partial}{i\partial\lambda}\right)\left[i\lambda^{\mathrm{T}}\varphi(y_t,t)+\right.\right.\\ &\quad\left.\left.+\chi(\psi(y_t,t)^{\mathrm{T}}\lambda;t)\right]\exp\left[i\lambda^{\mathrm{T}}y_t\right]p_\nu(y_t,u)\right\}_{\lambda=0}\times\\ &\quad\times w_1(y_t,u)\,dy_t,\end{aligned}\right\} \quad (13)$$

где $q_\kappa^m(y_t)$ — матрица-столбец производных полинома $q_\kappa(y_t)$ по компонентам вектора m_t; $q_\kappa^K(y_t)$ — квадратная матрица производных полинома $q_\kappa(y_t)$ по элементам матрицы K_t; $G_t = [G_{lj}(t)]$ — матрица интенсивностей векторного кругового белого шума V, причем

$$G_{lj}(t) = \left[\frac{\partial^2\chi(\mu;t)}{\partial(i\mu_l)\partial(i\mu_j)}\right]_{\mu=0};$$

$q_\kappa^m(\alpha)$ и $q_\kappa^K(\alpha)$ — результат замены одночленов вида $y_{1t}^{\rho_1}\cdots y_{rt}^{\rho_r}$ соответствующими начальными моментами $\alpha_{\rho_1,\ldots,\rho_r}$ которые, как известно, зависят от $c_{\kappa t}$.

Замечание 4. Уравнения (10) нелинейны относительно $c_{\kappa t}$. Если отказаться от требования совпадения первых моментов и задать эти моменты для эталонной плотности априори, то для $c_{\kappa t}$ получатся линейные уравнения. Эти уравнения проще чем (10), однако при этом придется взять большее N в (7).

Таким образом, при использовании полиномиальной биортонормальной системы $\{p_\nu(y_t,u), q_\mu(y_t,u)\}$ в основе математического обеспечения кругового ортогонального (7) разложения одномерной плотности КСтП Y_t в КСтС (6) лежат уравнения (8)–(10) при условиях (11)–(13).

Аналогично на основе результатов [6, 7] составляются уравнения для n-мерных плотностей (5) ($n \geq 2$).

В состав математического обеспечения «CStS-ANALYSIS» входят:

- уравнения для различных типов систем;
- типовые системы биортонормальных систем функций для одно- и многомерных плотностей;
- ортогональные разложения плотностей одно- и многомерных распределений;

– обыкновенные дифференциальные (разностные) уравнения для параметров одно- и многомерных распределений с соответствующими начальными условиями;
– выражения для вычисления типовых функционалов, определяющих задачи вероятностного кругового анализа многоканальной КСтС;
– набор тестовых задач.

6 Математическое обеспечение на основе метода «намотанной» нормальной аппроксимации

Обобщая результаты [2] на случай, когда коэффициент при круговом белом шуме в уравнении (6) зависит от состояния, приведем основные уравнения кругового МННА:

$$\dot m_t = A_0(m_t, K_t, t)\,,\ \ m_0 = m(t_0)\,; \qquad (14)$$

$$\dot K_t = B_0(m_t, K_t, t)\,,\ \ K_0 = K(t_0)\,; \qquad (15)$$

$$\left.\begin{aligned} &\frac{\partial K(t_1,t_2)}{\partial t_2} = \\ &= \frac{K(t_1,t_2)}{K(t_2)}\,B_{01}(m_{t_2}, K_{t_2}, t_2) \quad (t_1 < t_2)\,; \\ &K(t_1,t_1) = K_{t_1}\,. \end{aligned}\right\} \qquad (16)$$

Здесь $A_0(m_t, K_t, t)$, $B_0(m_t, K_t, t)$ и $B_{01}(m_t, K_t, t)$ определены в (11)–(12) для «намотанного» нормального распределения.

В состав МННА включено математическое обеспечение [2] для системы (6) при $\psi(y_t, t) = I_r$.

7 Математическое обеспечение на основе методов круговых начальных и центральных моментов

Из уравнений разд. 5 для круговых «намотанных» начальных моментов $\alpha_\rho = \alpha_\rho(t)$ порядка ρ, обобщая [6, 7], имеем следующие уравнения:

$$\dot\alpha_\rho = A_{0,\rho} + \sum_{k=3}^{N}\sum_{|\nu|=k} A_{\nu,\rho}c_\nu(\alpha)\ \ (c_\nu = q_\nu(\alpha))\,, \quad (17)$$

где

$$A_{0,\rho} = \int_{-\infty}^{\infty}\left\{\frac{\partial^{|\rho|}}{\partial(i\lambda_1)^{\rho_1}\cdots\partial(i\lambda_r)^{\rho_r}}\left[i\lambda^{\mathrm T}\varphi(y_t,t) + \right.\right.$$

$$\left.\left. + \chi(\psi(y_t,t)^{\mathrm T}\lambda; t)\right]\exp\left[i\lambda^{\mathrm T}y_t\right]\right\}_{\lambda=0} w_1(y_t,u)\,dy_t\,;$$

$$A_{\nu,\rho} = \int_{-\infty}^{\infty}\left\{\frac{\partial^{|\rho|}}{\partial(i\lambda_1)^{\rho_1}\cdots\partial(i\lambda_r)^{\rho_r}}\left[i\lambda^{\mathrm T}\varphi(y_t,t) + \right.\right.$$

$$\left.\left. + \chi(\psi(y_t,t)^{\mathrm T}\lambda; t)\right]\exp\left[i\lambda^{\mathrm T}y_t\right]\right\}_{\lambda=0} \times$$

$$\times\, p_\nu(y_t,u)w_1(y_t,u)\,dy_t$$

$$(|\rho| = \rho_1 + \cdots + \rho_r, \rho_1\,,\ \ldots\,,\ \rho_r = 0, 1\,,\ \ldots\,,\ N)\ .$$

Напомним, что $q_\nu(\alpha)$ представляет собой результат замены всех одночленов $y_{1t}^{\rho_1}\cdots y_{rt}^{\rho_r}$ в выражении полинома $q_\nu(\alpha)$ соответствующими моментами $\alpha_{\rho_1\,,\ldots\,,\,\rho_r}$.

Замечание 5. При составлении уравнений (7) в конкретных задачах следует иметь в виду, что число N_r^ρ моментов ρ-го порядка r-мерного вектора определяется формулой:

$$N_r^\rho = C_{r+\rho-1}^\rho = \frac{(r+\rho-1)!}{\rho!(r-1)!}\,,$$

а полное число моментов, не превосходящих N, для r-мерного вектора равно:

$$P_r^N = \sum_{l=1}^{N} N_r^l = \frac{(N+r)!}{N!r!} - 1\,.$$

Замечание 6. Уравнения (17) линейны относительно α_ρ ($|\rho| = 3\,,\ \ldots\,,\ N$) и нелинейны относительно моментов первого и второго порядка, поскольку эталонная плотность и полиномы $p_\nu(y_t,u)$ и $q_\nu(y_t,u)$ зависят от моментов первого и второго порядка.

Обобщая [6, 7] для математических ожиданий m_h ($h = 1\,,\ \ldots\,,\ r$) и центральных моментов μ_ρ порядка ρ, представим уравнения в виде:

$$\dot m_h = A_{0,h} + \sum_{k=3}^{\infty}\sum_{|\nu|=k} A_{\nu r}c_\nu\,;\ \ c_\nu = q_\nu(\alpha)\,; \qquad (18)$$

$$\dot\mu_\rho = B_{0,\rho} - \sum_{h=1}^{r}\rho_h B_{0,h}\mu_{\rho-e_h} +$$

$$+ \sum_{k=3}^{N}\sum_{|\nu|=k}\left[B_{\nu,\rho} - \sum_{h=1}^{r}\rho_h B_{\nu,h}\mu_{\rho-e_h}\right]c_\nu$$

$$(\rho_1\,,\ \ldots\,,\ \rho_r = 0, 1,\ ,\ \ldots\,,\ N\,;$$

$$|\rho| = 2\,,\ \ldots\,,\ N)\,. \qquad (19)$$

Здесь введены следующие обозначения:

$$e_h = \left[0\cdots0\underset{h}{1}0\cdots0\right]^{\mathrm T}\ ;$$

$$A_{0,h} = \int\limits_{-\infty}^{\infty} \varphi_h(y_t,t) w_1(y_t,u)\,dy_t\,;$$

$$A_{\nu,h} = \int\limits_{-\infty}^{\infty} \varphi_h(y_t,t) p_\nu(y_t,u) w_1(y_t,u)\,dy_t\,;$$

$$A_{0,\rho} = \int\limits_{-\infty}^{\infty} \left\{ \frac{\partial^{|\rho|}}{\partial(i\lambda_1)^{\rho_1}\cdots\partial(i\lambda_r)^{\rho_r}} \left[i\lambda^{\mathrm{T}}\varphi(y_t,t) + \right.\right.$$
$$\left. + \chi(\psi(y_t,t)^{\mathrm{T}}\lambda;t)\right] \times$$
$$\left.\times \exp\left[i\lambda^{\mathrm{T}}(y_t-m)\right]\right\}_{\lambda=0} w_1(y_t,u)\,dy_t\,;$$

$$B_{\nu,\rho} = \int\limits_{-\infty}^{\infty} \left\{ \frac{\partial^{|\rho|}}{\partial(i\lambda_1)^{\rho_1}\cdots\partial(i\lambda_r)^{\rho_r}} \left[i\lambda^{\mathrm{T}}\varphi(y_t,t) + \right.\right.$$
$$\left. + \chi(\psi(y_t,t)^{\mathrm{T}}\lambda;t)\right] \exp\left[i\lambda^{\mathrm{T}}(y_t-m)\right]\Big\}_{\lambda=0} \times$$
$$\times\, p_\nu(y_t,u) w_1(y_t,u)\,dy_t\,,$$

а $q_\nu(\alpha)$ должны быть выражены через центральные моменты.

Замечание 7. Уравнения (18) и (19) всегда нелинейны из-за наличия слагаемых вида $\mu_{\rho-e_h} q_\nu(\alpha)$.

Замечание 8. В практических задачах для полиномиальных функций $\varphi(y_t,t), \psi(y_t,t)$ и $p_\nu(y_t), q_\nu(y_t)$ непосредственно используются символьные вычисления [3].

8 Пример

Для одномерной нелинейной круговой системы

$$\dot{Y} + \varphi(Y) = V\,, \qquad (20)$$

($\varphi(Y)$ — скалярная нелинейная функция, V — круговой белый шум интенсивности G_t), уравнения МННА (14)–(16) имеют вид:

$$\dot{m}_t = -A_0(m_t,D_t);\;\; \dot{D}_t = -2k_1(m_t,D_t) + G_t\,;$$
$$\frac{\partial K(t_1,t_2)}{\partial t_2} = -k_1(m_t,D_t)K(t_1,t_2)\,,$$

где $A_0(m_t,D_t)$ и $k_1(m_t,D_t)$ — коэффициенты статистической линеаризации нелинейной функции $\varphi(Y)$ в (20) для «намотанного» нормального распределения с параметрами m_t и D_t.

При $N=4$ уравнения (18) и (19) имеют вид:

$$\dot{m}_t = $$
$$= A_0(m_t,D_t) + A_{13}(m_t,D_t)\mu_{3t} + A_{14}(m_t,D_t)\mu_{4t}\,;$$

$$\dot{D}_t = $$
$$= B_0(m_t,D_t) + B_{13}(m_t,D_t)\mu_{3t} + B_{14}(m_t,D_t)\mu_{4t}\,;$$

$$\dot{\mu}_{3t} = $$
$$= C_{30}(m_t,D_t) + C_{33}(m_t,D_t)\mu_{3t} + C_{34}(m_t,D_t)\mu_{4t}\,;$$

$$\dot{\mu}_{4t} = $$
$$= C_{40}(m_t,D_t) + C_{41}(m_t,D_t)\mu_{3t} + C_{44}(m_t,D_t)\mu_{4t} -$$
$$- 4A_{13}(m_t,D_t)\mu_{3t}^2 - A_{14}(m_t,D_t)\mu_{3t}\mu_{4t}\,.$$

Для различных нелинейных функций $\varphi(y)$ эти уравнения позволяют оценить точность МННА.

9 Заключение

Разработана прикладная теория анализа одно- и многомерных распределений в нелинейных многоканальных круговых стохастических системах на основе ортогональных разложений плотностей.

Для кругового эталонного «намотанного» нормального распределения описанное математическое обеспечение положено в основу инструментального программного обеспечения «CStS-ANALYSIS» в среде MATLAB.

В настоящее время в ИПИ РАН ведутся работы по созданию математического обеспечения для других эталонных распределений.

Литература

1. *Синицын И. Н.* Канонические разложения случайных функций и их применение в стохастических информационных технологиях научных исследований: Курс лекций // Распознавание образов и анализ изображений: новые информационные технологии. РОАИ-10-2010: Мат-лы 1-й Междунар. конф. — СПб., 2010.
2. *Синицын И. Н.* Стохастические информационные технологии для исследования нелинейных круговых стохастических систем // Информатика и её применения, 2011. Т. 5. Вып. 4. С. 2–5.
3. *Синицын И. Н., Корепанов Э. Р., Белоусов В. В. и др.* Развитие компьютерной поддержки статистических научных исследований систем высокой точности и доступности // Системы и средства информатики, 2011. Вып. 21. № 1. С. 7–37.
4. *Sinitsyn I. N., Belousov V. V., Konashenkova T. D.* Software tools for circular stochastic systems analysis // 29th Seminar (International) on Stability Problems for Stochastic

Models and 5th Workshop «Applied Problems in Theory of Probabilities and Mathematical Statistics Related to Modeling of Information Systems» (APTP + MS'2011): Book on Abstracts. — M.: IPIRAS, 2011. P. 86–87.

5. *Босов А. В., Будзко В. И., Захаров В. Н., Козмидиади В. А., Корепанов Э. Р., Синицын И. Н., Шоргин С. Я., Ушмаев О. С.* Информатика: состояние, проблемы, перспективы / Под. ред. И. А. Соколова. — М.: ИПИ РАН, 2009.
6. *Пугачев В. С., Синицын И. Н.* Стохастические дифференциальные системы. Анализ и фильтрация. — 2-е изд. доп. — М.: Наука, 1990.
7. *Пугачев В. С., Синицын И. Н.* Теория стохастических систем. — 2-е изд. — М.: Логос, 2004.
8. *Синицын И. Н.* Канонические представления случайных функций и их применения в задачах компьютерной поддержки научных исследований. — М.: ТОРУС ПРЕСС, 2009.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 19–26

ЗАДАЧИ АНАЛИЗА И ОПТИМИЗАЦИИ ДЛЯ МОДЕЛИ ПОЛЬЗОВАТЕЛЬСКОЙ АКТИВНОСТИ. ЧАСТЬ 2. ОПТИМИЗАЦИЯ ВНУТРЕННИХ РЕСУРСОВ

А. В. Босов[1]

Аннотация: Продолжено исследование математической модели описания активности пользователей, предложенной автором ранее. Сформулирована и решена задача оптимизации распределения «внутренних» ресурсов, используемых информационной системой, на основе квадратичного критерия качества. Предложены субоптимальные алгоритмы оптимизации.

Ключевые слова: информационная система; стохастическая система наблюдения; квадратичный критерий

1 Введение

В работе [1] предложена и исследована математическая модель описания пользовательской активности в некоторой информационной системе. Модель реализована в форме стохастической динамической системы наблюдения специального вида: ненаблюдаемое состояние x_t описывает число пользователей, формирующих запросы к узлу информационной системы, косвенные наблюдения y_t предполагаются линейными функциями состояния и задают число заданий, выполненных узлом в течение заданного промежутка времени. С описанием характера показателя пользовательской активности связано понятие текущего режима. Предполагается, что выделено определенное число характерных для изучаемого процесса x_t режимов, каждый из которых задается определенным диапазоном значений и изменяется при выходе x_t за границы текущего диапазона. Для соответствующей системы наблюдения решены задачи анализа, и на примере конкретного программного обеспечения — Информационного веб-портала [2] — проиллюстрированы, во-первых, физический смысл параметров модели, во-вторых, работоспособность процедур оценивания.

Традиционное применение такого рода моделей, в основном, заключается в разных формах анализа процессов в изучаемой системе, например с целью выявления «узких» мест, предельных распределений и пр. Однако не менее перспективными представляются варианты применения моделей функционирования для оптимизации работы элементов информационной системы и, прежде всего, компонентов ее программного обеспечения. Примеры подобной оптимизации хорошо известны. Это и управление страничным файлом операционной системы, и оптимизация запросов реляционной системы управления базами данных, и алгоритмы диспетчеризации задач в многопроцессорных средах, и многие другие. Относительно этих примеров можно отметить, что существенного применения математического аппарата в таких задачах немного, более привычными оказываются слабо формализуемые эвристические подходы. В качестве удачного примера обратного можно указать на класс задач, связанных с управляемым протоколом TCP (см., например, [3–5]), однако в целом подобных приложений немного.

Одной из причин такой ситуации, по-видимому, являются трудности не столько в моделировании конкретных процессов функционирования, сколько в корректной постановке задач оптимизации, включая выделение оптимизируемых характеристик, управляющих воздействий и критериев.

В данной работе применительно к вычислительным ресурсам, используемым узлом информационной системы, удается преодолеть указанные трудности, в том числе благодаря наличию модели пользовательской активности. При этом в интересах подтверждения прикладной значимости ассоциация рассмотренной далее оптимизационной постановки с портальной технологией будет сохранена, хотя результат, очевидно, допускает и более общую трактовку.

2 Используемые обозначения

Далее в работе будут использованы следующие обозначения:

$\triangleq$ — равенство по определению;

[1]Институт проблем информатики Российской академии наук, AVBosov@ipiran.ru

$\mathbf{M}[x]$ и $\mathbf{M}[x|\mathfrak{J}]$ — соответственно безусловное математическое ожидание случайной величины x и условное математическое ожидание x относительно σ-алгебры $\mathfrak{J}$;

x^{T} — операция транспонирования вектора (матрицы) x;

$\mathrm{col}\,(x_1,\ldots,x_n) \triangleq (x_1,\ldots,x_n)^{\mathrm{T}}$ — вектор-столбец с элементами $x_1,\ldots,x_n$;

$\mathrm{row}\,(x_1,\ldots,x_n) \triangleq (x_1,\ldots,x_n)$ — вектор-строка с элементами $x_1,\ldots,x_n$;

$\mathfrak{J}_t^y \triangleq \sigma\{y_\tau, \tau \le t\}$ — σ-алгебра, порожденная наблюдениями y_τ, $\tau \le t$;

$\overline{\psi}_x(x,t,j)$, $j = 1,2,\ldots,$ — условная плотность вероятности x_{t+j} относительно σ-алгебры $\mathfrak{J}_t^y$;

$\hat{\psi}_x(x,t)$ — условная плотность вероятности x_t относительно $\mathfrak{J}_t^y$.

3 Модель распределения «внутренних» ресурсов портала

Любая программная система, и портал в частности, при реализации собственной функциональности задействует различные вычислительные ресурсы. Программы в процессе работы используют ресурсы операционной системы, технической платформы, телекоммуникационной инфраструктуры. Принципиально возможно сформировать полный перечень такого рода ресурсов и даже как-то их классифицировать. Однако конструктивно влиять на их выделение, как правило, оказывается невозможным. Объясняется это, в основном, тем, что ресурсы программами запрашиваются у внешних (обслуживающих) систем, управлять которыми программы обычно не могут. Таким образом, предлагать формулировки задач оптимизации расходования ресурсов следует, прежде всего, для каких-либо «внутренних» ресурсов. К последним надо относить те программные объекты, которые создаются (временно используются) программами при росте нагрузки и освобождаются при ее уменьшении.

В работе Информационного веб-портала такой «внутренний» ресурс присутствует. Это так называемый пул запросов (подробнее см. [6]). Для параллельного выполнения поступающих запросов в составе подсистемы интеграции и поиска портала выделен компонент исполнения запросов. Этот компонент поддерживает заданное (фиксированное) число очередей. Каждая пользовательская команда преобразуется в набор запросов, которые распределяются по этим очередям. Для каждой очереди заранее создана программная нить, в которой и исполняются последовательно запросы из очереди. Такое решение ограничивает, с одной стороны, число одновременно работающих нитей, не позволяя системе «зависнуть», с другой — при поступлении сложных команд, порождающих большое число запросов, позволяет им исполняться параллельно.

Совокупность очередей с нитями, названная пулом запросов, и является тем самым «внутренним» ресурсом, возможностям оптимизации которого и посвящена данная статья. «Внутренним» пул назван потому, что его управление полностью зависит от программных приложений портала и никак не контролируется внешними обслуживающими системами. Принципиальные технические проблемы в обеспечении возможности динамического изменения размера пула отсутствуют, однако и программирование такой функциональности, и цена вычислительных затрат на ее реализацию существенно высоки, что требует, соответственно, строго обоснованного подхода к данной проблеме.

Требования к размеру пула определяет текущая пользовательская активность, исследование модели которой проведено в [1]. Показатель пользовательской активности x_t за интервал наблюдения $(t-1;t]$ описывается разностным стохастическим уравнением

$$x_t = a\theta(x_{t-1})x_{t-1} + q\Theta(x_{t-1}) + \\ + b\Theta(x_{t-1})v_t\,,\ \ t = 1,2,\ldots, \quad (1)$$

в предположении, что область значений x_t разбита на непересекающиеся интервалы Δ_k:

$$-\infty = a_1 < a_2 < \cdots < a_n < a_{n+1} = +\infty\,,$$
$$\Delta_k = (a_k, a_{k+1}],\ k = 1,\ldots,n-1,\ \Delta_n = (a_n, +\infty)$$

и текущий режим пользовательской активности задан индикаторной функцией $\Theta(x)$:

$$\Theta(x) = \mathrm{col}\,(I_{\Delta_1}(x),\ldots,I_{\Delta_n}(x))\ ;$$
$$I_{\Delta_k}(x) = \begin{cases} 1\,, & \text{если } x \in \Delta_k\,; \\ 0\,, & \text{если } x \notin \Delta_k\,; \end{cases} \quad (2)$$
$$a = \mathrm{row}\,(a_1,\ldots,a_n)\ ;\ q = \mathrm{row}\,(q_1,\ldots,q_n)\ ;$$
$$b = \mathrm{row}\,(b_1,\ldots,b_n)\ .$$

В качестве наблюдений за x_t, как и в [1], будем использовать число команд y_t, выполненных порталом за интервал наблюдения $(t-1;t]$:

$$y_t = c^y x_t + \sigma^y w_t^y\,. \quad (3)$$

Здесь параметр c^y определяет среднее число команд, направленных одним пользователем для

выполнения порталом; w_t^y — возмущение, характеризующее отклонение числа команд от среднего уровня; σ^y — среднее квадратическое отклонение этого возмущения (далее предполагается, что $\{w_t^y\}$ — стандартный дискретный белый шум второго порядка).

Кроме того, учтем, что на требуемый размер пула влияет не столько число выполняемых команд, сколько число сформированных из них запросов, поэтому в дополнение к наблюдениям (3) будем рассматривать и число выполненных на интервале $(t-1;t]$ запросов z_t, предполагая его пропорциональным числу команд:

$$z_t = c^z y_t + \sigma^z w_t^z \,. \tag{4}$$

Здесь параметр c^z определяет среднее число запросов к источникам, формируемых из одной команды; w_t^z — возмущение, характеризующее отклонение числа запросов от среднего уровня; σ^z — среднее квадратическое отклонение этого возмущения (далее предполагается, что $\{w_t^z\}$ — стандартный дискретный белый шум второго порядка).

Перейдем далее к формированию критерия оптимизации, т. е. к постановке задачи определения текущего размера пула запросов.

Пусть известно среднее время T выполнения одного запроса. Для максимального удовлетворения потребностей пользователей текущий размер пула $u = u_t$ следует положить равным $(l(t,t+1)/T)z_{t+1}$, где $l(t,t+1)$ — длина интервала наблюдения $(t,t+1]$. Проблема, однако, состоит в том, что на момент t принятия решения об изменении размера пула точное значение z_{t+1} еще не известно, поэтому можно было бы использовать некоторый прогноз $\overline{z}_{t+1}$. Таким образом, первый компонент целевой функции в задаче управления размером пула можно представить в виде штрафа за ошибку прогнозирования

$$\mathbf{M}\left[\left(z_{t+1} - \overline{z}_{t+1}\right)^2\right] . \tag{5}$$

Целевая функция (5) исходит из потребностей только одной (пользовательской) стороны, поскольку не включает штраф за выработанное управляющее воздействие. Если пользоваться стратегией оптимизации вида

$$u_t = \frac{l(t,t+1)}{T}\,\overline{z}_{t+1}\,, \tag{6}$$

то на каждом шаге наблюдений размер пула с большой вероятностью будет меняться, но изменения эти будут незначительны, а значит, и не скажутся принципиально на эффективности работы портала в целом, а только усложнят реализуемое управление. Кроме того, такое определение размера пула приведет к его потенциально бесконечному росту с увеличением текущего числа пользователей и в конечном итоге — к исчерпанию всех ресурсов обслуживающих систем. Усовершенствовать целевую функцию можно путем дополнения (5) штрафными аддитивными слагаемыми, вначале для учета штрафа за изменение размера пула:

$$\mathbf{M}\left[\left(z_{t+1} - G_t u_t\right)^2 + \left(u_t - u_{t-1}\right)^2\right] , \tag{7}$$

а затем ввести в (7) «плату» и за общий размер пула:

$$\mathbf{M}\left[\left(z_{t+1} - G_t u_t\right)^2 + \left(u_t - u_{t-1}\right)^2 + u_t^2\right] . \tag{8}$$

В (7) и (8) весовые коэффициенты G_t уместно задать, исходя из «пользовательского» варианта управления (6), т. е. в виде $G_t = T/l(t,t+1)$.

Для придания (8) окончательного вида предположим, что задан горизонт оптимизации N (например, сутки или неделя) и дополним выбранные слагаемые весовыми коэффициентами. Окончательно получаем целевую функцию следующего вида:

$$\mathbf{J}(u_0,\ldots,u_N) = \sum_{t=0}^{N} \mathbf{M}\left[S_t^{(1)}\left(z_{t+1} - G_t u_t\right)^2 + S_t^{(2)}\left(u_t - u_{t-1}\right)^2 + S_t^{(3)} u_t^2\right] . \tag{9}$$

4 Формальная постановка и решение задачи управления размером пула

Всюду далее будем предполагать, что $\{v_t\}$ из (1) — стандартный дискретный белый шум в узком смысле, сечения которого имеют плотность вероятности $\varphi_v(\cdot)$; x_0 — случайная величина, имеющая плотность вероятности $\psi_0(\cdot)$; $\{w_t^y\}$ из (3) — стандартный дискретный белый шум в узком смысле, сечения которого имеют плотность вероятности $\varphi_w^y(\cdot)$; $\{w_t^z\}$ из (4) — стандартный дискретный белый шум в узком смысле, сечения которого имеют плотность вероятности $\varphi_w^z(\cdot)$; $\{v_t\}$, x_0, $\{w_t^y\}$, $\{w_t^z\}$ независимы в совокупности; $\mathbf{M}[v_t^2] < \infty$; $\mathbf{M}[x_0^2] < \infty$; $\mathbf{M}[(w_t^y)^2] < \infty$; $\mathbf{M}(w_t^y)^2] < \infty$; параметры b_k, $k = 1,\ldots,n$, σ^y и σ^z неотрицательны.

Будем предполагать также, что параметры целевого функционала (9) $S_t^{(1)}$, $S_t^{(2)}$, $S_t^{(3)}$, G_t — известные неотрицательные функции t; класс допустимых управлений U_t содержит все $\mathfrak{J}_t^y$-измеримые функции u_t : $\mathbf{M}[u_t^2] < \infty$. Таким образом, целью оптимизации является поиск закона изменения размера пула u_t^*, удовлетворяющего потребности в ресурсах, описываемых выходом z_{t+1}, с минимизацией

затрат на управляющее воздействие, на изменение управляющего воздействия на текущем шаге и на всех последующих шагах вплоть до заданного горизонта N:

$$\operatorname{col}(u_0^*,\ldots,u_N^*) = \underset{(u_0,\ldots,u_N)\in U_0\cdots U_N}{\arg\min} \mathbf{J}(u_0,\ldots,u_N). \quad (10)$$

Теорема. *Пусть для целевого функционала* (9) *выполнено:* $S_t^{(2)} > 0{,}1 \le t \le N$, $S_0^{(2)} = 0$. *Тогда решение* u_t^* *задачи оптимизации* (10) *существует и определяется соотношениями:*

$$\left.\begin{aligned} u_t^* &= \frac{1}{R(t)}\left(S_t^{(2)}u_{t-1}^* + \sum_{j=1}^{N-t+1} Q_j(t)\overline{z}_{t+j,t}\right);\\ R(t) &= S_t^{(2)} + S_t^{(3)} + S_t^{(1)}G_t^2 + \\ &\quad + S_{t+1}^{(2)} - \frac{(S_{t+1}^{(2)})^2}{R(t+1)}, \quad 0 \le t < N;\\ R(N) &= S_N^{(2)} + S_N^{(3)} + S_N^{(1)}G_N^2;\\ Q_j(t) &= \frac{S_{t+1}^{(2)}}{R(t+1)}\,Q_{j-1}(t+1);\\ Q_1(t) &= S_t^{(1)}G_t, \quad 0 \le t < N,\\ &\qquad 1 \le j \le N-t+1;\\ Q_1(N) &= S_N^{(1)}G_N, \end{aligned}\right\} \quad (11)$$

где $\overline{z}_{t+j,t}$ *— оптимальные в среднем квадратическом прогнозы выхода* z_{t+j} *по наблюдениям* y_τ, $\tau \le t$.

Доказательство. Для решения задачи оптимизации (10) воспользуемся методом динамического программирования [7, 8]. Обозначим через

$$B(t) \triangleq \triangleq \min_{(u_t,\ldots,u_N)\in U_t\cdots U_N} \sum_{\tau=t}^{N} \mathbf{M}\left[S_\tau^{(1)}(z_{\tau+1} - G_\tau u_\tau)^2 + S_\tau^{(2)}(u_\tau - u_{\tau-1})^2 + S_\tau^{(3)}u_\tau^2 | \mathfrak{J}_t^y\right]$$

функцию Беллмана. При $t = N$ утверждение теоремы очевидно, так как из выражения

$$B(N) = \min_{u_N\in U_N} \mathbf{M}\left[S_N^{(1)}(z_{N+1} - G_N u_N)^2 + S_N^{(2)}(u_N - u_{N-1})^2 + S_N^{(3)}u_N^2 \,\middle|\, \mathfrak{J}_N^y\right]$$

после очевидных преобразований с учетом $\mathfrak{J}_N^y$-измеримости функций u_N и u_{N-1}, а также обозначения $\overline{z}_{N+1,N} = \mathbf{M}[z_{N+1}|\mathfrak{J}_N^y]$ получаем:

$$u_N^* = \underset{u_N\in U_N}{\arg\min}\left(\left(S_N^{(2)} + S_N^{(3)} + S_N^{(1)}G_N^2\right)u_N^2 - 2\left(S_N^{(2)}u_{N-1} + S_N^{(1)}G_t\overline{z}_{N+1,N}\right)u_N + S_N^{(2)}u_{N-1}^2 + S_N^{(1)}\mathbf{M}\left[z_{N+1}^2|\mathfrak{J}_N^y\right]\right),$$

откуда с учетом независимости последних двух слагаемых от u_N и положительности коэффициента при u_N^2 получаем:

$$u_N^* = \frac{S_N^{(2)}u_{N-1} + S_N^{(1)}G_N^2\overline{z}_{N+1,N}}{S_N^{(2)} + S_N^{(3)} + S_N^{(1)}G_N^2} = \frac{1}{R(N)}\left(S_N^{(2)}u_{N-1} + Q_1(N)\overline{z}_{N+1,N}\right).$$

Кроме того, получаем и выражение для функции Беллмана:

$$B(N) = S_N^{(2)}u_{N-1}^2 - \frac{1}{R(N)}\left(S_N^{(2)}u_{N-1} + Q_1(N)\overline{z}_{N+1,N}\right)^2 + \mathbf{M}[A(N)|\mathfrak{J}_N^y],$$

где обозначено $A(N) = S_N^{(1)}z_{N+1}^2$.

Предположим, что утверждение теоремы выполнено для u_{t+1}^* и для функции Беллмана $B(t+1)$ имеет место следующее выражение:

$$B(t+1) = S_{t+1}^{(2)}u_t^2 - \frac{1}{R(t+1)}\left(S_{t+1}^{(2)}u_t + \sum_{j=1}^{N-t} Q_j(t+1)\overline{z}_{t+j+1,t+1}\right)^2 + \mathbf{M}\left[A(t+1)|\mathfrak{J}_{t+1}^y\right];$$

$$A(t+1) = A(t+2) + S_{t+1}^{(1)}z_{t+2}^2 - \frac{1}{R(t+2)}\left(\sum_{j=1}^{N-t-1} Q_j(t+2)\overline{z}_{t+j+2,t+2}\right)^2.$$

Тогда уравнение Беллмана для $B(t)$ имеет следующий вид:

$$B(t) = \min_{u_t\in U_t}\mathbf{M}\left[S_t^{(1)}\left(y_{t+1}^{(2)} - G_t u_t\right)^2 + S_t^{(2)}(u_t - u_{t-1})^2 + S_t^{(3)}u_t^2 + S_{t+1}^{(2)}u_t^2 - \frac{\left(S_{t+1}^{(2)}u_t + \sum_{j=1}^{N-t} Q_j(t+1)\overline{z}_{t+j+1,t+1}\right)^2}{R(t+1)} + \mathbf{M}[A(t+1)|\mathfrak{J}_{t+1}^y]\,\middle|\,\mathfrak{J}_t^y\right].$$

Преобразовав это уравнение с учетом $\mathfrak{J}_t^y$-измеримости функций u_{t-1} и u_t, запишем:

$$B(t) = \min_{u_t \in U_t} \Bigg[\Bigg(S_t^{(1)} G_t + S_t^{(2)} + S_t^{(3)} + S_{t+1}^{(2)} - \frac{\left(S_{t+1}^{(2)}\right)^2}{R(t+1)} \Bigg) u_t^2 - 2 \Bigg(S_t^{(1)} G_t + S_t^{(2)} u_{t-1} + \frac{S_{t+1}^{(2)}}{R(t+1)} \times \sum_{j=1}^{N-t} Q_j(t+1) \mathbf{M}[\overline{z}_{t+j+1,t+1} | \mathfrak{J}_t^y] \Bigg) u_t + S_t^{(2)} u_{t-1}^2 + \mathbf{M} \Bigg[S_t^{(2)} z_{t+1}^2 - \frac{1}{R(t+1)} \left(\sum_{j=1}^{N-t} Q_j(t+1) \overline{z}_{t+j+1,t+1} \right)^2 + \mathbf{M}[A(t+1) | \mathfrak{J}_{t+1}^y] | \mathfrak{J}_t^y \Bigg] \Bigg] .$$

Применив в последнем выражении формулу полного математического ожидания и введенные в (11) обозначения, получим:

$$B(t) = \min_{u_t \in U_t} \Big[R(t) u_t^2 - 2 \Bigg(S_t^{(2)} u_{t-1} + \sum_{j=1}^{N-t+1} Q_j(t) \overline{z}_{t+j,t} \Bigg) u_t + S_t^{(2)} u_{t-1}^2 + \mathbf{M} \left[A(t) | \mathfrak{J}_t^y \right] \Big] .$$

Поскольку в полученном соотношении два последних слагаемых не зависят от u_t, то в предположении положительности $R(t)$ получается доказываемое соотношение (11) для u_t^*. Кроме того, подстановкой u_t^* подтверждается справедливость индуктивного предположения относительно функции Беллмана.

Для завершения доказательства остается показать, что $R(t) > 0$. Для этого достаточно показать, что $S_{t+1}^{(2)} - \left(S_{t+1}^{(2)}\right)^2 \Big/ R(t+1) \geq 0$. В свою очередь, последнее неравенство верно, если $S_{t+2}^{(2)} - \left(S_{t+2}^{(2)}\right)^2 \Big/ R(t+2) \geq 0$ и $R(t+1) > 0$. Выполнение, таким образом, индуктивного предположения вытекает из того, что $R(N) > 0$ и $S_N^{(2)} - \left(S_N^{(2)}\right)^2 \Big/ R(N) \geq 0$. Теорема доказана.

Замечание. В полученном утверждении используются оптимальные в среднем квадратическом прогнозы $\overline{z}_{t+j,t}$ выхода z_{t+j} по наблюдениям y_τ, $\tau \leq t$, $j = 1, 2, \ldots$ В теореме 2 работы [1] получены аналогичные прогнозы для y_{t+j}. Учитывая тривиальность уравнения (4), нетрудно увидеть, что соответствующие соотношения для $\overline{z}_{t+j,t}$ имеют вид:

$$\overline{z}_{t+j,t} = \sum_{k=1}^{n} \int_{\Delta_k} c^y c^z (a_k \xi + q_k) \overline{\psi}_x(\xi, t, j) \, d\xi \, , \quad j = 2, 3, \ldots ;$$

$$\overline{z}_{t+1,t} = \sum_{k=1}^{n} \int_{\Delta_k} c^y c^z (a_k \xi + q_k) \hat{\psi}_x(\xi, t) \, d\xi \, ,$$

где прогнозирующие плотности вероятности определяются соотношениями:

$$\overline{\psi}_x(x, t, j) = \sum_{k=1}^{n} \frac{1}{b_k} \int_{\Delta_k} \overline{\psi}_x(\xi, t, j-1) \times \varphi_v \left(\frac{x - a_k \xi - q_k}{b_k} \right) d\xi \, , \; j = 2, 3, \ldots ;$$

$$\overline{\psi}_x(x, t, 1) = \sum_{k=1}^{n} \frac{1}{b_k} \int_{\Delta_k} \hat{\psi}_x(\xi, t) \varphi_v \left(\frac{x - a_k \xi - q_k}{b_k} \right) d\xi ;$$

$$\hat{\psi}_x(x, t) = \Bigg(\varphi_{w^y} \left(\frac{y_t - c^y x}{\sigma^y} \right) \sum_{k=1}^{n} \frac{1}{b_k} \times \int_{\Delta_k} \hat{\psi}_x(\xi, t-1) \varphi_v \left(\frac{x - a_k \xi - q_k}{b_k} \right) d\xi \Bigg) \Bigg/ \Bigg(\sum_{k=1}^{n} \frac{1}{b_k} \times \int_{R^1} \varphi_{w^y} \left(\frac{y_t - c^y x}{\sigma^y} \right) \int_{\Delta_k} \hat{\psi}_x(\xi, t-1) \times \varphi_v \left(\frac{x - a_k \xi - q_k}{b_k} \right) d\xi \, dx \Bigg) .$$

5 Субоптимальные управления

Основной проблемой применения оптимальной стратегии оптимизации размера пула (11) является определение горизонта N.

Нетрудно видеть, что выбор больших значений N приводит к необходимости вычисления большого числа прогнозов, особенно на первых шагах алгоритма. Расчет же прогнозов является весьма вычислительно затратным. Малые же значения N не обеспечат должного учета динамического характера задачи, модельная и априорная информация окажутся фактически невостребованными.

Для преодоления данного противоречия предлагается использовать принцип локально-оптимального (адаптивного) управления [9]. Сохранив целевую функцию в виде (9), определим в качестве субоптимального решения задачи ее минимизации функцию u_t^L, доставляющую минимум функционалу

$$\mathbf{J}_t(u_t) = \mathbf{M}\Big[S_t^{(1)}(z_{t+1} - G_t u_t)^2 + \\ + S_t^{(2)}(u_t - u_{t-1})^2 + S_t^{(3)} u_t^2 + \\ + S_{t+1}^{(1)}(z_{t+2} - G_{t+1}u_{t+1})^2 + S_{t+1}^{(2)}(u_{t+1} - u_t)^2 + \\ + S_{t+1}^{(3)} u_{t+1}^2\Big] .$$

Таким образом, для локально-оптимального решения рассматриваемой задачи оптимизации предлагается двухшаговый вариант целевой функции вида (9), обновляемый на каждом следующем шаге алгоритма.

В целевую функцию $\mathbf{J}_t(u_t)$, как легко видеть, включены штрафы за ошибку прогнозирования, за изменения размера пула и за собственно размер пула на текущем и следующем шаге.

Отметим, что использование локально-оптимальных критериев управления в качестве подходящей альтернативы интегральным критериям показало свою эффективность в традиционных задачах управления состоянием стохастической динамической системы [10, 11].

Требующееся выражение для функции $u_t^L = \arg\min_{u_t \in U_t} \mathbf{J}_t(u_t)$ получаем непосредственно как частный случай (11):

$$u_t^L = \left(S_t^{(2)} u_{t-1}^L + S_t^{(1)} G_t \overline{z}_{t+1,t} + \right. \\ \left. + \frac{S_{t+1}^{(1)} S_{t+1}^{(2)} G_{t+1}}{S_{t+1}^{(2)} + S_{t+1}^{(3)} + S_t^{(1)} G_{t+1}^2} \overline{z}_{t+2,t} \right) \Bigg/ \left(S_t^{(2)} + S_t^{(3)} + \right. \\ \left. + S_t^{(1)} G_t^2 + S_{t+1}^{(2)} - \frac{\left(S_{t+1}^{(2)}\right)^2}{S_{t+1}^{(2)} + S_{t+1}^{(3)} + S_{t+1}^{(1)} G_{t+1}^2} \right) . \quad (12)$$

Наконец, самым простым вариантом возможного решения рассматриваемой задачи оптимизации является использование оптимальной программной стратегии u_t^P. Такое «усредненное» решение можно записать в виде:

$$u_t^P = \mathbf{M}[u_t^*] . \quad (13)$$

6 Результаты численных экспериментов

Для сравнения предложенных алгоритмов оптимизации использован модельный пример из [1], незначительно измененный для учета дополнительного уравнения (4) выхода (так, чтобы приведенные в [1] результаты прогнозирования сохранялись и здесь). Были заданы три интервала $\Delta_1 = (-\infty; 3]$, $\Delta_2 = (3; 7]$, $\Delta_3 = (7; +\infty)$ и следующие параметры уравнения (1):

a_1	a_2	a_3	q_1	q_2	q_3	b_1	b_2	b_3
0,3	0,4	0,7	1,4	3,0	3,0	0,9	1,5	2,5

Параметры наблюдений (3): $c^y = 1{,}0$, $\sigma^y = 3{,}0$, параметры выхода (4): $c^z = 3{,}0$, $\sigma^z = 1{,}0$. Распределения всех возмущений — стандартные гауссовские, распределение начального условия x_0 также предполагалось гауссовским со средним и дисперсией, равными соответствующим моментам предельного распределения (подробнее см. [1]).

Расчеты проводились для 10 шагов траектории: $t = 1, 2, \ldots, 10$, для вычисления значения целевой функции использовался пучок из 10 000 траекторий. Параметры целевой функции (9) выбраны следующим образом:

$$S_t^{(2)} = 0{,}1\,, \;\; 1 \le t \le 10\,; \\ S_t^{(1)} = S_t^{(3)} = 0{,}1\,, \;\; 0 \le t \le 10\,; \\ G_0 = 0{,}0\,; \;\; G_1 = G_2 = G_3 = 0{,}1\,; \\ G_4 = G_5 = G_6 = 1{,}0\,; \; G_7 = G_8 = G_9 = G_{10} = 4{,}0\,.$$

Для сравнения качества стратегий оптимизации (11)–(13) вычислялось значение целевой функции в каждый момент t, т. е. $\mathbf{J}(u_0, \ldots, u_t)$.

Результаты расчетов приведены на рис. 1 и 2. Рисунок 1 иллюстрирует характерные траектории компонентов системы наблюдения (1)–(4) и функций u_t^*, u_t^L и u_t^P. На рис. 2 приведены значения целевой функции: $\hat{J}_1$ (для u_t^*), $\hat{J}_2$ (для u_t^L) и $\hat{J}$ (для u_t^P).

Из приведенных результатов видно, что значения целевой функции, достигаемые при использовании оптимального решения u_t^* и локально-оптимального u_t^L, практически совпадают.

Преимущество алгоритма, использующего стратегию u_t^*, в сравнении с u_t^L в терминах целевой функции (9) составляет не более 1,5% к моменту достижения горизонта оптимизации. При этом очевидно, что вычислительные трудности, возникающие в связи с необходимостью расчета большого числа прогнозов, весьма велики, и

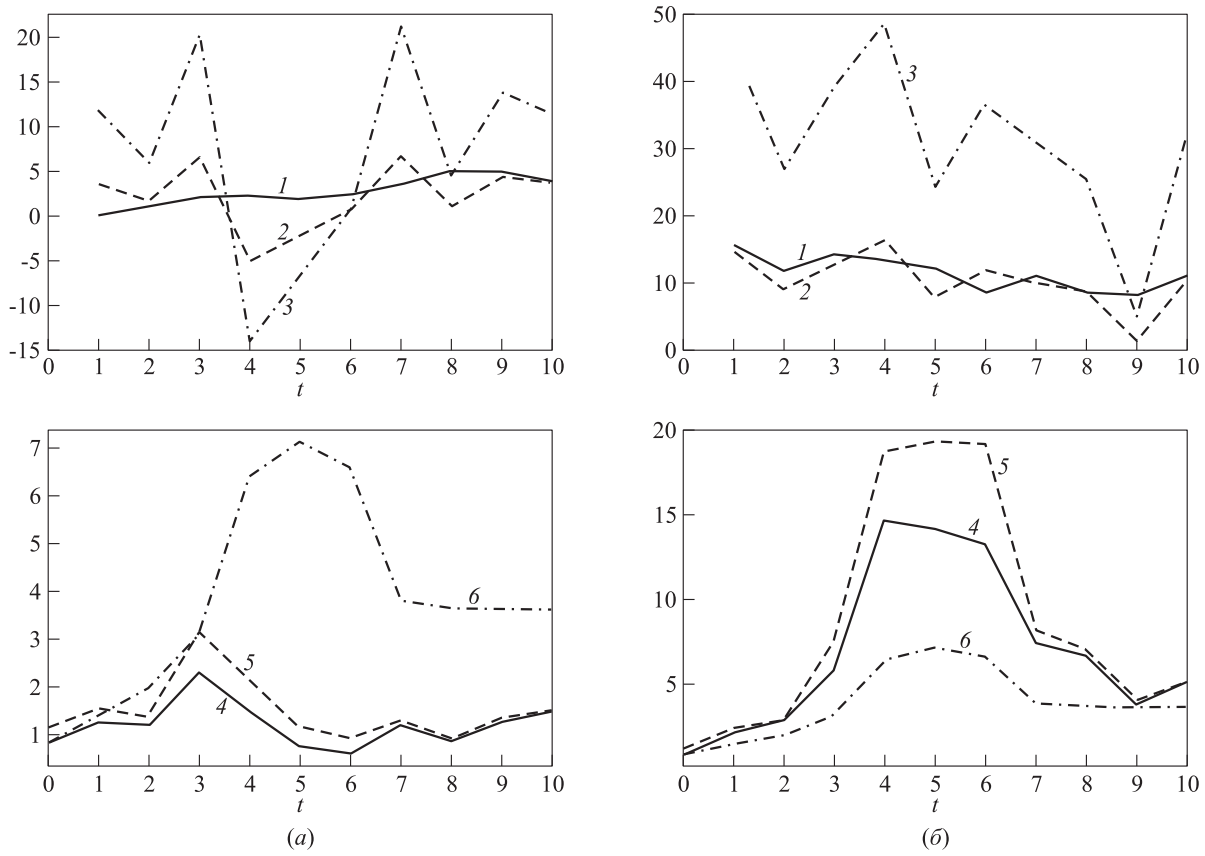


Рис. 1 Характерные траектории: *1* — x_t; *2* — y_t; *3* — z_t; *4* — u_t^*; *5* — u_t^L; *6* — u_t^P

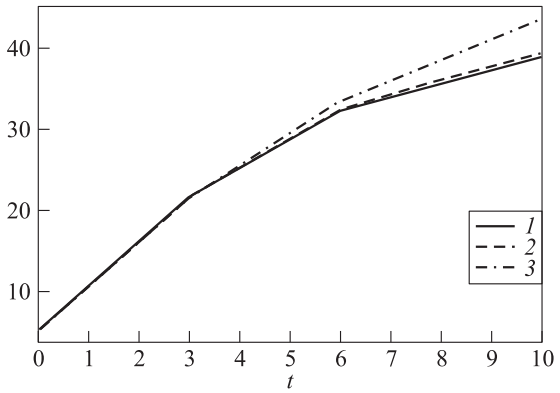


Рис. 2 Оценки целевых функций: *1* — $\hat{J}_1$ (для u_t^*); *2* — $\hat{J}_2$ (для u_t^L); *3* — $\hat{J}$ (для u_t^P)

можно считать, таким образом, что наиболее целесообразно использование локально-оптимального решения. Также можно отметить, что программное («усредненное») решение хотя и проигрывает оптимальным алгоритмам, но сохраняет приемлемое качество. При этом следует указать на тенденцию роста преимущества оптимальных решений с удалением горизонта оптимизации, т. е. с ростом N. Таким образом, практическое применение u_t^P в этой задаче вряд ли целесообразно, так как на практике интерес представляют расчеты с большим N.

7 Заключение

В статье продолжено исследование модели пользовательской активности, предложенной в [1]. Стохастическая динамическая система наблюдения (1)–(3), описывающая эволюцию числа пользователей, взаимодействующих с некоторой информационной системой, и косвенные наблюдения за ними — число поступивших для выполнения команд, расширено уравнением выхода (4), описывающим число сформированных из поступивших команд запросов. С текущим значением выхода связана величина, описывающая объем выделяемых системой вычислительных ресурсов. В рассмотренной прикладной постановке это размер пула, состоящего из множества очередей, поддерживаемых с целью параллельного выполнения

формируемых запросов. В соответствии с предложенной терминологией такой ресурс назван «внутренним», так как его обслуживание полностью контролируется самой программной системой.

Основная решенная задача состоит в оптимизации в процессе работы программы размера пула на основе квадратичного критерия качества, позволяющего учесть пользовательские потребности, расходы на управление пулом и его размер. В целом использованный подход вполне соответствует классической задаче динамического управления по квадратичному критерию качества, в том числе и по использованному аппарату динамического программирования. Предложенная постановка, однако, имеет существенное отличие: управляющее воздействие не влияет на фазовый процесс (текущее число пользователей) и связано с последним только косвенно, через критерий оптимизации. Такой результат вполне согласуется с физическим смыслом рассматриваемой задачи — определить наилучший алгоритм расходования вычислительных ресурсов на основании анализа состояния окружающей среды.

Аналогичный подход в последующих работах предполагается реализовать и для «внешних» ресурсов, выделяемых программе обслуживающими системами.

Литература

1. *Босов А. В.* Задачи анализа и оптимизации для модели пользовательской активности. Часть 1. Анализ и прогнозирование // Информатика и её применения, 2011. Вып. 4. Т. 5. С. 40–52.
2. Информационный Веб-портал: Свидетельство об официальной регистрации программы для ЭВМ № 2005612992 от 18.11.2005.
3. *Kelly F. P., Maulloo A., Tan D.* Rate control in communication networks: Shadow prices, proportional fairness and stability // J. Operational Research Soc., 1998. Vol. 49. P. 237–252.
4. *Low S. H., Paganini F., Doyle J. C.* Internet congestion control // IEEE Control Syst. Magazine, 2002. Vol. 22. No. 1. P. 28–43.
5. *Миллер Б. М., Миллер Г. Б., Семенихин К. В.* Методы синтеза оптимального управления марковским процессом с конечным множеством состояний при наличии ограничений // Автоматика и телемеханика, 2011. № 2. С. 111–130.
6. *Босов А. В.* Моделирование и оптимизация процессов функционирования Информационного веб-портала // Программирование, 2009. № 6. С. 53–66.
7. *Флеминг У., Ришел Р.* Оптимальное управление детерминированными и стохастическими системами. — М.: Мир, 1978.
8. *Бертсекас Д., Шрив С.* Стохастическое оптимальное управление. — М.: Наука, 1985.
9. *Коган М. М., Неймарк Ю. И.* Адаптивное локально-оптимальное управление // Автоматика и телемеханика, 1987. № 8. С. 126–136.
10. *Босов А. В., Панков А. Р.* Алгоритмы управления в системах с переключающимися каналами наблюдения // Изв. РАН. Сер. Теория и системы управления, 1996. № 2. С. 98–103.
11. *Босов А. В., Панков А. Р.* Алгоритмы управления для дискретных систем случайной структуры // Автоматика и телемеханика, 1997. № 10. С. 113–125.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 27–31

О ВИРТУАЛЬНОМ ВРЕМЕНИ ОЖИДАНИЯ В СИСТЕМЕ С ОТНОСИТЕЛЬНЫМ ПРИОРИТЕТОМ И ГИПЕРЭКПОНЕНЦИАЛЬНЫМ ВХОДЯЩИМ ПОТОКОМ*

А. В. Ушаков[1]

Аннотация: Найдены преобразования Лапласа–Стилтьеса виртуальных времен ожидания в одноканальной системе обслуживания с относительным приоритетом и рекуррентным входящим потоком с гиперэкспоненциальным распределением интервалов между поступлениями требований.

Ключевые слова: виртуальное время ожидания; относительный приоритет; гиперэкспоненциальный поток

1 Введение

При проектировании и анализе функционирования инфотелекоммуникационных систем в качестве математических моделей наиболее часто используются системы и сети массового обслуживания. С учетом сложности таких систем возникает необходимость разработки математических методов анализа систем обслуживания, учитывающих многие факторы: ненадежность каналов, нетерпеливость требований, наличие требований различной важности и т. д.

В работах [1–4] разработаны методы анализа одноканальных систем массового обслуживания с приоритетами, не допускающими прерывания уже начатого обслуживания (относительный приоритет, чередование приоритетов) и различными классами рекуррентных входящих потоков. Исследовано поведение одной из наиболее важных характеристик — вектора длин очередей из требований различных приоритетов. В настоящей статье методы [1–4] применены к анализу другой, не менее важной, характеристики — времени ожидания начала обслуживания.

2 Описание системы

Рассматривается одноканальная система обслуживания с r, $r \geqslant 1$, приоритетными классами требований. Длительности обслуживания — независимые в совокупности и не зависящие от входящего потока случайные величины с функцией распределения $B_i(x)$ для требований i-го класса.

Входящий поток требований — рекуррентный, определяемый плотностью распределения интервалов между поступлениями требований вида

$$a(x) = \begin{cases} \sum\limits_{j=1}^{N} c_j a_j \exp\left(-a_j x\right), & x \geqslant 0\,; \\ 0\,, & x < 0\,, \end{cases} \qquad (1)$$

где $a_i \neq a_j$ при $i \neq j$, $c_j > 0$, $\sum\limits_{i=1}^{N} c_i = 1$.

Поступившее требование направляется в i-й приоритетный класс с вероятностью p_i, $i = 1, \ldots, r$, независимо от остальных требований. Рекуррентный входящий поток, задаваемый плотностью распределения (1), эквивалентен следующему: интервалы времени между поступлениями требований независимы в совокупности и показательно распределены со случайным параметром a, принимающим значения a_i с вероятностями c_i, $i = 1, \ldots, N$, причем значение a определяется непосредственно перед началом отсчета времени до следующего поступления и не меняется между двумя поступлениями. Событие $\{j(t) = j\}$ будет означать, что $a = a_j$ в момент времени t.

Будем предполагать, что требования из класса с меньшим номером имеют относительный приоритет перед требованиями из класса с бóльшим номером. Для требований одного класса будут рассмотрены две дисциплины обслуживания: прямой порядок обслуживания (дисциплина FIFO — first in, first out) и инверсионный порядок обслуживания (дисциплина LIFO — last in, first out). Пусть, кроме того, в начальный момент $t = 0$ система свободна от требований.

*Работа выполнена при финансовой поддержке РФФИ (грант 11-07-00112а).

[1]Институт проблем информатики Российской академии наук, grimgnau@rambler.ru

3 Основные обозначения и определения

Как уже было указано выше, функцию распределения времени обслуживания требований из i-го приоритетного класса будем обозначать $B_i(x)$. Пусть, далее, $b_i(x)$, $\beta_i(s)$ и β_{ij} — соответственно плотность распределения, преобразование Лапласа–Стилтьеса и j-й момент случайной величины с функцией распределения $B_i(x)$.

Введем следующие случайные процессы:

– $w_i^{(0)}(t)$, $i = 1, \ldots, N$, — виртуальное время ожидания в момент времени t для требований i-го приоритетного класса при дисциплине FIFO при условии, что после t требования в систему не поступают; $w_0^{(0)}(t)$ — время с момента t до завершения обслуживания требования, находящегося в этот момент на приборе (если в момент t система свободна, то $w_0^{(0)}(t) = 0$);

– $w_i^{(1)}(t)$ и $w_i^{(2)}(t)$, $i = 1, \ldots, N$, — виртуальные времена ожидания для требований i-го приоритетного класса в момент времени t при дисциплинах FIFO и LIFO соответственно.

Положим

$$W_{ij}^{(k)}(s,t) = \int_0^\infty e^{-sy}\, d_y \mathbf{P}(w_i^{(k)}(t) < y, j(t) = j)\,;$$

$$\omega_{ij}^{(k)}(s,v) = \int_0^\infty e^{-vt} W_{ij}^{(k)}(s,t)\, dt,$$

$$i = 0, \ldots, r\,,\ j = 1, \ldots, N\,,\ k = 0, 1, 2\,;$$

$P_{0j}(t)$ — вероятность того, что в момент времени t система свободна и $j(t) = j$; $P_{mj}(t)\Delta + o(\Delta)$ — вероятность того, что в интервале времени $[t, t+\Delta)$ началось обслуживание требования m-го приоритетного класса и $j(t) = j$;

$$p_{0j}(v) = \int_0^\infty e^{-vt} P_{0j}(t)\, dt\,;$$

$$p_{mj}(v) = \int_0^\infty e^{-vt} P_{mj}(t)\, dt\,.$$

4 Вспомогательные результаты

Функции $W_{ij}^{(0)}(s,t)$ удовлетворяют системе дифференциальных уравнений

$$\frac{\partial W_{ij}^{(0)}(s,t)}{\partial t} = (s - a_j) W_{ij}^{(0)}(s,t) - sP_{0j}(t) - \sum_{m=i+1}^{r} (1 - \beta_m(s))\, P_{mj}(t) + c_j \sum_{k=1}^{N} a_k W_{ik}^{(0)}(s,t) \left(\sum_{m=1}^{i} p_m \beta_m(s) + \sum_{m=i+1}^{r} p_m \right), \quad i = 0, \ldots, r\,, \quad (2)$$

с начальным условием $W_{ij}^{(0)}(s,0) = c_j$.

Переходя в (2) к преобразованиям Лапласа, получаем:

$$\omega_{ij}^{(0)}(s,v) = \frac{c_j}{v - s + a_j} - \frac{s}{v - s + a_j} p_{0j}(v) - \sum_{m=i+1}^{r} \frac{1 - \beta_m(s)}{v - s + a_j}\, p_{mj}(v) + \frac{c_j}{v - s + a_j} \times \sum_{k=1}^{N} a_k \omega_{ik}^{(0)}(s,v) \left(\sum_{m=1}^{i} p_m \beta_m(s) + \sum_{m=i+1}^{r} p_m \right). \quad (3)$$

Умножая (3) на a_j и суммируя по j от 1 до N, находим:

$$\left(1 - \sum_{j=1}^{N} \frac{c_j a_j}{v - s + a_j} \left(\sum_{m=1}^{i} p_m \beta_m(s) + \sum_{m=i+1}^{r} p_m \right) \right) \sum_{k=1}^{N} a_k\, \omega_{ik}^{(0)}(s,v) = \sum_{j=1}^{N} \frac{c_j a_j}{v - s + a_j} - s \sum_{j=1}^{N} \frac{a_j p_{0j}(v)}{v - s + a_j} - \sum_{m=i+1}^{r} (1 - \beta_m(s)) \sum_{j=1}^{N} \frac{a_j p_{mj}(v)}{v - s + a_j}\,. \quad (4)$$

В первую очередь соотношения (4) используем для нахождения неизвестных функций $p_{0j}(v)$ и $p_{mj}(v)$.

Справедлива следующая лемма:

Лемма 1. *При каждом $i = 0, \ldots, r$ функциональное уравнение*

$$\sum_{j=1}^{N} \frac{c_j a_j}{v - s + a_j} \left(\sum_{m=1}^{i} p_m \beta_m(s) + \sum_{m=i+1}^{r} p_m \right) = 1$$

имеет N решений $s = \alpha_{li}(v)$, $l = 1, \ldots, N$, аналитических в области $\operatorname{Re} v > 0$.

Рассмотрим (4) при $i = r$:

$$\left(1 - \sum_{j=1}^{N} \frac{c_j a_j}{v - s + a_j} \sum_{m=1}^{r} p_m \beta_m(s)\right) \sum_{k=1}^{N} a_k \omega_{ik}^{(0)}(s,v) = \sum_{j=1}^{N} \frac{c_j a_j}{v - s + a_j} - s \sum_{j=1}^{N} \frac{a_j p_{0j}(v)}{v - s + a_j}\,.$$

В силу леммы 1 левая часть последнего соотношения обращается в нуль при $s = \alpha_{lr}(v)$, $l = 1, \ldots, N$. Отсюда получаем систему линейных уравнений для определения функций $p_{0j}(v)$:

$$\sum_{j=1}^{N} \frac{a_j\, p_{0j}(v)}{v - \alpha_{lr}(v) + a_j} = \alpha_{lr}^{-1}(v) \sum_{j=1}^{N} \frac{c_j a_j}{v - \alpha_{lr}(v) + a_j},$$

из которой находим:

$$a_k p_{0k}(v) = \sum_{l=1}^{N} \sum_{\nu=1}^{N} \frac{c_\nu a_\nu}{\alpha_{lr}(v)(a_\nu + v - \alpha_{lr}(v))} \times \times \frac{\prod_{j=1}^{N} ((a_k + v - \alpha_{jr}(v))(a_j + v - \alpha_{lr}(v)))}{(a_k + v - \alpha_{lr}(v)) \prod_{n \neq l} (\alpha_{nr}(v) - \alpha_{lr}(v)) \prod_{i \neq k} (a_k - a_i)}, \quad k = 1, \ldots, N\,.$$

Подставляя $s = \alpha_{li}(v)$ в (4) последовательно при $i = r - 1, r - 2, \ldots, 1, 0$, получаем системы линейных уравнений

$$\sum_{j=1}^{N} \frac{a_j p_{i+1\,j}(v)}{v - \alpha_{li}(v) + a_j} = f_{li}(v)\,,$$

где

$$f_{li}(v) = (1 - \beta_{i+1}(\alpha_{li}))^{-1} \left(\sum_{j=1}^{N} \frac{c_j a_j}{v - \alpha_{li}(v) + a_j} - \right.$$
$$- \alpha_{li}(v) \sum_{j=1}^{N} \frac{a_j p_{0j}(v)}{v - \alpha_{li}(v) + a_j} -$$
$$\left. - \sum_{m=i+2}^{r} (1 - \beta_m(\alpha_{li}(v))) \sum_{j=1}^{N} \frac{a_j p_{mj}(v)}{v - \alpha_{li}(v) + a_j} \right),$$

из которых находим систему рекуррентных соотношений для определения $p_{mj}(v)$, $m = r, r - 1, \ldots, 1$:

$$a_k p_{i+1k} = \sum_{l=1}^{N} f_{li}(v) \times \times \frac{\prod_{j=1}^{N} ((a_k + v - \alpha_{ji}(v))(a_j + v - \alpha_{li}(v)))}{(a_k + v - \alpha_{li}(v))} \times \times \prod_{n \neq l} (\alpha_{ni}(v) - \alpha_{li}(v))^{-1} \prod_{i \neq k} (a_k - a_i)^{-1}\,, \quad k = 1, \ldots, N\,,\ i = r - 1, \ldots, 0\,.$$

После того как найдены все функции $p_{0j}(v)$ и $p_{mj}(v)$, $m = 1, \ldots, r$, $j = 1, \ldots, N$, из (4) находим $\sum_{k=1}^{N} a_k \omega_{ik}^{(0)}(s,v)$ и из (3) $\omega_{ij}^{(0)}(s,v)$ для всех $i = 0, 1, \ldots, r$, $j = 1, \ldots, N$.

В дальнейшем понадобится совместная производящая функция числа требований всех приоритетных классов, поступивших в систему за заданное время. Обозначим через $n(t) = (n_1(t), \ldots, n_r(t))$ число требований приоритетов $1, \ldots, r$, поступивших в интервале $[0, t)$,

$$q_{\nu j}(z,t) = \mathbf{E}\left(z_1^{n_1(t)} \cdots z_r^{n_r(t)} I(j(t) = j) | j(0) = \nu\right), \quad z = (z_1, \ldots, z_r)\,.$$

Функции $q_{\nu j}(z,t)$ удовлетворяют следующей системе дифференциальных уравнений:

$$\frac{\partial q_{\nu j}(z,t)}{\partial t} = -a_j q_{\nu j}(z,t) + c_j(p,z) \sum_{k=1}^{N} a_k q_{\nu k}(z,t) \quad (5)$$

с начальным условием $q_{\nu j}(z,0) = \delta_{\nu,j}$, где $\delta_{\nu,j} = 1$ при $\nu = j$ и $\delta_{\nu,j} = 0$ при $\nu \neq j$, $(p,z) = \sum_{i=1}^{r} p_i z_i$.

Решение (5) имеет вид:

$$q_{\nu j}(z,t) = c_j a_\nu (p,z) \sum_{k=1}^{N} \frac{\prod_{i \neq \nu} (\mu_k(z) + a_i)}{\alpha_k(z)(\mu_k(z) + a_j)}\, e^{\mu_k(z) t}\,, \quad (6)$$

где $\alpha_k(z) = \prod_{i \neq k} (\mu_k(z) - \mu_i(z))$, а $\mu_1(z), \ldots, \mu_N(z)$ — корни многочлена

$$\prod_{i=1}^{N} (\mu + a_i) - (p,z) \sum_{j=1}^{N} c_j a_j \prod_{i \neq j} (\mu + a_i)\,.$$

И, наконец, понадобятся распределения различных промежутков занятости исследуемой системы. Пусть $\Pi^{(i)}\left(n_1^{(0)}, \ldots, n_i^{(0)}\right)$ — длительность периода занятости обслуживанием требований приоритетов $1, \ldots, i$, начавшегося с $n_1^{(0)}, \ldots, n_i^{(0)}$ требований этих приоритетов, т. е. случайного интервала времени, начинающегося с обслуживания

одного из $n_1^{(0)},\dots,n_i^{(0)}$ требований и кончающегося в момент первого после этого освобождения системы от требований приоритетов $1,\dots,i$. Обозначим

$$\Pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},t\right)\Delta+o(\Delta)=$$
$$=\mathbf{P}\left(\Pi^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)}\right)\in\right.$$
$$\left.\in[t,t+\Delta),j(t)=j|j(0)=\nu\right);$$
$$\pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)=$$
$$=\int_0^\infty e^{-st}\Pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},t\right)dt;$$
$$z_{ij}^{(k)}(s)=\beta_j(\alpha_{ki}(s));$$
$$\mu_{ki}^*(s)=\mu_k\left(z_{i1}^{(k)}(s),\dots,z_{ii}^{(k)}(s),1,\dots,1\right).$$

Аналогично [2] показывается, что функции $\pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)$ удовлетворяют системе линейных уравнений:

$$\sum_{j=1}^N\frac{a_j}{\mu_{ki}^*(s)+a_j}\pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)=$$
$$=\frac{a_\nu}{\mu_{ki}^*(s)+a_\nu}\prod_{l=1}^i(z_{il}^{(k)}(s))^{n_l^{(0)}},\ \ k=1,\dots,N.$$

Отсюда вытекают следующие леммы:

Лемма 2. *Функции* $\pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)$ *определяются по формулам:*

$$a_k\pi_{k\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)=$$
$$=\sum_{l=1}^N\frac{a_\nu}{\mu_{li}^*(s)+a_\nu}\prod_{m=1}^i(z_{im}^{(l)}(s))^{n_m^{(0)}}\times$$
$$\times\frac{\prod_{j=1}^N((a_k+\mu_{ji}^*(s))(a_j+\mu_{ki}^*(s)))}{(a_k+\mu_{li}^*(s))\prod_{n\neq l}(\mu_{li}^*(s)-\mu_{ni}^*(s))\prod_{q\neq k}(a_k-a_q)}.$$

Лемма 3. *Функции* $\pi_\nu^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)=\sum_{j=1}^N\pi_{j\nu}^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)$ *определяются по формулам:*

$$\pi_\nu^{(i)}\left(n_1^{(0)},\dots,n_i^{(0)},s\right)=\sum_{k=1}^N\prod_{m=1}^i(z_{im}^{(k)}(s))^{n_m^{(0)}}\times$$
$$\times\prod_{l\neq\nu}\frac{\mu_{ki}^*(s)+a_l}{a_l}\prod_{j\neq k}\frac{\mu_{ji}^*(s)}{\mu_{ji}^*(s)-\mu_{ki}^*(s)}. \quad (7)$$

5 Основные результаты

Теперь займемся изучением безусловных виртуальных времен ожидания при дисциплинах FIFO и LIFO. Во-первых, заметим, что виртуальное время ожидания для некоторого приоритетного класса не зависит от дисциплины обслуживания, принятой для других классов.

Сначала рассмотрим дисциплину FIFO. Очевидно, $W_{1j}^{(1)}(s,t)=W_{1j}^{(0)}(s,t)$. При $i\geqslant 1$ имеем

$$W_{i+1j}^{(1)}(s,t)=\sum_{\nu=1}^N\sum_{n_1=0}^\infty\dots$$
$$\dots\sum_{n_i=0}^\infty\int_0^\infty e^{-sy}\mathbf{P}(\text{в интервале времени }[t,t+y)$$
$$\text{поступило }n_k,\ k=1,\dots,i,\ \text{требований}$$
$$\text{приоритета }k,\ j(t+y)=\nu|j(t)=j)\times$$
$$\times\pi_\nu^{(i)}(n_1,\dots,n_i,s)\,d_y\mathbf{P}(w_{i+1}^{(0)}(t)<y,j(t)=j).$$

Подставляя $\pi_\nu^{(i)}(n_1,\dots,n_i,s)$ из (7), имеем:

$$W_{i+1j}^{(1)}(s,t)=\sum_{\nu=1}^N\sum_{n_1=0}^\infty\dots$$
$$\dots\sum_{n_i=0}^\infty\int_0^\infty e^{-sy}\mathbf{P}(\text{в интервале времени }[t,t+y)$$
$$\text{поступило }n_k,\ k=1,\dots,i,\ \text{требований}$$
$$\text{приоритета }k,\ j(t+y)=\nu|j(t)=j)\times$$
$$\times\sum_{k=1}^N\prod_{m=1}^i(z_{im}^{(k)}(s))^{n_m}\prod_{l\neq\nu}\frac{\mu_{ki}^*(s)+a_l}{a_l}\times$$
$$\times\prod_{p\neq k}\frac{\mu_{pi}^*(s)}{\mu_{pi}^*(s)-\mu_{ki}^*(s)}\,d_y\mathbf{P}(w_{i+1}^{(0)}(t)<y,j(t)=j).$$

Отсюда и из (6) получаем:

$$W_{i+1j}^{(1)}(s,t)=\sum_{\nu=1}^N\int_0^\infty e^{-sy}\sum_{k=1}^N\prod_{l\neq\nu}\frac{\mu_{ki}^*(s)+a_l}{a_l}\times$$
$$\times\prod_{p\neq k}\frac{\mu_{pi}^*(s)}{\mu_{pi}^*(s)-\mu_{ki}^*(s)}\,c_\nu a_j\times$$
$$\times\sum_{c=1}^N\frac{\prod_{q\neq j}(\mu_{ci}^{(k)}(s)+a_q)}{(\mu_{ci}^{(k)}(s)+a_\nu)\alpha_c(z_{i1}^{(k)}(s),\dots,z_{ii}^{(k)}(s),1,\dots,1)}\times$$
$$\times\left(\sum_{m=1}^i p_m z_{im}^{(k)}(s)+\sum_{m=i+1}^r p_m\right)\exp\left(\mu_{ci}^{(k)}(s)y\right)\,d_y\times$$
$$\times\mathbf{P}(w_{i+1}^{(0)}(t)<y,j(t)=j),$$

где $\mu_{ci}^{(k)}(s)=\mu_c(z_{i1}^{(k)}(s),\dots,z_{ii}^{(k)}(s),1,\dots,1)$.

Отсюда вытекает справедливость следующей теоремы:

Теорема 1. *При дисциплине FIFO* $W_{1j}^{(1)}(s,t) = W_{1j}^{(0)}(s,t)$ *и для* $i \geqslant 1$

$$W_{i+1j}^{(1)}(s,t) =$$

$$= \sum_{\nu=1}^{N}\sum_{k=1}^{N}\prod_{l\neq\nu}\frac{\mu_{ki}^{*}(s)+a_l}{a_l}\prod_{p\neq k}\frac{\mu_{pi}^{*}(s)}{\mu_{pi}^{*}(s)-\mu_{ki}^{*}(s)}\,c_\nu a_j \times$$

$$\times\sum_{c=1}^{N}\frac{\prod\limits_{q\neq j}(\mu_{ci}^{(k)}(s)+a_q)}{(\mu_{ci}^{(k)}(s)+a_\nu)\alpha_c(z_{i1}^{(k)}(s),\ldots,z_{ii}^{(k)}(s),1,\ldots,1)}\times$$

$$\times\left(\sum_{m=1}^{i}p_m z_{im}^{(k)}(s)+\sum_{m=i+1}^{r}p_m\right)\times$$

$$\times W_{i+1j}^{(0)}\left(s-\mu_{ci}^{(k)}(s),t\right).$$

При дисциплине LIFO связь между условными и безусловными виртуальными временами ожидания несколько другая:

$$W_{ij}^{(2)}(s,t)=\sum_{\nu=1}^{N}\sum_{n_1=0}^{\infty}\ldots$$

$$\ldots\sum_{n_i=0}^{\infty}\int_0^{\infty}e^{-sy}\mathbf{P}(\text{в интервале времени }[t,t+y)$$

$$\text{поступило } n_k,\ k=1,\ldots,i,\ \text{требований}$$

$$\text{приоритета } k, j(t+y)=\nu|j(t)=j)\times$$

$$\times\,\pi_\nu^{(i)}\,(n_1,\ldots,n_i,s)\;d_y\mathbf{P}(w_{i-1}^{(0)}(t)<y,j(t)=j)\,.$$

Отсюда вытекает

Теорема 2. *При дисциплине LIFO при всех* $i=1,\ldots,r$

$$W_{ij}^{(2)}(s,t)=\sum_{\nu=1}^{N}\sum_{k=1}^{N}\prod_{l\neq\nu}\frac{\mu_{ki}^{*}(s)+a_l}{a_l}\times$$

$$\times\prod_{p\neq k}\frac{\mu_{pi}^{*}(s)}{\mu_{pi}^{*}(s)-\mu_{ki}^{*}(s)}\,c_\nu a_j\times$$

$$\times\sum_{c=1}^{N}\frac{\prod\limits_{q\neq j}(\mu_{ci}^{(k)}(s)+a_q)}{(\mu_{ci}^{(k)}(s)+a_\nu)\alpha_c(z_{i1}^{(k)}(s),\ldots,z_{ii}^{(k)}(s),1,\ldots,1)}\times$$

$$\times\left(\sum_{m=1}^{i}p_m z_{im}^{(k)}(s)+\sum_{m=i+1}^{r}p_m\right)\times$$

$$\times\,W_{i-1j}^{(0)}\left(s-\mu_{ci}^{(k)}(s),t\right).$$

Литература

1. *Ушаков В. Г.* Система обслуживания с эрланговским входящим потоком и относительным приоритетом // Теория вероятности и ее примениния, 1977. Т. 22. С. 860–866.
2. *Матвеев В. Ф., Ушаков В. Г.* Системы массового обслуживания. — М.: МГУ, 1984.
3. *Ушаков В. Г.* Аналитические методы анализа системы массового обслуживания $GI|G_r|1|\infty$ с относительным приоритетом // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. и киберн., 1993. № 4. С. 57–69.
4. *Ушаков В. Г.* О длине очереди в однолинейной системе массового обслуживания с чередованием приоритетов // Вестн. Моск. ун-та. Сер. 15. Вычисл. матем. и киберн., 1994. № 2. С. 29–36.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 32–36

УТОЧНЕНИЕ НЕРАВНОМЕРНЫХ ОЦЕНОК СКОРОСТИ СХОДИМОСТИ В ЦЕНТРАЛЬНОЙ ПРЕДЕЛЬНОЙ ТЕОРЕМЕ ПРИ СУЩЕСТВОВАНИИ МОМЕНТОВ НЕ ВЫШЕ ВТОРОГО

С. В. Попов[1]

Аннотация: В статье уточняются неравномерные оценки скорости сходимости в центральной предельной теореме для сумм независимых случайных величин, у которых существуют моменты не выше второго.

Ключевые слова: центральная предельная теорема; оценка скорости сходимости; абсолютные константы

1 Введение

Многие исследователи отмечают, что при статистическом анализе тех или иных характеристик трафика в информационных системах возникают вероятностные распределения со столь тяжелыми хвостами, что можно предполагать наличие моментов лишь второго порядка. Как известно, к таким распределениям применима центральная предельная теорема, однако скорость вытекающей из нее сходимости к нормальному закону в таком случае может быть как угодно медленной и определяется поведением хвостов распределений исходных величин. В данной работе строятся неравномерные оценки скорости сходимости в центральной предельной теореме в терминах «квадратичных хвостов».

Пусть $X_1, X_2, \ldots$ — независимые случайные величины с $\mathsf{E}X_i = 0$ и $\mathsf{E}X_i^2 < \infty$. Для $n \in \mathbb{N}$ положим $W_n = X_1 + \cdots + X_n$. Предположим, что $\mathsf{D}W_n = \mathsf{E}X_1^2 + \cdots + \mathsf{E}X_n^2 = 1$. Пусть $\Phi(x)$ – стандартная нормальная функция распределения,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{x} e^{-z^2/2}\, dz\,, \quad x \in \mathbb{R}\,.$$

Для $x \in \mathbb{R}$ обозначим

$$\Delta_x = |\mathsf{P}(W_n < x) - \Phi(x)|\,.$$

В работе [1] показано, что существует положительная конечная постоянная C такая, что для любого $x \in \mathbb{R}$ выполнено неравенство:

$$\Delta_x \leqslant C \sum_{i=1}^{n} \left[\frac{\mathsf{E}X_i^2 \mathbb{I}(|X_i| \geqslant 1 + |x|)}{(1+|x|)^2} + \right.$$
$$\left. + \frac{\mathsf{E}|X_i|^3 \mathbb{I}(|X_i| < 1 + |x|)}{(1+|x|)^3} \right]. \quad (1)$$

В некоторых работах предпринимались попытки оценить значение константы C в неравенстве (1). Прежде всего в связи с этой задачей необходимо упомянуть недавние работы [2, 3]. Из результатов последней из указанных статей вытекает наилучшая (насколько известно автору) на сегодняшний день верхняя оценка константы $C \leqslant 76{,}17$.

Также необходимо отметить работу [4], в которой для случая одинаково распределенных слагаемых показано, что существуют положительные функции $C(x)$, $C_2(x)$ и $C_3(x)$ такие, что

$$\Delta_x \leqslant \frac{C_2(x)}{(1+|x|)^2} \sum_{i=1}^{n} \mathsf{E}X_i^2 \mathbb{I}(|X_i| \geqslant 1 + |x|) +$$
$$+ \frac{C_3(x)}{(1+|x|)^3} \sum_{i=1}^{n} \mathsf{E}|X_i|^3 \mathbb{I}(|X_i| < 1 + |x|)\,;$$

$$\Delta_x \leqslant C(x) \sum_{i=1}^{n} \left[\frac{\mathsf{E}X_i^2 \mathbb{I}(|X_i| \geqslant 1 + |x|)}{(1+|x|)^2} + \right.$$
$$\left. + \frac{\mathsf{E}|X_i|^3 \mathbb{I}(|X_i| < 1 + |x|)}{(1+|x|)^3} \right],$$

причем $C_2(x) \leqslant 14{,}262$, $C_3(x) \leqslant 41{,}229$, $C(x) \leqslant 39{,}317$, откуда вытекает, что в случае одинаково распределенных слагаемых неравенство (1) справедливо с $C \leqslant 39{,}317$.

В данной работе два последних приведенных неравенства рассматриваются для необязательно одинаково распределенных слагаемых. Будет показано, что в общем случае эти неравенства справедливы с константами $C_2(x) \leqslant 14{,}532$, $C_3(x) \leqslant 49{,}468$ и $C(x) \leqslant 47{,}648$.

[1]Факультет вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова, popovserg@yandex.ru

2 Основные результаты

Без потери общности достаточно ограничиться рассмотрением случая $x \geqslant 0$.

Для $x \geqslant 0$ и $n \in \mathbb{N}$ обозначим:

$$Y_{i,x} = X_i \mathbb{I}(|X_i| < 1+x)\,; \quad S_x = \sum_{i=1}^{n} Y_{i,x}\,;$$

$$\alpha_x = \sum_{i=1}^{n} \mathsf{E} X_i^2 \mathbb{I}(|X_i| \geqslant 1+x)\,;$$

$$\beta_x = \sum_{i=1}^{n} \mathsf{E}|X_i|^3 \mathbb{I}(|X_i| < 1+x)\,;$$

$$\overline{Y}_{i,x} = \frac{Y_{i,x} - \mathsf{E}Y_{i,x}}{\sqrt{\mathsf{D}S_x}}\,; \quad \overline{S}_x = \sum_{i=1}^{n} \overline{Y}_{i,x}\,;$$

$$\overline{\Delta}_x = \left| \mathsf{P}\left(\overline{S}_x \leqslant \frac{x - \mathsf{E}S_x}{\sqrt{\mathsf{D}S_x}} \right) - \Phi\left(\frac{x - \mathsf{E}S_x}{\sqrt{\mathsf{D}S_x}} \right) \right|.$$

Лемма 1. *Если $\alpha_x \leqslant A$ для некоторого $A \in (0,\, 1/2)$, то для любого $q \in [0, 1]$ справедливы неравенства:*

$$\overline{\Delta}_x \leqslant 0{,}5600 \left[\frac{4q(1+x)}{(1-2A)^{3/2}} \frac{\alpha_x}{(1+x)^2} + \right.$$
$$\left. + \frac{(K+q-Kq)(1+x)^3}{(1-2A)^{3/2}} \frac{\beta_x}{(1+x)^3} \right]; \quad (2)$$

$$\overline{\Delta}_x \leqslant 22{,}2460 \left[\frac{4s(x,A)}{(1-2A)^{3/2}(1+x)^2} \frac{\alpha_x}{(1+x)^2} + \right.$$
$$\left. + \frac{s(x,A)}{(1-2A)^{3/2}} \frac{\beta_x}{(1+x)^3} \right], \quad (3)$$

где

$$s(x,A) = \frac{(1+x)^3}{1 + (x - A/(1+x))^3}\,. \quad (4)$$

Доказательство. В статье [4] для одинаково распределенных слагаемых показаны неравенства:

$$\sum_{i=1}^{n} \mathsf{E}|\overline{Y}_{i,x}|^3 \leqslant$$
$$\leqslant \frac{1}{(1-2A)^{3/2}} \left((K+q-Kq)\beta_x + \frac{4q\alpha_x}{1+x} \right); \quad (5)$$

$$x - \frac{A}{1+x} \leqslant \frac{x - \mathsf{E}S_x}{\sqrt{\mathsf{D}S_x}} \leqslant \frac{x + A/(1+x)}{\sqrt{1-2A}}\,. \quad (6)$$

При этом отмечено (и в этом нетрудно убедиться), что эти неравенства вместе с доказательствами остаются справедливыми и для случая разнораспределенных слагаемых.

Поскольку $\overline{S}_x$ представляет собой сумму независимых случайных величин $\overline{Y}_{i,x}$ с $\mathsf{E}\overline{Y}_{i,x} = 0$, причем $\mathsf{D}\overline{S}_x = 1$, из неравенства Берри–Эссеена [5] с учетом (5) вытекает, что

$$|\mathsf{P}(\overline{S}_x < z) - \Phi(z)| \leqslant 0{,}5600 \sum_{i=1}^{n} \mathsf{E}|\overline{Y}_{i,x}|^3 \leqslant$$
$$\leqslant \frac{0{,}5600}{(1-2A)^{3/2}} \left((K+q-Kq)\beta_x + \frac{4q\alpha_x}{1+x} \right).$$

Последнее, очевидно, эквивалентно неравенству (2).

Применяя неравномерную оценку скорости сходимости в центральной предельной теореме, приведенную в работе [6], с учетом неравенств (5) и (6) получаем:

$$\overline{\Delta}_x \leqslant \frac{22{,}2417}{1 + \left((x - \mathsf{E}S_x)/\sqrt{\mathsf{D}S_x} \right)^3} \sum_{i=1}^{n} \mathsf{E}|\overline{Y}_{i,x}|^3 \leqslant$$
$$\leqslant \frac{22{,}2417}{1 + (x - A/(1+x))^3} \sum_{i=1}^{n} \mathsf{E}|\overline{Y}_{i,x}|^3 \leqslant$$
$$\leqslant \frac{22.2417 s(x,A)}{(1-2A)^{3/2}(1+x)^3} \left(\beta_x + \frac{4\alpha_x}{1+x} \right).$$

Последнее, очевидно, эквивалентно неравенству (3). Таким образом, лемма доказана.

Лемма 2. *Пусть $\alpha_x \leqslant A$ для некоторого $A \in (0,\, 1/2)$. Тогда*

$$\Delta_x \equiv |\mathsf{P}(W_n < x) - \Phi(x)| \leqslant \overline{\Delta}_x +$$
$$+ \frac{\alpha_x}{(1+x)^2} \left(D(x,A) + 1 \right),$$

где

$$D(x,A) = x(1+x)^2 e^{-x^2/2} \frac{B(A) e^A}{\sqrt{2\pi}} +$$
$$+ (1+x) e^{-x^2/2} \frac{(1 + AB(A))\, e^A}{\sqrt{2\pi}}. \quad (7)$$

Доказательство. См. лемму 4 в [4].

Терема 1. *Для любого $x \geqslant 0$ имеет место неравенство*

$$\Delta_x \leqslant C_2(x) \frac{\alpha_x}{(1+x)^2} + C_3(x) \frac{\beta_x}{(1+x)^3},$$

где $C_2(x)$ и $C_3(x)$ — положительные ограниченные функции, для которых справедлива каждая из следующих оценок:

1°. $C_2(x) \leqslant 2{,}0110(1+x)^2$; $C_3(x) \leqslant 2{,}0110(1+x)^3$.

2°. *Если* $\alpha_x \leqslant A$ *для некоторого* $A \in (0,\, 1/2)$, *то для любого* $q \in [0, 1]$

$$\text{(а) } C_2(x) \leqslant \frac{4 \cdot 0{,}5600q(1+x)}{(1-2A)^{3/2}} + D(x,A) + 1\,;$$
$$C_3(x) \leqslant \frac{0{,}5600(K+q-Kq)(1+x)^3}{(1-2A)^{3/2}}\,;$$
$$\text{(б) } C_2(x) \leqslant \frac{4 \cdot 22{,}2417s(x,A)}{(1-2A)^{3/2}(1+x)^2} + D(x,A) + 1\,;$$
$$C_3(x) \leqslant \frac{22{,}2417s(x,A)}{(1-2A)^{3/2}}\,,$$

где $s(x,A)$ *и* $D(x,A)$ *определены соответственно в* (4) *и* (7).

3°. *Если* $\alpha_x \geqslant A$ *для некоторого* $A \in (0,\, 1/2)$, *то*

$$\text{(а) } C_2(x) \leqslant \frac{0{,}541}{A}\,(1+x)^2\,;\quad C_3(x) = 0\,;$$
$$\text{(б) } C_2(x) \leqslant 1 + \frac{1}{x^4} + \frac{1}{x^4(1+x)^2} + \left[\frac{1}{x^4} + \frac{e^{-x^2/2}}{x\sqrt{2\pi}}\right]\frac{(1+x)^2}{A};$$
$$C_3(x) \leqslant \frac{(1+x)^4}{x^4} + \frac{(1+x)^2}{x^4}\,.$$

Доказательство пунктов 1° и 3° можно найти в [4] (см. теорему 1). Утверждение 2° следует непосредственно из лемм 1 и 2.

Теперь нетрудно понять, как следует использовать выводы теоремы 1 в численных расчетах. Для каждого $x > 0$ и $A \in (0,\, 1/2)$ следует определить наилучшую из оценок 2° (а) и 2° (б) и наилучшую из оценок 3° (а) и 3° (б). Далее при каждом x следует так выбирать параметр A, чтобы оптимизировать худшую из двух полученных оценок, которая и будет являться итоговой. Для улучшения итоговой оценки в некоторых случаях следует затем использовать оценку 1°.

В задаче поиска наилучшей оценки имеется два критерия качества: $C_2(x)$ и $C_3(x)$. Поэтому, как и в любой многокритериальной задаче, необходимо уделить внимание вопросу сравнения двух оценок. Сначала рассмотрим случай $q = 1$.

Нетрудно видеть, что при выполнении условия

$$\frac{22{,}2417}{1 + (x - A/(1+x))^3} > 0{,}5600 \qquad (8)$$

оценка 2° (а) точнее оценки 2° (б) (для каждой из функций $C_2(x)$ и $C_3(x)$). В противном случае следует предпочесть оценку 2° (б). Неравенство (8) можно преобразовать к виду:

$$\left(x - \frac{A}{1+x}\right)^3 < \frac{22{,}2417}{0{,}5600} - 1 = 38{,}7173\,.$$

Принимая во внимание область возможных значений A, можно сделать следующие выводы.

Вывод 1. При $x < 3{,}3830$ следует использовать оценку 2° (а). При $x > 3{,}4943$ следует использовать оценку 2° (б). В остальных случаях следует рассматривать обе оценки в зависимости от значения A.

Если выполнено условие

$$\frac{0{,}541}{A}\,(1+x)^2 \leqslant 1 + \frac{1}{x^4} + \frac{1}{x^4(1+x)^2} + \left[\frac{1}{x^4} + \frac{e^{-x^2/2}}{x\sqrt{2\pi}}\right]\frac{(1+x)^2}{A}\,, \qquad (9)$$

то оценка 3° (а), безусловно, точнее, чем 3° (б). Как показывают численные расчеты, это происходит при $x < 1{,}2593$ при любом допустимом A.

Неравенство, противоположное (9), выполняется опять же для всех допустимых значений A при $x > 1{,}3512$. В таком случае следует предпочесть оценку 3° (б), несмотря на то что оценка функции $C_3(x)$ становится грубее:

$$\sup_{x>1{,}2593} \frac{(1+x)^4}{x^4} + \frac{(1+x)^2}{x^4} \leqslant 12{,}39\,,$$

а эта величина меньше, чем значение функции $C_3(x)$ при применении оценок 2°.

Вывод 2. При $x < 1{,}2593$ следует использовать оценку 3° (а). При $x > 1{,}3512$ следует использовать оценку 3° (б). В остальных случаях следует рассматривать обе оценки в зависимости от значения A, но сравнение производить по значению функции $C_2(x)$.

Вычисления в пакете MatLab позволяют сформулировать следующий результат.

Теорема 2. *Для любого* $x \geqslant 0$ *имеет место неравенство*

$$\Delta_x \leqslant C_2(x)\frac{\alpha_x}{(1+x)^2} + C_3(x)\frac{\beta_x}{(1+x)^3}\,,$$

где $C_2(x)$ *и* $C_3(x)$ *— положительные ограниченные функции, для которых справедливы оценки, приведенные в табл.* 1.

Графики полученных оценок изображены на рис. 1.

Следствие 1. *Для любого* $x \geqslant 0$ *имеет место неравенство*

$$\Delta_x \leqslant 14{,}532\frac{\alpha_x}{(1+x)^2} + 49{,}468\frac{\beta_x}{(1+x)^3}\,.$$

Таблица 1 Верхние оценки функций $C_2(x)$ и $C_3(x)$

x	$C_2 \leqslant$	$C_3 \leqslant$	x	$C_2 \leqslant$	$C_3 \leqslant$	x	$C_2 \leqslant$	$C_3 \leqslant$
0	5,378	1,1308	3,3 ÷ 3,4	11,378	49,467	6,7 ÷ 6,8	3,2903	34,836
0,0 ÷ 0,1	5,9992	1,5163	3,4 ÷ 3,5	11,28	49,342	6,8 ÷ 6,9	3,2205	34,646
0,1 ÷ 0,2	6,6634	1,9843	3,5 ÷ 3,6	10,064	47,678	6,9 ÷ 7,0	3,1539	34,462
0,2 ÷ 0,3	7,3531	2,5522	3,6 ÷ 3,7	9,5333	46,915	7,0 ÷ 7,1	3,0902	34,284
0,3 ÷ 0,4	8,092	3,2345	3,7 ÷ 3,8	9,024	46,045	7,1 ÷ 7,2	3,0292	34,111
0,4 ÷ 0,5	8,8462	4,0295	3,8 ÷ 3,9	8,5783	45,367	7,2 ÷ 7,3	2,9709	33,944
0,5 ÷ 0,6	9,6318	4,9634	3,9 ÷ 4,0	8,1708	44,726	7,3 ÷ 7,4	2,915	33,781
0,6 ÷ 0,7	10,422	6,026	4,0 ÷ 4,1	7,7954	44,12	7,4 ÷ 7,5	2,8615	33,623
0,7 ÷ 0,8	11,233	7,2844	4,1 ÷ 4,2	7,4487	43,544	7,5 ÷ 7,6	2,8102	33,47
0,8 ÷ 0,9	12,077	8,6961	4,2 ÷ 4,3	7,128	42,998	7,6 ÷ 7,7	2,7609	33,321
0,9 ÷ 1,0	12,884	10,357	4,3 ÷ 4,4	6,8306	42,48	7,7 ÷ 7,8	2,7136	33,176
1,0 ÷ 1,1	13,741	12,257	4,4 ÷ 4,5	6,5543	41,986	7,8 ÷ 7,9	2,6682	33,035
1,1 ÷ 1,2	14,531	14,304	4,5 ÷ 4,6	6,2971	41,517	7,9 ÷ 8,0	2,6246	32,898
1,2 ÷ 1,3	14,531	14,601	4,6 ÷ 4,7	6,0572	41,069	8,0 ÷ 8,1	2,5827	32,765
1,3 ÷ 1,4	14,479	15,062	4,7 ÷ 4,8	5,8332	40,641	8,1 ÷ 8,2	2,5423	32,635
1,4 ÷ 1,5	13,61	15,698	4,8 ÷ 4,9	5,6236	40,233	8,2 ÷ 8,3	2,5035	32,509
1,5 ÷ 1,6	12,947	16,497	4,9 ÷ 5,0	5,4273	39,843	8,3 ÷ 8,4	2,4661	32,386
1,6 ÷ 1,7	12,416	17,422	5,0 ÷ 5,1	5,2329	39,345	8,4 ÷ 8,5	2,4301	32,266
1,7 ÷ 1,8	11,99	18,51	5,1 ÷ 5,2	5,0572	38,989	8,5 ÷ 8,6	2,3954	32,149
1,8 ÷ 1,9	11,648	19,69	5,2 ÷ 5,3	4,895	38,647	8,6 ÷ 8,7	2,3619	32,035
1,9 ÷ 2,0	11,357	20,971	5,3 ÷ 5,4	4,7422	38,32	8,7 ÷ 8,8	2,3296	31,924
2,0 ÷ 2,1	11,132	22,36	5,4 ÷ 5,5	4,5982	38,006	8,8 ÷ 8,9	2,2985	31,816
2,1 ÷ 2,2	10,965	23,947	5,5 ÷ 5,6	4,4623	37,704	8,9 ÷ 9,0	2,2684	31,71
2,2 ÷ 2,3	10,82	25,585	5,6 ÷ 5,7	4,3338	37,414	9,0 ÷ 9,1	2,2394	31,607
2,3 ÷ 2,4	10,716	27,364	5,7 ÷ 5,8	4,2123	37,135	9,1 ÷ 9,2	2,2104	31,408
2,4 ÷ 2,5	10,656	29,297	5,8 ÷ 5,9	4,2123	37,135	9,2 ÷ 9,3	2,182	31,31
2,5 ÷ 2,6	10,642	31,399	5,9 ÷ 6,0	3,9883	36,607	9,3 ÷ 9,4	2,1546	31,214
2,6 ÷ 2,7	10,692	33,583	6,0 ÷ 6,1	3,885	36,358	9,4 ÷ 9,5	2,1291	31,12
2,7 ÷ 2,8	10,768	35,961	6,1 ÷ 6,2	3,7868	36,117	9,5 ÷ 9,6	2,1046	31,028
2,8 ÷ 2,9	10,858	38,434	6,2 ÷ 6,3	3,6935	35,885	9,6 ÷ 9,7	2,0809	30,939
2,9 ÷ 3,0	10,986	41,131	6,3 ÷ 6,4	3,6935	35,885	9,7 ÷ 9,8	2,058	30,851
3,0 ÷ 3,1	11,115	43,846	6,4 ÷ 6,5	3,5205	35,444	9,8 ÷ 9,9	2,0358	30,766
3,1 ÷ 3,2	11,26	46,862	6,5 ÷ 6,6	3,4401	35,234	9,9 ÷ 10	2,0143	30,682
3,2 ÷ 3,3	11,378	49,467	6,6 ÷ 6,7	3,3634	35,032	$\geqslant 10$	2,0143	30,682

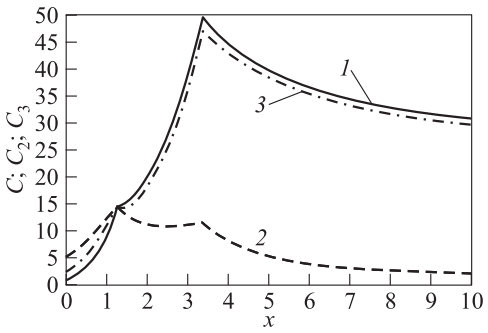


Рис. 1 Графики функций $C(x)$ (*1*), $C_2(x)$ (*2*), $C_3(x)$ (*3*)

Как видно из формулировки теоремы 1, «критическим» параметром в ней является значение α_x, так что предыдущие результаты получены с помощью выбора параметра A, оптимизирующего $C_2(x)$ при каждом x. Если в качестве критерия оптимальности рассмотреть $C(x) = \max\{C_2(x), C_3(x)\}$, то параметр A следует выбирать несколько иным способом. В таком случае на промежутке $x < 3{,}3830$ уже будет более целесообразным не фиксировать значение $q = 1$, а проводить оптимизацию по параметру q. Этот параметр следует выбирать так, чтобы $C_2(x) = C_3(x)$, если значение q лежит на отрезке $[0, 1]$. Если указанное равенство $C_2(x) = C_3(x)$ невозможно ни при каких допустимых значениях параметров, то надо, как и ранее, полагать $q = 1$. Такой подход приводит с следующему утверждению.

Теорема 3. *Для любого $x \geqslant 0$ имеет место неравенство*

Таблица 2 Верхние оценки функции $C(x)$

x	$C \leqslant$	x	$C \leqslant$	x	$C \leqslant$
0	2,6454	3,3 ÷ 3,4	47,647	6,7 ÷ 6,8	33,552
0,0 ÷ 0,1	3,251	3,4 ÷ 3,5	47,495	6,8 ÷ 6,9	33,368
0,1 ÷ 0,2	3,918	3,5 ÷ 3,6	45,848	6,9 ÷ 7,0	33,188
0,2 ÷ 0,3	4,6645	3,6 ÷ 3,7	45,097	7,0 ÷ 7,1	33,015
0,3 ÷ 0,4	5,482	3,7 ÷ 3,8	44,388	7,1 ÷ 7,2	32,847
0,4 ÷ 0,5	6,3743	3,8 ÷ 3,9	43,72	7,2 ÷ 7,3	32,684
0,5 ÷ 0,6	7,3317	3,9 ÷ 4,0	43,091	7,3 ÷ 7,4	32,526
0,6 ÷ 0,7	8,3775	4,0 ÷ 4,1	42,495	7,4 ÷ 7,5	32,373
0,7 ÷ 0,8	9,4707	4,1 ÷ 4,2	41,93	7,5 ÷ 7,6	32,224
0,8 ÷ 0,9	10,649	4,2 ÷ 4,3	41,396	7,6 ÷ 7,7	32,08
0,9 ÷ 1,0	11,863	4,3 ÷ 4,4	40,887	7,7 ÷ 7,8	31,939
1,0 ÷ 1,1	13,154	4,4 ÷ 4,5	40,405	7,8 ÷ 7,9	31,803
1,1 ÷ 1,2	14,449	4,5 ÷ 4,6	39,946	7,9 ÷ 8,0	31,669
1,2 ÷ 1,3	14,449	4,6 ÷ 4,7	39,509	8,0 ÷ 8,1	31,539
1,3 ÷ 1,4	14,418	4,7 ÷ 4,8	39,091	8,1 ÷ 8,2	31,414
1,4 ÷ 1,5	14,449	4,8 ÷ 4,9	38,693	8,2 ÷ 8,3	31,292
1,5 ÷ 1,6	14,962	4,9 ÷ 5,0	38,313	8,3 ÷ 8,4	31,172
1,6 ÷ 1,7	15,671	5,0 ÷ 5,1	37,948	8,4 ÷ 8,5	31,056
1,7 ÷ 1,8	16,56	5,1 ÷ 5,2	37,6	8,5 ÷ 8,6	30,943
1,8 ÷ 1,9	17,618	5,2 ÷ 5,3	37,267	8,6 ÷ 8,7	30,832
1,9 ÷ 2,0	18,836	5,3 ÷ 5,4	36,948	8,7 ÷ 8,8	30,724
2,0 ÷ 2,1	20,211	5,4 ÷ 5,5	36,64	8,8 ÷ 8,9	30,619
2,1 ÷ 2,2	21,736	5,5 ÷ 5,6	36,346	8,9 ÷ 9,0	30,516
2,2 ÷ 2,3	23,41	5,6 ÷ 5,7	36,063	9,0 ÷ 9,1	30,416
2,3 ÷ 2,4	25,23	5,7 ÷ 5,8	35,791	9,1 ÷ 9,2	30,318
2,4 ÷ 2,5	27,196	5,8 ÷ 5,9	35,791	9,2 ÷ 9,3	30,223
2,5 ÷ 2,6	29,303	5,9 ÷ 6,0	35,276	9,3 ÷ 9,4	30,13
2,6 ÷ 2,7	31,556	6,0 ÷ 6,1	35,033	9,4 ÷ 9,5	30,038
2,7 ÷ 2,8	33,953	6,1 ÷ 6,2	34,798	9,5 ÷ 9,6	29,949
2,8 ÷ 2,9	36,494	6,2 ÷ 6,3	34,572	9,6 ÷ 9,7	29,862
2,9 ÷ 3,0	39,182	6,3 ÷ 6,4	34,572	9,7 ÷ 9,8	29,777
3,0 ÷ 3,1	42,017	6,4 ÷ 6,5	34,144	9,8 ÷ 9,9	29,694
3,1 ÷ 3,2	45,001	6,5 ÷ 6,6	33,94	9,9 ÷ 10	29,613
3,2 ÷ 3,3	47,647	6,6 ÷ 6,7	33,742	$\geqslant 10$	29,613

$$\Delta_x \leqslant C(x)\left[\frac{\alpha_x}{(1+x)^2}+\frac{\beta_x}{(1+x)^3}\right],$$

где $C(x)$ — положительная ограниченная функция, для которой справедливы оценки, приведенные в табл. 2.

График полученной оценки приведен на рис. 1.

Следствие 2. *Если слагаемые $X_1,\ldots,X_n$ одинаково распределены, то неравенство* (1) *справедливо с* $C \leqslant 47{,}648$.

Литература

1. *Chen L. H. Y., Shao Q. M.* A non-uniform Berry–Esseen bound via Stein's method // Prob. Theory Related Fields, 2001. Vol. 120. P. 236–254.
2. *Thongtha P., Neammanee K.* Refinement of the constants in the non-uniform version of the Berry–Esseen theorem // Thai J. Math., 2007. Vol. 5. P. 1–13.
3. *Neammanee K., Thongtha P.* Improvement of the non-uniform version of the Berry–Esseen inequality via Paditz–Shiganov theorems // J. Inequalities Pure Appl. Math., 2007. Vol. 8. Iss. 4. Art. 92.
4. *Королев В. Ю., Попов С. В.* Уточнение оценок скорости сходимости в центральной предельной теореме при отсутствии моментов порядков, больших второго // Статистические методы оценивания и проверки гипотез. — Пермь: ПермГУ, 2011. С. 32–45.
5. *Шевцова И. Г.* Уточнение оценок скорости сходимости в теореме Ляпунова // Докл. РАН, 2010. Т. 435. Вып. 1. С. 26–28.
6. *Григорьева М. Е., Попов С. В.* О неравномерных оценках скорости сходимости в центральной предельной теореме // Докл. РАН, 2012 (в печати).

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 37–48

ОПТИМИЗАЦИЯ РАБОТЫ ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА С ПОМОЩЬЮ ИМИТАЦИОННОЙ МОДЕЛИ И АДАПТИВНЫХ АЛГОРИТМОВ*

М. Г. Коновалов[1]

Аннотация: Рассматривается проблема эффективного управления процессом выполнения заданий, поступающих в единый комплекс вычислительных ресурсов. Предлагается подход к решению проблемы, основанный на применении адаптивных стратегий в имитационной модели. На примере вычислительного комплекса излагается оригинальная методология построения и использования имитационных моделей. В качестве адаптивных стратегий используются алгоритмы, разработанные в теории частично наблюдаемого марковского процесса принятия решений. Приведены результаты вычислительного эксперимента.

Ключевые слова: системы вычислительных ресурсов; имитационные модели; адаптивные алгоритмы

1 Введение

Одно из направлений развития современных информационно-вычислительных систем заключается в их слиянии в мощные комплексы, объединяющие большое число разнородных, географически распределенных компьютеров и компьютерных систем. Примерами могут служить системы самого широкого спектра — от глобальных гридов и клаудов и заканчивая локальными специализированными вычислительными центрами, обладающими автономными парками вычислительных ресурсов. Для большинства таких систем характерна проблема эффективности использования вычислительной техники, которая лишь обостряется, несмотря на интенсивное развитие последней.

Общая краткая характеристика проблемы может быть выражена словами «что, где и когда вычислять», и в этом широком аспекте поиск ее решения является предметом интенсивных усилий. С некоторыми направлениями и результатами исследований можно ознакомиться в работах [1–3] и содержащихся в них обзорах. Данная работа также относится к указанной области.

Используемые в статье соображения вкратце сводятся к следующему. Всякий вычислительный комплекс, обслуживающий потоки заданий, характеризуется набором параметров. Некоторые из них могут трактоваться как статические (например, емкость и производительность процессоров), в то время как другие носят более выраженный динамический характер (например, параметры, определяющие размещение заданий на процессорах). Таких параметров очень много, и их взаимосвязь труднообозрима, что затрудняет построение чисто математических моделей и получение чисто математических решений, связанных с выбором оптимальных значений параметров. Имитационное компьютерное моделирование существенно расширяет возможности адекватного описания объекта. В частности, появляется возможность получать потенциально неограниченное количество траекторий функционирования системы, свойства которых зависят от упомянутых параметров. Используя эти траектории, можно попытаться оптимизировать значения параметров по выбранным критериям.

Изложенные соображения являются, в принципе, достаточно традиционными и широко используемыми. В данной работе, однако, делается акцент на двух сравнительно мало освещаемых обстоятельствах.

Во-первых, в подавляющем числе работ наличие имитационной модели лишь констатируется, а ее описание и, как следствие, степень соответствия описанию моделируемого объекта остаются как бы за кадром. Это в принципе затрудняет возможность воспроизводить результаты экспериментов с моделью и оценивать сделанные с ее помощью выводы. В предлагаемой работе уделяется значительное внимание способу задания имитационной модели.

Второй момент заключается в способе использования имитационной модели. Как правило, в экспериментах с моделью апробируются различные

* Работа выполнена при поддержке РФФИ, грант 11-07-00112.

[1] Институт проблем информатики Российской академии наук, mkonovalov@ipiran.ru

«готовые» алгоритмы, с тем чтобы выбрать лучший среди них. В данном случае предлагается «настраивать» алгоритмы, используя специальные стратегии адаптивной обработки информации, поступающей с имитируемых траекторий объекта.

Настоящая работа является продолжением работ [1, 3]. Объектом моделирования и оптимизации служит абстрактный вычислительный комплекс со свойствами, присущими реальным системам, который рассматривался также в [2]. Основное внимание уделяется построению имитационной модели и выбору параметров, подлежащих оптимизации. Используемые адаптивные алгоритмы базируются на теоретических предпосылках, изложенных в [4]. Приведены некоторые результаты численных экспериментов.

2 Общее описание

Вычислительный комплекс предназначен для обработки **потоков заданий**, постоянно поступающих на него. Комплекс состоит из совокупности **вычислительных устройств**, на которых происходит выполнение заданий, **центра управления**, а также **среды передачи**, через которую осуществляется сообщение между центром управления и вычислительными устройствами.

Вычислительные устройства, вообще говоря, представляют собой особым образом структурированную и организованную совокупность вычислительных ресурсов, имеющих к тому же достаточно сложную внутреннюю структуру. Однако в данном случае ограничимся представлением об упорядоченном массиве независимо работающих вычислительных устройств, которые будем называть для краткости также **компьютерами**. Вычислительные устройства не являются абсолютно надежными и могут частично или полностью выходить из строя.

Роль центра управления с точки зрения выполнения заданий заключается в реализации следующих функций:

(1) приемки заданий (извне);

(2) управления очередью заданий;

(3) управления процессом выполнения отдельного задания на вычислительных устройствах;

(4) управления средой передачи;

(5) отправки обработанных заданий (вовне);

(6) технического обслуживания вычислительных устройств.

Эти функции будут в определенной мере отражены в излагаемой ниже модели. Однако с точки зрения оптимизации из данного перечня будут представлять интерес, главным образом, второй и третий пункты.

Среда передачи является локальной коммуникационной сетью, которая в дальнейших рассмотрениях будет фигурировать в качестве источника задержек в обмене информацией между центром управления и компьютерами.

Задание состоит из некоторого числа (элементарных) **задач**, каждая из которых может выполняться независимо от остальных и на произвольном компьютере. (Это соответствует часто используемому в литературе понятию «делимой нагрузки».) Решение, или, по-другому, выполнение задания, заключается в решении (выполнении) задач. В некоторых случаях необходимо решить все задачи. Но может оказаться, что в задании существует одна или несколько элементарных задач, которые представляют особый интерес и которые будем называть **ключевыми** задачами. Поиск и решение ключевых задач представляют собой основную цель задания. Какие из задач являются ключевыми и существуют ли вообще ключевые задачи в данном задании, заранее не известно.

Пребывание задания в вычислительном комплексе складывается из чередования промежутков времени ожидания обслуживания и непосредственного решения на вычислительных устройствах. Промежуток времени второго типа, т. е. период, когда задание выполняется на компьютерах, будем называть **запуском**. Запусков одного и того же задания может быть несколько.

Любой запуск задания может оканчиваться следующими исходами:

– найдено заданное множество ключевых задач (одна, несколько или все);

– решены все задачи и установлено, что ключевые задачи отсутствуют (не найдены);

– истекло **время жизни** задания (наступил **дедлайн**);

– запуск задания принудительно прерван по решению центра управления;

– произошел сбой или отказ вычислительного комплекса или его элементов.

Весь процесс выполнения задания может оканчиваться одним из следующих исходов:

– задание выполнено (найдены ключевые задачи или решены все задачи);

– задание не выполнено, но наступил дедлайн;

– задание не выполнено, но удаляется из системы по решению вычислительного центра.

Окончание процесса выполнения задания при любом его варианте означает уход задания из вычислительного комплекса.

Организация процесса решения задач, составляющих задание, обладает следующими особенностями.

При подготовке задания к запуску центр управления разбивает задание на отдельные блоки, состоящие из некоторого количества элементарных задач. В дальнейшем для обозначения таких блоков используется термин **пакет**.

Одновременно составляется список вычислительных устройств, на которых будут обрабатываться пакеты, составленные из элементарных задач. В ходе конкретного запуска данного задания составленный список не может увеличиваться, а может лишь сокращаться (в случае выхода из строя отдельных компьютеров). В ходе выполнения задания центр управления периодически посылает на вычислительные устройства порции пакетов, по одному на каждый исправный компьютер из указанного списка. Для обозначения указанной порции пакетов используется термин **посылка**. Промежуток времени между отправлением посылки и приемом результатов ее выполнения называется **контрольным временем** и устанавливается в начале каждого запуска задания.

Пересылка пакетов к компьютерам осуществляется через среду передачи. На пересылку пакета затрачивается некоторое время, которое, вообще говоря, является случайным и соизмеримым со временем обработки пакета задач. Таким образом, отправление посылки занимает значимое время.

Предполагается, что результаты обработки пакетов из посылки доставляются центру управления за пренебрежимо малое время по истечении контрольного времени.

Обработка пакета на вычислительном устройстве происходит во многих отношениях независимо от работы остальной части системы и может заканчиваться одним из следующих исходов:

- выполнены все задачи из пакета (при этом обнаружены или не обнаружены ключевые задачи);
- истекло **контрольное время**, отведенное на выполнение задач из пакета, но задачи решены не все.

Невыполнение задач из пакета может наступать по следующим причинам:

- произошел **отказ** компьютера, вызвавший его полную остановку;
- произошел частичный отказ процессора (**сбой**), вследствие чего уменьшилась производительность;
- контрольного времени не хватило для решения всех задач из пакета по причинам, не связанным с отказами аппаратуры.

В любом случае неполного выполнения задач из пакета считается, что все входящие в пакет задачи должны быть выполнены заново без учета результатов предыдущей попытки и в составе новых пакетов.

3 Принципы формального описания имитационной модели

Под словами «формальная модель» будем понимать алгоритмически точное описание, которое позволяет независимому исследователю однозначно воспроизводить (в частности, на компьютере) задуманные и заложенные в модель особенности поведения системы. С этой точки зрения идеально подходил бы реализующий модель компьютерный код, но он занимает слишком много места, содержит не относящуюся к модели информацию и к тому же не очень нагляден. Можно воспользоваться одним из многочисленных и распространенных приемов описания (таких как псевдокод или блок-схема или более изощренных, таких как сети Петри, специальные языки моделирования и т. п.). Тем не менее здесь используется оригинальный подход, использующий, однако, сравнительно известные идеи.

Изложенную в предыдущем разделе модель можно представлять себе как совокупность взаимосвязанных параметров, которые изменяются вполне определенным образом в процессе моделирования. Исходная посылка для дальнейших рассуждений заимствована из обычного программирования и заключается в том, что моделирование системы на компьютере представляет собой чередование неких событий, наступление которых знаменуется пересчетом параметров. Формальное задание модели должно поэтому содержать два основных раздела: в одном должна быть указана совокупность событий и закономерность их чередования, а во втором — обработчики событий, т. е. алгоритмы пересчета параметров. Второй раздел, по-видимому, не требует особых комментариев, поскольку представляется идейно вполне ясным. Что касается первого раздела, то его обсуждение сейчас последует и оно основано на некоторых идеях, содержащихся в [5]. Впрочем, из теории Хоара взято немного, причем безоговорочно — только общее определение процесса, использование рекурсии и понятие параллельного взаимодействия процессов.

Определение 1. *Процессом называется пара символов* $\mathrm{P} = (\mathrm{A}, \Pi)$*, где* A *— не более чем счетное множество, называемое алфавитом, а* Π *— подмножество множества всех конечных и бесконечных последовательностей из* A*. Элементы алфавита будем называть также событиями, а элементы множества* Π *— протоколами.*

В дальнейшем будем иногда обозначать алфавит процесса P как $\alpha\mathrm{P}$, а множество протоколов — как $\pi\mathrm{P}$.

Интуитивно процесс соответствует представлению об объекте, который в процессе эволюции может принимать участие в событиях из своего алфавита, причем чередование событий обязано в точности соответствовать какому-нибудь протоколу. Иначе говоря, определение процесса, соответствующего некоторому объекту, задает потенциально возможное развитие этого объекта.

Задание протоколов путем перечисления не конструктивно, поэтому для этой цели используются специальные приемы, среди которых здесь используются рекурсия, композиция процессов в виде «выбора», параллельная композиция, а также переименование.

Определение 2. *Пусть* $\mathrm{P} = (\mathrm{A}, \Pi)$ *— некоторый процесс и пусть* a *— событие, необязательно содержащееся в алфавите процесса* P*. Новый процесс, обозначаемый* $\mathrm{P}_1 = a \to \mathrm{P}$ (*читается «*P *следует за* a*»*)*, определяется как* $\mathrm{P}_1 = (\mathrm{A}_1, \Pi_1)$*, где* $\mathrm{A}_1 = A \cup \{a\}$*, а* Π_1 *получается из множества* Π *добавлением в начало каждого протокола элемента* a*.*

Интерпретация процесса P_1 такова: «этот процесс вначале участвует в событии a, а затем ведет себя в точности, как процесс P».

Определение 2 можно обобщить, предлагая, например, для процесса P_1 в качестве первого события выбор из некоторого множества.

Отметим важное обстоятельство: в правой части уравнения может фигурировать тот же самый процесс, который определяется в левой части: $\mathrm{P} = a \to \mathrm{P}$. В этом случае определяемый процесс имеет единственный протокол: $(a, a, \ldots)$. Это пример так называемого рекурсивного задания процесса. Принципиально, что рекурсия может использоваться и во всех последующих конструкциях.

Определение 3. *Пусть* P_1 *и* P_2 *— некоторые процессы, такие что у любой пары протоколов* $p_1 \in \pi\mathrm{P}_1$ *и* $p_2 \in \pi\mathrm{P}_2$ *начальные элементы различны. Новый процесс, обозначаемый* $\mathrm{P} = \mathrm{P}_1|\mathrm{P}_2$ (*читается «*P *есть выбор между* P_1 *и* P_2*»*)*, определяется как процесс с алфавитом* $\alpha\mathrm{P} = \alpha\mathrm{P}_1 \cup \alpha\mathrm{P}_2$ *и множеством протоколов* $\pi\mathrm{P} = \pi\mathrm{P}_1 \cup \pi\mathrm{P}_2$*.*

Процесс P «ведет себя либо как процесс P_1, либо как процесс P_2, причем выбор зависит от начального события, которое определяется «окружением» процесса P, т. е. другими процессами, с которыми он взаимодействует».

Определение 4. *Пусть* P_1 *и* P_2 *— некоторые процессы. Новый процесс, обозначаемый* $\mathrm{P} = \mathrm{P}_1 \| \mathrm{P}_2$ (*читается «*P *есть параллельная композиция* P_1 *и* P_2*»*)*, определяется следующим образом. Его алфавит имеет вид* $\alpha\mathrm{P} = \alpha\mathrm{P}_1 \cup \alpha\mathrm{P}_2$*. Множество протоколов* $\pi\mathrm{P}$ *состоит из всех упорядоченных наборов событий из множеств* $\alpha\mathrm{P}_1$ *и* $\alpha\mathrm{P}_2$*, которые обладают следующим свойством: после удаления из набора всех символов, не принадлежащих алфавиту* $\alpha\mathrm{P}_1$ $(\alpha\mathrm{P}_2)$*, получается протокол, принадлежащий множеству* $\pi\mathrm{P}_1$ (*соответственно* $\pi\mathrm{P}_2$).

Параллельные процессы «обязаны принимать совместное участие во всех событиях, принадлежащих пересечению их алфавитов». В остальных случаях каждый процесс «ведет себя так, как будто другого не существует».

Полезным приемом, с помощью которого можно задавать новые процессы, является переименование. Пусть f — взаимно однозначная функция, отображающая алфавит A процесса во множество символов $f(\mathrm{A})$.

Определение 5. *Переименованием процесса* (A, Π) *с помощью функции* f *называется процесс* (A_f, Π_f)*, где* $\mathrm{A}_f = f(\mathrm{A})$*, а множество* Π_f *образовано из протоколов множества* Π *путем замены всех символов на их* f*-образы.*

Заданное последним определением переименование особенно полезно при создании групп сходных процессов, которые функционируют идентичным образом, никак не взаимодействуя друг с другом. Это означает, что все они должны иметь различные и взаимно непересекающиеся алфавиты. С этой целью каждый процесс снабжается меткой, которая добавляется к общему для всех процессов имени. Процесс с именем P и с меткой l обозначается $l : \mathrm{P}$. Каждое событие помеченного процесса имеет ту же метку и выглядит как $l.a$, где a — название события, а l — метка.

Определение 6. *Пусть* P *— процесс, а* l *— метка. Помеченный процесс* $l : \mathrm{P}$ *задается функцией* $f_l(a) = l.a$ *для всех* a *из алфавита процесса* P *и пометкой* $l : \mathrm{P} = f_l(\mathrm{P})$*.*

Приведенные определения можно обобщать, расширяя выразительные свойства данного языка. Например, если L — множество меток, то процесс $\mathrm{L} : \mathrm{P}$ «ведет себя как процесс $l : \mathrm{P}$ каждый раз, когда окружение процесса выбрало метку l». Другой пример — это процесс $\|_{l \in \mathrm{L}} l.\mathrm{P}$, который представляет собой параллельную композицию процессов из совокупности $\{l.\mathrm{P};\ l \in \mathrm{L}\}$.

Сформулированные в определениях 2–6 операции можно применять многократно, получая уравнения, у которых в левой части стоит вновь определяемый процесс, а в правой — сколь угодно сложная суперпозиция других процессов.

Вернемся к вопросу о способе описания имитационной модели.

Определение 7. *(Формальной) имитационной моделью некоторой системы назовем тройку символов*

$$\mathbb{M} = (\mathfrak{P}, \mathrm{P}, \mathrm{H}) \ ,$$

где $\mathfrak{P}$ *— множество параметров модели (системы);* P *— процесс (в смысле определения* 1*);* $\mathrm{H} = \{\mathrm{H}_{a,\mathfrak{p}},\ a \in \alpha\mathrm{P},\ \mathfrak{p} \in \mathfrak{P}\}$ *— семейство операторов, каждый из которых действует из множества значений соответствующего параметра в то же множество. Совокупность* $\mathrm{H}_a = \{\mathrm{H}_{a,\mathfrak{p}},\ \mathfrak{p} \in \mathfrak{P}\}$ *будем называть обработчиком события* $a \in \alpha\mathrm{P}$.

Приведенное определение имитационной модели соответствует представлению о ее функционировании как о чередовании событий из множества $\alpha\mathrm{P}$. Допустимые последовательности событий — это протоколы процесса P. При наступлении события $a \in \alpha\mathrm{P}$ текущее значение x каждого параметра $\mathfrak{p} \in \mathfrak{P}$ заменяется на (вообще говоря, новое) значение $x' = \mathrm{Q}_{a,\mathfrak{p}}(x)$.

4 Имитационная модель вычислительного комплекса

4.1 Основные параметры

Множество параметров $\mathfrak{P}$ естественным образом разбивается на подмножества, относящиеся к составным частям объекта, как они были описаны в разд. 2, как-то: компьютеры, задания и т. д. В этом подразделе задаются основные элементы каждого из этих подмножеств.

Вычислительные устройства

Множество всех вычислительных устройств (компьютеров) обозначается через $\mathcal{C}$. Каждый компьютер $c \in \mathcal{C}$ характеризуется следующими параметрами:

$c.v$ — емкость, т. е. максимальное количество элементарных задач, которое данное вычислительное устройство может единовременно принять для обработки;

$c.r$ —производительность, под которой подразумевается безразмерный коэффициент, показывающий, во сколько раз скорость обработки элементарной задачи на данном процессоре отличается от скорости обработки на стандартном компьютере;

$c.s$ — состояние, которое может принимать одно из трех значений: 0 (operable), 1 (fail), 2 (fault).

$c.t_{\mathrm{trans}}$ — время перехода в следующее состояние.

Переходы между состояниями процессора регулируются матрицей вероятностей, которая имеет вид:

$$\begin{pmatrix} 0 & c.\mathrm{P}_{01} & 1 - c.\mathrm{P}_{01} \\ c.\mathrm{P}_{10} & 0 & 1 - c.\mathrm{P}_{10} \\ 0 & 0 & 1 \end{pmatrix} ,$$

а продолжительности пребывания в каждом из состояний образуют последовательность условно независимых случайных величин с функциями распределения — компонентами вектора $c.\mathrm{H} = (c.\mathrm{H}_0, c.\mathrm{H}_1, c.\mathrm{H}_2)$.

Задания

Пусть j обозначает задание. Следующие параметры задания определяются в момент его возникновения и остаются неизменными в течение всего времени пребывания задания в системе:

$j.f$ — поток, породивший задание;

$j.t_{1t}$ — время жизни задания;

$j.k$ — ключ — индикатор наличия ключевой задачи в задании. Предполагается, что в каждом задании имеется не более одной ключевой задачи. Если значение данного параметра равно 1, то ключевая задача существует; если параметр равен 0, то ключевой задачи в задании нет;

$j.l$ — длина, которая определяется как количество элементарных задач в задании;

$j.t_{\mathrm{prep}}$ — время, необходимое для подготовки задания к очередному запуску. Этот параметр устанавливается в момент поступления задания в систему и остается неизменным независимо от числа запусков.

Кроме того, задание характеризуется набором параметров, которые изменяются в процессе выполнения задания:

$j.\mathcal{C}$ — список вычислительных устройств, на которых выполняется задание;

$j.\mathcal{P}$ — список пакетов, составляющих очередную посылку, т. е. партия пакетов, направляемых единовременно для обработки на компьютеры. Число пакетов в посылке совпадает с количеством компьютеров в предыдущем списке;

$j.t_{\mathrm{check}}$ — контрольное время, через которое происходит проверка результатов выполнения всех пакетов задач из посылки.

Потоки заданий

Множество всех потоков обозначается через $\mathcal{F}$. Пусть $t_0 = 0$, а $0 \le t_1 \le t_2 \le \ldots$ — последовательные моменты поступления заданий некоторого потока $f \in F$. Тогда последовательность $t_n - t_{n-1}$, $n = 1, 2, \ldots$, интервалов между последовательными поступлениями заданий является последовательностью независимых случайных величин, имеющих одинаковое распределение $f.\mathcal{D}_{\text{in}}$. Текущее значение промежутка времени между поступлениями заданий обозначается через $f.t_{\text{in}}$. Неизменяемые параметры вновь поступившего задания данного потока определяются с помощью набора функций распределения $(f.\mathcal{D}_{1t}, f.\mathcal{D}_{\text{key}}, f.\mathcal{D}_{\text{len}}, f.\mathcal{D}_{\text{prep}})$. В качестве значений параметров «время жизни», «ключ», «длина» и «время подготовки» задание получает реализации случайных величин с соответствующими функциями распределения.

Еще два статических параметра потока $f.\mu$ и $f.\sigma$ определяют соответственно среднее значение и дисперсию времени выполнения элементарной задачи на стандартном вычислительном устройстве.

Очередное задание, порождаемое потоком f, обозначается через $f.j$.

Пакеты

Пакетом называется набор задач, который отправляется единовременно на определенный процессор в составе посылки — совокупности пакетов, образованных из одного задания. Параметры пакета:

$p.j$ — задание, из которого сформирован пакет;

$p.c$ — компьютер, на котором выполняется пакет;

$p.l$ — длина, т. е. число входящих в пакет задач;

$p.t_{\text{send}}$ — время передачи пакета на выбранный компьютер, которое определяется функцией распределения $\mathcal{D}_{\text{send}}$, общей для всей системы и имеющей в качестве аргумента длину пакета;

$p.t_{\text{exe}}$ — время выполнения пакета заданий на выбранном компьютере при условии, что последний находится в исправном состоянии (об этом параметре ниже).

Центр управления

Работа центра управления фактически определяется совокупностью правил принятия решений, касающихся следующих аспектов.

1. Управление очередью заданий (запуск, прерывание запуска, удаление из системы), с которым связаны следующие параметры:

   t_{dec} — момент принятия решения об обслуживании очереди заданий;

   Δt_{dec} — интервал времени между последовательными моментами принятия решения об обслуживании очереди;

   $\mathcal{J}_{\text{out}}$ — множество заданий, удаляемых из системы как выполненных или просроченных;

   $\mathcal{J}_{\text{break}}$ — множество заданий, запуск которых прерывается;

   $\mathcal{J}_{\text{run}}$ — множество заданий, выбранных для запуска.

2. Управление запуском задания (назначение вычислительных устройств, обслуживающих задание, установление размеров пакетов, определение контрольного времени для проверки результатов посылок). Соответствующие параметры были определены в разделах, относящихся к заданиям и пакетам.

3. Техническое обслуживание компьютеров (ремонт, замена). Этот вид деятельности центра управления в данной работе не рассматривается. В дальнейшем предполагается, что вышедшие из строя компьютеры могут либо самопроизвольно восстановиться через некоторое время, либо окончательно ломаются и больше не принимают участия в работе.

4.2 События и процессы

В этом пункте формулируется основной процесс P, определяющий допустимые последовательности рассматриваемых событий в вычислительном комплексе. Вначале выделяются определенные события, с которыми связано функционирование объекта. Затем устанавливаются потенциально возможные траектории событий с помощью системы уравнений, определяющих вспомогательные процессы. Эти уравнения основаны на конструкциях из разд. 3 и включают рекурсию.

Условимся рассматривать следующие события в системе:

decision — момент принятия решения центром управления об обслуживании очереди задач;

$f.\text{input}$ — поступление в систему задания из потока f;

$j.\text{run}$ — подготовка к запуску задания j;

$j.\text{package}$ — отправление посылки из задания j на вычислительные устройства;

$j.\text{replay}$ — проверка результатов посылки из задания j;

$j.\text{break}$ — прерывание запуска задания j;

$j.\text{output}$ — уход из системы задания j;

$p.\text{start}$ — начало выполнения пакета p;

p.finish — завершение выполнения пакета p;

c.fail — сбой компьютера c;

c.fault — отказ компьютера c;

c.reneval — восстановление компьютера c.

Совокупность перечисленных событий составляет множество событий $\alpha\mathrm{P}$ процесса P.

Далее используются дополнительные соглашения и обозначения. Считается, что операция $\to$ связывает аргументы сильнее, чем операции $|$ и $\|$, которые имеют одинаковый ранг и, если не расставлены скобки, выполняются слева направо. Например, записи $a \to \mathrm{P}_1|b \to \mathrm{P}_2\|\mathrm{P}_3$ и $(a \to \mathrm{P}_1)|\,((b \to \mathrm{P}_2)\,\|\mathrm{P}_3)$ равносильны.

Множественная пометка перед уравнением процесса означает, что задано множество уравнений, в каждом из которых все события, подпроцессы и параметры имеют одинаковую метку. Например, запись $\mathcal{G}:\ \mathrm{P}_1 = a \to \mathrm{P}_2$ равносильна системе $g.\mathrm{P}_1 = g.a \to g.\mathrm{P}_2,\ g \in \mathcal{G}$.

Обозначение $[\![\ldots]\!]$ используется для перечисления событий, которые все должны произойти, но в произвольном порядке. Например, запись $[\![a, b]\!] \to \to \mathrm{P}$ равносильна выражению $a \to b \to \mathrm{P}|b \to \to a \to \mathrm{P}$.

Обозначение $\langle\ldots\rangle$ используется для перечисления событий, которые предоставляются в качестве возможного, но необязательного выбора. Например, если протоколы процесса P не начинаются с символа b, то запись $a \to \langle b\rangle \to \mathrm{P}$ равносильна выражению $a \to (\mathrm{P}|b \to \mathrm{P})$.

Символ $\boxdot$ указывает на завершение процесса.

Через $\mathcal{J}$ обозначается множество всех потенциально возможных заданий, поступающих в систему.

Аналогично $\mathcal{P}$ — множество всех потенциальных пакетов, которые могут быть образованы в системе.

Все потенциальные траектории событий в модели задаются следующим набором соотношений:

$$
\begin{aligned}
&\mathrm{P} = \mathrm{COMPUTER\_SYSTEM} = \mathrm{FLOWS}\|\mathrm{COMPUTERS}\|\mathrm{CONTROL\_CENTER}\\
&\mathrm{FLOWS} = \|_{f\in\mathcal{F}} f.\mathrm{FLOW}\\
\mathcal{F}:\ &\mathrm{FLOW} = \mathrm{input} \to (j.\mathrm{JOB}\|\mathrm{FLOW})\\
\mathcal{J}:\ &\mathrm{JOB} = \mathrm{run} \to \mathrm{EXECUTION}|\mathrm{output} \to \boxdot\\
\mathcal{J}:\ &\mathrm{EXECUTION} = \mathrm{package} \to (\mathrm{PACKAGE}\|\mathrm{replay} \to \mathrm{EXECUTION})\,|\mathrm{BREAK}\\
\mathcal{J}:\ &\mathrm{PACKAGE} = \|_{p\in\mathcal{P}} p.\mathrm{PACKET}\\
\mathcal{J}:\ &\mathrm{BREAK} = \mathrm{break} \to (\mathrm{JOB}|\mathrm{output} \to \boxdot)\\
\mathcal{P}:\ &\mathrm{PACKET} = \mathrm{start} \to \langle \mathrm{finish}, c.\mathrm{fail}, c.\mathrm{fault}\rangle \to j.\mathrm{replay} \to \boxdot\\
&\mathrm{COMPUTERS} = \|_{c\in\mathcal{C}} c.\mathrm{COMPUTER}\\
\mathcal{C}:\ &\mathrm{COMPUTER} = p.\mathrm{PACKET}\|\mathrm{CYCLE}\\
\mathcal{C}:\ &\mathrm{CYCLE} = \mathrm{fail} \to \mathrm{reneval} \to \mathrm{COMPUTER}|\mathrm{fail} \to \mathrm{fault} \to \boxdot|\mathrm{fault} \to \boxdot\\
&\mathrm{CONTROL\_CENTER} = \mathrm{decision} \to \mathrm{DECISION}\|\mathrm{CONTROL\_CENTER}\\
&\mathrm{DECISION} = [\![j.\mathrm{output},\ j \in \mathcal{J}_{\mathrm{out}}]\!] \to [\![j.\mathrm{break},\ j \in \mathcal{J}_{\mathrm{break}}]\!] \to [\![j.\mathrm{run},\ j \in \mathcal{J}_{\mathrm{run}}]\!] \to \boxdot
\end{aligned}
$$

4.3 Обработка событий

Обработка события связана с выполнением определенных действий, которые по аналогии с обычным программированием будем называть процедурами. Обработчик H_a события a представляет собой процедуру, объединяющую эти действия.

Опишем кратко обработчики некоторых из определенных выше событий, опуская не самые существенные подробности.

Обработка события decision связана с принятием решений относительно обслуживания очереди заданий. Предполагается, что эта процедура выполняется в определенные моменты времени (в п. 4.1 обозначенные с помощью переменного параметра t_{dec}), причем назначение этих моментов также является составной частью принимаемого решения. Обработка заключается в выполнении следующих процедур:

$$
\mathrm{H}_{\mathrm{decision}} = \{\mathsf{BreakDecision};\ \mathsf{RunDecision};\ \mathsf{CheckDecision};\ \mathsf{PeriodDecision}\}\ .
$$

Процедура BreakDecision определяет множества заданий $\mathcal{J}_{\mathrm{out}}$ и $\mathcal{J}_{\mathrm{break}}$.

Процедура RunDecision определяет множество заданий $\mathcal{J}_{\mathrm{run}}$ и определяет множества компьютеров $j.\mathcal{C},\ j \in \mathcal{J}_{\mathrm{run}}$, на которых эти задания будут выполняться. При составлении списка $j.\mathcal{C}$ используется информация о степени готовности (исправности) компьютеров, а также об их загруженности. Компьютер может быть включен в указанный список

только в том случае, если он не занят решением задач и находится в состоянии operable. Этот список может обновляться при каждом новом запуске задания. В течение одного запуска данный список может лишь уменьшаться (в случае выхода из строя вычислительных устройств).

Процедура CheckDecision определяет размер посылаемых пакетов, а также контрольное время, через которое будут проверяться результаты посылок.

Процедура PeriodDecision устанавливает промежуток времени Δt_{dec} до следующего момента принятия решения.

Обработчик события $f.\mathrm{input}$ порождает новое задание $f.j$ и устанавливает его параметры: $(f.j).t_{1t}$ (время жизни), $(f.j).k$ (ключ), $(f.j).l$ (длину), $(f.j).t_{\mathrm{prep}}$ (время на подготовку запуска). Пребывание задания j в системе оканчивается с наступлением события $j.\mathrm{output}$.

Событие $j.\mathrm{run}$ означает начало запуска задания j.

Обработка события $j.\mathrm{package}$ заключается в образовании множества пакетов $j.\mathcal{P}$ и отправке их на соответствующие компьютеры из множества $j.\mathcal{C}$.

Событие $j.\mathrm{replay}$ влечет обработку результатов выполнения партии пакетов. Выполненные пакеты уменьшают размер невыполненной части задания. В случае решения ключевой задачи задание считается выполненным. Если задание не выполнено, то организуется новая посылка.

Событие $p.\mathrm{start}$ означает, что пакет p доставлен на соответствующее вычислительное устройство. При обработке этого события выполняется процедура ProcessingTime, которая определяет время $p.t_{\mathrm{exe}}$ выполнения пакета. Событие $p.\mathrm{finish}$ означает, что пакет p успешно выполнен.

Обработка события $j.\mathrm{break}$ заключается в прерывании запуска задания. Освобождаются все компьютеры, занятые его выполнением.

Наступление события $c.\mathrm{fail}$ ($c.\mathrm{fault}$) означает переход компьютера c в состояние 1 (2). Если этот компьютер был закреплен за заданием j, то он вычеркивается из списка $j.\mathcal{C}$ и в дальнейшем не участвует в обслуживании заданий. Событие $c.\mathrm{reneval}$, которое может наступить после события $c.\mathrm{fail}$, означает восстановление компьютера (переход в состояние 0).

4.4 Реализация

Процесс P, построенный в п. 4.2, задает множество возможных траекторий (протоколов), по которым могут реализовываться события из алфавита $\alpha\mathrm{P}$. Однако для имитации поведения системы необходимо выделять из этого множества единственную траекторию, которую будем обозначать $\pi = (a_0, a_1, \ldots) \in \pi\mathrm{P}$, $a_n \in \alpha\mathrm{P}$.

Считаем, что имитационная траектория разворачивается в непрерывном времени $t \geq 0$. С каждым событием a_n из протокола π свяжем обозначение $\top(a_n)$, указывающее момент времени, в который оно происходит (или должно произойти). Сопоставим моменту $\top(a_n)$ множество

Таблица 1 Построение множества «ближайших» событий

Событие a_n	$a \in \mathcal{B}_n$	$\top(a) - \top(a_n)$	
decision	decision	Δt_{dec}	Определяется процедурой PeriodDecision
	$j.\mathrm{output}$	t_ε	Константа в той же процедуре
	$j.\mathrm{break}$		
	$j.\mathrm{run}$		
$j.\mathrm{input}$	$j.\mathrm{input}$	$(j.f).t_{\mathrm{in}}$	Реализация случайной величины с распределением $(j.f).\mathcal{D}_{\mathrm{in}}$
$j.\mathrm{run}$	$j.\mathrm{package}$	$j.t_{\mathrm{prep}}$	Реализация случайной величины с распределением $f.\mathcal{D}_{\mathrm{prep}}$
$j.\mathrm{package}$	$j.\mathrm{replay}$	$j.t_{\mathrm{check}}$	Определяется процедурой CheckDecision
	$p.\mathrm{start}$, $p \in j.\mathcal{P}$	$p.t_{\mathrm{send}}$	Реализация случайной величины с распределением $\mathcal{D}_{\mathrm{send}}$
$p.\mathrm{start}$	$p.\mathrm{finish}$	$p.t_{\mathrm{exe}}$	Определяется процедурой ProcessingTime
$c.\mathrm{fail}$	if $\alpha = 1$ then $c.\mathrm{reneval}$ else $c.\mathrm{fault}$	$c.t_{\mathrm{trans}}$	α — бернуллиевская случайная величина с распределением $c.\mathrm{P}_{10}$, t_{trans} — реализация случайной величины с распределением $c.\mathrm{H}_1$
$c.\mathrm{reneval}$	if $\alpha = 1$ then $c.\mathrm{fail}$ else $c.\mathrm{fault}$	$c.t_{\mathrm{trans}}$	α — бернуллиевская случайная величина с распределением $c.\mathrm{P}_{01}$, t_{trans} — реализация случайной величины с распределением $c.\mathrm{H}_0$

«ближайших» событий $\mathcal{A}_{n+1} \subset \alpha\mathrm{P}$, которые должны произойти непосредственно вслед за событием a_n. Каждый элемент множества $a \in \mathcal{A}_{n+1}$ обладает характеристикой $\top(a_n)$, причем обязательно $\top(a) > \top(a_n)$. При этом первое за моментом $\top(a_n)$ событие a_{n+1} определяется как элемент $a_{n+1} = a \in \in \mathcal{A}_{n+1}$ с минимальным значением $\top(a)$.

Множество $\mathcal{A}_{n+1}$ образуется из множества $\mathcal{A}_n$ следующим образом: $\mathcal{A}_{n+1} = (\mathcal{A}_n \backslash \{a_n\}) \cup \mathcal{B}_n$, причем состав множества добавляемых событий $\mathcal{B}_n$ определяется событием a_n. В это множество $\mathcal{B}_n$ входят события, которые обязательно должны произойти непосредственно за событием a_n в соответствии с протоколами процесса P. В табл. 1 показан состав множества $\mathcal{B}_n$ в зависимости от события a_n.

5 Оптимизация

5.1 Параметры процедур принятия решений

Рассмотрим подробнее процедуры, которые связаны с принятием решений и входят в обработчик события decision (см. п. 4.3).

Процедура BreakDecision. Эта процедура устанавливает, какие задания следует удалить из системы, а для каких следует прервать запуск. Первая составляющая вполне ясна: удаляются задания, выполненные или просроченные к моменту наступления очередного события decision. Вопрос с прерыванием более интересен. Дело в том, что по мере выполнения задания (без обнаружения ключевой задачи) может увеличиваться вероятность того, что задание вообще ее не содержит, т. е. имеет малую ценность. Возможно, выгоднее прервать его запуск и обратиться к другим заданиям. Работа процедуры определяется параметром, который обозначается p_{break}. Если доля выполненной части задания превысила порог p_{break}, а ключевая задача не обнаружена, то запуск задания прерывается (задание включается в множество $\mathcal{J}_{\mathrm{break}}$).

Процедура RunDecision. Рассмотрим следующую процедуру выбора запускаемых заданий. Упорядочим множество заданий согласно некоторому «рейтингу» r_{run} (параметр процедуры), который вычисляется в момент принятия решения о запуске. Множество запускаемых заданий $\mathcal{J}_{\mathrm{run}}$ составляется из первых n заданий с наибольшим рейтингом. Величина n зависит от целочисленного параметра процедуры n_{run}:

$$n = \begin{cases} N\,, & \text{если } n_{\mathrm{run}} = 0\,; \\ \min\{n_{\mathrm{run}}, N, M\}\,, & \text{если } n_{\mathrm{run}} > 0\,, \end{cases}$$

где N — количество заданий, ожидающих запуска; M — число свободных компьютеров.

Процедура CheckDecision. Пусть j — выполняемое задание. Время выполнения пакета p, имеющего длину $p.l$ на процессоре $c \in j.\mathcal{C}$ составляет $(j.f).\mu \cdot (p.l)/c.r$, где $(j.f).\mu$ — среднее время выполнения элементарной задачи для данного задания (на стандартном процессоре), а $c.r$ — производительность компьютера. Потребуем, чтобы указанное время было одинаковым для всех компьютеров из множества $j.\mathcal{C}$, и обозначим его через ϑ. Учтем ограничения на размеры пакетов, связанные с емкостью компьютеров: $1 \le p.l \le c.v$. Для возможных значений ϑ получим соотношения $(j.\mu) \cdot \max\limits_{c\in\mathcal{C}}(1/(c.r)) \le \vartheta \le (j.\mu) \cdot \min\limits_{c\in\mathcal{C}}((c.v)/(c.r))$. Таким образом, если обозначить левую и правую части последнего неравенства соответственно через m и M, то $\vartheta = m + k^1_{\mathrm{check}} M$, где $k^1_{\mathrm{check}} \in [0,\, 1]$ — параметр процедуры.

Если зафиксировано значение ϑ из указанного диапазона, то размер пакета p определяется как $p.l = \vartheta \cdot (c.r)/((j.f).\mu)$. Кроме того, значение ϑ является основанием для выбора контрольного времени проверки результатов посылки, которое устанавливается в момент запуска. Полагаем, что $j.t_{\mathrm{check}} = = k^2_{\mathrm{check}}\vartheta$, где k^2_{check} — еще один параметр процедуры.

Процедура PeriodDecision. Эта процедура характеризуется единственным параметром Δt_{dec}, который показывает, через какое время произойдет очередное принятие решения по управлению очередью заданий и выбором параметров запуска.

5.2 Целевая функция

Как изложено в предыдущем пункте, принятие решений центром управления определяется набором параметров

$$\mathcal{P} = \left(p_{\mathrm{break}}, r_{\mathrm{run}}, n_{\mathrm{run}}, k^1_{\mathrm{check}}, k^2_{\mathrm{check}}, \Delta t_{\mathrm{dec}}\right)\,,$$

входящих в определение процедур из обработчика $\mathrm{H}_{\mathrm{decision}}$. Соответственно, оптимизация работы центра управления сводится к оптимизации выбора этих параметров.

Любой из параметров $p \in \mathcal{P}$ может, вообще говоря, изменяться в ходе работы системы, и в общем случае зависимость от времени может быть не только явная, но и опосредованная через зависимость от предыстории наблюдаемых параметров системы. Совокупность $\{p = p(t),\ p \in \mathcal{P}\}$ зависимостей параметров от времени будем обозначать буквой $\mathcal{S}$ и называть стратегией центра управления.

Для того чтобы охарактеризовать эффективность стратегии центра управления, введем следующие обозначения:

$x_1(t)$ — количество найденных ключевых задач к моменту t;

$x_2(t)$ — количество выполненных к моменту t заданий, в которых не оказалось ключевой задачи;

$x_3(t)$ — общее количество решенных к моменту t элементарных задач;

$x_4(t)$ — количество незавершенных заданий, которые превысили время жизни и были удалены из системы к моменту t.

Определим «доход» за стратегию $\mathcal{S}$ формулой

$$w(\mathcal{S}) = \sum_{i=1}^{4} \mathcal{C}_i \varliminf_{T\to\infty} T^{-1} \sum_{t=0}^{T} \mathbf{E}_{\mathcal{S}} x_i(t)\,,$$

где $\mathbf{E}_{\mathcal{S}}$ — оператор усреднения, порожденный стратегией $\mathcal{S}$, а $\mathcal{C}_i$ — весовые коэффициенты, соответствующие «стоимости» различных показателей x_i.

Требуется найти стратегию, обеспечивающую доход, близкий к величине (или равный) $w = \sup_{\mathcal{S}\in\mathbf{S}} w(S)$, где $\mathbf{S}$ — некоторое заданное множество стратегий.

5.3 Адаптивная стратегия

Ограничимся задачей оптимизации целевой функции на следующем множестве статических стратегий $\mathbf{S}$. Предположим, что множества допустимых значений каждого из параметров $p \in \mathcal{P}$ конечны. Пусть s_p означает вероятностное распределение на множестве значений параметра p, $\mathcal{S} = \{s_p,\ p \in \mathcal{P}\}$. Совокупность $\mathcal{S}$ можно трактовать как стратегию, при которой выбор параметра p каждый раз осуществляется в виде реализации случайной величины с распределением s_p. Положим $\mathbf{S} = \{\mathcal{S}\}$.

Стратегия S может быть представлена в виде вектора в конечномерном пространстве. Соответственно, максимизацию функции $w(\mathcal{S})$ на множестве $\mathbf{S}$ можно трактовать как классическую задачу оптимизации и решать ее, например, с помощью алгоритма проекции градиента. Однако в данном случае такой подход затруднен, поскольку невозможно получить явное аналитическое представление функции w и ее производных. Выход заключается в том, чтобы вместо точного значения градиента использовать его оценку по результатам наблюдений за имитируемой траекторией. И на этом пути имеются принципиальные трудности, в частности, из-за того, что значения функции w, определяемой как предел, не наблюдаемы. Для построения алгоритмов максимизации функции w, в которых вектор $\mathcal{S}$ пересчитывается на основе доступных наблюдений, можно воспользоваться теорией, изложенной в [4]. Подобные алгоритмы, которые действуют в условиях минимальной априорной информации об объекте, трактуются как адаптивные стратегии управления частично наблюдаемыми марковскими цепями. Изложение этих алгоритмов не предусмотрено форматом статьи, поэтому ограничимся численным примером использования адаптивной стратегии для оптимизации дохода в построенной модели вычислительного комплекса.

6 Пример

6.1 Параметры модели

Имеются 10 вычислительных устройств c_i, $i = 1, \ldots, 10$, и 2 потока заданий, f_1 и f_2. Эти объекты имеют следующие параметры.

Емкость: $c_1.v = 100$, $c_2.v = \cdots = c_5.v = 50$, $c_6.v = \cdots = c_{10}.v = 10$.

Производительность: $c_1 = 10$, $c_2 = c_3 = 5$, $c_4 = c_5 = 2$, $c_6 = \cdots = c_{10} = 1$.

Распределение времени пребывания в исправном состоянии: $c_i.\mathrm{H}_0$ — экспоненциальное с параметром 0,01. Вероятность сбоя в исправном состоянии: $c_i.P_{01} = 1$, $i = 1, \ldots, 10$. Распределение времени пребывания в состоянии сбоя: $c_i.\mathrm{H}_1$ — экспоненциальное с параметром 0,5. Вероятность восстановления после сбоя: $c_i.\mathrm{P}_{10} = 1, i = 1, \ldots, 10$. (Таким образом, предполагается, что компьютеры время от времени оказываются в несправном состоянии, а затем самопроизвольно восстанавливаются.)

Распределение промежутка между поступлениями заданий: $f_1.\mathcal{D}_{\mathrm{in}}$ — экспоненциальное с параметром 0,7; $f_2.\mathcal{D}_{\mathrm{in}}$ — экспоненциальное с параметром 0,3.

Распределение времени жизни задания ($f_1.\mathcal{D}_{1t}$ и $f_2.\mathcal{D}_{1t}$): равномерное на отрезке [10, 30].

Параметры $f_1.\mathcal{D}_{\mathrm{key}}$ и $f_2.\mathcal{D}_{\mathrm{key}}$: в заданиях из потока f_1 всегда присутствует (единственная) ключевая задача, ее местонахождение в задании распределено равномерно и заранее неизвестно; в заданиях из потока f_2 ключевые задачи отсутствуют. Принадлежность заданий какому-либо потоку заранее не известна.

Распределение длины заданий ($f_1.\mathcal{D}_{\mathrm{len}}$ и $f_2.\mathcal{D}_{\mathrm{len}}$): равномерное на отрезке $[10^4,\ 2 \cdot 10^4]$.

Распределение времени подготовки задания к запуску ($f_1.\mathcal{D}_{\mathrm{prep}}$ и $f_2.\mathcal{D}_{\mathrm{prep}}$): фиксированное значение, равное 0,01.

Распределение времени передачи пакета ($\mathcal{D}_{\text{send}}$): равномерное на отрезке $[10^{-3},\, 2\cdot 10^{-3}]$.

Время выполнения пакета моделируется следующим образом. Пусть $g = (p.j).f$ —поток, породивший задание, из которого сформирован пакет длиной $p.l$. Пусть $\tilde{z}$ — сумма $p.l$ независимых положительных случайных слагаемых со средним значением $g.\mu = 1{,}5\cdot 10^{-3}$ и дисперсией $g.\sigma = 10^{-4}$, а z — случайная величина с нормальным распределением, аппроксимирующим распределение $\tilde{z}$. Тогда $p.t_{\text{exec}} = \max(0,\, z)$.

Константа t_ε в процедуре PeriodDecision равна 10^{-7}.

Коэффициенты целевой функции: $\mathcal{C}_1 = 20, \mathcal{C}_2 = = 2, \mathcal{C}_3 = 10^{-3}, \mathcal{C}_4 = 10$.

Множества значений параметров, отвечающих за принятие решений и определенных в п. 5.1, представляются в виде конечных наборов следующим образом.

Порог прерывания запуска: $p_{\text{break}} \in \in \{0{,}5; 0{,}6; 0{,}7; 0{,}8; 0{,}9; 1\}$.

Рейтинг предпочтения выбора задания для запуска: $r_{\text{run}} \in \{R_{\max rs}, R_{\min rs}, R_{\max crs}, R_{\min crs}$. Здесь $R_{\max rs}$ ($R_{\min rs}$) — упорядочение по наибольшему (наименьшему) числу невыполненных задач; $R_{\max crs}$ ($R_{\min crs}$) — упорядочение по наибольшему (наименьшему) относительному числу невыполненных задач по отношению к первоначальной длине задания.

Параметр, определяющий количество одновременно запускаемых заданий: $n_{\text{run}} \in \{0, \ldots, 4\}$.

Параметры, определяющие контрольное время проверки результатов посылок: $k^1_{\text{check}} \in \{0, \ldots, 4\}$, $k^2_{\text{check}} \in \{1; 1{,}5; 2; 2{,}5; 3\}$.

Интервал между последовательными принятиями решений центром управления: $\Delta t_{\text{dec}} \in \in \{0{,}1; 0{,}2; \ldots; 1\}$.

Выбор значений оптимизируемых параметров в процессе имитации производится с помощью распределений на указанных множествах значений.

6.2 Эксперимент

Достижение цели, указанной в п. 5.2, сводится к поиску максимума целевой функции на произведении симплексов $\mathbf{S}$ общей размерностью $6+4+5+5+ +10 = 30$. Традиционными методами оптимизации воспользоваться невозможно, поскольку нет явного аналитического представления функции, а ее значения принципиально не наблюдаемы при имитации. Интуитивные соображения относительно расположения точки максимума отсутствуют, поскольку система является достаточно сложной и гетерогенной. В этих условиях адаптивный подход представляется единственно возможным способом рационального поведения.

Теоретически адаптивная стратегия сходится в пределе к максимуму [4], и это подтверждается экспериментально на более простых примерах, в которых доступно нахождение решения иными методами. Поскольку в данном случае «правильный ответ» не известен по указанным выше причинам, то пришлось сравнивать показатели адаптивной стратегии с результатами, полученными случайным перебором точек из континуального множества $\mathbf{S}$. Заметим, что последний метод крайне неэффективен, потому что «испытание» каждой точки требует прогона имитационной траектории, а одних только крайних точек множества $\mathbf{S}$ насчитывается $6 \cdot 4 \cdot 5 \cdot 5 \cdot 10 = 6000$.

Эксперимент проводился на персональном компьютере с тактовой частотой 2,6 ГГц. Работа адаптивного алгоритма заняла около 10 мин моделирования траектории смены событий в системе и завершилась получением устойчивого значения стратегии (точки) из множества $\mathbf{S}$ (с точностью до третьего знака в вероятностях выбора значений параметров). Эта стратегия обозначена через $\mathcal{S}_{\text{адап}}$. Случайный перебор производился в течение примерно 3 ч. Полученные этим способом наилучшая и наихудшая стратегии обозначены соответственно $\mathcal{S}_{\text{макс}}$ и $\mathcal{S}_{\text{мин}}$. Показатели, соответствующие различным стратегиям, приведены в табл. 2, в которой данные по решенным и потерянным заданиям и задачам приведены в процентах к их общему количеству, поступившему за время моделирования.

Как видно из табл. 2, разброс между наилучшим и наихудшим вариантами, полученными случайным перебором, оказался достаточно велик. Это означает, что сама постановка оптимизационной

Таблица 2 Показатели адаптивной и переборной стратегий

Стратегия	Целевая функция	Решено ключевых задач, %	Выполнено заданий, %	Решено задач, %	Потери, %
$\mathcal{S}_{\text{адап}}$	47,76	68,75	96,71	63,88	3,22
$\mathcal{S}_{\text{макс}}$	42,09	67,49	94,58	69,08	5,39
$\mathcal{S}_{\text{мин}}$	25,40	48,15	68,41	35,36	31,55

задачи оправдана, поскольку система существенно управляема с помощью выбранных параметров. (Более того, в ходе эксперимента выяснилось, что целевая функция и другие показатели оказались чувствительными к изменению любого из выбранных шести управляющих параметров при фиксированных остальных.)

Самый важный итог эксперимента заключается в том, что с помощью случайного перебора, занявшего на порядок больше времени, получены результаты хуже, чем с помощью адаптивной стратегии. Это относится ко всем показателям, кроме второстепенного «общего количества решенных задач».

Интересно также отметить, что стратегия $\mathcal{S}_{\text{адап}}$, полученная в результате работы адаптивного алгоритма, оказалась внутренней точкой множества $\mathbf{S}$, т. е. она предписывает выбирать управляющие параметры с помощью невырожденных распределений.

7 Заключение

Рассмотрена проблема эффективного управления процессом обслуживания заданий в специализированном вычислительном комплексе. Использованный подход заключается в построении имитационной модели объекта с последующим использованием адаптивных алгоритмов. Имитационная модель отражает необходимые аспекты устройства и функционирования системы, которая является слишком сложной для исследования чисто математическими методами. В модели выделены управляющие параметры, значения которых влияют на производительность вычислительного комплекса. В ходе реализации имитационной траектории осуществляется адаптивная коррекция параметров, которая приводит к оптимизирующим значениям.

При моделировании были использованы специальные определения и подходы. В частности, существенную роль сыграли конструкции языка взаимодействующих последовательных процессов Хоара, которые были модернизированы с учетом потребностей имитационного моделирования. Примененные средства описания имитационной модели обладают такими преимуществами, как компактность и наглядность, удобство для перевода в компьютерный код, возможность трансформации и развития модели.

Оптимизационные алгоритмы основаны на результатах адаптивного варианта теории частично наблюдаемых управляемых марковских цепей. Особенностью их работы является отсутствие необходимости в априорной информации о характеристиках объекта. Вычислительный эксперимент показал, что адаптивные стратегии способны решать многомерную оптимизационную задачу с фактически неизвестной и ненаблюдаемой целевой функцией.

Сфера применения представленной в работе методологии не ограничивается конкретным примером рассмотренного вычислительного комплекса. Аналогичным образом можно действовать с целью оптимизации самых разнообразных информационно-телекоммуникационных, производственных и иных систем.

Литература

1. *Коновалов М. Г., Малашенко Ю. Е., Назарова И. А.* Модели и методы управления заданиями в системах распределенных вычислительных ресурсов. — М.: ВЦ РАН, 2009. 110 с. (Сообщения по прикладной математике.)
2. *Голосов П. Е., Козлов М. В., Малашенко Ю. Е., Назарова И. А., Ронжин А. Ф.* Модель системы управления специализированным вычислительным комплексом. — М.: ВЦ РАН, 2010. 48 с. (Сообщения по прикладной математике.)
3. *Коновалов М. Г., Малашенко Ю. Е., Назарова И. А.* Управление заданиями в гетерогенных вычислительных системах // Известия РАН. Теория и системы управления, 2011. № 2. С. 72–90.
4. *Коновалов М. Г.* Методы адаптивной обработки информации и их приложения. — М.: ИПИ РАН, 2007. 212 с.
5. *Хоар Ч.* Взаимодействующие последовательные процессы. — М.: Мир, 1989.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 49–58

ВЫЯВЛЕНИЕ ИМПЛИЦИТНОЙ ИНФОРМАЦИИ ИЗ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ: ПРОБЛЕМЫ И МЕТОДЫ

И. П. Кузнецов[1], Н. В. Сомин[2]

Аннотация: Рассматривается семантико-ориентированный лингвистический процессор (ЛП), осуществляющий глубинный анализ текстов естественного языка (ЕЯ) и формирующий на этой основе структуры знаний. Одно из направлений развития таких процессоров связано с выявлением имплицитной информации, которая рассматривается в узком плане — как извлечение из текстов информационных объектов, их свойств и связей, заданных в неявном виде. Предлагаются методики, обеспечивающие такое извлечение на различных уровнях анализа текстов — лексико-морфологическом, синтактико-семантическом и структурном.

Ключевые слова: лингвистические процессоры; извлечение знаний; имплицитная информация

1 Постановка задачи выявления имплицитной информации

1.1 Цели проекта «Лингво-ИИ»

На протяжении 20 лет в ИПИ РАН развивается направление, связанное с автоматической обработкой потоков (корпусов) текстов на ЕЯ с целью выявления из текстов информационных объектов, их свойств и связей. В результате формируются структуры, которые служат основой для выполнения различных видов объектных (семантических) поисков, а также экспертных решений, составляющих круг задач пользователей [1–5]. Это направление связано с формализацией текстов и относится к области ***извлечения знаний*** (*knowledge extraction*). При этом важно, чтобы знания были представлены в форме, предусматривающей характер последующей обработки.

Следует учитывать особенности ЕЯ, носители которого обладают такими возможностями, до моделирования которых науке еще нужно пройти очень большой путь. Это, прежде всего, видение мира. За текстами ЕЯ человек видит картины внешнего мира, которые несут гораздо больше информации, чем сам текст. Человек способен по отдельным компонентам, присутствующим в тексте, восстанавливать эти картины, дополнять их, использовать причинно-следственные зависимости для прослеживания последующих изменений, динамики. Такая возможность выходит далеко за рамки моделей, основанных на логическом выводе. Отсюда следует особенность текстов ЕЯ. Как правило, в них умалчивается то, что известно адресатам, для которых предназначен текст, и что легко восстанавливается по тексту.

Другими словами, большое количество нужной пользователю информации дается в текстах ЕЯ в скрытом виде. Такая ***информация*** называется ***имплицитной***. Помимо этого, в текстах имеет место множество ***неопределенностей***, когда имеет место несколько вариантов анализа и требуется выбор одного из них. Многие неопределенности человек просто не замечает, но при автоматизации требуется разработка специальных методик и процедур для их разрешения. Важной научной и практической проблемой в области извлечения знаний из текстов ЕЯ является представление такой информации в явном виде: преобразование имплицитной информации в эксплицитную и устранение возникающих при этом неопределенностей, что является важным фактором в плане повышения качества решения пользовательских задач.

В связи с многоплановостью проблемы ее решение возможно только при существенных ограничениях. Речь будет идти о такой имплицитной информации, которую можно восстановить путем глубинного анализа текстов и логического вывода. Экстралингвистическая информация останется за пределами рассмотрения.

Работа является логическим продолжением исследований, имеющих целью создание нового класса интеллектуальных систем, основанных на автоматической формализации текстов ЕЯ с формированием структур знаний для решения логико-аналитических задач, по проектам ИПИ РАН «Криминал», «Аналитик», «Поток», «Лингвопроцессор». В рамках этих проектов созданы новые

[1]Институт проблем информатики Российской академии наук, igor-kuz@mtu-net.ru
[2]Институт проблем информатики Российской академии наук, somin@post.ru

методы формализации и извлечения знаний из текстов ЕЯ, разработан и постоянно совершенствуется уникальный ***семантико-ориентированный ЛП***, выделяющий информацию для пользователей, которые интересуются конкретными объектами, их свойствами и связями (другое название — объектно-ориентированный ЛП). Такая информация отображается на структуры знаний. Лингвистический процессор реализован средствами языка ДЕКЛ и управляется лингвистическими знаниями (ЛЗ) в виде предметных словарей, средств параметрической настройки, а также правил выделения объектов и связей [1–5]. С помощью ЛЗ осуществляется настройка ЛП на соответствующие категории пользователей и корпусы текстов. В результате возникает конкретная реализация. Таким образом, речь идет о средствах построения класса процессоров нового типа.

Проект «Лингво-ИИ» ставит целью дальнейшее развитие таких процессоров, совершенствование методик и средств автоматизации для более точного и полного выявления объектов, их признаков и связей, устранения неопределенностей на всех уровнях формализации, дополнения структур знаний новой информацией, отсутствующей или заданной в неявном виде.

1.2 Виды имплицитной информации

В проекте «Лингво-ИИ» затрагивается только та часть имплицитной информации, которая поддается автоматизации в рамках процедур, обеспечивающих работу ЛП и решение задач на основе технологии баз знаний. В реальности имплицитная информация далеко выходит за рамки такой интерпретации.

Понятие «*имплицитный*» возникло от латинского слова *implicito*, которое переводится как «*внутри заложенное*», и применительно к информации означающее «*скрытый, подразумеваемый, неявный*». В лингвистике ***имплицитной*** называется ***информация***, которая в явном виде не выражается, но извлекается адресатом при интерпретации сообщения [6, 7]. Существуют различные подходы к классификации имплицитной информации. В частности, некоторые из них представлены в [6–11]. Рассмотрим, какие виды имплицитной информации различаются в лингвистике.

Пресуппозиция — это термин лингвистической семантики. Различают семантическую, прагматическую и лексическую пресуппозиции [7, 11]. Семантическая пресуппозиция (в логике — импликация) предполагает элементы логического вывода для порождения новых знаний на основе имеющейся информации. Как правило, такое порождение осуществляется на уровне суждений или фактов, которые описываются на ЕЯ с помощью глаголов и управляемых ими форм. Со многими глаголами связаны действия, которые вызывают определенные изменения ситуации. Например, «*Купить вещь*» (означает, что вещь будет у субъекта действия, но количество денег у него уменьшится), «*Взять книгу у* A_1» (означает, что книга будет у субъекта действия и ее не будет у A_1) и т. д. Описанные изменения задаются с помощью правил, которые являются основой логического вывода. Другие примеры: «*Мы работаем, чтобы сохранить Ваше доверие*» (означает, что такое доверие было), «*Воссоединение Белоруссии и России*» (означает, что раньше они были вместе). Глаголы типа «*видеть*», «*знать*» подразумевают истинность суждения и т. д.

Прагматическая пресуппозиция учитывает знания и убеждения адресата. Суждение P является прагматической пресуппозицией суждения S, если, высказывая суждение S, адресант считает P само собой разумеющимся и известным адресату. Лексическая пресуппозиция предполагает выводы на уровне лексического анализа. Например, из «*истерический*» следует «*нервный*», «*больной*», «*псих*» и т. д.

Анафоры (от греч. *anapheren* — относить назад) являются разновидностью имплицитной информации. Они задаются в текстах с помощью анафорических местоимений, связок «*тот, который*», кратких имен и отличительных свойств. Например, «. . . *Медведев*. . . *Он* (или *президент*, или *который*). . . ». Разрешение анафор — это соотнесение местоимений с соответствующими лицами или объектами. Различаются синтаксические анафоры, для разрешения которых достаточно морфологических признаков, и семантические анафоры, где учитываются семантические категории слов и возможность участия соответствующих объектов в тех или иных действиях.

Коммуникативные импликатуры учитывают коммуникативное воздействие языка на человека [8]. Это, прежде всего, жанровые и стилистические смещения, которые в наибольшей степени проявляются в скрытой рекламе. Например, когда рекламное сообщение о лекарствах маскируется под рекомендации врача или больной говорит об их положительном воздействии при лечении. Коммуникативные импликатуры при манипулировании сознанием человека учитывают многие его свойства. Человек лучше запоминает информацию в начале и в конце текстового материала, при повторах. Критичность к сообщению снижается, если имеет место доверие к носителю информации, если оно по каким-то причинам нравится (эффект эмоциональности), если человек предрасположен к ее восприятию.

Подобная классификация далеко не полная. Но она иллюстрирует всю сложность языка и его восприятия. Многие виды имплицитной информации доступны только для человека. У компьютера нет фоновых знаний (как у человека). Компьютером невозможно манипулировать перечисленными выше способами. Вне сферы автоматизации остаются метафоры, аналогии, многие сравнительные конструкции и др. Поэтому и само понятие «имплицитный» трансформируется с учетом возможностей и задач ЛП и баз знаний.

1.3 Проблемы извлечения имплицитной информации

Проект «Лингво-ИИ» направлен на разработку методик автоматического извлечения имплицитной информации в рамках существующего инструментария — языка ***расширенных семантических сетей*** (РСС) и средств их обработки (***язык ДЕКЛ***). Язык РСС состоит из фрагментов, которые в простейшем случае имеют вид предикатов. В отличие от предикатов каждый фрагмент имеет свой уникальный код, который может стоять на аргументных местах других фрагментов. Это необходимо для представления семантических составляющих ЕЯ, когда действия включают в себя объекты или другие действия и т. д. Возникают сложные структуры, выходящие за рамки языка логики предикатов. При этом логический вывод осуществляется с помощью правил преобразования таких структур, реализованных в инструментальной среде ДЕКЛ [3].

Понятие ***имплицитный*** рассматривается с точки зрения дополнения и уточнения информационных объектов и связей, которые выделяются ЛП в процессе формализации текстов ЕЯ и которые необходимы для решения задач. Остаются в стороне многие виды пресуппозиций, коммуникативные импликатуры и др. При этом акцент смещается в сторону импликатур, которые порождаются с помощью логического вывода, осуществляемого путем анализа и преобразования структур знаний.

Отметим два важных момента. Во-первых, на основе логического вывода осуществляется принятие многих решений, в том числе экспертных. В результате формируются экспертные знания, которые в явном виде не присутствуют в текстах документов и которые будем считать разновидностью имплицитной информации. А во-вторых, при работе ЛП (на всех уровнях анализа) требуются специальные методики для автоматического устранения разного рода неопределенностей — лексической, морфологической, синтаксической и семантической. Это необходимо для повышения качества работы ЛП при формировании структур знаний, на основе которых выявляется имплицитная информация.

Итак, автоматическое извлечение имплицитной информации связано с решением ряда достаточно сложных лингвистических задач: выявлением подразумеваемых объектов и связей, идентификацией на основе анафорических ссылок, разрешением различного рода полисемии и неопределенностей и др. Для этих задач требуются нетривиальные механизмы принятия решений и соответствующая техника логического вывода. Их наличие существенно повышает научный уровень исследований в области создания ЛП. Для уточнения задач рассмотрим структуру ЛП, разрабатываемых в ИПИ РАН.

Семантико-ориентированный ЛП состоит из четырех основных компонентов.

1. ***Блок лексико-морфологического анализа*** (ЛМА). Выделяет из документа слова и предложения и выдает в виде семантической сети, представляющей собой последовательность компонентов (слов в нормальной форме, чисел, знаков) и их основные признаки — лексические, морфологические и др. [12, 13]. Такая сеть названа ***пространственной структурой*** (***ПС***) ***документа***. Более того, блок использует набор предметных словарей (стран, регионов России, имен, профессий и др.) для придания словам и словосочетаниям дополнительных семантических признаков [14].

2. ***Блок синтактико-семантического анализа*** (ССА). Путем анализа ПС документа он выделяет объекты и связи. На их основе строит другую семантическую сеть, представляющую семантическую структуру (СС) документа, называемую ***содержательным портретом документа*** [2–5]. В СС документа представляются не только объекты и связи, но и их участие в действиях, из каких предложений взяты тексты их описания и многое другое. По СС документа можно восстановить сам текст.

Содержательные портреты образуют структуры знаний, которые запоминаются в базе знаний. Блок управляется ЛЗ, за счет которых обеспечивается: извлечение информационных объектов (лиц, организаций, событий, мест и др.), выявление связей объектов (каким образом лица связаны с организациями, адресами и др.), анализ глагольных форм, причастных и деепричастных оборотов с выявлением фактов участия объектов в тех или иных действиях, идентификация объектов (с учетом анафорических ссылок и сокращенных наименований), выявление связей действий с местом или временем, анализ причинно-следственных и временны́х связей между действиями и событиями.

Этот блок включает в себя базу ЛЗ, которая содержит правила анализа текста во внутреннем представлении (РСС). Они определяют работу ЛП.

3. ***Блок экспертных решений***. Анализирует структуры знаний, решает логико-аналитические задачи и формирует дополнительную (экспертную) информацию, необходимую пользователю.

4. ***Обратный лингвистический процессор*** (ОЛП). Преобразует структуры знаний в тексты ЕЯ, которые должны быть выданы пользователю.

Выявление имплицитной информации и устранение неопределенностей осуществляется (в рамках ЛП) на всех уровнях преобразования текстов документов в СС документов с их последующей обработкой.

1.4 Задачи выявления имплицитной информации

Автоматическое выявление имплицитной информации на основе анализа текстов ЕЯ и разработанных методов их формализации требует проведения следующих работ.

- Совершенствование блока ЛМА. Разработка методик (с доработкой соответствующих алгоритмов и программ) для устранения неопределенностей при следующих видах анализа:
  - разбиении текста на словоформы и предложения (неопределенности вызваны наличием в корпусах текстов лексем, содержащих буквы, цифры и разделители практически в произвольной последовательности);
  - присвоении словам морфологических и ряда семантических признаков за счет анализа составных частей словоформы (выделение фамилий);
  - ранжировании вариантов ЛМА (разрешение лексической полисемии);
  - присвоении словам семантических признаков на основе предметных каталогов (в случае наличия несколько вариантов такого присвоения, взятых из различных каталогов);
  - выделении объектов фиксированной структуры (адресов, мейлов, имен сайтов и др.);
  - приведении выделенных объектов к стандартной форме (для адресов).
- Разработка и реализация методик выявления объектов и их ролевых функций (потерпевший, преступник, террорист, сотрудник милиции и др.) по косвенным признакам и контексту. Создание правил такого выявления в структуре ЛЗ блока ССА. Проверка их работоспособности на документах области «Криминалистика».
- Разработка и реализация методик выявления объектов, заданных в неявном виде, при отсутствии характеристических признаков объекта. Использование предположений о возможном их появлении. Создание правил такого выявления в структуре ЛЗ блока ССА.
- Разработка и реализация методик выявления связей объектов путем предположения их наличия (например, если выявлена автомашина, то поиск ее обладателя и т. д.). Создание правил такого выявления. Совершенствование блока ССА для поддержки этих правил.
- Разработка методов идентификации объектов с учетом анафорических ссылок (местоимений) и их краткого описания. Создание правил идентификации в структуре ЛЗ. Совершенствование блока ССА и предметных словарей для поддержки этих правил.
- Исследование явления переноса объектов (когда объект отсутствует, но подразумевается) и возможности его реализации в рамках ЛП.
- Разработка и реализация методик анализа происшествий и событий, представленных в виде структуры знаний (СС документов), с выявлением их значимых признаков и особенностей, отсутствующих в тексте описания.
- Разработка экспертных систем, использующих структуры знаний для порождения новой информации об объектах. Создание соответствующей оболочки и ее применение для классификации организаций («Место учебы», «Место работы», «Курсы»), оценки степени знания языков и др.
- Разработка методик классификации объектов по текстам их описания на примере распознавания профессиональной области лица по описанию его функциональных обязанностей.
- Разработка ОЛП для выдачи объектов и результатов, представленных в виде РСС (в СС документов), на ЕЯ. Разработка блока, обеспечивающего выдачу описаний объектов в нормальной форме (в единственном числе, именительном падеже).

В данной статье рассматриваются методы решения ряда таких задач, предложенных в рамках проекта «Лингво-ИИ».

2 Методы и алгоритмы устранения лексической полисемии

2.1 Проблемы лексической полисемии

Читая текст, человек легко определяет в нем абзацы, предложения, лексемы и прочие элементы текста. Однако при разработке алгоритмов их автоматического распознавания возникают проблемы, вызванные наличием различного рода неоднозначностей. Например, знак «.» (точка) может выступать как конец предложения, как признак сокращения («*г.*», «*прил.*»), как инициалы в ФИО (типа «*И.*», «*А.*»), как разделитель целой и дробной части числа в английских текстах (*3.14*), как разделитель в датах, как элемент электронного или Интернет-адреса и в ряде других ролей.

В то же время для выявления имплицитной информации крайне важным является корректное определение начала и конца предложения или абзаца. Абзац является той максимальной рамкой, в которой имеет смысл искать имплицитную информацию для уже найденных объектов, но не имеющих достаточного количества характеристик. Для ряда важных характеристик такой рамкой служит более узкий контекст — предложение.

Однако именно аккуратное определение границ предложения является наиболее проблематичным. Как видно из приведенного примера, точка «.» не может служить надежным признаком конца предложения. Более того, в современных текстах в качестве признака конца предложения зачастую используются другие знаки. Это может быть «конец ячейки» таблицы, который при преобразованиях потерялся, или совершенно неожиданная комбинация символов. Отсюда следует необходимость разработки специальных методик.

2.2 Методики снятия неопределенности на лексическом уровне

Опыт разработки и использования ЛП показал, что главным способом борьбы с лексической полисемией является правильная классификация лексических единиц. Классификация должна помогать в решении основной задачи — выявления в тексте информационных (семантических) объектов. Но поскольку этот процесс многоуровневый, то хорошая классификация должна быть ориентирована не только на семантический анализ, но и на промежуточные уровни — морфологический и синтаксический анализ.

В разработанных ЛП классификация включает в себя более 20 лексических типов: слово из русских или латинских букв, в кавычках, с большой буквы или из больших букв, с точкой в конце и т. д.

Определение конца предложения осуществляется еще до лексического анализа и определения типологии лексем. Уточнение, является ли данная точка концом предложения, осуществляется после проведения морфологического анализа лексем с привлечением лексической и морфологической информации. Для этого в рамках блока ЛМА разработаны соответствующие рекурсивные алгоритмы.

Для корректной фиксации границ предложений, «точкой» не заканчивающихся, наиболее эффективным оказалось использование операторов настройки алгоритмов на особенности задачи и предметной области [12, 13, 15].

Примерами таких операторов являются следующие:

- NEW_SENT (произвольное число аргументов). Семантика: если указанное во фрагменте слово записано с прописной буквы и находится в начале строки текста, то оно рассматривается как начало нового предложения. Допустимы знаки «*», заменяющие окончание или указание части речи, типа *V, *T. Пример записи: NEW_SENT(ANALYSIS, ASSUR*). Действие: если слово «*Analysis*» или «*Assurance*» стоит в начале строки, то оно рассматривается как начало предложения;

- END_SENT (произвольное число аргументов). Семантика: если в тексте встречается одно из указанных слов (символов, знаков), то оно считается концом предложения. Пример записи: END_SENT(‘;’). Действие: точка с запятой «;» рассматривается как конец предложения;

- ABBR (произвольное число аргументов). Список сокращений с точками на конце, которые считаются цельными словами, и точки не рассматриваются как конец предложения. Пример записи: ABBR(Inc., Ltd.). Действие: словосочетания «*Inc.*» и «*Ltd.*» рассматриваются как сокращения;

- SEPARATOR (произвольное число аргументов). Семантика: указание символов, которые всегда являются разделителями. Пример записи: SEPARATOR (‘+’, ‘:’).

Полную систему операторов параметрической настройки можно найти в [12].

3 Методы устранения неопределенностей морфологического анализа

3.1 О проблеме морфологической омонимии

Выявление имплицитной информации связано с глубинным анализом текста ЕЯ. И немаловажную роль в этом процессе играет устранение омонимии морфологического анализа. Схема и особенности используемого в предлагаемом ЛП морфологического анализа описаны в [13, 16]. Дело в том, что сам по себе морфологический анализ принципиально омонимичен (многовариантен). Например, лексема «*стекло*» может означать и существительное, и глагол. Лексема «*связи*» дает несколько вариантов морфологического анализа с разными падежами и числом. Более того, для многих лексем возможно несколько вариантов морфологического анализа — их число может превышать 20. Случаи однозначного морфологического анализа являются исключениями. Однако человек умеет из всех вариантов уверенно выбирать единственно правильный. Для этого требуется анализ контекста — лексического, синтаксического, семантического и ситуационного. Ниже будут рассматриваться методы, которые используются в блоке ЛМА.

В первую очередь полнота морфологического анализа обеспечивается использованием широкой номенклатуры морфологических признаков (см. п. 3.2).

Другой используемый метод — комбинаторный анализ, заключающийся в определении только допустимых комбинаций. Для этого авторами разработаны алгоритмы, основанные на эвристических решениях, пусть не всегда безупречных, но срабатывающих в ряде самых значимых случаев (см. п. 3.3).

Однако наиболее эффективным методом устранения неопределенностей, как показала практика, является учет контекста. Для этого в блоке морфологического анализа широко используются средства частичного синтаксического анализа (см. п. 3.4).

3.2 Система морфологических признаков

Блок морфологического анализа обеспечивает выделение множества морфологических признаков — их более 100 (часть речи, род–число–падеж, форма глагола, указатель мейла и многое другое). Для одной лексемы этот блок выдает несколько признаков. Их набор и характеризует морфологический тип. Кроме чисто морфологических блок выдает еще несколько лексических признаков, а также ряд фонетических признаков, которые могут быть использованы для синтеза речи.

Разработанная в ИПИ РАН система морфологических признаков традиционна и в то же время обладает достаточной полнотой.

Особое место занимает признак «#». Он означает, что данный набор признаков сформирован «по аналогии», т. е. была найдена словоформа с таким же окончанием, как у данной лексемы, и набор признаков словоформы приписан данной лексеме. Варианты разбора «по аналогии» применяются для лексем, которых нет в морфологическом словаре.

3.3 Устранение морфологической омонимии методами комбинаторного анализа

Правило 1. Если есть два альтернативных варианта морфологического разбора, несовместимых между собой или практически несовместимых, то оставляется только один из них. Например, если один из вариантов разбора имеет признак «f» — фамилия, то все варианты с признаком «#» вычеркиваются.

Правило 2. После сравнения двух вариантов разбора один из них ранжируется как «старший», т. е. ставится на первое место. Отметим, что в принципе все варианты разбора равноправны. Однако для некоторых задач (например, генерации текстов) используется только один — старший вариант разбора. Поэтому далеко не безразлично, какой именно вариант станет старшим.

Например, если какой-либо вариант разбора имеет признак «г» — географическое название, то он ставится на первое место.

Отметим, что если ни одно из такого рода правил не срабатывает, то по умолчанию старший вариант разбора высчитывается по специальному алгоритму, учитывающему достоверность морфологического анализа.

Правило 3. Склеивание вариантов. Если варианты разбора совпадают с точностью до падежа, то они склеиваются в один вариант, где присутствуют оба падежа. Склеивание, по сути дела, является технической процедурой сокращения записи. Однако оно начинает играть существенную роль с учетом алгоритмов по первым двум правилам.

Опыт использования комбинаторных алгоритмов, которых разработано уже около двух десятков, показал их высокую эффективность.

3.4 Устранение неопределенностей методами синтаксического анализа

Другим эффективным методом устранения морфологической омонимии является использование элементов синтаксического анализа. Хорошо известно, что омонимию слова можно устранить в контексте словосочетаний. Так, если говорят «*большое стекло*», то вариант анализа последнего слова как глагола «*стекать*» отпадает. Исходя из этой идеи, было предложено:

- проверять на полное согласование (по роду, числу и падежу) существительное со стоящими перед ним прилагательными или причастиями. Если указанная связь обнаруживается, то у обеих лексем оставлять только варианты разбора, совпадающие по роду, числу и падежу;
- проверять на наличие «генитивной цепочки» существительное (или группу существительных) и стоящие за ним дополнения в родительном падеже. Если такая связь обнаруживается, то у дополнения оставлять варианты разбора только с родительным падежом.

Последнее правило можно проиллюстрировать следующим примером. Если слово «*связи*» стоит в словосочетании «*лейтенант связи*», то морфологически многозначное слово «связи» (это существительное в родительном, дательном, предложном падежах единственного числа и винительном, именительном падежах множественного числа) приводится к однозначному разбору — родительный падеж единственного числа.

Использование элементов синтаксического анализа для устранения морфологической омонимии является очень эффективным методом, резко повышающим качество разбора.

3.5 Особенности распознавания имен и фамилий

Используемый морфологический словарь содержит около 500 различных имен, отчеств и фамилий — как русских, так и иностранных. Однако ясно, что этого явно недостаточно для уверенного распознавания этих очень важных элементов текста. Поэтому в рамках «постморфологического» анализа действует специальная программа распознавания фамилий. Она основана на анализе окончаний и суффиксов, характерных для русских фамилий («*ов*», «*ев*», «*ин*», «*ын*» и др.), а также фамилий, часто встречающихся в русскоязычных текстах.

Были выявлены все встречающиеся в фамилиях суффиксы и с каждым суффиксом сопоставлена парадигма возможных окончаний. Отметим, что «суффиксы» и «окончания» — условные названия хвостов лексем, играющие определенную роль в распознавании фамилий.

Алгоритм программы сводится к следующему. Выявляется слово с прописной буквы. Для него в массивах окончаний ищется подходящее окончание, а для данного окончания — подходящий суффикс. Если эти проверки (плюс некоторые дополнительные) прошли успешно, то слову присваивается признак «f» — фамилия и с помощью суффикса формируется каноническая форма этой фамилии.

С помощью данного алгоритма удается выявить основную массу встречающихся в текстах русских фамилий (по предварительным оценкам — до 90%). Однако фамилии европейского типа или фамилии восточных и среднеазиатских народов (например, *Смит*, *Линкольн*, *Абу-Оглы*) этим алгоритмом не охватываются. Тем не менее, в связи с нарастающей глобализацией количество такого рода фамилий увеличивается с каждым годом. Поэтому в разработках ИПИ РАН применяются дополнительные словари тюркских и западных имен, которые увеличивают вероятность распознавания ФИО, но не могут охватить множество возможных вариаций.

4 Семантические методы извлечения имплицитной информации

Автоматическое извлечение из текстов ЕЯ имплицитной информации связано с решением целого ряда сложных задач: выявлением информационных объектов и связей, в том числе заданных в неявном виде; выявлением действий, в которых участвуют объекты; дополнением объектов новыми признаками на основе классификации и экспертных решений; идентификацией объектов путем анализа анафорических ссылок и др.

Решение данных задач осуществляется на синтактико-семантическом уровне: в процессе построения содержательного портрета (СС документа) и его последующего анализа.

Еще раз отметим, что качество решения во многом определяется блоком ЛМА — методами устранения неопределенностей (см. разд. 2 и 3). Любые ошибки и неоднозначности на этом уровне сказываются на решении вышеупомянутых задач.

4.1 Задача «оценки» и «окраски» информационных объектов

Задача «оценки» и «окраски» связана с порождением новых признаков или свойств информаци-

онных объектов на основе текстов ЕЯ. Например, оценка стабильности предприятия по информации из Интернета, окраска политических деятелей (положительная или отрицательная) в зависимости от высказываний в прессе, оценка качества изделия по высказываниям пользователей и т. д. Часто напрямую не говорится: это плохо, а это хорошо. Как правило, в текстах ЕЯ описываются события, ситуации, в которых участвовал тот или иной информационный объект. По ним и делается оценка, которая зачастую представляется в виде нового (порожденного) свойства объекта. Частным случаем этой задачи является выявление ролевых функций объектов.

Для решения данной задачи используются различные методы [10, 17]. Наиболее распространенный — метод выявления новых свойств объектов путем использования ***синтактико-семантических форм***. Например:

<что — лекарство> вызывает аллергию у <кого — человека>. . . ;

<что — лекарство> имеет побочные эффекты. . . ;

<кто — человек> учинил скандал. . . и т. д.

Применение таких форм к текстам ЕЯ заключается в поиске «оценочных» или «характеристических» слов (типа «*скандал*») или словосочетаний типа «*вызывает аллергию*» («*может вызывать аллергию*»), «*имеет побочные эффекты*» («*побочные воздействия*»), «*учинить скандал*» («*скандалить*»). . . И затем анализируется окрестность, т. е. слова, стоящие слева и справа, их семантические классы (по ним распознаются объекты) и падежные формы. В результате даются оценки информационных объектов. По первым двум формам — это «качество лекарств», а по последней — человек совершил «хулиганские действия» или что он «подозреваемый».

Использование синтактико-семантических форм связано с определенными трудностями, вызванными особенностями ЕЯ: наличием в текстах причастных, деепричастных оборотов, различных пояснений, факультативных компонентов (время, место, цель), анафорических ссылок и многого другого. В результате информационные объекты часто оказываются на значительном расстоянии от оценочных слов. Отсюда — значительные потери, влияющие на качество оценивания.

Пример 1 (текст взят из сводок происшествий ГУВД г. Москвы):

. . . Горелов Петр Сергеевич, 01.03.76 г/р, прож.: г. Москва, ул. Юных Ленинцев, д. 71-6-12, не работает, 01.02.1998 г. в 14.30 у своего дома из хулиганских побуждений в состоянии алкогольного опьянения учинил скандал и разбил оконное стекло в квартире Литвиновой Галины Ивановны, 20.07.1961 г/р. . .

В данном примере оценочные (характеристические) слова «*учинил скандал*» и «*разбил оконное стекло*» находятся на значительном расстоянии от оцениваемого лица — «*Горелов Петр Сергеевич*». Это ограничивает возможности применения форм. Требуется первоначальное выделение компонентов, которые не должны учитываться в формах: годов рождения, адресов, свойств («*не работает*», «*в состоянии алкогольного опьянения*»), времени, места и др., что предполагает достаточно глубокий анализ текста с выделением объектов, их свойств и атрибутов.

В связи со сказанным более перспективным представляется другой метод — когда оценивание осуществляется на уровне структур знаний. Для их построения используется семантико-ориентированный ЛП, который осуществляет глубинный анализ текстов ЕЯ с приведением синонимичных групп к одному виду, выявлением объектов и их свойств, идентификацией объектов, выявлением и унификацией различных форм, представляющих события или действия (в том числе форм с отглагольными существительными, причастных и деепричастных оборотов), которые связываются с временем и местом. В результате формируются структуры знаний, в которых объекты напрямую связываются с событиями и действиями, что исключает потери, о которых говорилось выше. Последующий анализ осуществляется с помощью правил языка ДЕКЛ, ориентированных на обработку таких структур (РСС), что делает простым процесс разработки программ «оценки» и реализации соответствующих правил анализа и вывода. При этом структура знаний не изменяется, а только пополняется новыми (полезными) фрагментами.

Проиллюстрируем предлагаемый метод применительно к задаче выявления ролевых функций лиц из сводок происшествий, взятых из области «Криминалистика». Имеется в виду задача присвоения лицам (по их участию в различного рода деяниях) свойств — «потерпевший», «подозреваемый» или «преступник», «заложник», если описание таких свойств отсутствует в тексте в явном виде. Например, если в тексте говорится «*потерпевший Иванов И. И.*», то возникает другая задача — выявление свойства в процессе лингвистического анализа и формирование соответствующего фрагмента в структуре знаний.

Как уже говорилось, в рамках предлагаемой методики (вместо применения синтактико-семантических форм к документам) используются правила логического вывода и преобразования структур знаний (СС документов), в которых нет морфо-

логических признаков (типа *кто*, *кого*. . .), но с помощью фрагментов РСС представлены объекты и их участие в действиях. Имена таких фрагментов представляют характер действий. Например, в примере 1, где фигурантом является «*Горелов Петр Сергеевич*», его свойства и деяния представляются в виде фрагментов:

ПЬЯНЫЙ(<код фигуранта>)
БЕЗРАБОТНЫЙ(<код фигуранта>)
УЧИНИТЬ(<код фигуранта>,СКАНДАЛ)
РАЗБИТЬ(<код фигуранта>,ОКОННЫЙ,СТЕКЛО).

Выявление ролевых функций фигуранта сводится к анализу таких фрагментов. Анализ осуществляется с помощью ***логико-семантической оболочки***, которая осуществляет необходимые преобразования фрагментов РСС и логический вывод. Оболочка состоит из продукций языка ДЕКЛ и управляется фрагментами РСС, образующими ***управляющие знания***. Пример управляющих фрагментов:

РАЗБИТЬ(ОКНО,СТЕКЛО,ДВЕРЬ. . .)
УЧИНИТЬ(ССОРА,СКАНДАЛ. . .)

Первый фрагмент означает, что если фигурант разбил окно, стекло или дверь, то ему присваивается свойство, связанное с этим фрагментом, например «подозреваемый». Правило реализуется в рамках оболочки, которая осуществляет поиск в СС документа фрагмента с именем РАЗБИТЬ и наличием в нем одного из аргументов — ОКНО, СТЕКЛО, ДВЕРЬ. . . Если данное условие выполняется, то к СС документа добавляется фрагмент ПОДОЗРЕВАЕМЫЙ(<код фигуранта>). Это простейший случай.

В более сложных случаях учитываются отрицания, отношения принадлежности, совокупность действий. Например, «. . . *ушла из дому*. . . *не вернулась*. . . » или «. . . *автомашина*. . . *под управлением*. . . *выехала на полосу встречного движения*. . . *произошло столкновение*. . . » В последнем происшествии в действиях участвует автомашина, а «нарушитель» — это человек, который ею управляет.

Отметим, что в приведенных примерах (для простоты понимания) фрагменты записаны в виде предикатов. В реальной системе каждый фрагмент РСС имеет свой уникальный код. Такие коды используются для представления классов слов, словосочетаний и указания их связи с ролевыми функциями.

4.2 Выявление объектов и связей, заданных в неявном виде

Выявление объектов и связей осуществляется в процессе ССА — преобразования ПС документа в структуру знаний, т. е. СС документа (см. п. 1.3). Такой анализ заключается в последовательном применении правил выделения объектов или их компонентов из текстов ЕЯ. Каждое правило ориентировано на выделение объектов определенного типа (фигурантов, адресов, организаций. . .). Выделение объектов начинается с поиска ***характеристических слов***. Например, для объектов типа «адрес» такими словами являются «*город*», «*улица*», «*дом*» и др. Далее анализируется окрестность этих слов, выбираются допустимые слова, которые и составляют объект.

Довольно часто характеристические слова в тексте отсутствуют — подразумеваются. В таких случаях возникают трудности выделения объектов. Например, если в тексте встречаются лица с иностранными ФИО. У английских фамилий (*Буш*, *Блэк*. . .) нет характерных суффиксов, как в русском языке. Более того, в качестве фамилии может фигурировать любое слово, называющее или определяющее какой-либо предмет внешнего мира. При анализе текстов ЕЯ такие фамилии вносят элементы неопределенности — омонимии. В азиатских языках компоненты ФИО — это просто слова с большой буквы (*Ден Сяо Пин*, *Хун Вай*. . .). В таких ФИО отсутствуют характеристические слова. Требуются другие методики выделения. Аналогично адреса могут иметь вид — «*Семеновская, 2-44*». Сказанное относится и к другим объектам.

Для выявления объектов без характеристических слов предлагается методика, основанная на принципе ***ожидания***. Учитывается тот факт, что часто в ЕЯ после одних слов или объектов ожидается наличие других. Например, если после слова «*инженер*» стоит слово с большой буквы, то, скорее всего, оно относится к ФИО. Таким образом, начинается выделение объектов, у которых не распознаны компоненты ФИО.

Реализация соответствующей методики осуществляется в процессе ССА. При этом используется оператор следующего вида:

$$\mathrm{GO\_}(\langle\text{правило 1}\rangle, \langle\text{правило 2}\rangle)\,,$$

где правило 1 выявляет в тексте соответствующий объект. И если оно применилось (объект выявлен), то вызывается правило 2, выявляющее ожидаемый объект.

Методика «ожидания» используется и при выделении ***связей между объектами***, которые в явном виде не задаются. В текстах ЕЯ многие связи подразумеваются и привязаны к типу выявленных объектов. Например, если выявлен адрес, то, скорее всего, он относится к какому-либо определенному лицу (или учреждению), которое нужно искать.

При результативном поиске формируется новая связь.

На этом основана методика формирования новых связей. Она заключается в следующем. В процессе анализа текста строятся «временные» фрагменты, представляющие связи выявленных объектов с пока что неизвестными объектами, которые специальным образом отмечаются. В дальнейшем осуществляется их поиск. Если соответствующий объект не найден, то «временный» фрагмент удаляется из СС документа. Если найден, то фрагмент остается и вводится в структуру СС документа.

Поиск неизвестных объектов осуществляется на одном из этапов ССА и управляется с помощью фрагментов, посредством которых задается направление поиска, число шагов и условия окончания поиска — недопустимые слова, знаки или объекты.

Более детализированное описание методики, а также другие семантические методы выявления имплицитной информации предполагается рассмотреть в последующих работах.

5 Заключение

Автоматическое извлечение из текстов ЕЯ имплицитной информации — это область искусственного интеллекта, связанная с развитием моделей языка, ЛП, методов устранения неопределенностей и принятия решений. Успешное решение этой сложнейшей задачи возможно лишь при комплексном подходе, когда анализ не сосредоточен в какой-то одной точке, а совершается постоянно, на всех уровнях работы ЛП.

В данной статье рассмотрен ряд методик, позволивших существенно продвинуться в данном направлении [18].

Практическая ценность выполненных работ определяется возрастающей потребностью автоматической формализации быстро растущих потоков документов на ЕЯ, особенно в среде всемирной сети Интернет.

Литература

1. *Kuznetsov I., Kozerenko E.* The system for extracting semantic information from natural language texts // Conference (International) on Machine Learning (MLMTA-03) Proceedings. — Las Vegas, 2003. P. 75–80.
2. *Кузнецов И. П.* Семантико-ориентированная система обработки неформализованной информации с выдачей результатов на естественном языке // Системы и средства информатики. — М.: Наука, 2006. Вып. 16. С. 235–253.
3. *Кузнецов И. П., Мацкевич А. Г.* Семантико-ориентированные системы на основе баз знаний. — М.: МТУСИ, 2007. 173 с.
4. *Кузнецов И. П.* Объектно-ориентированная система, основанная на знаниях в виде XML-представлений // Системы и средства информатики. — М.: Наука, 2008. Вып. 18. С. 96–118.
5. *Kuznetsov I. P., Kozerenko E. B.* Linguistic processor Semantix for knowledge extraction from natural texts in Russian and English // Conference (International) on Artificial Intelligence (ICAI 2008) Proceedings. — Las Vegas: CSREA Press, 2008. P. 835–841.
6. *Падучева Е. В.* Высказывание и его соотнесенность с действительностью. — М.: Наука, 1985.
7. *Кондрашова Д. С.* К проблеме классификации типов имплицитной информации // Cognitive Modelling in Linguistics: Мат-лы VIII Междунар. конф. — Варна, 2005. Т. 1. С. 245–252.
8. *Пирогова Ю. К.* Имплицитная информация как средство коммуникативного воздействия и манипулирования // Проблемы прикладной лингвистики. — М., 2001. С. 209–227.
9. *Asher N., Lascarides A.* Logics of conversation. — Cambridge: Cambridge University Press, 2003.
10. *Clark P., Harrison P., Thompson J.* A knowledge-driven approach to text meaning processing // HLT-NAACL 2003 Workshop on Text Meaning Proceedings, 2003. P. 1–6.
11. *Анохина Н. В.* Роль пресуппозиции и импликации в процессе понимания научно-популярного текста // Вестник Башкирского ун-та, 2009. Т. 14. № 1. С. 92–94.
12. *Кузнецов И. П., Сомин Н. В.* Средства настройки семантико-ориентированной системы на выделение и поиск объектов // Системы и средства информатики. — М.: Наука, 2008. Вып. 18. С. 119–143.
13. *Сомин Н. В., Кузнецов И. П., Мацкевич А. Г., Николаев В. Г.* Методы и средства настройки морфо-лексического анализатора на предметную область // Системы и средства информатики. — М.: Наука, 2009. Вып. 19. С. 96–118.
14. *Кузнецов И. П., Сомин Н. В.* Англо-русская система извлечения знаний из потоков информации в интернет-среде // Системы и средства информатики. — М.: Наука, 2007. Вып. 17. С. 236–254.
15. *Кузнецов И. П., Сомин Н. В.* Особенности лексико-морфологического анализа при извлечении информационных объектов и связей из текстов естественного языка // Компьютерная лингвистика и интеллектуальные технологии: по мат-лам междунар. конф. «Диалог 2010». — М.: РГГУ, 2010. Вып. 9(16). С. 254–264.
16. *Сомин Н. В., Соловьева Н. С., Шарнин М. М.* Система морфологического анализа: опыт эксплуатации и модификации // Системы и средства информатики. — М.: Наука, 2005. Вып. 15. С. 20–30.
17. *Banko M., Cafarella M., Soderland S., Broadhead M., Etzioni O.* Open information extraction from the Web // 20th Joint Conference (International) on Artificial Intelligence (IJCAI-07) Proceedings, 2007. P. 2670–2676.
18. Лаборатория компьютерной лингвистики ИПИ РАН: Официальный сайт. www.IpiranLogos.com.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 59–69

УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ И ПРАВАМИ ДОСТУПА ПОЛЬЗОВАТЕЛЕЙ В ЦЕНТРАХ ОБРАБОТКИ ДАННЫХ ВЫСОКОЙ ДОСТУПНОСТИ

М. В. Бендерина[1], С. В. Борохов[2], В. И. Будзко[3], П. В. Степанов[4], А. П. Сучков[5]

Аннотация: Изложены функционально-организационные схемы (ФОС) и принципы управления учетными записями и правами пользователей, разработанные для двух стратегий защиты информации, которые принимаются организацией или сообществом облачных вычислений. Определен порядок организации работ по созданию централизованной системы управления учетными записями и правами пользователей в составе системы обеспечения информационной безопасности (СОИБ) коллективных центров обработки данных (ЦОД) высокой доступности (ВД).

Ключевые слова: информационная безопасность; высокая доступность; центр обработки данных

1 Введение

Консолидация обработки данных на коллективных вычислительных центрах, используемых многими пользователями, — современная мировая тенденция применения средств обработки и передачи данных. Консолидация обработки данных позволяет существенно снизить финансовые затраты на обеспечение выполнения приложений, в частности за счет увеличения коэффициента полезного использования оборудования и уменьшения требуемого ИТ-персонала, а также существенно снизить количество нештатных ситуаций в системотехнической инфраструктуре, которые отражаются на конечных пользователях.

К централизованной консолидированной обработке данных перешли крупные мировые компании и организации, создавая корпоративные хранилища данных с единой системой сбора, накопления, верификации и хранения для всех приложений корпорации, что, в свою очередь, повысило достоверность, актуальность и точность получаемой на основе этих данных информации для поддержки принятия решений.

Многие небольшие компании не в состоянии самостоятельно создавать и поддерживать собственную необходимую для решения всех задач информационно-технологическую инфраструктуру, обеспечивающую требуемые вычислительные ресурсы и организационно-технические условия для осуществления безопасной технологии обработки данных. Им дешевле покупать (арендовать) необходимые услуги и ресурсы у ЦОД, который находится в промышленной эксплуатации и может их предоставить в требуемом объеме и при надлежащем качестве обслуживания.

Динамическая масштабируемость современных вычислительных и телекоммуникационных средств позволяет наращивать ресурсы с ростом нагрузок со стороны прикладных систем (ПС) без остановки обработки, а наличие средств виртуализации — обеспечивать их эффективное использование.

Реализация «облачных вычислений» также основана на использовании мощных ЦОД, которые позволяют хранить большие объемы данных и осуществлять исполнение тысяч приложений различных пользователей одновременно.

Яркий пример применения ЦОД — создание катастрофоустойчивой территориально распределенной информационно-телекоммуникационной системы централизованной обработки банковской информации в Банке России [1, 2].

Современному ЦОД присущи следующие архитектурные особенности:

- разделение одних и тех же вычислительных ресурсов множеством параллельно исполняемых ПС;
- централизованное управление функционированием центра на уровне ресурсов, осуществляемое в интересах всех эксплуатируемых ПС;

[1]Институт проблем информатики Российской академии наук, mbenderina@ipiran.ru
[2]Институт проблем информатики Российской академии наук, sborokhov@ipiran.ru
[3]Институт проблем информатики Российской академии наук, vbudzko@ipiran.ru
[4]Институт проблем информатики Российской академии наук, pvstepanov@ipiran.ru
[5]Институт проблем информатики Российской академии наук, asuchkov@ipiran.ru

– индивидуальное управление функционированием каждой ПС на прикладном уровне;
– удаленный доступ пользователей к своим ПС.

Однако создание и использование ЦОД в ИТ-инфраструктуре компании имеет свои особенности, в том числе и в части обеспечения ИБ. Составная часть обеспечения ИБ — задача управления учетными записями и правами доступа пользователей, для решения которой разработаны, доступны и применяются разнообразные аппаратно-программные комплексы различных фирм-изготовителей. Наиболее распространенные из них обладают приблизительно равными возможностями и наборами функций. Выбор конкретного комплекса должен делаться на основе общей стратегии организации в области защиты информации, существующей модели угроз, используемых подходов, опыта внедрения и эксплуатации средств и решений по обеспечению ИБ и других факторов.

Центру обработки данных присущ ряд особенностей, которые необходимо учитывать при создании СОИБ ЦОД, а в данном случае — подсистемы управления учетными записями и правами пользователей. Указанные особенности влияют прежде всего на распределение зон ответственности между различными структурными подразделениями организации, связанными с эксплуатацией ЦОД. Также требуется более четко определять иерархическую структуру и порядок взаимодействия между ее различными элементами.

В настоящей статье предполагается рассмотреть особенности обеспечения ИБ ЦОД, ФОС систем управления учетными записями и правами пользователей, построенные на основе выбранной стратегии защиты информации, и основные проблемы, возникающие при создании и/или вводе подобных систем в эксплуатацию.

2 Описание центра обработки данных

2.1 Общее описание центра обработки данных

Центр обработки данных — это организационно-технический комплекс, предназначенный для создания высокопроизводительной, отказоустойчивой информационной инфраструктуры [3].

С ростом централизации хранения и обработки информации возрастает значение адекватного проектирования и эксплуатации ЦОД, на котором сосредоточиваются основные информационные ресурсы и решаются задачи информационной поддержки деятельности подразделений одной или нескольких организаций и их отдельных сотрудников. Центр обработки данных становится существенно более критичным элементом системы, чем при распределенной обработке. Увеличивается потенциальный ущерб, который может быть нанесен в результате прекращения функционирования или/и преодоления системы безопасности ЦОД.

Выделяется особый класс систем — системы ВД, требуемое время восстановления которых при любых причинах прерывания работы не должно превышать относительно небольшого значения (нескольких минут).

Такие системы должны обладать свойством катастрофоустойчивости — способностью сохранять критически важные информационные и программные ресурсы и продолжать выполнение своих функций (возможно, с определенными ограничениями) в условиях деградации ФОС, вызванной массовым уничтожением элементов системы, а также связей между ними в результате стихийных бедствий, техногенных аварий и катастроф, целенаправленного воздействия людей или групп людей (включая террористические акты) [4–6].

Как правило, ЦОД ВД состоят из идентичных площадок (вычислительных комплексов – ВК), расположенных в территориально разнесенных вычислительных центрах. Минимальное число таких площадок — 2. Выделяют семь уровней катастрофоустойчивости ЦОД. Разделение на уровни происходит в зависимости от времени восстановления работоспособности ЦОД, объема предварительных мероприятий и других факторов, подробно изложенных в [7].

В настоящей статье рассматриваются проблемы обеспечения ИБ ЦОД ВД с катастрофоустойчивостью не ниже пятого уровня.

2.2 Особенности обеспечения информационной безопасности центра обработки данных

Обеспечение ИБ любой автоматизированной системы предполагает наличие оперативно-технического управления комплексом мер и средств защиты и управления системой безопасности [7–11].

Первое, как правило, связано непосредственно с управлением средствами обеспечения ИБ (штатными или дополнительными) и реализацией организационных мер. Второе подразумевает реализацию общего управления процессом обеспечения ИБ в организации, разработку и совершенствование нормативной и нормативно-методической базы организации, анализ текущего уровня ИБ организации.

На практике оперативно-техническое управление реализуется силами структурного подразделения из состава эксплуатирующего персонала ЦОД. А управление системой безопасности является обязанностью отдельного подразделения в организации, ответственного за поддержание режима ИБ организации.

Назначение ЦОД — предоставление информационно-вычислительных ресурсов и услуг для различных ПС. В ЦОД могут функционировать несколько ПС, совместно использующих его вычислительные ресурсы. Данные этих ПС обрабатываются с использованием одних и тех же программных и технических средств. При этом сами прикладные системы могут функционировать в интересах различных структурных подразделений организации – владельца ЦОД или/и его «арендаторов».

Эксплуатация и сопровождение этих систем также осуществляются различными структурными подразделениями организации, а в ряде случаев — с привлечением внешних организаций.

Сам ЦОД по отношению к прикладным системам — лишь отдельный элемент их системотехнической инфраструктуры, причем не целиком, а в части объема предоставляемых им услуг и ресурсов. Поэтому задача обеспечения ИБ ЦОД при консолидации обработки распадается на две подзадачи:

(1) обеспечение ИБ ПС;

(2) обеспечение ИБ ресурсов и услуг ЦОД.

Порядок обеспечения ИБ конкретной ПС на прикладном уровне определяется особенностями самой системы. При этом необходимо обеспечивать взаимодействие и согласование порядка обеспечения ИБ ПС с порядком обеспечения ИБ ЦОД.

Обеспечение ИБ ресурсов ЦОД осуществляется в интересах всех ПС, функционирующих в ЦОД. Данная задача, как правило, решается отдельным структурным подразделением в составе службы эксплуатации ЦОД. При этом сама СОИБ может входить в состав ЦОД, а может быть самостоятельной системой, смежной ЦОД.

Обеспечение взаимодействия систем безопасности ПС между собой и с системой безопасности ЦОД является основной задачей управления информационной безопасностью в организации.

Одной из задач, решаемых при обеспечении ИБ ЦОД, является управление учетными записями и правами пользователей в рамках СОИБ. Далее будут рассмотрены особенности ее решения.

3 Функционально-организационная схема системы управления учетными записями и правами пользователей

Системы управления учетными записями и правами пользователей могут создаваться на базе решений различных производителей программного обеспечения и технических средств, иметь различный масштаб и режим работы. Однако в основе ФОС, используемых при их создании, лежат три стратегии защиты информации:

(1) оборонительная;

(2) наступательная;

(3) упреждающая.

Упреждающая стратегия предполагает тщательное исследование возможных угроз системе и разработку мер по их нейтрализации еще на стадии проектирования и создания системы.

Оборонительная стратегия используется тогда, когда не допускается вмешательство в процесс функционирования защищаемого объекта. В этом случае обычно реализуются организационные меры защиты, направленные, прежде всего, на противодействие наиболее опасным угрозам.

Наступательная стратегия занимает промежуточное положение. При ее реализации уже на начальной стадии создания системы решаются вопросы обеспечения ИБ.

Рассмотрим ФОС системы управления учетными записями и правами пользователей, построенные на базе наступательной и упреждающей стратегий. По мнению авторов статьи, данные стратегии представляют наибольших интерес, так как позволяют разработчикам системы безопасности и службе безопасности организации адекватно реальным условиям и требованиям выстраивать методы и средства защиты и формировать организацию и технологии этой системы. Учитывая сказанное, далее будут рассмотрены:

- ФОС системы, построенная на основе наступательной стратегии;
- ФОС системы, построенная на основе упреждающей стратегии.

При реализации управления учетными записями и правами пользователей представленные подходы различаются, прежде всего, разделением обязанностей между администратором системы и аудитором. При использовании ФОС, основанной

на наступательной стратегии, в обязанности администратора системы входит поддержание системы в работоспособном состоянии, выполнение всех задач администрирования в соответствии с установленным в организации регламентом. В части управления учетными записями и правами пользователей администратор имеет все полномочия по созданию, удалению и изменению учетной записи; он может предоставлять права доступа, изменять их, включать в группу и т. п.

Аудитор системы — роль, предназначенная для контроля функционирования системы, работы эксплуатирующего персонала и функциональных пользователей, соответствия текущих настроек установленной в организации политике ИБ. При управлении учетными записями и правами пользователей аудитор выполняет функции контроля соответствия набора прав и полномочий функциональных пользователей системы и эксплуатирующего персонала выполняемым ими функциям (должностным обязанностям) и установленной политике ИБ. В случае выявления нарушений аудитор проводит расследование и выполняет другие действия в соответствии с установленным регламентом.

Функционально-организационная схема, основанная на упреждающей стратегии, предполагает, что аудитор выполняет все обязанности, приведенные для предыдущей ФОС, и дополнительно наделяется частью прав администратора системы в соответствии с установленной в организации политикой ИБ и другими нормативно-методическими документами. При рассмотрении подобной ФОС принято говорить уже не о роли аудитора системы, а о роли администратора ИБ системы.

Далее каждый вид ФОС будет рассмотрен применительно к консолидированной обработке в ЦОД.

3.1 Функционально-организационная схема, построенная на основе наступательной стратегии защиты информации

Применительно к консолидированной обработке в ЦОД ФОС системы управления учетными записями и правами пользователей, построенная на основе наступательной стратегии защиты информации, имеет вид, представленный на рис. 1.

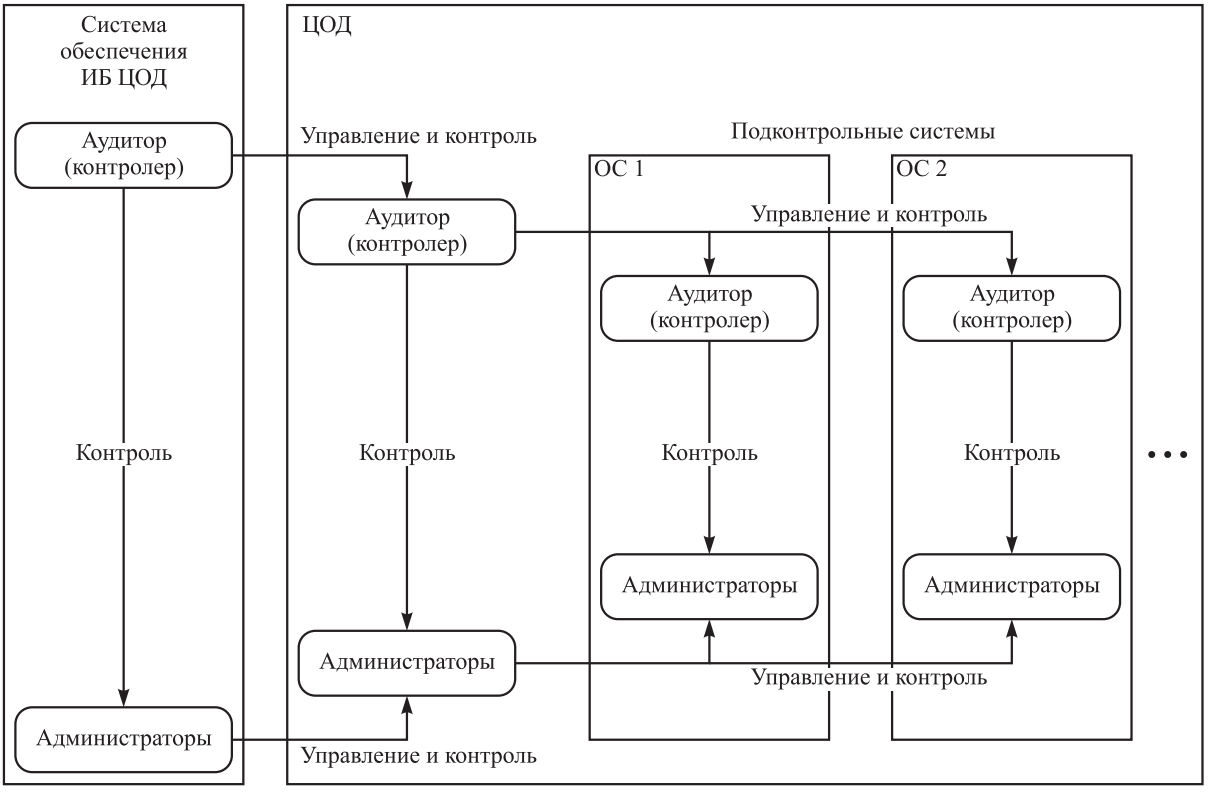


Рис. 1 Функционально-организационная схема системы управления учетными записями и правами пользователей, основанная на наступательной стратегии ИБ

В системе обеспечения ИБ присутствуют следующие роли: администратор и аудитор системы. Предполагается следующее разделение полномочий между выделенными ролями[1].

Аудитор СОИБ ЦОД в рамках всего комплекса в целом[2] осуществляет общее руководство процессом обеспечения ИБ, включая решение следующих задач:

- разработку общей схемы управления ИБ, включая вопросы управления учетными записями и правами пользователей; разработку нормативных и нормативно-методических документов, определяющих процесс обеспечения ИБ всего комплекса;
- координацию работ по ведению и настройке средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей;
- контроль и управление деятельностью аудиторов ЦОД и ПС;
- контроль поддержания заданного уровня ИБ комплекса в целом и разработку предложений по его совершенствованию;
- интегрированный мониторинг ИБ всего комплекса в целом.

В обязанности аудитора может входить:

- контроль деятельности администратора СОИБ ЦОД;
- контроль функционирования СОИБ ЦОД в части вопросов ИБ;
- периодическая сверка и анализ данных регистрации событий ИБ, накапливаемых средствами обеспечения ИБ, а также ручными журналами учета.

Администратор СОИБ в рамках всего комплекса в целом отвечает за его функционирование в целом, включая решение следующих задач:

- разработку общей схемы обеспечения ИБ, включая основные принципы управления учетными записями и правами пользователей;
- координацию работ по ведению средств обеспечения ИБ, в том числе средств управления учетными записями и правами пользователей;
- периодический анализ настройки средств обеспечения ИБ и сверку этих данных с установленными в организации правилами доступа к ресурсам;
- управление и контроль деятельности администраторов ЦОД и ПС.

Администратор системы также выполняет все функции по администрированию системы, включая вопросы управления учетными записями и правами пользователей.

В составе эксплуатирующего персонала ЦОД выделяются следующие роли.

Аудитор ЦОД отвечает за поддержание режима ИБ ЦОД. В его должностные обязанности также входит определение конкретной схемы контроля доступа к ресурсам в соответствии с установленными правилами, контроль заданных настроек средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей. Кроме того, он осуществляет контроль регистрации действий пользователей и администраторов и оперативный мониторинг событий ИБ в зоне своей ответственности. Аудитор ЦОД осуществляет управление деятельностью аудиторов ПС. В части задачи управления учетными записями и правами пользователей он отвечает за контроль процесса создания и изменения учетных записей, состава учетных записей администратором ЦОД. Аудитор ЦОД проводит анализ соответствия состава ролей и прав доступа каждой учетной записи выполняемым пользователем служебным обязанностям, а также анализ состава ролей и их полномочий на соответствие установленной в организации схеме в зоне своей ответственности.

Администратор ЦОД отвечает за функционирование ЦОД. В его должностные обязанности входят определение конкретной схемы разграничения доступа к защищаемым ресурсам в зоне своей ответственности, настройка и сопровождение средств обеспечения ИБ. При управлении учетными записями и правами пользователей в зоне своей ответственности администратор ЦОД выполняет следующие действия:

- создает, удаляет и изменяет учетные записи пользователей;
- назначает и изменяет права доступа пользователей к защищаемым ресурсам;
- создает, удаляет и изменяет роли в ЦОД в соответствии с установленной схемой;
- приписывает роль (группы ролей) учетной записи, изменяет состав ролей учетной записи.

В ПС, функционирующих в ЦОД, присутствуют следующие роли.

[1] Рассматриваемый набор ролей представляется наиболее интересным с точки зрения разделения полномочий эксплуатирующего персонала и ни в коем случае не является полным перечнем ролей эксплуатирующего персонала ЦОД.

[2] Здесь и далее под словосочетанием «весь комплекс в целом» будет пониматься совокупность ЦОД, СОИБ ЦОД и ПС, функционирующих на ЦОД.

Аудитор ПС отвечает за поддержание режима ИБ ПС. В рамках своих должностных обязанностей в зоне своей ответственности определяет конкретную схему контроля доступа к ресурсам, контролирует настройки средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей, непрерывность регистрации действий пользователей и администратора ПС. Осуществляет оперативный мониторинг событий ИБ. В части решения задачи управления учетными записями и правами пользователей в его обязанности входит:

- контроль процесса создания и изменения учетных записей, состава учетных записей администратором ПС;
- контроль соответствия состава ролей и прав доступа каждой учетной записи выполняемым пользователем служебным обязанностям;
- анализ состава ролей и их полномочий на соответствие установленной в организации схеме.

В обязанности администратора ПС входит поддержание функционирования ПС. Кроме того, в зоне своей ответственности он определяет конкретную схему разграничения доступа к ресурсам, устанавливает, сопровождает и настраивает прикладное программное обеспечение (ППО) в соответствии с установленными в организации правилами. При управлении учетными записями и правами пользователей отвечает:

- за создание, удаление и изменение учетных записей пользователей;
- за назначение и изменение прав доступа пользователей в ПС;
- за создание, изменение и удаление ролей в ПС;
- за приписывание роли (группы ролей) учетной записи, изменение состава ролей учетной записи.

3.2 Функционально-организационная схема системы, основанная на упреждающей стратегии защиты информации

Как уже отмечалось ранее, основным различием между рассматриваемыми ФОС является разделение функций между администратором СОИБ и администратором ИБ. В зависимости от политики ИБ организации и ее технических возможностей указанное разделение может быть реализовано на организационном уровне или на организационном и техническом уровнях.

В случае консолидированной обработки на ЦОД ФОС системы управления учетными записями и правами пользователей, основанная на упреждающей стратегии защиты информации, имеет вид, представленный на рис. 2.

Рассматриваемая ФОС предполагает следующее разделение обязанностей между персоналом, участвующим в эксплуатации ЦОД.

Для СОИБ ЦОД выделяются следующие роли эксплуатирующего персонала:

- администратор СОИБ ЦОД;
- администратор ИБ СОИБ ЦОД.

Администратор СОИБ отвечает за поддержание работоспособности СОИБ ЦОД. Он осуществляет контроль функционирования СОИБ ЦОД, а также всего комплекса ЦОД в целом в части вопросов обеспечения ИБ. В обязанности администратора СОИБ также входит разработка общей схемы обеспечения ИБ, включая основные принципы управления учетными записями и правами пользователей, координация работы по ведению средств обеспечения ИБ, в том числе средств управления учетными записями и правами пользователей, периодический анализ их настроек и сверка этих данных с установленными в организации правилами доступа к ресурсам. Кроме того, он осуществляет управление и контроль деятельности администраторов ЦОД и ПС в зоне своей ответственности. При управлении учетными записями и правами пользователей администратор СОИБ имеет ограниченные возможности по администрированию. В зависимости от политики ИБ организации он может иметь возможность создавать, удалять и изменять учетные записи пользователей СОИБ[1], однако может не иметь возможности назначать им права доступа, изменять имеющиеся или аннулировать права доступа пользователя.

Администратор ИБ СОИБ ЦОД в рамках представленной схемы отвечает за поддержание режима ИБ СОИБ и всего ЦОД в целом. Он разрабатывает общую схему управления ИБ, включая вопросы управления учетными записями и правами пользователей, нормативные и нормативно-методические документы, определяющие процесс обеспечения ИБ. Администратор ИБ осуществляет общее руководство процессом обеспечения ИБ. В рамках своих должностных обязанностей он координирует работы по ведению и настройке средств обеспечения ИБ, включая средства управления учетными

[1] В зависимости от масштаба ЦОД функции администратора СОИБ и администратора ЦОД могут разделяться между несколькими ролями. В таком случае общие вопросы функционирования систем и обеспечения ИБ решаются администраторами ЦОД и СОИБ, а непосредственное администрирование осуществляется более мелкими ролями.

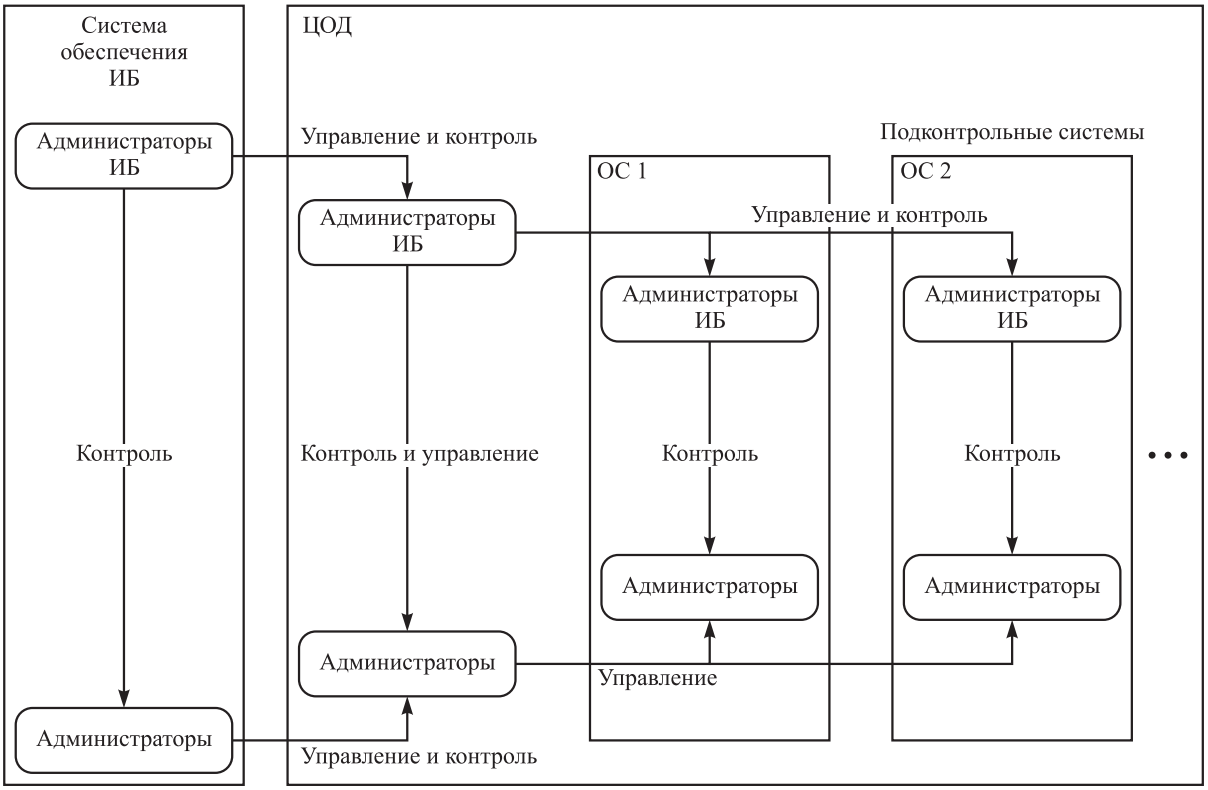


Рис. 2 Функционально-организационная схема системы управления учетными записями и правами пользователей, основанная на упреждающей стратегии

записями и правами пользователей, периодически анализирует и сверяет данные регистрации событий ИБ, накапливаемые средствами обеспечения ИБ, а также ручными журналами учета. Он осуществляет интегрированный мониторинг событий ИБ на всех уровнях. Кроме того, в обязанности администратора ИБ СОИБ ЦОД входит контроль и управление деятельностью администраторов ИБ ЦОД и администратора СОИБ ЦОД в соответствии с установленным в организации порядком. В части решения задачи управления учетными записями и правами пользователей в рамках СОИБ ЦОД в обязанности администратора ИБ может входить назначение и/или изменение состава ролей внутри СОИБ ЦОД.

Для ЦОД выделяются следующие роли:

- администратор ЦОД;
- администратор ИБ ЦОД;
- администраторы ПС, функционирующих в ЦОД;
- администраторы ИБ ПС, функционирующих в ЦОД.

Администратор ЦОД отвечает за поддержание ЦОД в работоспособном состоянии. В его обязанности также входит определение конкретной схемы разграничения доступа к защищаемым ресурсам, настройка и сопровождение средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей, проведение необходимой настройки при установке нового прикладного и системного программного обеспечения в зоне своей ответственности. В части решения задачи управления учетными записями в зависимости от политики ИБ организации администратор ЦОД может иметь полномочия на создание, удаление и изменение учетных записей пользователей в зоне своей ответственности. Кроме того, в его обязанности может входить создание новых ролей и/или изменение уже существующих. Однако ему может быть запрещено назначать и/или изменять права доступа пользователей к защищаемым ресурсам, приписывать роль (группы ролей) учетной записи, изменять состав ролей учетной записи.

Администратор ИБ ЦОД отвечает за поддержание режима ИБ ЦОД. В обязанности администратора ИБ ЦОД входят определение конкретной схемы контроля доступа к ресурсам в соответствии с установленными правилами, контроль заданных настроек средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей, в соответствии с установленной схемой

в зоне своей ответственности. Кроме того, администратор ИБ ЦОД осуществляет контроль регистрации действий пользователей и администраторов, оперативный мониторинг событий ИБ в зоне своей ответственности. В части задачи управления учетными записями и правами пользователей он осуществляет контроль процесса создания и изменения администратором ЦОД учетных записей, состава учетных записей; анализ состава ролей и их полномочий на соответствие установленной в организации схеме; контроль соответствия состава ролей и прав доступа каждой учетной записи выполняемым пользователем служебным обязанностям в зоне своей ответственности. Администратор ИБ ЦОД может также назначать и/или изменять права доступа пользователей, приписывать роли (группы ролей) учетной записи, изменять состав ролей учетной записи.

Администратор ПС, функционирующей в ЦОД, отвечает за поддержание функционирования ПС. В обязанности администратора входит определение конкретной схемы разграничения доступа к ресурсам в зоне своей ответственности, установка, настройка и сопровождение ППО в соответствии с установленными правилами. В части решения задачи управления учетными записями и правами пользователей на администратора ПС может быть возложено создание, удаление и изменение учетных записей пользователей; создание, изменение и удаление ролей в ПС в зоне своей ответственности. Однако может быть запрещено назначать и изменять права доступа пользователей в ПС, приписывать роль (группы ролей) учетной записи, изменять состав ролей учетной записи в ПС.

Администратор ИБ ПС, функционирующей в ЦОД, отвечает за поддержание режима ИБ данной системы. Аналогично администратору ИБ ЦОД, он в зоне своей ответственности определяет конкретную схему контроля доступа к ресурсам в соответствии с установленными правилами, осуществляет контроль заданных настроек средств обеспечения ИБ, включая средства управления учетными записями и правами пользователей; контроль регистрации действий пользователей и администратора ПС; оперативный мониторинг событий ИБ. В части задачи управления учетными записями и правами пользователей он осуществляет контроль процесса создания и изменения администратором ПС учетных записей, состава учетных записей ПС; анализ состава ролей и их полномочий на соответствие установленной в организации схеме; контроль соответствия состава ролей и прав доступа каждой учетной записи выполняемым пользователем служебным обязанностям в зоне своей ответственности. Администратор ИБ ПС может также назначать и/или изменять права доступа пользователей, приписывать роли (группы ролей) учетной записи, изменять состав ролей учетной записи.

В зависимости от политики ИБ организации и режима работы ЦОД и ПС, функционирующих в ЦОД, также могут выделяться роли администратора нештатного режима и оператора (оператора ИБ).

Роль администратора нештатного режима предполагает наличие в системе учетной записи, используемой в случае перехода системы в нештатный режим функционирования. Данная роль имеет максимальные права по доступу к ресурсам системы. При штатном режиме функционирования системы вход с использованием учетной записи данной роли в систему, как правило, запрещен.

Оператор системы имеет минимальные полномочия в соответствующей системе. Данная роль предназначена для непрерывного наблюдения за работой системы (ПС или самого ЦОД) и, в случае нарушения нормального функционирования, принятия оперативных действий в соответствии с установленным в организации регламентом.

В зависимости от масштаба и сложности системы (прикладной или самого ЦОД), а также от потребностей организации состав ролей может меняться.

4 Порядок организации работ по созданию централизованной системы управления учетными записями и правами пользователей

Для создания эффективной системы централизованного управления учетными записями и правами пользователей в составе СОИБ ЦОД еще на стадии проектирования необходимо решение ряда вопросов. Далее будут приведены некоторые из них, способные оказать значительное влияние на создаваемую систему централизованного управления учетными записями и правами пользователей.

4.1 Определение зон ответственности

Как уже отмечалось в разд. 2 настоящей статьи, задача обеспечения ИБ ЦОД включает в себя два направления: обеспечение ИБ ПС, функционирующих в ЦОД, и обеспечение ИБ ресурсов самого ЦОД.

На практике задачи решаются различными подразделениями и на различных уровнях информационной инфраструктуры. Однако очевидно, что оба процесса не могут идти абсолютно независимо друг от друга.

Как правило, СОИБ ЦОД функционирует на нижних уровнях информационной инфраструктуры ЦОД и обеспечивает поддержание заданного уровня ИБ ресурсов ЦОД, находящихся в совместном пользовании ПС. При этом сами прикладные системы, погружаемые в среду ЦОД, представляются для СОИБ ЦОД некоторыми «черными ящиками».

Система ИБ ПС предназначена для обеспечения и поддержания требуемого уровня ИБ внутри ПС. Для эффективного функционирования каждой из систем в рамках консолидированной обработки на ЦОД необходимо для каждого подразделения организации, участвующего в эксплуатации ЦОД и/или СОИБ ЦОД, а также для подразделений, эксплуатирующих прикладные системы, определить их зоны ответственности. Должны быть определены задачи, решаемые каждым подразделением, схема и порядок их взаимодействия при работе ЦОД в штатном и нештатном режимах функционирования. Также дополнительно может быть определен приоритет подразделения (или ПС) при распределении вычислительных ресурсов ЦОД.

В соответствии с установленными зонами ответственности определяются необходимые каждому подразделению функции программных продуктов, предполагаемых к использованию в части управления учетными записями и правами пользователей.

Другой немаловажный момент при определении зон ответственности — распределение обязанностей между ролями эксплуатирующего персонала в соответствии с установленной в организации политикой ИБ. Для построения эффективной СОИБ ЦОД на начальных этапах ее создания необходимо определить (или разработать по необходимости) порядок разделения обязанностей эксплуатирующего персонала в части управления учетными записями и правами доступа. Как было показано в разд. 3 настоящей статьи, существуют два принципиально различных подхода к решению данной задачи.

После того как сделан выбор в пользу одного из вариантов, необходимо сформировать перечни ролей и функций для каждой роли. На основе полученных материалов проводится разработка технических решений и предложений по организационным мерам в части тех требований, которые невозможно реализовать техническими средствами в рамках предполагаемых к использованию решений.

4.2 Формирование иерархической структуры

В большинстве продуктов, предназначенных для централизованного управления учетными записями и правами пользователей, организационное обеспечение СОИБ ЦОД представляется в виде определенной структуры. Данная структура представляет собой дерево, отображающее присутствующие в организации подразделения и их взаимосвязь. Ввиду этого при проектировании системы для каждого подразделения организации необходимо сформировать иерархическую структуру групп пользователей. Кроме того, для нормального функционирования системы централизованного управления учетными записями и правами пользователей необходимо также определить регламенты (*workflow*), в соответствии с которыми должны будут осуществляться действия, связанные с созданием, управлением и удалением учетных записей в системе. Обычно данные регламенты содержат не только саму цепочку действий, направленных на решение определенной задачи (создание нового пользователя, предоставление прав доступа к ресурсу, снятие блокировки учетной записи и т. д.), но и предполагают наличие вариантов действий системы в случае возникновения различных проблем (таких как болезнь лица, подтверждающего/разрешающего предоставление прав, создание новой учетной записи, истечение периода реакции ответственного лица на заявку пользователя и т. п.).

Грамотное решение данного вопроса позволяет создать гибкую и практически автономную систему централизованного управления учетными записями и правами пользователей, учитывающую все особенности установленного порядка.

4.3 Формирование перечня разрешенных и запрещенных к использованию функций системы

Существующие решения по централизованному управлению учетными записями и правами пользователей, как правило, предоставляют пользователям широкий набор функций (включая функции самообслуживания, сброса и переустановки собственных паролей, просмотра списка доступных ресурсов с возможностью запроса необходимых полномочий, возможность удаленного администрирования). Наличие некоторых из этих функций у определенных подразделений и/или групп пользователей может противоречить существующей в

организации политике ИБ или другим нормативным документам. Ввиду этого уже на стадии проектирования необходимо четко сформировать перечень разрешенных функций для подразделений и/или групп пользователей и перечень функций, которые должны быть запрещены (или заблокированы).

Кроме того, хорошей практикой является определение минимального перечня прав и полномочий, которыми должны обладать вновь создаваемые учетные записи (или явное указание того, что они не имеют никаких прав), а также все имеющиеся в системе роли (администратор, администратор ИБ, оператор и т. п.).

4.4 Определение порядка реализации организационных мер обеспечения информационной безопасности

Анализ требований, предъявляемых некоторыми организациями к системе централизованного управления учетными записями и правами пользователей, показывает, что ряд требований не покрывается функциональными возможностями используемых программных продуктов. Очень важно уже на ранних стадиях создания системы провести анализ соответствия между требованиями организации в области обеспечения ИБ и функциями предполагаемых к использованию программных продуктов. Необходимо определить все механизмы реализации каждого требования, а также требования, которые не реализуются средствами программного продукта и предполагают использование дополнительных средств или реализации организационных мер. Для каждого требования (или группы требований), не реализуемого функциями предполагаемых к использованию программных продуктов, необходимо разработать организационные меры и порядок их реализации в подразделениях, имеющих доступ к ресурсам ЦОД.

4.5 Определение порядка использования технологических учетных записей

Существующие программные продукты осуществляют централизованное управление учетными записями и правами пользователей в подконтрольных системах посредством технологических учетных записей. Внутри подконтрольной системы такая учетная запись имеет максимальные права по доступу к ресурсам. При этом на практике защите технологических учетных записей уделяется мало внимания (или не уделяется вовсе).

При создании системы централизованного управления учетными записями и правами пользователей необходимо подробно изучить вопрос взаимодействия программных продуктов, предполагаемых к использованию, с подконтрольными им системами (ЦОД, возможно ПС, функционирующими в ЦОД). В результате изучения должен быть сформирован перечень необходимых для взаимодействия технологических учетных записей, их права в системах и порядок использования, а также описание мер их защиты. Оптимальным вариантом во многих случаях считается запрет интерактивного входа в подконтрольную систему с использованием идентификаторов технологических учетных записей.

4.6 Определение порядка внесения изменений в учетные записи пользователей

При изучении функциональных возможностей программных продуктов для системы централизованного управления учетными записями и правами пользователей необходимо также исследовать возможности отслеживания изменений в учетных записях пользователей в подконтрольной системе средствами самих продуктов. Предлагаемые сегодня на рынке программные продукты можно условно разделить на два класса — обеспечивающие возможность отслеживания изменений, внесенных в учетные записи пользователей подконтрольной системы средствами самой системы, и не обеспечивающие такой возможности.

В зависимости от того, какой программный продукт/продукты предполагается к использованию в системе централизованного управления учетными записями и правами пользователей, должен решаться вопрос о разработке дополнительных организационных мер. В частности, при использовании продуктов, не обеспечивающих отслеживание изменений в учетных записях, представляется целесообразной разработка порядка внесения изменений в учетные записи пользователей только посредством программного продукта, предназначенного для централизованного управления учетными записями и правами пользователей.

5 Заключение

Консолидация обработки данных на коллективных ЦОД, используемых многими разнообразными ПС и их пользователями, требует обеспечения необходимого уровня ИБ. Центр обработки данных,

входящий в состав системы ВД, обладает рядом особенностей, которые необходимо учитывать при создании СОИБ ЦОД и ее подсистемы управления учетными записями и правами пользователей. В статье изложены ФОС организации управления учетными записями и правами пользователей, разработанные для двух стратегий защиты информации, которые принимаются организацией или сообществом облачных вычислений. Определен порядок организации работ по созданию централизованной системы управления учетными записями и правами пользователей в составе СОИБ ЦОД.

Литература

1. *Будзко В. И., Сенаторов М. Ю., Михайлов С. Ф., Курило А. П., Соколов И. А.* Направления совершенствования и развития информационно-телекоммуникационной системы Банка России // Информационная безопасность России в условиях глобального информационного общества: Мат-лы 5-й Всеросс. конф. — М., 2003. С. 207–211.
2. *Беленков В. Г., Будзко В. И., Быстров И. И., Козлов А. Н., Кудряшов А. А., Курило А. П., Михайлов С. Ф., Нагибин С. Я., Сенаторов М. Ю., Шмид А. В.* Катастрофоустойчивая территориально распределенная информационно-телекоммуникационная система централизованной обработки банковской информации // Системы высокой доступности, 2011. Т. 7. № 3. С. 6–47.
3. *Заенц Д.* Введение в ЦОД (дата-центр). Опубликовано 30.08.2009. http://dcnt.ru/?p=325.
4. *Будзко В. И., Синицын И. Н., Соколов И. А.* Построение информационно-телекоммуникационных систем высокой доступности // Системы высокой доступности, 2005. Т. 1. № 1. С. 6–14.
5. *Борохов С. В., Будзко В. И., Киселев Э. В., Кейер П. А.* Функциональное структурирование и критерии оптимизации построения и функционирования информационно-телекоммуникационных систем высокой доступности // Системы высокой доступности, 2005. Т. 1. № 1. С. 15–25.
6. *Беленков В. Г., Борохов С. В., Будзко В. И., Киселев Э. В., Кейер П. А.* Экономические основы консолидации обработки в информационно-телекоммуникационных системах высокой доступности // Системы высокой доступности, 2005. Т. 1. № 1. С. 26–37.
7. *Беленков В. Г., Будзко В. И., Кейер П. А.* Катастрофоустойчивые решения в информационно-телекоммуникационных системах высокой доступности // Системы высокой доступности, 2005. Т. 1. № 1. С. 57–69.
8. *Будзко В. И., Соловьев А. В.* Вопросы защиты от угроз со стороны обслуживающего персонала в центрах обработки данных // Вопросы защиты информации, 2003. № 2(61). С. 33–39.
9. *Борохов С. В., Будзко В. И., Курило А. П.* ФОС и принципы построения системы безопасности информационно-телекоммуникационных систем высокой доступности // Системы высокой доступности, 2005. Т. 1. № 1. С. 38–45.
10. *Борохов С. В., Будзко В. И., Капырин А. Ю.* Опыт применения динамического контроля целостности в информационно-телекоммуникационных системах высокой доступности // Системы высокой доступности, 2006. Т. 2. № 1. С. 46–50.
11. *Борохов С. В., Будзко В. И.* Информационная безопасность при консолидированной обработке на мейнфрейме // Информационные технологии и математическое моделирование систем 2009–2010: Тр. Междунар. науч.-техн. конф. — М., 2010. С. 182–183.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 70–77

EXTENDING INFORMATION INTEGRATION TECHNOLOGIES FOR PROBLEM SOLVING OVER HETEROGENEOUS INFORMATION RESOURCES*

L. A. Kalinichenko[1], S. A. Stupnikov[2], and V. N. Zakharov[3]

Abstract: This position paper is an attempt to match up the emerging challenges for problem solving over heterogeneous distributed information resources. State-of-the-art in subject mediation technology reached at IPI RAN is presented. The technology is aimed at filling the widening gap between the users (applications) and heterogeneous resources of data, knowledge, and services. Also, the paper affects the semantic-based information integration technologies challenges including investigation of application-driven approach for problem solving in the subject mediator environment, a provision for support of executable declarative specifications of the applications over the mediator, enhancement of presence of knowledge-based facilities at the mediator level, and mediation of databases with nontraditional data models motivated by the need of large data support.

Keywords: subject mediation; heterogeneous information resources; scientific problem solving; information integration; application-driven approach; rule-based languages; nontraditional data models

1 Introduction

This position paper[4] is an attempt to match up the emerging challenges for problem solving over heterogeneous information resources. Methods and tools for integration of data-oriented information systems within the Internet where heterogeneous databases are designed in different application domain contexts, modeled and implemented independently are well studied [1]. The trend in data integration is in the direction of providing a unified data access and manipulation interface over a mediated schema. Each data resource in such middleware architecture is represented as a view over the mediated schema (the approach known as "view-based data manipulation" in the settings known as LAV (Local As View) and GLAV (Global/Local As View) [2]). In such settings, the mediated schema is expressed in the context of the application domain applying the respective concept definitions. Semantic conflicts between the contexts of resources and of the application domain should be resolved. Such approach is called "application-driven mediation" in contrast to the "resource-driven mediation" intrinsic for the GAV (Global As View) setting.

The information integration technologies challenges touched in this paper include an emphasis on semantic issues requiring, in particular, a provision for support of executable declarative specification of the application over the mediator, enhancement of presence of knowledge-based facilities at the mediator level, compliance with the proliferation of new kinds of data sources in the data space, specifically with the need in the big data reflected in development of various nontraditional database systems and data models to be integrated.

2 State-of-the-Art in Subject Mediation Reached at IPI RAN

2.1 Basic principles

Subject mediation technology is aimed to fill the widening gap between the users (applications) and heterogeneous distributed information resources of data, knowledge, and services and to provide methods and tools for problem solving over the multitude of such resources. Basic principles of subject mediation based organization of problem solving are the following [3]:

- independence of definition of problem domain (the mediator definition) of the existing information resources;
- definition of a mediator as a result of consolidated efforts of the respective scientific community;

*This work has been partially supported by the RFBR grants 10-07-00342-a and 11-07-00402-a and by the project 4.2 of the Program for basic research No. 16 of the Presidium of RAS.

[1]Institute of Informatics Problems, Russian Academy of Sciences, leonidk@synth.ipi.ac.ru

[2]Institute of Informatics Problems, Russian Academy of Sciences, ssa@ipi.ac.ru

[3]Institute of Informatics Problems, Russian Academy of Sciences, vzakharov@ipiran.ru

[4]The paper has been prepared in connection with the first international call for Research Center proposals in frame of the MIT/SkTech initiative (http://web.mit.edu/SkTech/rc-call/).

– independence of user interfaces of the multiple information resources involved: the mediator users should know only the definition of a problem domain in the mediator (definition of concepts, structures, and behavior of the problem domain objects);

– information about new resources can be published at any time independently of mediators acting at that time, such new resources (if relevant) can be integrated in the existing mediators without changing their (mediators) specifications;

– GLAV-based setting for relevant information resources integration at the mediator;

– integrated access to the information resources in the process of problem solving; and

– recursive structure of a mediator: each mediator is published as a new information resource that might be used while solving the problems belonging to the intersection of various subject domains.

2.2 Canonical information model synthesis

To provide an integration of heterogeneous resources in the mediators, it is required to develop the *canonical information model* [4] serving for adequate reflection of semantics of various information models used in the environment. The main principle of the *canonical model synthesis* is the *extensibility* of the canonical model kernel. A kernel itself is fixed. For each specific information model M (called source model) of the environment, an extension of the kernel (target model) is defined so that this extension together with the kernel is refined by M. Such refining transformation of models should be provably correct. The canonical model for the environment is synthesized as the union of extensions, constructed for all models M of the environment.

2.3 Resources identification and integration

Identification of resources relevant to the mediator specification is based on three models:

(i) metadata model (characterizing resource capabilities represented in external registries);

(ii) ontological model (providing for definition of mediator concepts); and

(iii) canonical model (providing for conceptual definition of structure and behavior of mediator objects).

Reasoning in canonical model is based on the semantics of the canonical model and facilities for proof of the refinement. Reasoning in the metadata model is a heuristic one based on nonfunctional requirements for the resources needed in application.

A process of integration of relevant information resources in a subject mediator (registration) follows GLAV that combines two approaches: LAV and GAV [2]. Such integration technique provides for stability of application problem specification during any modifications of specific information resources and of their actual presence as well as for scalability of mediators with regard to the number of resources integrated.

2.4 Subject mediation infrastructure

The subject mediation infrastructure is multilayered including resource layer, computation, and information resource environments (that can include *grids* or *clouds*), wrapper layer used for technical interface interoperability, semantic mediation middleware layer representing subject area semantics and defining unified mappings of various resource information models, application problem domain layer (Fig. 1).

2.5 Results obtained at IPI RAN in the area of semantic information integration and mediation

1. A prototype of the subject mediation infrastructure has been developed [5, 6]. The prototype is used for problem solving over multiple distributed information resources in astronomy problem domain. One of the possible environments for resource organization used is the AstroGrid environment.

2. Methods and tools for mapping and transformation of information models of heterogeneous resources intended for their *unification* in mediation middleware have been developed. A resource model is said to be unified if its refining mapping into the canonical information model has been constructed. As a canonical model kernel, a specific hybrid semistructured and object oriented data model (the SYNTHESIS language [3]) is used. Model mappings are verified using B-technology formalizing specification refinement. The Model Unifier prototype tool aimed at partial automation of heterogeneous information models unification has been implemented. First version is based on term-rewriting technology [7, 8]. The second version as an Eclipse platform application based on model transformation languages is under implementation.

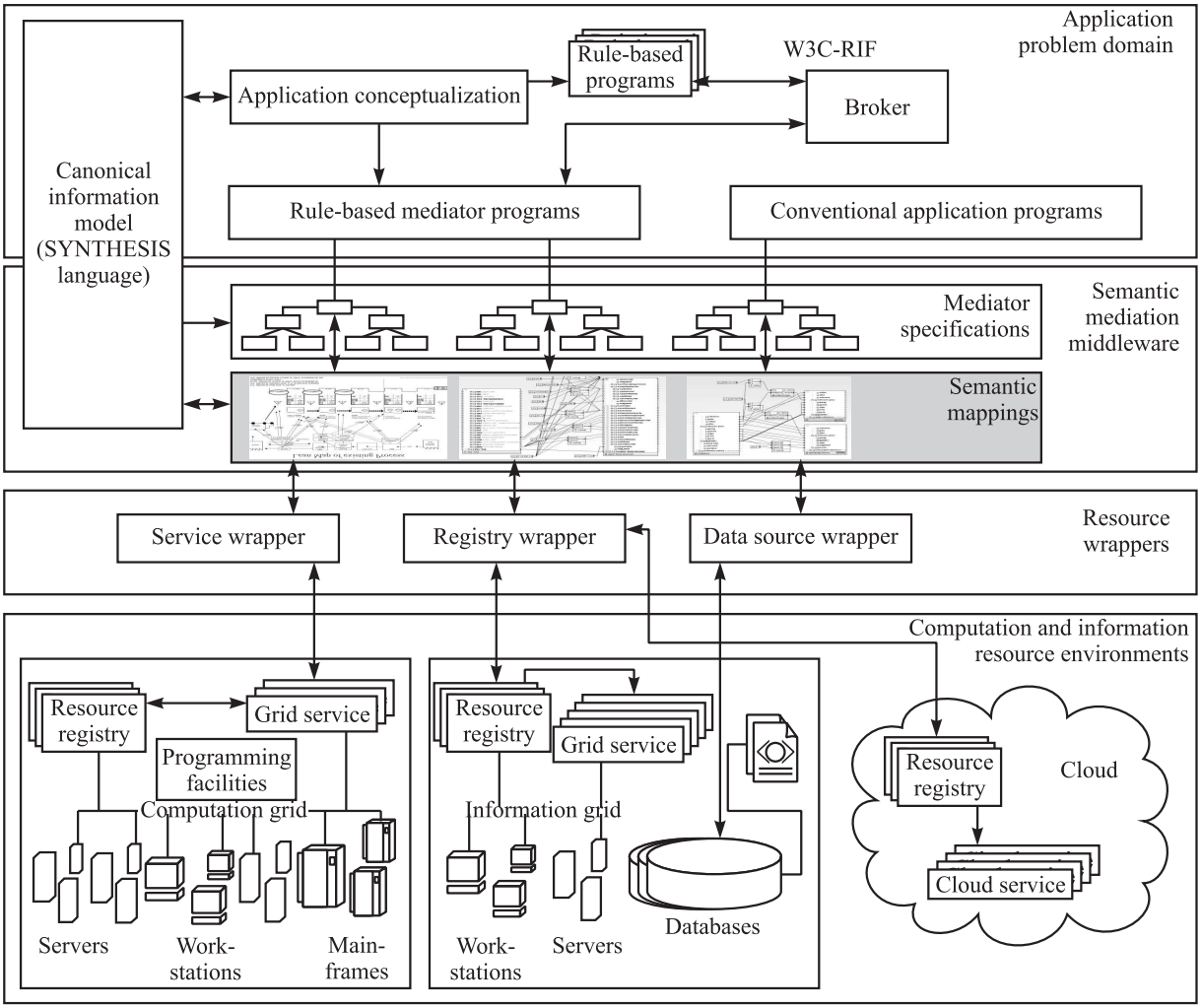


Figure 1 Subject mediation infrastructure

3. Methods for information resources semantic interoperability support in a context of application problem domain have been investigated. Tools for identification of resources relevant to a problem on the basis of and ontological descriptions of problem domain as well as tools for registration of the relevant resources in the mediator have been implemented.

4. Methods and tools for rewriting of nonrecursive mediator programs into resource partial programs have been developed. The methods are oriented on applying of object schemas of resources and mediators and typed GLAV-views.

5. A method for optimizing planning of resource partial programs execution over distributed environment has been developed. The method takes into account capabilities of the resources, assigns places of operation's execution on the basis of estimative samples, applying an interface for interoperation of planner and wrappers which allows dynamic estimations of plans efficiency.

6. Methods for dispersed organization of problem solving in the mediation environment have been developed. An implementation of a problem in mediation environment may be dispersed among programming systems, mediators, GLAV-views, wrappers and resources. Methods and tools for representation, manipulation, and estimation of efficiency of dispersed organization are provided. Algorithms for construction of efficient dispersed organization have been implemented.

7. An original approach for binding of programming languages with declarative mediator rule language has been implemented. The approach combines static and dynamic binding overcoming impedance mismatch and allowing dynamic result types.

Most of the methods developed so far are applicable not only for data integration, but for data exchange [9] as well.

3 Directions of Research and Development

3.1 Investigation of application-driven approach for scientific problem solving in the subject mediator environment

According to the application-driven approach, on the basis of a problem domain, an ontology (concepts and relationships among them) of the domain is created. After that, a conceptual schema (including data structures and methods required for problem solving) is created. Thus, a semantic specification of a problem independent of concrete resources is developed.

Integration of resources relevant to a problem in a subject mediator requires semantic schema mapping methods and tools to be used for construction of mappings between mediator and resources schemas.

The authors got an experience of applying the application-driven approach to several problems, one of them being the *problem of secondary standards search for photometric calibration of optical components of gamma-ray bursts* formulated by the Institute of Space Research of RAS and specified as text in a natural language. The idea of the problem is to find a set of standard stars (stars with well-known parameters) in some area around a gamma-ray burst. The problem was formalized and implemented applying the subject mediation. To do that, a glossary of the problem domain was manually extracted from the textual specification and astronomical literature. After that, an ontology required for problem solving was constructed. Data structures (abstract data types), methods, and functions (e. g., for cross-match, color index calculation, star variability checking) constituting problem domain schema were defined.

Resources relevant to the problem were identified in the Astrogrid and VizieR information grids. A set of distributed resources includes SDSS, USNO B-1, 2MASS, GSC, and UCAC catalogs used for extraction of standards and VSX, ASAS, GCVS, and NSVS catalogs used for star variability checking. The resources were registered in the mediator and corresponding GLAV-views were obtained. The problem was formulated as a program consisted of a set of declarative rules over the mediator schema.

The implemented mediator is a basis for an application monitoring in real time the e-mails informing about the gamma-ray bursts. The application extracts standards located in the area of a burst and e-mails them to subscribers [6].

The issues requiring further investigations include:

- semantic identification of resources relevant to a mediator;
- construction of semantic source to target schema mapping in the presence of constraints reflecting specificity of various data models; and
- development of mediator program rewriting algorithms in the presence of source and mediator constraints over the classes of objects.

3.2 Heterogeneous multidialect mediator infrastructure for data, knowledge, and services semantic integration

3.2.1 An approach for the infrastructure

Recently, the World Wide Web Consortium (W3C) adopted RIF (Rule Interchange Format, http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/) standard oriented on providing of interoperability of declarative programs represented in different languages and rule-based programming (inference) systems. The standard is oriented not only on Semantic Web, but also on a creation of the intellectual information systems as well as on a knowledge representation in different application areas. This standard motivated the following investigation: to find a solution of the complicated problem of integration of multilanguage knowledge representations and rule-based declarative programs, heterogeneous databases, and services on the basis of unified languages and multidialect mediation infrastructure. The methods and tools developed are aimed at scientific problems solving over heterogeneous distributed information resources. The methods and tools to be developed are intended for combining two paradigms of extensible unified languages construction. The first one is W3C RIF standard paradigm. The second one is a paradigm based on the GLAV approach built on the extensible canonical information model idea, applicable for database, service and process languages unification, and mediation.

The idea of the proposed approach consists in developing of a modular mediator infrastructure in which alongside with the modules representing data and services in the GLAV setting, the modules representing knowledge and declarative rule-based programs over various resources will be introduced. The infrastructure is planned to be based on the following principles:

- *the multidialectal construction of the canonical model*. Mediators are represented as a functional composition of declarative specification of modules, each based on its own dialect with an appropriate semantics. Semantic of a conceptual definition in such setting becomes a multidialect one;

– *the mediator modules as peers*. Rule-based modules become the mediator components alongside with the GLAV-based modules. Interoperability of the modules is based on peer-to-peer (P2P) and W3C RIF techniques;
– *combination of integration and interoperability*. The information resource integration can be provided in the scope of an individual mediator module. The integration approaches in different modules can be different. The interoperability is provided between the modules supporting different dialects; and
– *rule-based specifications on different levels of the infrastructure*. Rule-based, inference providing modules are used for declarative programming over the mediators, to support various modules of a mediator, to support schema mapping for semantic integration of the information resources in the mediator, etc.

3.2.2 Example

The idea of the multidialect mediation infrastructure is demonstrated on example (Fig. 2) of finding an optimal assignment of applicants among universities. The program calculating such assignment is defined in DLV (Answer Set Programming). The required information resources are integrated in a SYNTHESIS mediator. OntoBroker communicates with users and applying its ontologies, formulates the queries to the mediator and after collecting the required data, initiates a program in DLV. The assignment problem is formulated as follows. A set of n applicants is to be assigned among m universities, where q_i is the quota of the ith college. Applicants (universities) rank the universities (the applicants) in the order of their preference. The aim is to find optimal assignment from the quotas of the colleges and the two sets of orderings.

An assignment of applicants to colleges is *unstable* if there are two applicants α and β who are assigned to colleges A and B, respectively, although β prefers A to B and A prefers β to α; otherwise, an assignment is *stable*.

A stable assignment is called *optimal* if every applicant is at least as well off under it as under any other stable assignment.

3.2.3 Issues to be investigated and prototyped

1. Approaches for constructing the rule-based dialect mappings.
2. Methods for justification of semantic preservation by the mappings (e. g., as a preserving of entailment of initial formulae by the mapping justified by test case checking; reducing entailment to refinement; manual proof using structural induction over constructs of a dialect, etc.).
3. Investigation of approaches for modular representation of knowledge in the multidialect mediation environment.
4. Investigation of approaches for providing the interoperability of the mediator multidialect modules.
5. Infrastructure design and prototyping.
6. Real problems solving in a scientific subject domains chosen.
7. Expansion of the experience into the Semantic Web area.

3.3 Mediation of databases with nontraditional data models

3.3.1 Motivation for the proposed research and development

The objective is to develop an approach providing for semantic integration of information resources represented in frame of the traditional as well as nontraditional data models aiming at problem solving over such integrated ensemble of resources.

During last years, nonrelational data models are being intensively developed called here collectively as nontraditional data models. Classes of such data models include NoSQL (not only Structured Query Language), graph data models, triple-based data models, ontological data models, "scientific" data models, etc.

The NoSQL data models are oriented on the support of extra large volumes of data applying a "key-value" technology for vertical storage. The examples of such systems include Dynamo, BigTable, HBase, Cassandra, MongoDB, CouchDB.

In the class of *graph data models*, one can find such systems as Neo4j, InfiniteGraph, DEX, InfoGrid, HyperGraphDB, Trinity, and supporting flexible data structures.

The triple-based data model (expressible in RDF (Resource Description Framework) and RDFS (RDF Schema)) is used for representing information about the Web resources and is used together with the OWL profiles, logic inference terchniques, SPARQL query language. Systems belonging to this class include Virtuoso, OWLIM, 5Store, and Bigdata.

One of the OWL profiles (OWL QL) is oriented on support of *ontological modeling* over relational databases. Recently, it was found an equivalent to OWL-inherent description logics mechanism expressible by data dependencies used together with declarative query languages (such as Datalog).

For the support of "*scientific*" data, several approaches are being developed (e. g., SciDB applying a multidimensional array data model).

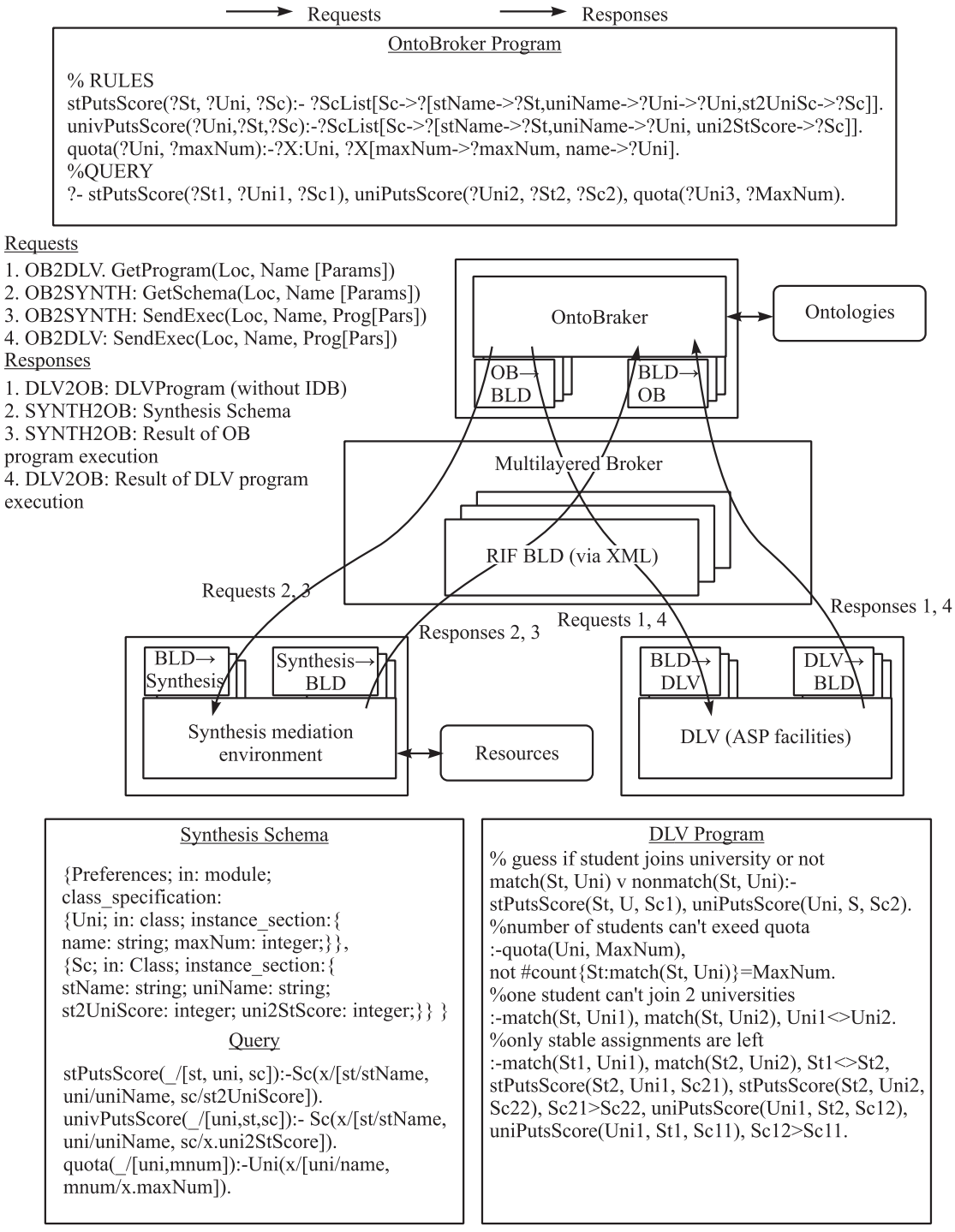


Figure 2 Example of a problem specification in the multidialect mediation infrastructure

It is remarkable that for the most of these data models, the standards still do not exist. The most of these data models and systems are oriented on "big data" support applying massive parallel technique of the MapReduce kind.

Such diversity of nontraditional data models and systems makes urgent the investigation of a possibility of such data stores usage in frame of the subject mediation paradigm in the GLAV setting. Such investigation includes the methods of mapping and transformation into the canonical information mediation model of the nontraditional data models, techniques of interpretation of canonical model facilities in data manipulation languages of the nontraditional classes, as well as an efficiency of such interpretation.

3.3.2 The results of research to be obtained

1. Development of information preserving methods of mapping and techniques for design of transformations of various classes of nontraditional data models into the canonical one, design of such mappings, and transformations for specific data models and of adequate extensions of the canonical data model.
2. Investigation of techniques for interpretation of canonical model data manipulation language (DML) (including query language) in the DMLs of different classes of nontraditional data models and approaches for their implementation.
3. Obtaining of architectural decisions on implementation of the massive parallel techniques on the level of mediators, evaluation of performance growth that can be reached.
4. Evaluation of suitability and efficiency of integration of nontraditional data models of different classes in the GLAV mediation infrastructure for various problem domains.

3.4 Storage of very large volumes of data

The objective is to develop a *novel distributed parallel fault-tolerant file system* possessing the following capabilities:

- storage of data volumes of petabyte scale;
- unlimited period of storage;
- scalability;
- efficient multiuser access support in different kinds of networks; and
- usage of different storage types (e. g., hard disk drive and flash memory).

The experience of existing file systems vendors should be taken onto account:

- ReFS (Windows Server 8) by Microsoft;
- VMFS by VMware;
- Lustre;
- ZFS by Sun Microsystems;
- zFS (z/OS) by IBM; and
- OneFS by Isilon.

4 Concluding Remarks

The paper discusses the research and development directions required to match up the information integration technologies challenges including an emphasis on semantic issues requiring a provision for support of executable declarative specification of the application over the mediator, enhancement of presence of knowledge-based facilities at the mediator level, compliance with the proliferation of new kinds of data sources in the data space, specifically with the need in the big data reflected in development of various nontraditional database systems and data models to be integrated.

The paper has been prepared for the discussions in connection with the Information Session on Sk-Tech International Research Centers Call for Proposals, February 9–10, 2012, at the Massachusetts Institute of Technology (http://web.mit.edu/sktech/news-events/pr-save-the-date.html).

Acknowledgements

The authors express deep gratitude to Dmitry Kovalev for the example formulation (paragraph 3.2.2).

References

1. *Kalinichenko L. A., Briukhov D. O., Martynov D. O., Skvortsov N. A., Stupnikov S. A.* Mediation framework for enterprise information system infrastructures // 9th Conference (International) on Enterprise Information Systems ICEIS 2007 Proceedings: Volume databases and information systems integration. — Funchal, 2007. P. 246–251.
2. *Briukhov D. O., Kalinichenko L. A., Martynov D. O.* Source registration and query rewriting applying LAV/GLAV techniques in a typed subject mediator // 9th Conference (Russian) on Digital Libraries RCDL'2007 Proceedings. — Pereslavl-Zalesskij: Pereslavl University, 2007. P. 253–262.
3. *Kalinichenko L. A., Stupnikov S. A., Martynov D. O.* SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. — M.: IPI RAN, 2007. 171 p.

4. *Kalinichenko L. A.* Canonical model development techniques aimed at semantic interoperability in the heterogeneous world of information modeling // Knowledge and model driven information systems engineering for networked organizations: CAiSE INTEROP Workshop Proceedings. — Riga: Riga Technical University, 2004. P. 101–116.
5. *Briukhov D. O., Vovchenko A. E., Zakharov V. N., Zhelenkova O. P., Kalinichenko L. A., Martynov D. O., Skvortsov N. A., Stupnikov S. A.* The middleware architecture of the subject mediators for problem solving over a set of integrated heterogeneous distributed information resources in the hybrid grid-infrastucture of virtual observatories // Informatics and Applications, 2008. Vol. 2. Is. 1. P. 2–34.
6. *Vovchenko A. E., Kalinichenko L. A., Stupnikov S. A.* Mediation based semantic grid. distributed computing and grid-technologies in science and education // 4th Conference (International) Proceedings. — Dubna: JINR, 2010. P. 309–318.
7. *Zakharov V. N., Kalinichenko L. A., Sokolov I. A., Stupnikov S. A.* Development of canonical information models for integrated information systems // Informatics and Applications, 2007. Vol. 1. Iss. 2. P. 15–38.
8. *Kalinichenko L. A., Stupnikov S. A.* Constructing of mappings of heterogeneous information models into the canonical models of integrated information systems // Advances in Databases and Information System: 12th Conference (East-European) Proceedings. — Pori: Tampere University of Technology, 2008. P. 106–122.
9. *Fagin R., Kolaitis P., Miller R., Popa L.* Data exchange: Semantics and query answering // Theor. Computer Sci., 2005. Vol. 336. No. 1. P. 89–124.

РАЗВИТИЕ ТЕХНОЛОГИЙ ИНТЕГРАЦИИ ИНФОРМАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ НАД НЕОДНОРОДНЫМИ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ

Л. А. Калиниченко[1], С. А. Ступников[2], В. Н. Захаров[3]
[1]Институт проблем информатики Российской академии наук, leonidk@synth.ipi.ac.ru
[2]Институт проблем информатики Российской академии наук, ssa@ipi.ac.ru
[3]Институт проблем информатики Российской академии наук, vzakharov@ipiran.ru

Аннотация: Рассмотрены актуальные проблемы в области решения задач над неоднородными распределенными информационными ресурсами. Изложены основные достижения технологии предметных посредников, предназначенной для заполнения увеличивающегося разрыва между пользователями (приложениями) и неоднородными ресурсами данных, знаний и сервисов. Рассмотрены также актуальные проблемы технологии семантической интеграции информации, включающие исследование движимого приложениями подхода к решению задач в среде предметных посредников; обеспечение поддержки исполняемых декларативных спецификаций приложений над посредниками; расширение применения ориентированных на знания средств на уровне посредников; применение технологии предметных посредников для баз данных, основанных на нетрадиционных моделях данных и мотивированных потребностями поддержки «больших данных».

Ключевые слова: предметные посредники; неоднородные информационные ресурсы; решение научных задач; интеграция информации; движимый приложениями подход; языки правил; нетрадиционные модели данных

ТЕМАТИЧЕСКИЙ РАЗДЕЛ

Обработка изображений и распознавание образов

Настоящий раздел носит тематический характер и посвящен различным актуальным прикладным проблемам в области обработки изображений и распознавания образов. Материалы раздела содержат полные версии докладов, представленных на Международной конференции по компьютерной графике и зрению и Всероссийской конференции «Математические методы распознавания образов».

Статья К. В. Рудакова и И. Ю. Торшина «Анализ информативности мотивов на основе критерия разрешимости в задаче распознавания вторичной структуры белка» посвящена актуальной проблеме развития методов и алгоритмов распознавания вторичной структуры белка по данным о первичной структуре. В статье И. Н. Белых, А. И. Капустина, А. В. Козлова, А. И. Лохановой, Ю. Н. Матвеева, Т. С. Пеховского, К. К. Симончика и А. К. Шулипы «Система идентификации дикторов по голосу для конкурса *NIST SRE 2010*» изложены результаты испытаний биометрической технологии текстонезависимой идентификации диктора «Центра речевых технологий» на международных испытаниях NIST SRE, где технология получила очень высокие оценки по точности распознавания. В статье В. Ю. Гудкова и М. В. Бокова «Быстрая обработка изображений отпечатков пальцев» затронут вопрос ускорения обработки дактилоскопических изображений, что является критическим фактором для встраиваемых систем биометрической идентификации. В статье Ю. В. Визильтера, В. С. Горбацевича, С. Л. Каратеева, Н. А. Костромова «Обучение алгоритмов выделения кожи на цветных изображениях лиц» представлено оригинальное решение задачи цветовой сегментации кожи человека на изображениях. Статья А. В. Куракина «Распознавание жестов ладони в реальном времени на основе плоских и пространственных скелетных моделей» содержит описание экспериментального комплекса распознавания жестов ладони, реализующего разработанный автором метод определения координат кончиков пальцев на бинарном изображении ладони посредством анализа его скелетного представления. Статья Д. В. Мурашова «Комбинированный подход к локализации различий многомодальных изображений» посвящена вопросу одновременного исследования мультиспектральных изображений, получаемых при реставрации живописных произведений. Предложенные методы и алгоритмы позволяют существенно автоматизировать некоторые этапы реставрационных работ. В статье О. С. Ушмаева и В. В. Кузнецова «Алгоритмы защищенной биометрической верификации на основе бинарного представления топологии отпечатков пальцев» представлены результаты исследования по совмещению криптографических конструкций и методов биометрической идентификации по отпечаткам пальцев.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 79–90

АНАЛИЗ ИНФОРМАТИВНОСТИ МОТИВОВ НА ОСНОВЕ КРИТЕРИЯ РАЗРЕШИМОСТИ В ЗАДАЧЕ РАСПОЗНАВАНИЯ ВТОРИЧНОЙ СТРУКТУРЫ БЕЛКА*

К. В. Рудаков[1], И. Ю. Торшин[2]

Аннотация: Представлено развитие формального описания задачи распознавания вторичной структуры белка. Вводятся ключевые понятия (мотив, оценка информативности мотива, порядок на мотивах), позволяющие использовать разрабатываемый формализм для анализа реально существующих множеств прецедентов. Приведены результаты экспериментов по тестированию разрешимости задачи. Показано, что анализ разрешимости позволяет проводить эффективный отбор наиболее информативных мотивов.

Ключевые слова: алгебраический подход; биоинформатика; локальность; разрешимость; теория классификации значений признаков

1 Введение

Распознавание вторичной структуры белка на основе его первичной структуры (аминокислотной последовательности) — одна из важнейших задач современной теоретической биологии [1–3]. Актуальность задачи обусловлена значительным объемом данных по первичной структуре белка (миллионы аминокислотных последовательностей) и в сотни раз меньшим количеством экспериментальных данных по третичной и, следовательно, вторичной структуре белка. Это позволяет рассматривать накопленный материал о третичном и вторичном уровнях структуры белка как обучающую выборку для задачи распознавания вторичной структуры белка по его первичной структуре. В рамках настоящего исследования данная задача рассматривается как перевод последовательности символов из одного алфавита в другой [3] (рис. 1).

Данная статья является продолжением работы [3], где были подробно рассмотрены мотивация и постановка проблемы. Приведем краткое введение в проблемную область.

Клетка — мельчайшая структурная единица организма, а белки — активные молекулярные образования, поддерживающие жизнь клетки. В современной биологии любой белок рассматривается с нескольких точек зрения: (1) как одномерная аминокислотная последовательность (так называемая «первичная структура белка», 1D); (2) как одномерная последовательность характерных локальных конфигураций («вторичная структура», 2D); (3) как трехмерный объект («третичная структура», «пространственная структура», 3D) и (4) как особый механизм, выполняющий определенную роль в функционировании клетки [1]. Одной из основных задач биоинформатики считается установление закономерностей, определяющих взаимосвязь первичной, вторичной и третичной структур.

Следует отметить, что имеющиеся данные о первичном, вторичном и третичном уровнях описания структуры белка получены на основании существенно различных экспериментов.

Первичная структура (последовательность символов в 20-буквенном алфавите) устанавливается посредством «секвенирования» (*досл.* «установления последовательности») — процедуры последовательной химической деградации молекулы белка. Вторичная структура (последовательность локальных конфигураций) и третичная структура (набор координат атомов) устанавливаются дифракционными методами (как правило, рентгеноструктурным анализом) или посредством исследования внутримолекулярного спин-спинового расщепления с использованием ЯМР (ядерного магнитного резонанса).

В то время как точность секвенирования определяется однозначно как совпадение–несовпадение символов и достигает 100%, в различных экспериментах по определению структуры одного и того же белка устанавливаются отличающиеся друг от друга наборы координат атомов молекул этого белка. Эти

*Работа выполнена при поддержке РФФИ (гранты 09-07-12098, 09-07-00212-а и 09-07-00211-а) и Минобрнауки РФ (контракт № 07.514.11.4001).

[1]Вычислительный центр Российской академии наук им. А. А. Дородницына; Московский физико-технический институт, rudakov@ccas.ru

[2]Московский физико-технический институт; Центр систем прогнозирования и распознавания (ЦСПР), tiy135@yahoo.com

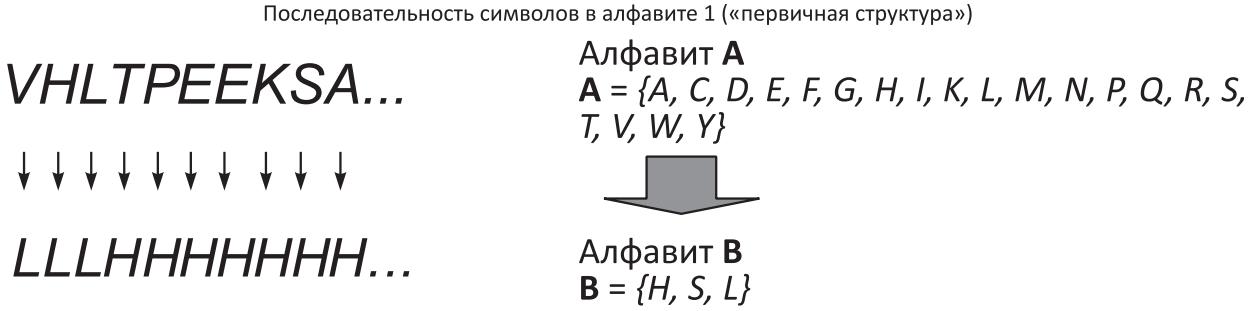


Рис. 1 Задача распознавания вторичной структуры белка

отличия зависят от многочисленных условий проведения структурного эксперимента: выбора метода (дифракция, ЯМР), температуры, кислотности среды (pH), качества кристалла (в случае дифракционного метода), присутствия других молекул и др.

Экспертный анализ третичных структур белков указал на существование ряда характерных пространственных конфигураций локальных участков молекулы белка: «спиралей», «стрэндов» и «петель». Последовательности этих пространственных конфигураций были названы «вторичной структурой белка». Вторичная структура как последовательность символов некоего алфавита (различные способы определения этого алфавита рассмотрены далее) — результат интерпретации набора координат атомов молекулы белка. Так как координаты атомов и межатомные расстояния подвержены вариациям вследствие упомянутых выше особенностей структурного эксперимента, то и вторичная структура как последовательность символов также подвержена вариациям. Это обуславливает противоречивость имеющихся выборок экспериментальных данных (в соответствии с данными из PDB, *Protein Data Bank* [4]) и приводит к необходимости формирования непротиворечивых множеств прецедентов.

Применение современных физических методов для исследования структуры и свойств белковых молекул позволяет предположить *локальный характер зависимости вторичной структуры от первичной*. Гипотеза о локальности подразумевает, что вторичная структура в данной позиции последовательности определяется не всей аминокислотной последовательностью, а некой окрестностью данной позиции (локальным контекстом).

Таким образом, противоречивость экспериментальных данных, обусловленная изменениями структуры белка в различных условиях и особенностями структурного эксперимента, и необходимость систематического исследования гипотезы о локальном характере взаимосвязи между первичной и вторичной структурой указали на целесообразность разработки специализированного формализма для корректной постановки изучаемой проблемы. В работах [2, 3] было предложено точное описание данной задачи распознавания и рассмотрена ее разрешимость, регулярность и локальность. Введение ключевых понятий для анализа локальности (окрестность, маска, система масок, монотонность и тупиковость систем масок) позволило предложить метод построения безызбыточных систем масок.

Одним из основных результатов работ [2, 3] является формулировка критериев разрешимости задачи распознавания вторичной структуры. В настоящей статье представлено дальнейшее развитие формализма и результаты вычислительных экспериментов по тестированию разрешимости. Осуществляется переход к разрешимости на мотивах; для сокращения полного перебора при комбинаторном тестировании условия разрешимости вводится понятие информативности мотива. На основе информативности предлагается метод формирования непротиворечивых множеств прецедентов. Приведены результаты экспериментов для отбора множеств информативных мотивов, которые являются основой для построения корректных алгоритмов в рамках алгебраического подхода к распознаванию. Эксперименты проводились на подвыборках общедоступных экспериментальных данных по первичной, вторичной и третичной структурам белков [4].

2 Критерий разрешимости на объектах и мегасловах

В рамках разрабатываемого формализма используются два алфавита: алфавит **A** для описания первичной структуры белка («верхнего слова») и алфавит **B** для описания вторичной структуры

(«нижнего слова»). Пусть $\mathbf{A} = \{a_1, a_2, \ldots, a_{n(\mathbf{A})}\}$, $n(\mathbf{A}) = |\mathbf{A}| > 0$ и $\mathbf{B} = \{b_1, b_2, \ldots, b_{n(\mathbf{B})}\}$, $n(\mathbf{B}) = |\mathbf{B} > 0$. Алфавит $\mathbf{A}$ (однобуквенные обозначения аминокислот) обычно определяется как $\mathbf{A} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, R, S, T, V, W, Y\}$. Алфавит $\mathbf{B}$ может быть определен разными способами [3]. Для целей настоящей работы вполне приемлем трехбуквенный алфавит $\mathbf{B} = \{S, H, L\}$, описывающий три принципиально различных вида последовательной организации пространственных структур белков: «стрэнды» (S, англ. *strand*), «спирали» (H, *helix*), и «петли» (L, *loop*).

Произвольное слово в алфавите $\mathbf{A}$ будем обозначать $V = v_1 v_2 \ldots v_{n(V)}$, в алфавите $\mathbf{B}$ — $W = w_1 w_2 \ldots w_{n(W)}$, где $n(V)$ и $n(W)$ — длины слов. Критерий локальной разрешимости с использованием отдельных масок (выражение (6″) в работе [3]) был сформулирован следующим образом:

$$\underset{\mathbf{Pr}}{\forall} \left(V^1, W^1\right), \left(V^2, W^2\right) \forall (i, j): \left(\bigvee_{k=1}^{|\mathbf{M}|} \hat{m}_k : \eta\left(i, \hat{m}_k, V^1\right) = \eta\left(j, \hat{m}_k, V^2\right) \right) \Rightarrow \Rightarrow w_i^1 = w_j^2, \; l(\mathbf{M}) < i \le |V^1| - r(\mathbf{M}), \; l(\mathbf{M}) < j \le |V^2| - r(\mathbf{M}), \; i \ne j, \quad (1)$$

где $\left(V^1, W^1\right)$ и $\left(V^2, W^2\right)$ — произвольные элементы *множества прецедентов* $\mathbf{Pr}$; i и j — *ведущие позиции* в прецедентах; $\mathbf{M} = \{\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_{|\mathbf{M}|}\}$ — *множество (система) масок*; $\hat{m}_k = \{\mu_1^k, \mu_2^k, \ldots, \mu_{m(k)}^k\}$ — k-*я маска* ($\mu_i^k \in Z$, $\mu_1^k < \mu_2^k < \ldots < \mu_{m(k)}^k$, $k = 1, \ldots, |\mathbf{M}|$); μ_i^k — i-я позиция k-й маски; $m(k) = |\hat{m}_k|$ — *размерность маски* $\hat{m}_k$; η — *оператор выбора подслова*; $l(\mathbf{M})$ и $r(\mathbf{M})$ — границы для описания краевых эффектов, $l(\mathbf{M}) = \max\left(-\max_{k=1,|\mathbf{M}|} \mu_1^k, 0\right)$, $r(\mathbf{M}) = \max\left(\min_{k=1,|\mathbf{M}|} \mu_{|m(k)|}^k, 0\right)$. В дальнейшем предполагается выполнение указанных в (1) ограничений по $l(\mathbf{M})$ и $r(\mathbf{M})$ на значения i и j. Будем также использовать обозначение $[\hat{m}_k]$ — *протяженность* маски $\hat{m}_k$: $[\hat{m}_k] = \mu_{m(k)}^k - \mu_1^k + 1$.

Утверждение (1) соответствует локальной форме задачи распознавания вторичной структуры, т. е. существованию функции $f: \mathbf{A}^{|\hat{m}_\Sigma(\mathbf{M})|} \to \mathbf{B}$, где $\hat{m}_\Sigma(\mathbf{M})$ — *объединенная маска* $\mathbf{M}$, $\hat{m}_\Sigma(\mathbf{M}) = \bigcup_{k=1}^{|\mathbf{M}|} \hat{m}_k$ [3].

Элементарными объектами q (в дальнейшем просто объектами) будем называть элементы множества $\mathbf{Q} = \mathbf{A}^{|\hat{m}_\Sigma(\mathbf{M})|} \times \mathbf{B}$. Множество объектов $\mathbf{Q}(\mathbf{Pr}, \mathbf{M}) = \{q_1, q_2, \ldots, q_N\}$, $N = |\mathbf{Q}(\mathbf{Pr}, \mathbf{M}|$ однозначно получается при переборе всех допустимых значений i в прецедентах, причем допустимость i определяется упомянутыми выше $l(\mathbf{M})$ и $r(\mathbf{M})$. Иначе говоря, элементами *наблюдаемых множеств объектов* $\mathbf{Q}(\mathbf{Pr}, \mathbf{M})$ являются пары $q_i^j = (\eta(i, \hat{m}_\Sigma(\mathbf{M}), V^j), w_i^j)$; каждая пара есть совокупность подслова, выбранного по $\hat{m}_\Sigma(\mathbf{M})$ в i-й ведущей позиции верхнего слова ($V^j = v_1^j v_2^j \ldots v_{n(V^j)}^j$) и i-й литеры нижнего слова ($W^j = w_1^j w_2^j \ldots w_{n(W^j)}^j$) j-го прецедента.

В i-й позиции произвольного прецедента (V, W) по системе масок $\mathbf{M}$ фактически считывается *мегаслово* или *вектор подслов* $\vec{V}_i = (V_1(i), V_2(i), \ldots, V_{|\mathbf{M}|}(i))$, где $V_k(i) = \eta(i, \hat{m}_k, V)$. Мегаслово также может быть считано с нулевой позиции произвольного объекта $q_j \in \mathbf{Q}(\mathbf{Pr}, \mathbf{M})$, $q_j = (V_j, w_j)$, так что $\vec{V}_j = (V_{j,1}, V_{j,2}, \ldots, V_{j,|\mathbf{M}|})$, где $V_{j,k} = \eta(0, \hat{m}_k, V_j)$. Введение элементарных объектов и мегаслов позволяет значительно упростить запись критерия разрешимости, заменив в (1) сравнение прецедентов сравнением объектов, а сравнение подслов — сравнением мегаслов.

Назовем *мотивами* κ элементы множества $\mathbf{K} = \{(\hat{m}, V) | \hat{m} \in \mathbf{M}, n(V) = |\hat{m}|\}$. Будем говорить, что мотив $\kappa = (\hat{m}, V')$ *присутствует в объекте* $q = (V, w)$, если $\eta(0, \hat{m}, V) = V'$. Обозначим принадлежность мотива объекту q как $\kappa \in^* q$. Аналогично мотив $\kappa = (\hat{m}, V)$ *присутствует в мегаслове* $\vec{V}_i = (V_1(i), V_2(i), \ldots, V_{|\mathbf{M}|}(i))$, если для k-й маски $\hat{m} = \hat{m}_k$ и $V = V_k(i)$. Вхождение мотива в мегаслово обозначим $\kappa \in^* \vec{V}_i$. Для произвольной пары объектов q_1 и q_2 мотив κ назовем *отличающим*, если κ присутствует в одном из объектов и отсутствует во втором. Пусть $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ — множество всех мотивов, присутствующих в словах из $\mathbf{Pr}$ при данной системе масок $\mathbf{M}$.

Теорема 1. *Условие локальной разрещимости задачи выполнено тогда и только тогда, когда для каждой пары объектов* $q_1 = (V_1, w_1)$ *и* $q_2 = (V_2, w_2)$ *при* $w_1 \ne w_2$ *существует хотя бы один отличающий мотив.*

Доказательство. Запишем критерий разрешимости (1) с использованием мегаслов:

$$\underset{\mathbf{Pr}}{\forall} \left(V^1, W^1\right), \left(V^2, W^2\right) \forall (i, j): \left(V_1^1(i), V_2^1(i), \ldots, V_{|\mathbf{M}|}^1(i)\right) = \left(V_1^2(i), V_2^2(i), \ldots, V_{|\mathbf{M}|}^2(i)\right) \Rightarrow w_i^1 = w_j^2.$$

Заменяя перебор различных ведущих позиций по всем парам прецедентов на перебор пар объектов из $\mathbf{Q}(\mathbf{Pr}, \mathbf{M})$, а сравнение мегаслов прецедентов — на сравнение мегаслов объектов, получим:

$$\underset{\mathbf{Q}(\mathbf{Pr},\mathbf{M})}{\forall}(i,j):\ \left(V_{i,1},V_{i,2},\ldots,V_{i,|\mathbf{M}|}\right)= \\ =\left(V_{j,1},V_{j,2},\ldots,V_{j,|\mathbf{M}|}\right)\Rightarrow w_i=w_j\,. \quad (2)$$

Будем считать, что $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ — непротиворечиво, т. е. для всех пар объектов выполнено условие (2). Применив логический оператор НЕ и обозначив мегаслова i-го и j-го объектов как $\vec{V}_i = \left(V_{i,1},V_{i,2},\ldots,V_{i,|\mathbf{M}|}\right)$ и $\vec{V}_j=\left(V_{j,1},V_{j,2},\ldots,V_{j,|\mathbf{M}|}\right)$, получим обратную форму утверждения (2):

$$\underset{\mathbf{Q}(\mathbf{Pr},\mathbf{M})}{\forall}(i,j):\ w_i\neq w_j\Rightarrow\vec{V}_i\neq\vec{V}_j\,. \quad (2')$$

Очевидно, что два мегаслова в выражении (2′) различны, если при попарном сравнении подслов в соответствующих позициях мегаслов не совпадает хотя бы одно из этих подслов, т. е.

$$\underset{\mathbf{Q}}{\forall}(i,j):\ w_i\neq w_j\Rightarrow\overset{|\mathbf{M}|}{\underset{k=1}{\exists}}k:\ V_{i,k}\neq V_{j,k}\,, \quad (2'')$$

где $V_{j,k}=\eta(0,\hat{m}_k,V_j)$. Очевидно, что k-я позиция в j-м мегаслове соответствует маске $\hat{m}_k$ и подслову $V_{j,k}$, т. е. некоторому мотиву $\kappa=(\hat{m}_k,V_{j,k})$, $\kappa\in\mathbf{K}(\mathbf{Pr},\mathbf{M})$. Таким образом, условие (2″) записывается как критерий *разрешимости на множестве мотивов*:

$$\underset{\mathbf{Q}(\mathbf{Pr},\mathbf{M})}{\forall}(i,j):w_i\neq w_j\Rightarrow\underset{\mathbf{K}(\mathbf{Pr},\mathbf{M})}{\exists}\kappa:\ \left(\kappa\in^*\vec{V}_i\right)\neq \\ \neq\left(\kappa\in^*\vec{V}_j\right)\,. \quad (3)$$

Выражение (3) доказывает необходимость. Достаточность доказывается от противного. Предположим, что (3) не выполнено и для определенной пары объектов q_1, q_2 из $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ при $w_1\neq w_2$ не существует отличающего мотива. Тогда мегаслова для данной пары объектов будут равны (2″) и в соответствии с (2′) одному и тому же мегаслову будут соответствовать различные литеры нижнего слова. Последнее противоречит критерию разрешимости (1), (2). Теорема доказана.

Теорема 1 имеет принципиальное значение как для дальнейшего развития разрабатываемого формализма, так и для его практических приложений. Утверждение (3) соответствует переходу от задачи $\mathbf{Z}(\mathbf{Pr},\mathbf{M})$ [3] к эквивалентной задаче $\mathbf{Z}(\mathbf{Q},\mathbf{K})$, в которой в качестве параметров выступают множество объектов $\mathbf{Q}=\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ и множество мотивов $\mathbf{K}=\mathbf{K}(\mathbf{Pr},\mathbf{M})$.

Для практического применения разрабатываемого формализма особый интерес представляет поиск минимальных наборов мотивов, гарантирующих разрешимость. Подобно тому, как в работе [3] анализировалась монотонность условия разрешимости по системам масок и исследовалась тупиковость систем масок, здесь рассматривается монотонность условия разрешимости (3) по отношению «быть подмножеством» на множестве всех подмножеств множества мотивов.

3 Монотонность условия разрешимости и тупиковые системы мотивов

Разрешимость задачи $\mathbf{Z}(\mathbf{Q},\mathbf{K})$ по (3) зависит, естественно, от $\mathbf{Q}$ и $\mathbf{K}$. Множество объектов $\mathbf{Q}$ *непротиворечиво* при определенной $\hat{m}_\Sigma(\mathbf{M})$, если $\mathbf{Q}$ удовлетворяет условию (2′). Способы формирования непротиворечивых $\mathbf{Q}$ будут рассмотрены отдельно. Ниже рассматриваются возможности варьирования множества мотивов $\mathbf{K}$ при непротиворечивых множествах объектов.

Варьирование $\mathbf{K}$ сводится к добавлению или удалению отдельных мотивов. Добавление мотивов к $\mathbf{K}$ при постоянных $l(\mathbf{M})$ и $r(\mathbf{M})$ не нарушает истинности (3), т. е. *условие* (3) *монотонно по* $\mathbf{K}$ *при* $\mathbf{K}\subseteq\mathbf{K}'$.

Монотонность условия (3) важна для нахождения безызбыточных и тупиковых множеств мотивов. Действительно, множество мотивов $\mathbf{K}$, при котором условие (3) выполнено на всех парах объектов из $\mathbf{Q}$, может быть избыточно в том смысле, что разрешимость сохранится при удалении некоторых мотивов. Если условие (3) выполнено для $\mathbf{K}$, но не выполнено для любого $\mathbf{K}'\subset\mathbf{K}$, то такое множество мотивов назовем *тупиковым*.

Для исследования монотонности условия разрешимости (1) в работе [3] были введены понятия 0-тупиковости, тупиковости и ядерности систем масок. Было показано, что любая тупиковая $\mathbf{M}$ (потеря разрешимости при удалении любой маски из $\mathbf{M}$) является также 0-тупиковой (изменение $\hat{m}_\Sigma(\mathbf{M})$ при удалении любой маски) и ядерной (каждая маска однозначно соответствует уникальной позиции в $\hat{m}_\Sigma(\mathbf{M})$). Были сформулированы принципы построения алгоритма поиска безызбыточных систем масок.

Важно отметить, что изучение монотонности условия разрешимости на множествах мотивов — гораздо более «тонкий» исследовательский инструмент. Действительно, удаление даже одной маски из системы масок $\mathbf{M}$ соответствует удалению всех мотивов, порожденных данной маской. Пусть, например, M_n^m — система масок, образованная всеми сочетаниями m позиций из n возможных в соответствующей объединенной маске (т. е. m — раз-

мерность каждой маски M_n^m, а n — протяженность объединенной маски), так что $|M_n^m| = C_n^m$. Удаление любой маски из $\mathbf{M}$ повлечет за собой удаление всех $|\mathbf{A}|^m$ мотивов, порожденных данной маской. В практически интересных случаях $m = 3$ или $m = 4$, $|\mathbf{A}| = 20$, так что $|\mathbf{A}|^3 = 8000$, $|\mathbf{A}|^4 = 160\,000$. В то же время изучение монотонности критерия разрешимости (3) на множествах мотивов позволяет удалять отдельные мотивы.

Вообще говоря, определение тупиковых множеств мотивов $\mathbf{K}$ безызбыточной системы масок $\mathbf{M}$ — задача, разрешимая полным перебором. Однако полный перебор подмножеств множества из $C_n^m|\mathbf{A}|^m$ мотивов не представляется возможным практически. Редукция множества мотивов $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ и нахождение тупиковых множеств мотивов $\mathbf{K}$ может рассматриваться как частный случай выделения информативных значений признаков в теории классификации значений признаков [5, 6]. В рамках этой теории значения признаков объектов в задачах обучения по прецедентам рассматриваются как объекты некоторой задачи классификации, в которой требуется выделить во множестве всех значений всех исследуемых признаков подкласс «информативных».

4 Эвристические оценки информативности мотивов

При исследовании монотонности условия разрешимости возникает очевидный вопрос: какие мотивы следует удалять, а какие — оставлять? В духе теории классификации значений признаков можно сказать, что следует оставлять мотивы с «высокой информативностью» и удалять мотивы с «достаточно низкой» информативностью. При этом критерий разрешимости задачи распознавания (3) служит условием, предотвращающим удаление отличающих мотивов, обеспечивающих разрешимость задачи. Редукцию множества $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ можно, в частности, проводить на базе эвристических оценок информативности.

Оценка информативности мотивов $D: \mathbf{K} \to \mathbf{R}_+$ может быть введена различными способами так, чтобы бóльшая «информативность» мотива соотвествовала бóльшим значениям D. Строгое теоретико-множественное обоснование формы соответствующего функционала является отдельным направлением исследований и лежит за рамками настоящей статьи. Здесь вводится несколько эвристических оценок информативности мотивов, основанных на частоте их встречаемости в различных классах объектов.

Пусть $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ — множество мотивов для заданных $\mathbf{Pr}$ и $\mathbf{M}$, а $\mathbf{Q} = \mathbf{Q}(\mathbf{Pr}, \mathbf{M})$ — множество объектов. Каждый мотив $\kappa_\alpha \in \mathbf{K}(\mathbf{Pr}, \mathbf{M})$ входит в состав N_Σ^α объектов из $\mathbf{Q}$. При этом $N_\Sigma^\alpha = \sum\limits_{l=1}^{m=|\mathbf{B}|} N_l^\alpha$, где N_l^α соответствует числу объектов $q = (\vec{a}, b)$, у которых $b = b_l$, $b_l \in B$, так что мотиву κ_a поставлен в соответствие вектор $(N_1^\alpha, N_2^\alpha, \ldots, N_m^\alpha, N_\Sigma^\alpha)$. *Частота встречаемости* каждого значения $b_l \in \mathbf{B}$ определяется как $\nu_l^\alpha = N_l^\alpha / N_\Sigma^\alpha$, и, таким образом, мотиву κ_α оказывается сопоставлен вектор частот $(\nu_1^\alpha, \nu_1^\alpha, \ldots, \nu_m^\alpha, N_\Sigma^\alpha)$.

Пусть частоты встречаемости литер $b_l \in \in \mathbf{B}$ во всем множестве объектов $\mathbf{Q}$ составляют $(\nu_1^0, \nu_2^0, \ldots, \nu_m^0)$. Будем считать, что «информативность» мотива κ_α по данному b_l монотонна по $|\nu_l^\alpha - \nu_l^0|$: т. е. чем сильнее отличается ν_l^α от ν_l^0 — частоты b_l в среднем по $\mathbf{Q}$, тем более информативен мотив по литере b_l. Тогда D_l^α, оценку информативности α-го мотива по литере b_l (или по l-му классу вторичной структуры) естественно определить как некоторую V-образную функцию с единственным минимумом при $v_l^\alpha = \nu_l^0$ и такую, что $D_l^\alpha = 1$ при $v_l^\alpha = 1{,}0$ и 0. Этим требованиям удовлетворяет, например, кусочно-линейная функция

$$D_l^\alpha = \begin{cases} 1 - \dfrac{\nu_l^\alpha}{\nu_l^0} & \text{при } \nu_l^\alpha \le \nu_l^0\,; \\ \dfrac{\nu_l^\alpha - v_l^0}{1 - \nu_l^0} & \text{при } \nu_l^\alpha > \nu_l^0\,. \end{cases}$$

Величина D_l^α, т. е. информативность a-го мотива по литере b_l, указывает, насколько чаще данный мотив встречается в l-м классе объектов, чем в других, или, иначе говоря, отражает распределение вхождений мотива в объекты разных классов. Например, $D_l^\alpha = 1{,}0$ соответствует тому, что мотив встречается только среди объектов l-го класса или ни разу не встречается в данном классе. Отметим, что важным вариантом оценки D_l^α является оценка

$$D_l^\alpha = \begin{cases} \nu_l^\alpha \le \nu_l^0: & 0\,; \\ \nu_l^\alpha > \nu_l^0: & \dfrac{\nu_l^\alpha - \nu_l^0}{1 - \nu_l^0}\,. \end{cases}$$

Кроме сравнительных оценок распределения объектов между классами на информативность мотива влияет частота его встречаемости среди объектов. Иначе говоря, при фиксированном D_l^α более информативным будем считать мотив с бóльшим N_Σ^α. Используя введенные обозначения, можно предложить, по меньшей мере, три способа общей оценки информативности α-го мотива:

$$D_1(\alpha) = \sum_{l=1}^{m} D_l^\alpha \,;$$

$$D_2(\alpha) = N_\Sigma^\alpha D_1(\alpha) = N_\Sigma^\alpha \sum_{l=1}^{m} D_l^\alpha \,;$$

$$D(\alpha, D_0) = \begin{cases} N_\Sigma^\alpha & \text{при } D_1(\alpha) > D_0 \,; \\ 0 & \text{при } D_1(\alpha) \le D_0 \,. \end{cases}$$

Помимо сформулированных выше эвристических оценок информативности мотива могут быть предложены и другие. Интуитивно ясно, что «информативный» мотив должен выделять «достаточно много» объектов l-го класса N_l^α и «достаточно мало» объектов всех остальных классов $N_\Sigma^\alpha - N_l^\alpha$ [7]. В работе [8] приведено около 20 различных эвристических оценок информативности, представляющих собой разного рода эвристические функции от пары величин, аналогичных N_l^α и $N_\Sigma^\alpha - N_l^\alpha$, таких как энтропийный критерий «информационного выигрыша» (*information gain*), общеизвестные статистические критерии хи-квадрат, точный тест Фишера и др. [7, 8].

Эвристические оценки информативности мотивов необходимы для нахождения тупиковых множеств мотивов с учетом критерия разрешимости задачи. Кроме того, оценки информативности также могут быть использованы для формирования непротиворечивых множеств объектов.

5 Информативность мотивов и условие разрешимости

Пусть D — эвристическая оценка информативности мотивов, $D : \mathbf{K} \to \mathbf{R}_+$. Функция D ставит в соответствие каждому мотиву множества $\mathbf{K}$ его информативность из определенного подмножества $\mathbf{R}_+$. Отношение порядка на $\mathbf{R}_+$ порождает линейный порядок на множестве мотивов $\mathbf{K}$.

При наличии упорядоченного множества мотивов отбор наиболее информативных может быть осуществлен (1) как определение границы информативности такой, что при удалении «менее информативных» мотивов сохраняется разрешимость, или (2) как отбор достаточного для разрешимости количества «наиболее информативных» мотивов.

Определение скалярной границы и удаление «менее информативных» мотивов подразумевает введение некоторой системы пороговых значений на информативность мотивов. Мотивы, удовлетворяющие данным ограничениям, являются «информативными». В простейшем случае ограничением является введение некоторого порога $D_{\min}$ для значения используемой оценки информативности D. Порог $D_{\min}$ может быть вычислен с использованием итеративной процедуры с фиксированным инкрементом или же методом дихотомии. На каждом шаге истинность выражения (3) вычисляется для текущего значения D до достижения сходимости.

Возможно использование нескольких функций оценок информативности (D_1, N_Σ, D_2 и т. д.), и в качестве критериев отбора мотивов будут выступать несколько пороговых значений. Для нахождения пороговых значений могут быть использованы жадные алгоритмы или семейства поверхностей, подобных поверхности Парето в пространстве $\mathbf{R}^n$, где n — число используемых функций D.

Отбор «наиболее информативных» мотивов. Введение линейного порядка на можестве мотивов позволяет использовать данные об информативности мотивов при тестировании условия разрешимости в форме (3). Принцип отбора мотивов состоит в том, что для каждой пары объектов из $\mathbf{Q}$ находится различающий мотив с наивысшей информативностью. Отобранные таким образом мотивы образуют некоторое *множество различающих мотивов* $\mathbf{K}^0$ *с наивысшей информативностью* такое, что $\mathbf{K}^0 \subseteq \mathbf{K}(\mathbf{Pr}, \mathbf{M})$.

Теорема 2. *Множество* $\mathbf{K}^0$ *является тупиковым тогда и только тогда, когда для каждого мотива из* $\mathbf{K}^0$ *в* $\mathbf{Q}$ *существует хотя бы одна пара объектов, для которой данный мотив — единственный различающий.*

Доказательство. Сначала докажем достаточность. Любые два мотива $\kappa_\alpha = (\hat{m}_\alpha, V_\alpha)$ и $\kappa_\beta = (\hat{m}_\beta, V_\beta)$ могут быть упорядочены в соответствии со значениями $D(\alpha)$и $D(\beta)$. Перенумеруем все элементы $\mathbf{K} = \mathbf{K}(\mathbf{Pr}, \mathbf{M})$ так, чтобы линейный порядок мотивов соответствовал убыванию значений D: $\kappa_1, \kappa_2, \kappa_3, \ldots, \kappa_\alpha \ldots, \kappa_{|\mathbf{K}|}$, $D(\kappa_\alpha) \ge D(\kappa_{\alpha+1})$.

Пусть на исходном множестве мотивов $\mathbf{K}$ выполнено условие разрешимости (3). Определим функцию $K_f(i,j)$, находящую единственный мотив с максимальным D (и, следовательно, с минимальным номером мотива α), который позволит различить i-й и j-й объекты из $\mathbf{Q}$:

$$K_f(i,j) = \min_{1,\ldots,|\mathbf{K}|} \alpha : \ (\kappa_\alpha \in^* V_i) \ne (\kappa_\alpha \in^* V_j) \ . \quad (4)$$

Тогда минимальное множество мотивов $\mathbf{K}^0$, на котором сохраняется разрешимость, $\mathbf{K}^0 \subseteq \mathbf{K}(\mathbf{Pr}, \mathbf{M})$, определяется через *характеристическую функцию* $T(\alpha)$ следующим образом:

$$T(\alpha) = \begin{cases} 1 \equiv \exists_{\mathbf{Q}}(i,j) : \ (K_f(i,j) = \alpha) \,; \\ 0 \ \text{в противном случае} \,. \end{cases} \quad (5)$$

Для каждой пары из i-го и j-го объектов множества Q функция $K_f(i,j)$ находит наиболее информативный различающий мотив. Для всех таких мотивов $T(\alpha)=1$, т. е. эти мотивы образуют $\mathbf{K}^0$. После вычисления $T(\alpha)$ для всех пар объектов из $\mathbf{Q}$ каждому i-му объекту из $\mathbf{Q}$ соответствует n_i^{rm} различающих мотивов из $\mathbf{K}^0$, $n_i^{rm}=|\{T(\alpha)=1\}_i|$. Объекты с $n_i^{rm}=0$ назовем «0-*объектами*», а с $n_i^{rm}=1$ — «1-*объектами*». Очевидно, что различающий мотив единствен только в парах объектов, составленных из 0-объекта и 1-объекта (т. е. $n_i^{rm}+n_j^{rm}=1$).

Теперь представим, что из $\mathbf{K}^0$ удаляется k-й мотив, найденный в N_Σ^k объектах. Если $n_i^{rm}>1$ для всех N_Σ^k объектов, то и $n_i^{rm}+n_j^{rm}>1$ и удаление мотива может и не приводить к потере разрешимости. Когда же $n_i^{rm}=1$ для одного из N_Σ^k объектов, то при сравнении этого объекта с произвольным 0-объектом другого класса k-й мотив будет единствен в этой паре объектов и удаление этого мотива неизбежно приведет к потере разрешимости. Множество $\mathbf{K}^0$ не может не быть тупиковым, когда последнее утверждение справедливо для всех мотивов.

Необходимость доказывается от противного. Пусть $\mathbf{K}^0$ — тупиковое множество мотивов. Условием тупиковости $\mathbf{K}^0$ является потеря разрешимости при удалении произвольного мотива. В соответствии с (3) разрешимость теряется, когда при $W_i\neq W_j$ не существует различающих мотивов, т. е. $n_i^{rm}+n_j^{rm}=0$. Пусть произвольный k-й мотив из тупикового $\mathbf{K}^0$ встречается в N_Σ^k объектах и для всех этих объектов $n_i^{rm}>1$ (иными словами, для k-го мотива не существует пары объектов, для которой данный мотив — единственный различающий). Тогда при удалении k-го мотива $n_i^{rm}+n_j^{rm}>0$, т. е. возможно удаление из $\mathbf{K}^0$ произвольного мотива без потери разрешимости, и, следовательно, $\mathbf{K}^0$ не является тупиковым. Теорема доказана.

Следствие 1. *В общем случае множество* $\mathbf{K}^0$ *не является тупиковым*. Множество $\mathbf{K}^0$, вычисленное по выражению (5), будет тупиковым только при условии соответствия каждому мотиву, по крайней мере, одной пары объектов с единственным различающим мотивом. Различающий мотив единствен только в частном случае, когда пара объектов состоит из 0-объекта и 1-объекта различных классов. Построение $\mathbf{K}^0$ по (5) не гарантирует существования этого частного случая для каждого мотива.

Следствие 2. *Тупиковое множество мотивов может быть найдено путем итеративного удаления из* $\mathbf{K}^0$ *мотивов с наименьшей информативностью*. Существование в $\mathbf{Q}$ пар «0-объект – 1-объект» различных классов для произвольного мотива — условие тупиковости $\mathbf{K}^0$. Каждый α-й мотив встречается в N_Σ^α объектах, каждому из этих объектов соответствует число найденных в нем различающих мотивов (n_i^{rm} для i-объекта). Мотивы, входящие в объекты с $n_i^{rm}=1$, не могут быть удалены из $\mathbf{K}^0$ без потери разрешимости. В то же время удаление мотивов для всех объектов, у которых $n_i^{rm}>1$, не нарушает условия тупиковости. Так как целью является нахождение тупиковых множеств мотивов с *наибольшей информативностью*, то удаляться должны мотивы с наименьшими D.

Следствие 3. *Наличие всех 0-объектов в одном классе — необходимое условие разрешимости задачи*. Предположим, что все 0-объекты, кроме i-го, сосредоточены в классе 1, а i-й 0-объект — в классе 2. Тогда при сравнении пар объектов $W_i^2\neq W_j^1$ будет происходить потеря разрешимости для всех j, соответствующих 0-объектам ($n_j^{rm}=0$).

Следствие 4. *Наличие всех нуль-объектов в одном классе — необходимое условие тупиковости множества мотивов* $\mathbf{K}^0$. Тупиковость $\mathbf{K}^0$ подразумевает разрешимость задачи. При нарушении необходимого условия разрешимости (следствие 3) о тупиковости не может идти и речи.

Следствие 5. *Тупиковость множества* $\mathbf{K}^0$ *гарантирована только при постановке задачи в двухклассовой форме*. Предположим, что задача распознавания поставлена для трех классов ($|\mathbf{B}|=3$ — типичный пример). Из следствий 3 и 4 очевидно, что все 0-объекты должны быть сосредоточены в одном классе. Пусть это будет класс 3. Классы 1 и 2 содержат только 1-объекты и объекты с $n_i^{rm}>1$. Тогда при попарном сравнении объектов классов 1 и 2 необходимое условие тупиковости $n_i^{rm}+n_j^{rm}=1$ никогда не будет выполнено.

Теорема 2 и ее следствия позволяют не только вычислить минимальное и тупиковое множества мотивов максимальной информативности, но и накладывают существенные структурные ограничения на процедуры тестирования разрешимости. При заданных непротиворечивом $\mathbf{Q}$ и функции D для оценки информативности мотивов следствие 2 позволяет определить тупиковое множество мотивов $\mathbf{K}^0$, используя, по сути дела, «жадный» алгоритм.

По следствию 5, для анализа тупиковости необходимо исследование разрешимости задачи в двухклассовой форме. При $|\mathbf{B}|=3$ и более это требование соответствует сведению задачи поиска тупиковых $\mathbf{K}^0$ к исследованию разрешимости таких задач, как «H/не H», «S/не S» и т. д. по отдельности. При этом $\mathbf{K}^0=\sup\limits_{l=1}^{|\mathbf{B}|}\mathbf{K}_l^0$; для $\mathbf{B}=\{S,H,L\}$ $\mathbf{K}^0=$ $=\mathbf{K}^0_{H/\neg H}\cup\mathbf{K}^0_{S/\neg S}\cup\mathbf{K}^0_{L/\neg L}$. В теории классифика-

ции значений признаков [5, 6, 9], разбиение $\mathbf{K}^0$ на подмножества $\mathbf{K}^0_l$ соответствует существованию ядерной эквивалентности функций-предикторов, отображающих множества значений признаков во множества классов.

В основе разрабатываемого формализма лежат два принципиальных допущения, анализ которых представляет собой отдельные направления дальнейших исследований. Во-первых, разрешимость на множестве мотивов определяется через введение D-функций, эвристических оценок информативности мотивов. Необходимо проведение строгого теоретико-множественного обоснования возможных форм соответствующего функционала, порождающего D-функции.

Во-вторых, условие $D(\kappa_\alpha) \geq D(\kappa_{\alpha+1})$ в процедуре вычисления $K_f(i,j)$ (выражение (4)) соответствует некоторому произволу в выборе мотива при $D(\kappa_\alpha) = D(\kappa_{\alpha+1}) = D(\kappa_{\alpha+2})$ и т. д. Произвол в выборе мотива поднимает вопрос о взаимосвязи тупиковости $\mathbf{K}^0$, построенных на разных выборках объектов, и проблем переобучения при построении алгоритмов распознавания. Варьирование встречаемости мотивов в различных выборках объектов также делает необходимым введение комбинаторных оценок значений D.

6 Об оценках информативности и непротиворечивых множествах объектов

Помимо нахождения тупиковых множеств мотивов эвристические оценки информативности мотивов могут также использоваться для решения важной промежуточной задачи — формирования непротиворечивых множеств объектов.

Пусть в произвольном $\mathbf{Q}(\mathbf{Pr}, \mathbf{M})$ имеются объекты с одинаковыми верхними словами — ситуация, типичная для множеств объектов, полученных на основе реальных экспериментальных данных по структуре белка [3]. Некоторое k-е подмножество объектов из $\mathbf{Q}$, в котором верхние слова равны определенному слову V, назовем *кластер-объектом* qc_k, $qc_k = \{(V, w_1), (V, w_2), \ldots, (V, w_m)\}$, $m = |qc_k|$. Множество $\mathbf{Q}$ разбивается на N_{qc} кластер-объектов, так что $\mathbf{Q} = \bigcup\limits_{k=1}^{N_{qc}} qc_k$.

В произвольном кластер-объекте каждому верхнему слову соответствует множество литер $\mathbf{B}$-алфавита. В непротиворечивом множестве объектов все эти литеры попарно равны. В противоречивых множествах объектов некоторые из литер будут попарно различны. *Условие непротиворечивости* $\mathbf{Q}$ *по кластер-объектам* записывается следующим образом:

$$\bigvee_{k=1}^{N_{qc}} qc_k \bigvee_{i,j=1}^{|qc_k|} q_i, q_j : \; w_i = w_j \,. \tag{6}$$

С физической точки зрения $|qc_k|$ является числом независимых экспериментов, в которых наблюдались объекты, составляющие кластер-объект qc_k. Вследствие рассмотренных ранее особенностей структурного эксперимента [3] выполнение условия (6) будет наблюдаться не для всех кластер-объектов, так что наряду с числом экспериментов $|qc_k|$ имеет смысл характеризовать кластер-объекты и по степени их непротиворечивости. Для этого также могут применяться эвристические оценки информативности мотивов.

Действительно, любой объект из $\mathbf{Q}$ можно рассматривать как мотив, выбранный по объединенной маске $\hat{m}_\Sigma(\mathbf{M})$ в определенной ведущей позиции верхнего слова некоторого прецедента. При этом $N^\alpha_\Sigma = \sum\limits_{l=1}^{|\mathbf{B}|} N^\alpha_l$, N^α_l — число объектов $q = (V, w)$ с $w = b_l$ в α-м кластер-объекте. Как и любой другой мотив, кластер-объект характеризуется вектором $(N^\alpha_1, N^\alpha_2, \ldots, N^\alpha_{|\mathbf{B}|}, N^\alpha_\Sigma)$, что позволяет вычислять предложенные ранее оценки информативности мотивов $D_1(\alpha)$, $D_2(\alpha) = N^\alpha_\Sigma D(\alpha)$ и др. Полученные значения D характеризуют «степень непротиворечивости» или же «информативность» кластер-объектов. С использованием $D_1(\alpha)$ запись условия непротиворечивости $\mathbf{Q}$ (6) упрощается:

$$\bigvee_{\alpha=1}^{N_{qc}} qc_\alpha : \; D_1(\alpha) = |\mathbf{B}| \,.$$

Для формирования непротиворечивых множеств объектов на практике наиболее приемлемой представляется упомянутая ранее стратегия исключения «менее информативных» (или «наиболее противоречивых») кластер-объектов путем введения скалярных границ. В качестве ограничений могут выступать фиксированное пороговое значение D, диаграмма Парето или же совокупность значений порогов для различных D:

$$\left.\begin{array}{l} \bigvee\limits_{\alpha=1}^{N_{qc}} qc_a : \; D_1(\alpha) > D^{\text{obj}}_{\min}\,; \\ \bigvee\limits_{\alpha=1}^{N_{qc}} qc_a : \; N^\alpha_\Sigma > N^{\text{obj}}_{\min}\,. \end{array}\right\} \tag{7}$$

Параметр $D^{\text{obj}}_{\min}$ характеризует минимально допустимую информативность объекта, а $N^{\text{obj}}_{\min}$ — минимальное число независимых структурных экспериментов, в которых наблюдался объект.

7 Приложение формализма к тестовым выборкам экспериментальных данных[1]

Для проведения экспериментов по тестированию условия разрешимости (3) и вычисления характеристических функций множеств наиболее информативных мотивов (5) прежде всего необходимо формирование непротиворечивого множества объектов $\mathbf{Q}^{\text{нп}}$ в соответствии с (7). Для обоснованного выбора значений параметров $D^{\text{obj}}_{\min}$ и $N^{\text{obj}}_{\min}$ необходимо ввести некоторое формальное описание «качества» получаемого $\mathbf{Q}^{\text{нп}}$.

Критерии (7) позволяют выбрать в исходном $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ некоторое подмножество кластер-объектов $\mathbf{Q}^0$. Формирование непротиворечивого $\mathbf{Q}^{\text{нп}}$ из кластер-объектов $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ — это, по сути дела, выбор множества представителей (т. е. каждому кластер-объекту в $\mathbf{Q}^0 \subseteq \mathbf{Q}(\mathbf{Pr},\mathbf{M})$ соответствует единственный объект в $\mathbf{Q}^{\text{нп}}$). На интуитивном уровне понятно: качество $\mathbf{Q}^{\text{нп}}$ тем выше, чем больше кластер-объектов из $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$ представлено в $\mathbf{Q}^{\text{нп}}$ и чем меньше число противоречий между экспериментами (т. е. выше информативность каждого объекта в $\mathbf{Q}^{\text{нп}}$).

Формальными показателями «качества» множества объектов $\mathbf{Q}^{\text{нп}}$, получаемого отображением $h : \mathbf{Q}^0 \subseteq \mathbf{Q}(\mathbf{Pr},\mathbf{M}) \to \mathbf{Q}^{\text{нп}}$, служат, во-первых, соотношение между числом отобранных кластер-объектов ($|\mathbf{Q}^{\text{нп}}|$) и общим числом кластер-объектов N_{qc} в исходном множестве $\mathbf{Q}(\mathbf{Pr},\mathbf{M})$, $|\mathbf{Q}^{\text{нп}}|/N_{qc} \to \max$; во-вторых, соотношение между числом объектов в $\mathbf{Q}^0$, включаемых в $\mathbf{Q}^{\text{нп}}(N_{\text{вкл}})$, и общим числом объектов в $\mathbf{Q}^0$, $N_{\text{вкл}}/|\mathbf{Q}^0| \to \max$.

Критерий разрешимости задачи распознавания (1) и (2) и следующий из него критерий непротиворечивости (6) требуют однозначного отнесения любого объекта (кластер-объекта) к определенной литере алфавита $\mathbf{B}$. Естественно определить $\mathbf{B}$-литеру α-го кластер-объекта как класс, соответствующий большинству объектов. Тогда $N_{\text{вкл}}$ определяется суммированием по α величин $N^{\alpha}_{\max} = \max\limits_{l=1,\ldots,|\mathbf{B}|} N^{\alpha}_{l}$, а $|\mathbf{Q}^0|$ — суммированием N^{a}_{Σ}. Задача нахождения непротиворечивого множества объектов $\mathbf{Q}^{\text{нп}}$ может быть сформулирована как

$$\arg\max_{D^{\text{obj}}_{\min},N^{\text{obj}}_{\min}} \left(\frac{|\mathbf{Q}^{\text{нп}}|}{N_{qc}} + \frac{N_{\text{вкл}}}{|\mathbf{Q}^0|} \right). \qquad (8)$$

В проведенных экспериментах в качестве множества прецедентов были использованы все 165 000 прецедентов, представленных в базе данных PDB [4] на октябрь 2010 г. При этом число кластер-объектов превысило 5 млн ($N_{qc} = 5{,}42 \cdot 10^6$). *Формирование непротиворечивых множеств объектов* осуществлялось решением задачи (8). При оптимальных значениях параметров $D^{\text{obj}}_{\min} = 2{,}5$ и $N^{\text{obj}}_{\min} = 4$ число отобранных объектов составило $|\mathbf{Q}^{\text{нп}}| = 2{,}01 \cdot 10^6$.

На основе множества $\mathbf{Q}^{\text{нп}}$ формировались тестовые множества $\mathbf{Q}$ для вычисления характеристических функций $T(\alpha)$. Были исследованы выборки размером 10 000, 20 000, 30 000, 50 000, 100 000 и 200 000 объектов, сформированные путем случайного отбора объектов без возвращения. В общей сложности было проанализировано 60 различных $\mathbf{Pr}$, т. е. по 10 выборок для каждого из шести приведенных выше значений $|\mathbf{Q}|$. Исследование бóльших выборок объектов в настоящее время не представляется возможным вследствие значительных вычислительных трудностей.

Каждая из *использованных систем масок* имела фиксированную размерность всех масок. Максимальная протяженность масок во всех системах составила 8 позиций. Изученные системы масок были получены на основе системы масок с размерностью всех масок, равной $m = 2$ (система M^1_8) и $m = 3$ (M^3_8), в которых нулевая позиция каждой маски соответствовала позиции $\lfloor n/2 \rfloor + 1$ в верхнем слове каждого объекта. Очевидно, что $|M^2_8| = C^2_8$ и $|M^3_8| = C^3_8$. Была произведена частичная редукция систем масок путем удаления сдвиг-эквивалентных масок, и для вычисления $T(\alpha)$ использовались системы масок $M'^2_8, |M'^2_8| = 11$ и $M'^3_8, |M'^3_8| = 25$. Ниже представлены результаты для системы масок M'^3_8.

Была исследована целесообразность использования трех *эвристических оценок информативности мотивов* $D_1(\alpha)$, $D_2(\alpha)$ и $D_{D0}(\alpha)$. Предварительные эксперименты показали, что оценка $D_2(\alpha) = N_{\Sigma} D_1(\alpha)$ приводит к тупиковым множествам мотивов наименьшей размерности, так что в дальнейших экспериментах использовалась именно $D_2(\alpha)$.

Вычисления $T(\alpha)$ показали, что предложенный формализм позволяет значительно сократить множество мотивов $\mathbf{K}(\mathbf{Pr},\mathbf{M})$ без потери разрешимости. Например, в $\mathbf{MO}(\mathbf{Pr}(|\mathbf{Q}| = 200\,000), M'^3_8)$ содержалось 201 000 мотивов. Число отобранных мотивов (т. е. $|\mathbf{MO}^0| = |\{T(\alpha) = 1\}|$) составило $\sim 25\,000$. Логарифмический характер зависимости числа отобранных мотивов от $|\mathbf{Q}|$ (рис. 2) позволяет предположить, что высокая эффективность редукции множества мотивов сохранится и при значительно бóльших $\mathbf{Q}$. Отметим, что оценки $|\mathbf{K}^0|$, полученные на разных выборках одного размера, отличались не более чем на 5%.

[1] Экспериментальная часть работы полностью выполнена И. Ю. Торшиным.

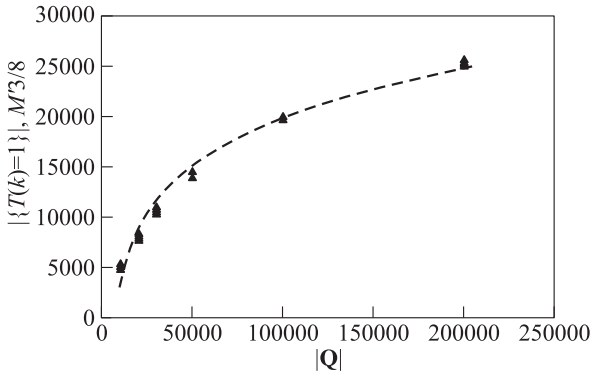


Рис. 2 Зависимость числа отобранных мотивов от размера выборки объектов: $y = 6989{,}9 \ln x - 60\,548$; $R^2 = 0{,}9897$. Было исследовано по 10 независимых выборок объектов каждого размера

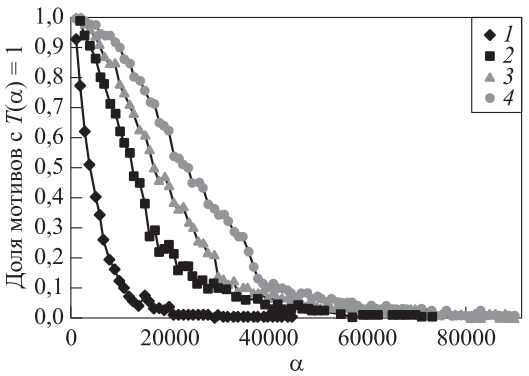


Рис. 3 Распределение мотивов тупиковых $\mathbf{K}^0$ среди мотивов $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ (α — порядковый номер мотива) для выборок объектов различного размера: *1* — 10 000; *2* — 50 000; *3* — 100 000; *4* — 200 000

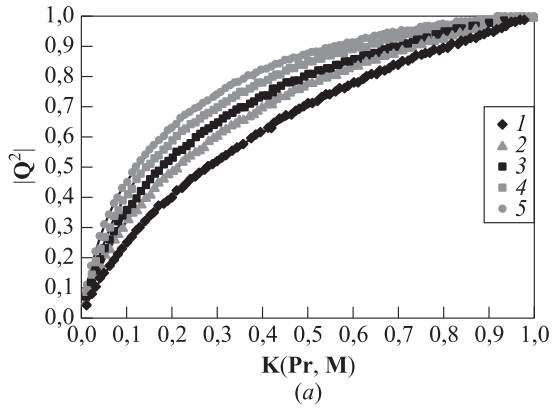
Исследуем структуру получаемых $\mathbf{K}^0$ более подробно. В $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ мотивы упорядочены по убыванию информативности. Как и следовало ожидать, мотивы $\mathbf{K}^0$ (т. е. мотивы с $T(\alpha) = 1$) встречаются наиболее часто среди мотивов с высокой информативностью (наименьшими α, рис. 3). Во множествах с 200 000 объектов практически все 8000 наиболее информативных мотивов входят в $\mathbf{K}^0$.

Представляет интерес рассмотрение зависимости числа пар объектов, на которых достигнута разрешимость (максимально $|\mathbf{Q}^2|$), от числа мотивов с максимальной информативностью. Так как очевидно, что число мотивов в $\mathbf{K}^0$ зависит от числа объектов (см. рис. 2), то для сравнения результатов следует использовать процентные соотношения (рис. 4).

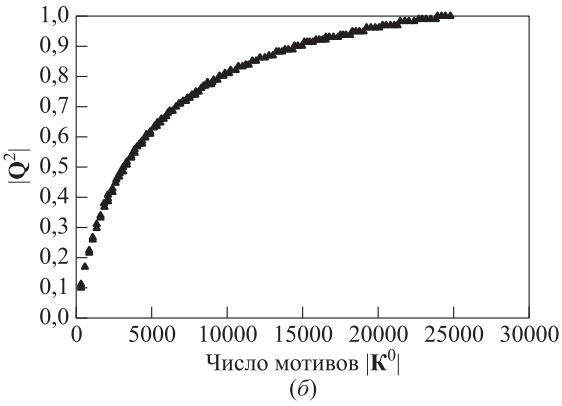
Результаты, представленные на рис. 4, указывают на существование некоторого «ядра» в тупиковых множествах мотивов. Мотивы, входящие в такое «ядро», обеспечивают разрешимость на большинстве пар объектов. Например, в выборках по 200 000 объектов 50% наиболее информативных мотивов в тупиковом $\mathbf{K}^0$ обеспечивают разрешимость почти на 90% пар объектов («90% ядро»).

Логарифмический характер зависимости $|\mathbf{Q}^2|$–$|\mathbf{K}^0|$ указывает на то, что полная разрешимость достигается добавлением к «ядру» некоторых низкоинформативных мотивов, каждый из которых обеспечивает разрешимость всего лишь на нескольких парах объектов. Действительно, число пар объектов на мотив резко падает по мере увеличения порядкового номера мотива в $\mathbf{K}^0$ (рис. 5). Согласно данным рис. 5, *б*, мотивы 90% «ядра» ($|\mathbf{Q}| = 200\,000$) обеспечивают разрешимость более чем на 5 парах объектов.

Можно предположить, что мотивы, входящие в «ядра», будут гораздо чаще встречаться в произвольных обучающих выборках. Исследуем *заполненность* (англ. *occupancy*, термин заимствован из физики твердого тела) мотивов во множествах $\mathbf{K}^0$,



(*а*)



(*б*)

Рис. 4 Число распознаваемых пар объектов для выборок разных размеров: (*а*) выборки разного размера (*1* — из 10 000 объектов; *2* — из 30 000; *3* — из 50 000; *4* — из 100 000; *5* — из 200 000 объектов); (*б*) выборки из 200 000 объектов

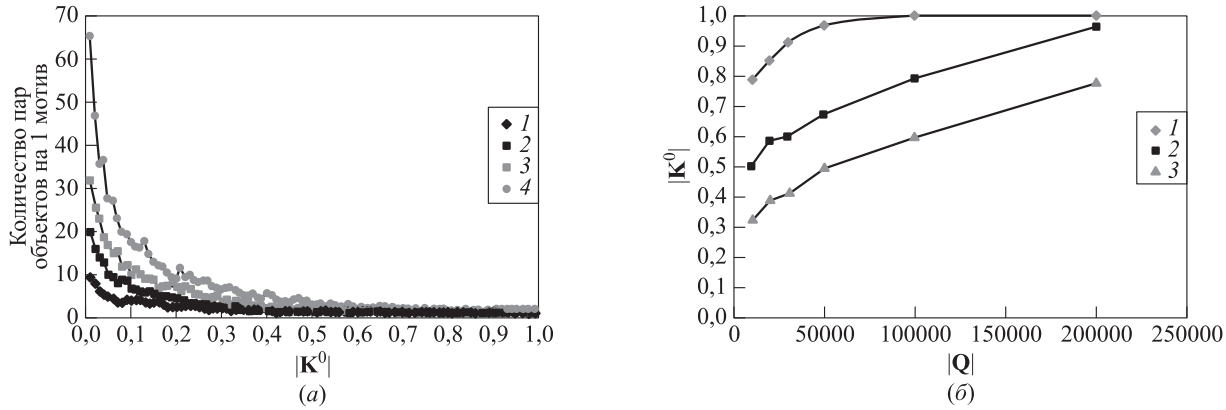


Рис. 5 Зависимость среднего числа «разрешаемых» по условию (3) пар объектов на один мотив от информативности мотива (его порядкового номера в $\mathbf{K}^0$): (*а*) количество пар объектов на 1 мотив (*1* — 10 000; *2* — 50 000; *3* — 100 000; *4* — 200 000); (*б*) доля мотивов в $\mathbf{K}^0$, имеющих заданное число пар объектов на мотив (*1* — >1 п/м; *2* — >5; *3* — >10 п/м)

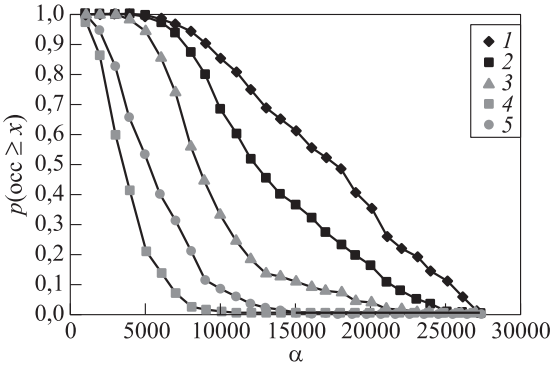


Рис. 6 Информативность и заполненность мотивов. Мотивы с более низкими порядковыми номерами мотива в $\mathbf{K}(\mathbf{Pr}, \mathbf{M})$ α чаще входят в $\mathbf{K}^0(\mathbf{Q})$ при произвольном $\mathbf{Q}$: *1* — $p(\text{occ} \leq 0{,}8)$, 200 000; *2* — $p(\text{occ} \leq 0{,}9)$, 200 000; *3* — $p(\text{occ} = 1)$, 200 000; *4* — $p(\text{occ} = 1)$, 100 000; *5* — $p(\text{occ} = 1)$, 50 000

полученных для разных $\mathbf{Q}$ одного размера (если мотив имел $T(\alpha) = 1$ в 8 множествах $\mathbf{Q}$ из 10, то его заполненность 0,8 и т. д.). Сравнение множеств мотивов, полученных на различных $\mathbf{Q}$ (например, $10\mathbf{Q}$, $|\mathbf{Q}| = 200\,000$), показало: чем выше информативность мотива (т. е. чем ниже k), тем более вероятно, что мотив входит в тупиковое $\mathbf{K}^0$, построенное на произвольном $\mathbf{Q}$ (рис. 6). Например, во множествах объектов с $|\mathbf{Q}| = 200\,000$ первые 5000 наиболее информативных мотивов встречались в каждом полученном $\mathbf{K}^0$. Мотивы «95% ядра» имеют занятость не менее 0,8, а мотивы с заполненностью 1,0 (т. е. встречающиеся в $\mathbf{K}^0$, построенном на произвольном $\mathbf{Q}$) обеспечивают разрешимость 90% пар объектов (так как образуют «90% ядро»).

8 Выводы

В настоящей работе проведено развитие формализма для исследования разрешимости задачи распознавания вторичной структуры белка на множествах аминокислотных мотивов. Показано, что введение порядка на множестве мотивов посредством эвристических оценок информативности позволяет проводить эффективное сокращение множества мотивов без потери разрешимости задачи. Разработанный формализм позволил провести эксперименты по нахождению тупиковых множеств наиболее информативных мотивов. Установлены перспективные направления дальнейших исследований: создание теоретико-множественного обоснования оценок информативности D, введение комбинаторных оценок значений D, исследование ядерной эквивалентности функций-предикторов, построенных на мотивах. Разработанный формализм также позволяет провести систематическое исследование для рационального выбора значений параметров используемых систем масок. Нахождение тупиковых множеств наиболее информативных мотивов существенно важно для следующего этапа представленного исследования — синтеза алгоритмов в рамках алгебраического подхода к распознаванию.

Литература

1. *Torshin I. Y.* Bioinformatics in the post-genomic era: The role of biophysics. — N.Y.: Nova Biomedical Books, 2006.
2. *Рудаков К. В., Торшин И. Ю.* О разрешимости формальной задачи распознавания вторичной структуры белка // ММРО-14, Суздаль, 2009. С. 596–597.

3. *Рудаков К. В., Торшин И. Ю.* Вопросы разрешимости задачи распознавания вторичной структуры белка // Информатика и её применения, 2010. Т. 4. Вып. 2. С. 25–35.
4. *Berman H. M., Henrick K., Nakamura H.* Announcing the worldwide Protein Data Bank // Nature Structural Biology, 2003. Vol. 10. No. 12. P. 980–982.
5. *Рудаков К. В.* Универсальные и локальные ограничения в проблеме коррекции эвристических алгоритмов // Кибернетика, 1987. № 2. С. 30–35.
6. *Рудаков К. В.* О проблемах классификации значений признаков в задачах распознавания // Интеллектуализация обработки информации (ИОИ-8): VIII Междунар. конф. (Пафос, Кипр): Сб. докл. — М.: МАКС Пресс, 2010. С. 81–82.
7. *Воронцов К. В.* Комбинаторная теория надежности обучения по прецедентам. Дис. . . . докт. физ.-мат. наук. — М.: ВЦ РАН, 2010. 271 с.
8. *Furnkranz J., Flach P. A.* Roc'n' rule learning — towards a better understanding of covering algorithms // Machine Learning, 2005. Vol. 58. No. 1. P. 39–77.
9. *Журавлев Ю. И., Рудаков К. В.* Об алгебраической коррекции процедур обработки (преобразования) информации // Проблемы прикладной математики и информатики. — М.: Наука, 1987. С. 187–198.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 91–98

СИСТЕМА ИДЕНТИФИКАЦИИ ДИКТОРОВ ПО ГОЛОСУ ДЛЯ КОНКУРСА *NIST SRE 2010*

И. Н. Белых[1], А. И. Капустин[2], А. В. Козлов[3], А. И. Лоханова[4], Ю. Н. Матвеев[5], Т. С. Пеховский[6], К. К. Симончик[7], А. К. Шулипа[8]

Аннотация: Приведено описание системы идентификации дикторов по голосу, разработанной для конкурса по оцениванию систем распознавания дикторов *NIST SRE 2010*.

Ключевые слова: биометрическая идентификация; идентификация диктора; распознавание по голосу; GMM; SVM; NIST

1 Введение

Системы идентификации (распознавания, верификации) дикторов по голосу относятся к классу биометрических систем, достоинством которых является то, что они чаще всего не требуют дополнительного оборудования для регистрации голоса и могут быть реализованы с использованием телефонных сетей или устройств ввода–вывода разных типов (микрофонов).

Область использования такого рода приложений обширна:

- автоматическая идентификация подозреваемого по телефонному каналу;
- автоматическая верификация клиентов при удаленном доступе по телефонному каналу;
- обработка речевых баз данных;
- криминалистические исследования.

В данной работе представлено описание текстонезависимой системы автоматической идентификации дикторов по голосу, разработанной ООО «Центр речевых технологий» для участия в международном конкурсе по оцениванию систем распознавания дикторов *NIST SRE 2010*.

В профессиональной среде *NIST SRE* (speaker recognition evaluation) называют неофициальным чемпионатом мира по голосовой идентификации. Начиная с 1996 г. этот конкурс ежегодно проводится американским Национальным институтом стандартов и технологий (National Institute of Standards and Technology, NIST). Его основная цель — оценить уровень существующих технологий и определить перспективные направления развития индустрии. Регулярно в конкурсе принимают участие ведущие компании, университеты и лаборатории со всего мира. В 2010 г. в конкурсе участвовало 46 научных команд.

Первой особенностью оценивания *NIST SRE 2010* являлось использование баз речевых данных, собранных по различным каналам связи и в акустике помещений, а потому характеризующихся широким диапазоном значений отношения сигнал–шум и уровня реверберации (рис. 1). Точка на скаттерограмме обозначает присутствие в речевом корпусе фонограммы с определенным соотношением сигнал/шум (ОСШ) и временем реверберации.

Второй особенностью *NIST SRE 2010* в сравнении с предыдущими конкурсами стало то, что организаторы задали новую функцию минимизации ошибки идентификации, суть которой состояла в крайне высокой стоимости ошибки ложного пропуска нецелевого диктора:

$$\mathrm{DCF} = \mathrm{FR} + 999\mathrm{FA}\,,$$

где FR (false rejection error rate) — вероятность ошибки ложного отклонения; FA (false acceptance error rate) — вероятность ошибки ложного пропуска.

Введение новых значений весов параметров потребовало значительных объемов данных для калибровки порога принятия решения системы идентификации в связи с тем, что число попыток

[1]«Центр речевых технологий», Санкт-Петербург, belykh@speechpro.com
[2]«Центр речевых технологий», Санкт-Петербург, kapustin@speechpro.com
[3]«Центр речевых технологий», Санкт-Петербург, kozlov-a@speechpro.com
[4]«Центр речевых технологий», Санкт-Петербург, lohanova@speechpro.com
[5]«Центр речевых технологий», Санкт-Петербург, matveev@speechpro.com
[6]«Центр речевых технологий», Санкт-Петербург, tim@speechpro.com
[7]«Центр речевых технологий», Санкт-Петербург, simonchik@speechpro.com
[8]«Центр речевых технологий», Санкт-Петербург, shulipa@speechpro.com

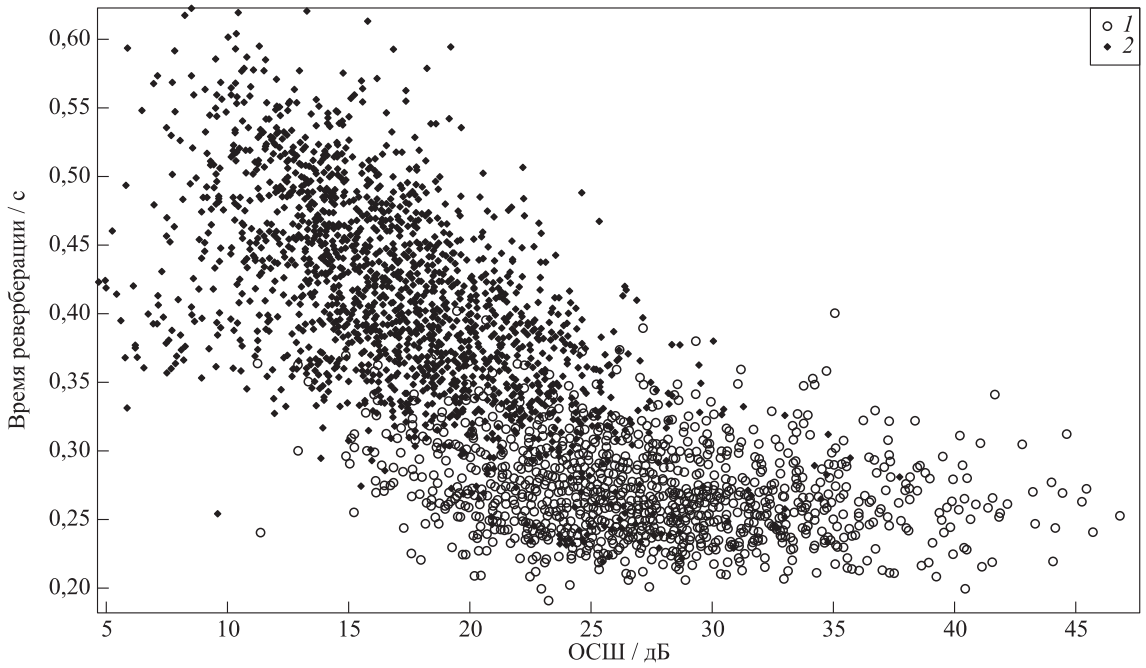


Рис. 1 Скаттерограмма корпусов речевых данных NIST различных годов: сравнительная характеристика параметров речевых данных телефонного корпуса SRE 2010 г. (*1*) и в акустике помещений SRE 2005–2010 гг. (*2*)

идентификации нецелевого диктора должно быть достаточно большим для статистически устойчивой оценки DCF.

Точность калибровки особенно важна при применении голосовой идентификации в реальных условиях и зачастую играет критическую роль, так как позволяет максимально точно адаптироваться под прикладные задачи.

В данной работе описывается система, которая показала один из лучших результатов по качеству идентификации, в том числе заняла первое место по уровню калибровки среди коммерческих систем.

2 Методы идентификации диктора

Принцип работы системы идентификации диктора основан на выделении речи из фонограмм и последующем попарном сравнении биометрических признаков (содержащихся в голосе индивидуальных, идентификационно значимых признаков личности).

Выделение и сравнение биометрических признаков производится с использованием различных языко- и текстонезависимых методов идентификации дикторов по голосу. Система распознавания диктора называется текстонезависимой, если она не содержит информации о том, что именно диктор будет произносить (система обучается и тестируется на произвольных речевых данных).

На данный момент наиболее распространенным подходом к решению задач текстонезависимой идентификации является подход на основе использования моделей гауссовых смесей (*Gaussian mixture models*, GMM) [1]. В качестве речевых признаков в подавляющем большинстве систем идентификации используются мэл-частотные кепстральные коэффициенты (mel-frequency cepstral coefficients, MFCC) [2].

В текстонезависимых приложениях идентификации диктора наибольшую надежность показывают GMM-системы, основанные на использовании совместного факторного анализа (*joint factor analysis*, JFA), предложенного и исследованного в работах [3–5].

В этих системах решение идентификации диктора основано на использовании отношения правдоподобия. Модели дикторов вычисляются или с помощью классической MAP-адаптации [6] GMM-модели от UBM-модели (*universal background model*), или с использованием более мощных методов создания каналонезависимых моделей диктора [4, 5].

Очень перспективным для идентификации дикторов является метод опорных векторов (*support vector machine*, SVM) [7]. Метод SVM дискриминантный, в отличие от порождающего метода GMM.

Современное развитие данного направления показало, что вариант гибридной системы GMM–SVM, где SVM действует не в пространстве акустических векторов, а в модельном пространстве супервекторов средних GMM, оказывается самым эффективным для задачи идентификации диктора. О супервекторе средних GMM можно говорить как об отображении совокупности векторов MFCC произнесения диктора $O = \{\vec{o}_1, \ldots, \vec{o}_t, \ldots, \vec{o}_T\}$ в высокоразмерный вектор $\vec{\mu}$.

В данной работе для GMM-подсистемы представляется редуцированная версия JFA (без собственных голосов [4]), а именно: ML (*maximum likelihood*) модификация [8] версии метода Фогта [5] (для краткости ML_Vogt).

Как и в работе [8], будем исходить из следующего представления супервектора средних для h-й сессии s-го диктора:

$$\vec{\mu}(s) = \vec{\mu}_0 + \hat{D} \cdot \vec{z}(s)\,,$$
$$\vec{\mu}(s,h) = \vec{\mu}(s) + \hat{U} \cdot \vec{x}(s,h)\,.$$

Здесь $\vec{\mu}_0$ — супервектор средних дикторонезависимой UBM-модели; $\vec{z}(s)$ — вектор диктора (размерностью MF); $\vec{\mu}(s)$ — супервектор сессионезависимой модели диктора s; $\vec{x}(s,h)$ — низкоразмерный вектор подпространства каналов-сессий (размерностью R_x), который находится методом MLED (*maximum likelihood eigen-decomposition*) [9].

Матрица собственных каналов $\hat{U}$ определяется методом MLES (*maximum likelihood eigen-space*) [9]. Диагональная матрица $\hat{D}$ (размерностью $MF \times MF$) находится из уравнения:

$$\hat{I} = \tau \hat{D}^T \Sigma^{-1} \hat{D}\,,$$

где τ — фактор релевантности в классической MAP-адаптации; M — размерность GMM; F — размерность MFCC-вектора.

Метод [8], как и метод Фогта [5], благодаря использованию информации в $\hat{U}$-матрице собственных каналов корпуса позволяет получать GMM-супервектор диктора, свободный от эффектов канала корпуса (и даже на одном эталоне диктора).

Для SVM-подсистемы в данной работе используется SVM-классификатор, работающий в пространстве супервектора средних GMM двух классов — *Target* (целевой диктор) и *Imposter* (нецелевой диктор). В терминах ядра SVM-классификатор данных двух классов может быть построен согласно

$$\Lambda = f_a(\vec{\mu}(b)) = \sum_{q=1}^{Q} \lambda_q y_q K(\vec{\mu}(b), \vec{\mu}_q(a)) - w_0\,, \quad (1)$$

где $\vec{\mu}_q$ — GMM-супервекторы (это Q опорных векторов, полученных при обучении SVM-модели диктора *a*), y_q — целевые значения двух классов: $\{+1\}$ для класса *Target* и $\{-1\}$ для класса *Imposter* к заданному диктору.

Тогда $f(\vec{\mu}(b))$ дает расстояние (с учетом знака y_q) от разделяющей гиперплоскости диктора *a*, определяемой набором $\{\vec{\mu}_q, \lambda_q, w_0\}$, до GMM-супервектора спорной записи $\vec{\mu}(b)$. Здесь в (1) используется линейное ядро, предложенное Кампбеллом [10]:

$$K(\vec{\mu}(b)\vec{\mu}(a)) = \\ = \sum_{m=1}^{M} \left(\sqrt{a_m}\,\hat{\Sigma}_m^{-1} \cdot \vec{\mu}_m(b)\right) \left(\sqrt{a_m}\,\hat{\Sigma}_m^{-1} \vec{\mu}_m(a)\right)^{\mathrm{T}}\,.$$

Это линейное ядро основано на верхней границе дивергенции Кульбака–Лейбнера между двумя GMM. Выбор простого линейного ядра оправдан высокой размерностью GMM-супервектора (порядка 40 000).

Таким образом, работа представленной системы начинается с генерации GMM-системой каналонезависимых супервекторов средних $\vec{\mu}$ для каждой спорной и эталонной фонограммы. Далее на основе супервектора эталонной фонограммы (класс целевого диктора) и аналогичных супервекторов класса базы независимых импостеров (их число порядка 1000–2000) для каждого целевого диктора строится SVM-модель — своя разделяющая SVM-гиперплоскость. И, наконец, в качестве результирующей оценки спорного произнесения рассчитывается SVM-дистанция (2) между супервектором спорной фонограммы и SVM-гиперплоскостью целевого диктора.

На основании таких оценок по всем парам сравнения на тестовой базе строится итоговый DET-график, по которому можно сделать оценку мощности исследуемой системы идентификации.

Значения равновероятной ошибки (*equal error rate*, EER) принятия чужого и отбрасывания своего диктора для метода на основе MFCC–GMM зависят от длительности сравниваемых речевых фрагментов и могут достигать величины 4%–5%.

В настоящее время рассматриваемый метод является преобладающим в системах текстонезависимой идентификации диктора. Стоит отметить, что существуют и альтернативные методы, такие как спектрально-формантный (СФ) [11] и метод идентификации на основе статистик основного тона (СОТ) [12]. Однако только в случае коротких и сильно зашумленных фонограмм они обеспечивают точность, сопоставимую с методом MFCC–GMM–SVM. Сравнительные характеристики перечисленных методов приведены в табл. 1 (число знаков «+» отражает степень зависимости метода от параметров сигнала).

Таблица 1 Сравнительные характеристики методов идентификации дикторов

Метод	Параметры сигнала		
	Длительность	Качество	Физическое и эмоциональное состояние диктора
СФ	+++	++	+
СОТ	++	+	++++
MFCC–GMM–SVM	++	+	++

3 Описание системы

3.1 Структура системы

В системе автоматической идентификации личности по голосовым признакам в естественной речи осуществляется сравнение одной или нескольких записей (моделей) голоса неизвестного диктора с одной или несколькими записями (моделями) голоса известного диктора. В результате такого сравнения определяется, насколько похож голос неизвестного диктора на голос известного и, следовательно, принадлежат ли записи речи одному человеку или разным людям.

Если тестируемая фонограмма речи диктора может не принадлежать ни одному из кандидатов, то в систему дополнительно вводится модель «самозванца» (в настоящей системе это 1000–2000 супервекторов SVM-импостеров) и она называется системой идентификации дикторов на открытом множестве. Схема такой системы представлена на рис. 2.

Как уже было сказано ранее, для каждого целевого диктора строится SVM-модель — своя разделяющая SVM-гиперплоскость. Для каждого диктора-кандидата проводится сравнение речевого сигнала с SVM-моделью голоса данного диктора и получается оценка сравнения в виде SVM-дистанции между супервектором спорной фонограммы и SVM-гиперплоскостью целевого диктора. Если полученная оценка оказывается выше порога принятия решения, то спорная фонограмма кандидата приписывается целевому диктору. Если иначе, то заявленный кандидат признается самозванцем.

3.2 Описание системы

Представленная на конкурс *NIST SRE 2010* система состояла из 6 различных гендеро- и каналозависимых подсистем. Подсистемы адаптировались под различные каналы получения фонограмм:

- телефонные;
- микрофонные;
- смешанные (телефон–микрофон).

Кроме того, в рамках одного канала производилось дополнительное деление подсистем на две гендерозависимые (для женских и мужских голосов) подсистемы.

В качестве обучающих данных были взяты речевые базы NIST SRE прошлых лет (2004, 2006 и 2008 гг.) общим объемом более 20 тыс. фонограмм.

Для повышения надежности системы в качестве дополнительных речевых признаков были использованы линейно-частотные кепстральные коэффициенты (*linear-frequency cepstral coefficients*, LFCC) [2], что обеспечило повышение качества идентификации в микрофонном канале.

Результирующее решение (*decision*) по обоим признакам (MFCC и LFCC) получалось путем вычисления «обобщенного решения» (*fusion*), получаемого методом взвешенного голосования, когда

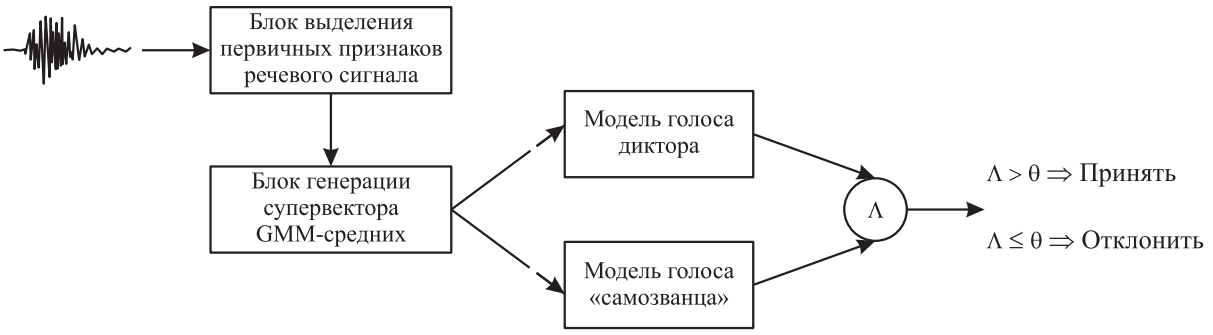


Рис. 2 Структурная схема GMM–SVM–MFCC-системы идентификации дикторов по голосу на открытом множестве

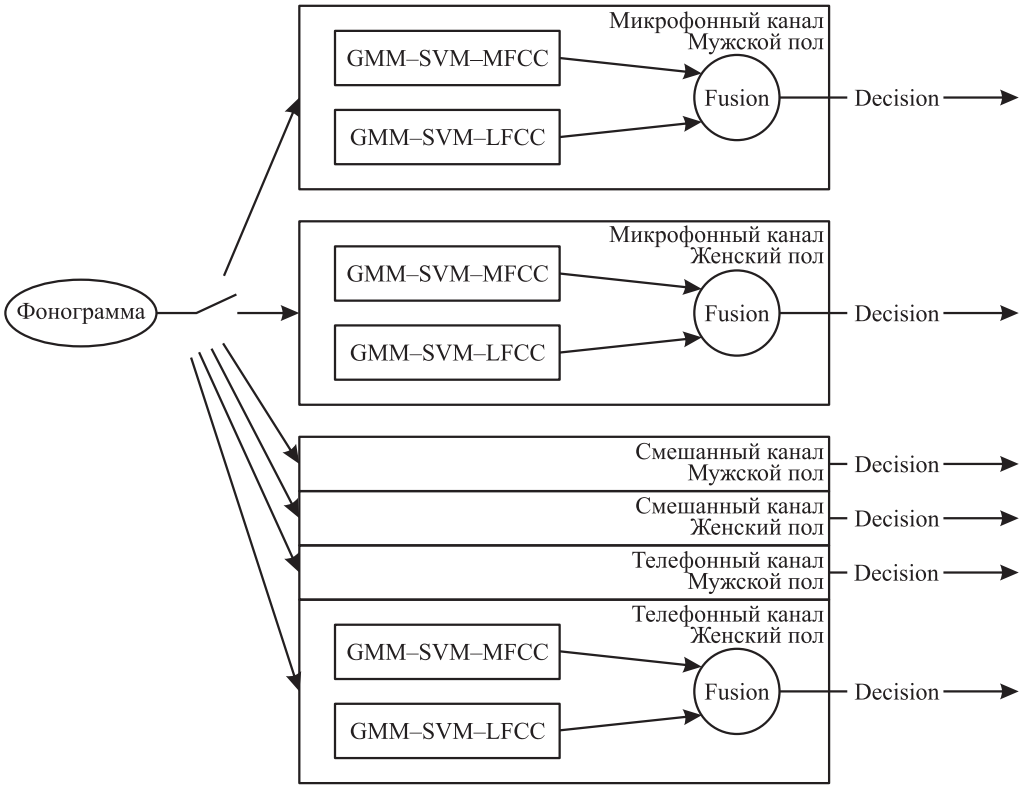


Рис. 3 Структура системы идентификации диктора ООО «Центр речевых технологий»

результату работы каждой подсистемы присваивается некоторый вес (рис. 3). Для определения этих весов на этапе обучения системы использовался инструментарий собственной разработки. Обучение и точная калибровка системы производилась на речевой базе NIST SRE 2005.

GMM-супервекторы дикторов обучались методом максимального правдоподобия ML_Vogt [8] с использованием компенсации канальных искажений. Размерность пространства собственных каналов для разных подсистем варьировалось от 50 до 80. В качестве классификатора использовался SVM с классической zt-нормализацией [5], которая представляла собой нормирование выходной дистанции SVM по случайным произнесениям дикторов из речевой базы объемом 1000–2000 фонограмм.

4 Смешивание подсистем и систем

4.1 Основные характеристики систем и подсистем

На конкурс *NIST SRE 2010* были предоставлены три системы (SVID-1, SVID-2 и SVID-3), которые отличались корпусами речевых данных, используемых для обучения и настройки систем.

4.1.1 Primary system (SVID-1)

Базовая (*primary*) система являлась комбинацией двух подсистем, каждая из которых строилась на отдельных наборах речевых признаков:

(1) первая подсистема — на базе 39-мерных векторов признаков, составленных из 13 MFCC-коэффициентов, дополненных их первыми и вторыми производными. Для каждого из векторов применялась процедура вычитания кепстрального среднего (CMS);

(2) вторая подсистема — на базе 39-мерных векторов признаков, составленных из 13 LFCC-коэффициентов, дополненных их первыми и вторыми производными. Для каждого из векторов применялась процедура вычитания кепстрального среднего (CMS).

Каждая из подсистем, в свою очередь, имела 6 гендеро- и каналозависимых UBM. При обучении UBM использовались 1024-компонентные гауссовы смеси. База обучения UBM состояла из речевых корпусов Switchboard II Phases 2 & 3, Switchboard Cellular Parts 1 & 2, NIST SRE 2004, 2006 и 2008, из

Таблица 2 Характеристики базы обучения

Каналы	Общее количество			
	Дикторов		Фонограмм	
	Мужской пол	Женский пол	Мужской пол	Женский пол
Телефон–телефон	788	6546	1042	8589
Микрофон–микрофон	158	1516	203	1955
Телефон–микрофон	280 (153 тел + 158 мик)	2290	371 (201 тел + 203 мик)	3050

Таблица 3 Характеристики базы импостеров

Каналы	Общее количество дикторов	
	Мужской пол	Женский пол
Телефон	1070	1227
Микрофон	1450	2236
Телефон–микрофон	1000	1000

которых отбирались фонограммы дикторов, имеющих по 5–10 сессий записи речи. Характеристики базы обучения представлены в табл. 2.

Импостеры отбирались из РБД NIST SRE 2006, 2008. Характеристики базы импостеров представлены в табл. 3.

Для сегментации дикторов использовалась информация из ASR (*automatic speech recognition*) транскрипции, предоставленной NIST.

Для получения обобщенного по всем подсистемам результирующего решения использовался инструментарий собственной разработки для смешивания по критерию минимизации функции стоимости DCF.

4.1.2 Secondary system (SVID-2)

Вторичная (*secondary*) система отличается использованием на этапе обучения UBM речевой базы NIST SRE 2005 вместо NIST SRE 2008.

4.1.3 Secondary system (SVID-3)

Еще одна вторичная (*secondary*) система была сформирована путем комбинирования первичной (SVID-1) и вторичной (SVID-2) систем.

4.2 Смешивание подсистем

Обобщенное решение по всем подсистемам было основано на методе взвешенного голосования:

$$d(x) = \sum_{i=1}^{S} \alpha_i d_i(x) + \Theta\,,$$

где $d_i(x)$ — выходное значение i-й подсистемы; α_i — весовой коэффициент для i-й подсистемы; Θ — пороговое значение; S — число подсистем.

Калибровка общего решения производилась на речевой базе NIST SRE 2005, которая не использовалась для обучения базовых GMM–SVM-подсистем.

В связи с тем, что оценка DCF при высокой стоимости ошибки ложного пропуска не является статистически устойчивой, оптимизация коэффициентов α_i производилась простейшим методом перебора. При этом в качестве функции минимизации ошибки использовалась непосредственно функция DCF.

5 Результаты

В табл. 4 и 5 приведены характеристики систем идентификации дикторов ООО «Центр речевых технологий» до конкурса и представленной на конкурс *NIST SRE 2010*.

Как следует из приведенных данных, было обеспечено:

- повышение точности идентификации (снижение EER) в 3–4 раза;
- улучшение робастности идентификации в различных условиях (шумы, реверберация);

Таблица 4 Характеристики систем идентификации дикторов на различных корпусах

Каналы	EER, %	
	Система до конкурса	Система после конкурса
Микрофон–микрофон (различные микрофоны)	15–18	6,0
Микрофон–телефон (различные телефонные каналы)		4,9
Телефон–телефон (различные телефонные трубки и каналы)		5,0

Таблица 5 Характеристики систем идентификации дикторов на смешанном корпусе микрофон–GSM-канал (EER, %)

Система до конкурса				Система после конкурса			
Длительность чистой речи целевого диктора, с	Длительность чистой речи диктора-кандидата, с			Длительность чистой речи целевого диктора, с	Длительность чистой речи диктора-кандидата, с		
	16	32	80		17	29	77
16	15,2	14,0	14,0	17	8,0	6,3	3,8
32		12,9	11,4	29		4,5	2,7
80			8,9	77			1,3

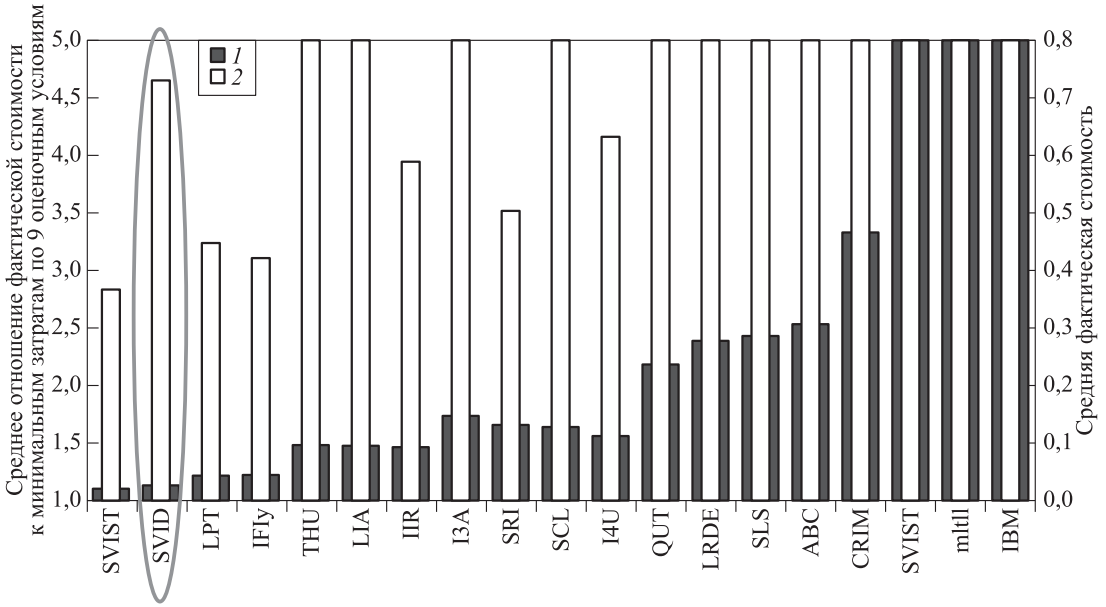


Рис. 4 Среднее отношение фактической стоимости системы (определяется участником) к минимальным затратам (определяется NIST) (*1*) и средняя фактическая стоимость системы (*2*)

– сохранение достигнутых параметров точности и робастности при кросс-канальных сравнениях.

Исходя из официально предоставленных NIST материалов (рис. 4), система идентификации дикторов ООО «Центр речевых технологий» заняла на конкурсе *NIST SRE 2010*:

– 2-е место по уровню калибровки (1-е место среди коммерческих компаний);
– 7-е место по фактической стоимости (*actual cost*) технологии — официальной метрике NIST (2-е место среди коммерческих компаний).

6 Заключение

В рамках подготовки к конкурсу *NIST SRE 2010* была произведена модернизация системы идентификации дикторов ООО «Центр речевых технологий», что обеспечило значительное повышение точности идентификации (снижение EER) — в 3–4 раза, повышение робастности идентификации в различных условиях (шумы, реверберация) и повышение скорости системы при построении голосовых моделей по фонограмме.

В настоящее время предложенный в рамках *NIST SRE 2010* подход к идентификации дикторов используется в системе ведения и автоматизации национального фоноучета Мексики.

Литература

1. *Bimbot F., Bonastre J.-F., Fredouille C., et al.* A tutorial on text-independent speaker verification // EURASIP J. Appl. Signal Processing, 2004. No. 4. P. 430–451.

2. *Reynolds D.* Experimental evaluation of features for robust speaker identification // IEEE Trans. Speech Audio Processing, 1994. Vol. 2. No. 4. P. 639–643.
3. *Burget L., Matejka P., Glembek O., et al.* Analysis of feature extraction and channel compensation in GMM speaker recognition system // IEEE Trans. Audio Speech Language Processing, 2007. Vol. 15. Iss. 7. P. 1979–1986.
4. *Kenny P., Ouellet P., Dehak N., et al.* A study of interspeaker variability in speaker verification // IEEE Trans. Audio Speech Language Processing, 2008. Vol. 16. Iss. 5. P. 980–988.
5. *Vogt R., Sridharan S.* Explicit modelling of session variability for speaker verification // Computer Speech Language, 2008. Vol. 22 (I). P. 17–38.
6. *Reynolds D. A., Quatieri T. F., Dunn R. B.* Speaker verification using adapted Gaussian mixture models // Digital Signal Processing, 2000. No. 10. P. 19–41.
7. *Vapnik V.* The nature of statistical learning theory. — Springer, 1995.
8. *Pekhovsky T., Oparin I.* Maximum likelihood estimations in the session-independent modelling of the speaker // Speech and Computer (SpeCom'09): XIII Conference (International) Proceedings. — St.-Petersburg, 2009. P. 267–270.
9. *Pekhovsky T., Oparin I.* Eigen channel method for text-independent Russian speaker verification // Speech and Computer (SpeCom'08): XII Conference (International) Proceedings. — Moscow, 2008. P. 385–390.
10. *Campbell W., Sturim D., Reynolds D.* Support vector machines using GMM supervectors for speaker verification // IEEE Signal Processing Lett., 2006. Vol. 13. No. 5. P. 308–311.
11. *Коваль С. Л., Лабутин П. В., Раев А. Н.* Метод распознавания диктора и устройство для его осуществления. Патент РФ 2230375 от 10.06.2004.
12. *Коваль С. Л., Лабутин П. В., Малая Е. В., Прощина Е. А.* Идентификация дикторов на основе сравнения статистик основного тона голоса // Информатизация и информационная безопасность правоохранительных органов: Мат-лы XV Междунар. научн. конф. — М.: Академия управления МВД России, 2006. С. 324–327.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 99–107

БЫСТРАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ ОТПЕЧАТКОВ ПАЛЬЦЕВ

В. Ю. Гудков[1], М. В. Боков[2]

Аннотация: Предложена последовательность методов распознавания частных признаков на изображении отпечатка пальца с жесткими ограничениями на время обработки. Частные признаки сохраняются в шаблоне изображения. По шаблонам выполняется идентификация изображений.

Ключевые слова: отпечаток пальца; обработка изображений; матрица потоков; матрица периодов; частные признаки

1 Введение

Исследования в области биометрии начались более ста лет назад с разработки методов сравнения отпечатков пальцев. С развитием вычислительной техники появилась возможность учета лиц в электронных системах. Функционирование таких электронных систем, подобно деятельности эксперта-криминалиста, опирается на модель дактилоскопического изображения (ДИ) в виде частных признаков и отношений между ними [1]. Среди электронных систем наиболее известны системы криминального и гражданского назначения. Если для первых систем основным показателем эффективности служит величина ошибки идентификации подозреваемого лица, то для вторых наравне с величиной ошибки аутентификации пользователя не менее важна и производительность системы [2]. Это оказывает сильное влияние на выбор методов обработки ДИ, например в системах контроля и управления доступом к объекту.

На рис. 1 на узоре левой петли выделены частные признаки в виде окончания и разветвления, распознавание которых простыми методами неэффективно [1, 3]. Поэтому быстрая обработка ДИ реализуется в виде последовательности специальных методов измерения, анализа и синтеза параметров изображения. Настройка и обучение этих методов минимизируют влияние дефектов изображения. Тем не менее жесткие ограничения по времени сужают класс ДИ, пригодных для быстрой обработки, преимущественно до изображений хорошего и среднего качества.

Рис. 1 Изображение отпечатка пальца

2 Постановка задачи

Обычно при распознавании частных признаков выполняются этапы предварительной обработки и повышения качества ДИ. Для этого изображение представляется в прямоугольной области G мощностью $|G| = x_0 y_0$ в виде $F = \{f(x,y) \in \in 0,\dots,2^b-1 | (x,y) \in X \times Y\}$, где b — глубина изображения; $X = 0,\dots,x_0-1$ и $Y = 0,\dots,y_0-1$.

Обработка изображения структурно представляется в виде пирамиды $\mathfrak{R}$ взаимосвязанных иерархий [3–5], в которых сегментация изображения производится для любого слоя произвольной иерархии. Например, l-й слой k-й иерархии $F_k^{(l)}$ разбивается на $x_h y_h$ квадратных сегментов $S_{hk}^{(l)}(x,y)$ с длиной стороны 2^{h-k} и вершинами $(x,y) \in X_h \times Y_h$, где $k < h$ и h — номер иерархии; $X_h = 0,\dots,x_h-1$ и $Y_h = 0,\dots,y_h-1$.

Доступ к каждой точке сегмента $S_{hk}(x,y)$ по [3] записывается в координатах $(u,v) \in \overline{X}_{hk} \times \overline{Y}_{hk}$:

[1]Челябинский государственный университет, diana@sonda.ru
[2]Южно-Уральский государственный университет, guardian@mail.ru

$$\left.\begin{aligned}\overline{X}_{hk} &= \{u + x2^{h-k} | x \in \\ &\in X_h \wedge u \in 0, \ldots, 2^{h-k} - 1\}\ ; \\ \overline{Y}_{hk} &= \{v + y2^{h-k} | y \in \\ &\in Y_h \wedge v \in 0, \ldots, 2^{h-k} - 1\}\ .\end{aligned}\right\} \quad (1)$$

Доступ к центральной точке сегмента $S_{hk}(x, y)$ записывается в координатах $(u, v) \in \hat{X}_{hk} \times \hat{Y}_{hk}$:

$$\left.\begin{aligned}\hat{X}_{hk} &= \{2^{h-k-1} + x2^{h-k} | x \in X_h\}\ ; \\ \hat{Y}_{hk} &= \{2^{h-k-1} + y2^{h-k} | y \in Y_h\}\ .\end{aligned}\right\} \quad (2)$$

Размер области h-й иерархии: $x_h = \lceil x_0/2^h \rceil$ и $y_h = \lceil y_0/2^h \rceil$, где $\lceil a \rceil$ — наименьшее целое число, превышающее вещественную величину a.

При иерархической сегментации сегменты слоя F_k отображаются на вершины сегментов слоя F_h пирамиды $\mathfrak{R}$, где $k < h$. Соответственно, вершины сегментов отображаются на сегменты, расположенные ближе к основанию пирамиды [3]. Размер сегмента заметно влияет на время и качество обработки. Далее положим $S_h(x, y) = S_{h0}(x, y)$ и вершины $S_h(x, y) \in F_h$.

Слои пирамиды можно представить множеством действительных чисел, а исходное изображение — множеством неотрицательных действительных чисел [4, 5]. Это снимает необходимость утомительного целочисленного представления сигнала и упрощает выражения, однако дискретизация изображения (слоев пирамиды $\mathfrak{R}$) в пространстве сохраняется.

Для компактной математической формализации методов классификационного анализа (КА) широко применяется аппарат апертур. Ключевую роль при этом играют прямолинейные щелевые $A_h(x, y, \alpha, w)$ и $A_h^-(x, y, \alpha, w)$ и круговая $A_h(x, y, w)$ апертуры, представляемые множеством точек слоя данных h-й иерархии и связанными с ними углами в виде элементов упорядоченных троек (u, v, β). Эти апертуры определяются по формулам:

$$\left.\begin{aligned}A_h(x, y, \alpha, w) &= \{(u, v, \beta) = \\ &= (x+]w\cos(\alpha)[, y+]w\sin(\alpha)[, \beta) | w \in Z_w\}\ ; \\ A_h^-(x, y, \alpha, w) &= \{(u, v, \beta) = \\ &= (x+]w\cos(\alpha)[, y+]w\sin(\alpha)[, \beta) | w \in Z_w^-\}\ ;\end{aligned}\right\} \quad (3)$$

$$A_h(x, y, w) = \bigcup_{\alpha \in Z^*} A_h(x, y, \alpha, w)\,, \quad (4)$$

где $(x, y) \in X_h \times Y_h$ — центр апертуры; $(u, v) \in \in X_h \times Y_h$ — точка апертуры; w — размер апертуры; $Z_w = 1, \ldots, w$; $Z_w^- = -w, \ldots, -1 \cup 1, \ldots, w$; α — угол направления апертуры; $]a[$ — ближайшая целая часть вещественного числа a. Угол, определяющий направление из центра апертуры (x, y) в точку (u, v), находится в виде:

$$\beta = \operatorname{arctg}\left(\frac{v-y}{u-x}\right) + \pi n \ \text{ при } n \in 0, \ldots, 1\,.$$

Для задачи распознавания частных признаков этапы предварительной обработки и повышения качества ДИ должны удовлетворять требованию на ограничение по времени. При этом алгоритм должен обеспечивать приемлемое качество распознавания частных признаков, которое проверяется на тестовой базе ДИ. Список частных признаков формируется в виде:

$$L_m = \{M_i = \{x_i, y_i, \alpha_i, t_i\} | i \in 1, \ldots, n\}\ , \quad (5)$$

где M_i — частный признак, индексированный номером i; $n = |L_m|$ — мощность списка частных признаков; (x_i, y_i) — координаты частного признака M_i; $\alpha_i \in 0, \ldots, 359$ — направление M_i как угол; $t_i \in \{0, 1\}$ — тип M_i (окончание или разветвление).

Компромиссным решением задачи, устраняющим противоречие качество–скорость, может служить реализация шести этапов обработки ДИ: (1) построение матрицы потоков; (2) сглаживание вдоль потоков; (3) построение матрицы плотности линий; (4) сегментация; (5) бинаризация; (6) скелетизация и распознавание частных признаков.

3 Быстрая обработка

Большинство алгоритмов КА отпечатков пальцев нацелено на распознавание частных признаков [1]. Решение задачи быстрой обработки продемонстрируем на примере исходного изображения $F_0^{(0)} = \left\{f_0^{(0)}(x, y)\right\}$ (см. рис. 1).

3.1 Интегральное изображение

Интегральное изображение I позволяет вычислить сумму элементов прямоугольной области изображения с постоянным числом операций независимо от размера области. Для вычисления интегрального изображения используется следующая формула:

$$\begin{aligned}I(x, y) &= f(x, y) + \\ &+ I(x-1, y) + I(x, y-1) - I(x-1, y-1)\,. \quad (6)\end{aligned}$$

Результат вычисления интегрального изображения показан на рис. 2. Однажды вычислив интегральное изображение, можно найти сумму элементов любой прямоугольной области изображения с

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

(*а*)

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

(*б*)

Рис. 2 Прямоугольная область изображения (*а*) и интегральное изображение (*б*)

левым верхним углом (x_1, y_1) и правым нижним углом (x_2, y_2) за постоянное время, используя следующее уравнение:

$$\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} f(x,y) = I(x_2,y_2) - I(x_1-1,y_2) - \\ - I(x_2,y_1-1) + I(x_1-1,y_1-1)\,. \quad (7)$$

3.2 Построение матрицы потоков

Это базовый этап обработки, от которого зависит точность распознавания частных признаков. Он состоит из двух последовательно выполняемых процедур обработки ДИ. Самый простой подход к вычислению матрицы потоков основан на вычислении градиента.

Измерение матрицы потоков. Суть метода заключается в разбиении изображения на сегменты с точками $(u,v) \in S_h(x,y)$ при $h = 3$ (8×8) по (1) и вычислении для вершины каждого сегмента величины угла $0^\circ \le \delta_h^{(0)}(x,y) < 180^\circ$ как элемента матрицы потоков $\Lambda_h^{(0)}$ по формуле:

$$\Lambda_h^{(0)} = \left\{\delta_h^{(0)}(x,y)\right\} = \\ = \left\{\frac{\pi}{2} + \frac{1}{2}\,\mathrm{arctg}\left(\frac{2J_{12}(x,y)}{J_{22}(x,y) - J_{11}(x,y)}\right)\right\},$$

где

$$J_{12}(x,y) = \sum_{(u,v)\in S_h(x,y)} \nabla_x\nabla_y\,;$$
$$J_{11}(x,y) = \sum_{(u,v)\in S_h(x,y)} \nabla_x\nabla_x\,;$$
$$J_{22}(x,y) = \sum_{(u,v)\in S_h(x,y)} \nabla_y\nabla_y\,.$$

Компоненты градиента в отсчетах $(u,v) \in \overline{X}_{hk} \times \times \overline{Y}_{hk}$ по (1) основания сегмента $S_h(x,y)$ вычисляются в виде $\nabla_x = \mathbf{H}_x * * f_0^{(0)}(u,v)$, $\nabla_y = \mathbf{H}_y * * f_0^{(0)}(u,v)$, где ядра двумерной свертки как оптимизированные по величине ошибки угла ориентации операторы Собела [5] находятся в виде:

$$\mathbf{H}_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}; \quad \mathbf{H}_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix}.$$

Таким образом, предварительно необходимо вычислить интегральные изображения $IG_{xy}(x,y)$, $IG_{xx}(x,y)$ и $IG_{yy}(x,y)$ по (6) для определения тензоров $J_{12}(x,y)$, $J_{11}(x,y)$ и $J_{22}(x,y)$ соответственно и расчета потока независимо от размера апертуры. Здесь

$$IG_{xy}(x,y) = G_{xy}(x,y) + IG_{xy}(x-1,y) + \\ + IG_{xy}(x,y-1) - IG_{xy}(x-1,y-1)\,,$$

где $G_{xy}(x,y) = \nabla_x(x,y)\nabla_y(x,y)$.

Интегральные изображения $IG_{xx}(x,y)$ и $IG_{yy}(x,y)$ вычисляются аналогично $IG_{xy}(x,y)$. Затем определяем $J_{12}(x,y)$, $J_{11}(x,y)$ и $J_{22}(x,y)$ по следующим формулам:

$$J_{12}(x,y) = IG_{xy}(x_2,y_2) - IG_{xy}(x_1-1,y_2) - \\ - IG_{xy}(x_2,y_1-1) + IG_{xy}(x_1-1,y_1-1)\,;$$
$$J_{11}(x,y) = IG_{xx}(x_2,y_2) - IG_{xx}(x_1-1,y_2) - \\ - IG_{xx}(x_2,y_1-1) + IG_{xx}(x_1-1,y_1-1)\,;$$
$$J_{22}(x,y) = IG_{yy}(x_2,y_2) - IG_{yy}(x_1-1,y_2) - \\ - IG_{yy}(x_2,y_1-1) + IG_{yy}(x_1-1,y_1-1)\,,$$

где точка (x_1,y_1) — левый верхний угол заданной прямоугольной области изображения, а точка (x_2,y_2) — правый нижний угол.

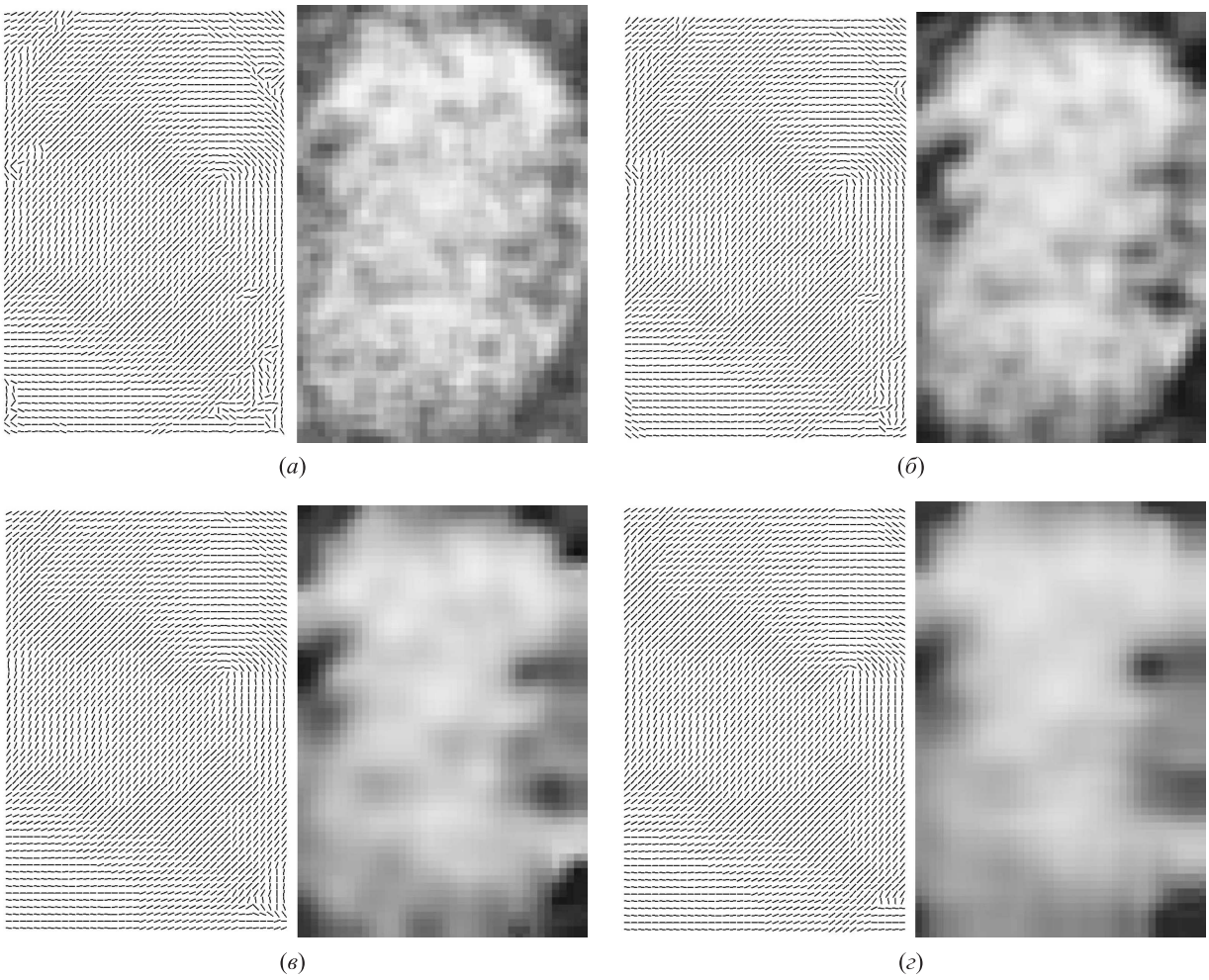
Фактически элементы из Λ_h вычисляются сглаживанием в сегментах $\{S_h(x,y)\}$ компонент поточечного структурного тензорного оператора [5], записываемого в виде:

$$J = \begin{bmatrix} J_{11} + J_{22} \\ J_{22} - J_{11} \\ 2J_{12} \end{bmatrix}. \quad (8)$$

Анализ и коррекция матрицы потоков. В иерархии $h = 3$ на основе (8) для $(x,y) \in X_h \times Y_h$ рассчитывается когерентность потоков по формуле:

$$M_h^{(0)} = \left[\mu_h^{(0)}(x,y)\right] = \\ = \left[\frac{\sqrt{(J_{22}(x,y) - J_{11}(x,y))^2 + 4J_{12}^2(x,y)}}{J_{11}(x,y) + J_{22}(x,y)}\right]. \quad (9)$$

Когерентность для идеальных линий равна единице, а для изотропной структуры — нулю [5]. На



(*а*) (*б*) (*в*) (*г*)

Рис. 3 Результаты вычислений Λ_h (справа) и соответствующих им M_h (слева): (*а*) $\Lambda_h^{(0)}$ и $M_h^{(0)}$; (*б*) $\Lambda_h^{(1)}$ и $M_h^{(1)}$; (*в*) $\Lambda_h^{(2)}$ и $M_h^{(2)}$; (*г*) $\Lambda_h^{(3)}$ и $M_h^{(3)}$

основе ранее вычисленных интегральных изображений находим $\Lambda_h^{(0)}$, $\Lambda_h^{(1)}$, $\Lambda_h^{(2)}$, $\Lambda_h^{(3)}$ и $M_h^{(0)}$, $M_h^{(1)}$, $M_h^{(2)}$, $M_h^{(3)}$ при $h = 3$ и апертурах $A_h(x, y, 8)$, $A_h(x, y, 24)$, $A_h(x, y, 40)$, $A_h(x, y, 56)$ соответственно. Результаты вычислений представлены на рис. 3.

Затем матрицы потоков $\Lambda_h^{(0)}$, $\Lambda_h^{(1)}$, $\Lambda_h^{(2)}$, $\Lambda_h^{(3)}$ и матрицы когерентностей $M_h^{(0)}$, $M_h^{(1)}$, $M_h^{(2)}$, $M_h^{(3)}$ анализируют. Для этого рассчитывают матрицу производных $N_n^{(0)}$ и матрицу потоков $O_h^{(0)}$ в виде:

$$N_h^{(0)} = \left[v_h^{(0)}(x,y)\right] = \\ = \left[\frac{1}{n}\sum_{l=1}^{n-1}\left(\mu_h^{(l)}(x,y) - \mu_h^{(l-1)}(x,y)\right)\right];$$

$$O_h^{(0)} = \lfloor o_h^{(0)}(x,y)\rfloor = \lfloor \delta_h^{(c)}(x,y)\rfloor,$$

где $c = \arg\max_l \left(\mu_h^{(l)}(x,y)\right)$; n — количество слоев (в реализации 4). Матрица $N_h^{(0)}$ отображает изменение когерентности потоков, а $O_h^{(0)}$ — лучшие потоки.

На основе матриц $N_h^{(0)}$ и $O_h^{(0)}$ вычисляют матрицу потока $\Lambda_h^{(4)}$:

$$\Lambda_h^{(4)} = \left\{\delta_h^{(4)}(x,y)\right\}. \qquad (10)$$

Здесь

$$\delta_h^{(4)}(x,y) = \begin{cases} \delta_h^{(3)}(x,y), & \text{если } v_h^{(0)}(x,y) > T; \\ o_h^{(0)}(x,y) & \text{иначе}, \end{cases}$$

где T — пороговое значение (в реализации 0.1).

Элементы из $\Lambda_h^{(4)}$ корректируют по формулам:

$$\Lambda_h^{(5)} = \left\{\delta_h^{(5)}(x,y)\right\} = \left\{\frac{1}{2}\operatorname{arctg}\left(\frac{\operatorname{Im}_h^{(0)}(x,y)}{\operatorname{Re}_h^{(0)}(x,y)}\right)\right\};$$

$$\operatorname{Re}_h^{(0)}(x,y) = \sum_{(u,v)\in A_h(x,y,1)} \mu_h^{(0)}(u,v)\cos\left(2\delta_h^{(4)}(u,v)\right);$$

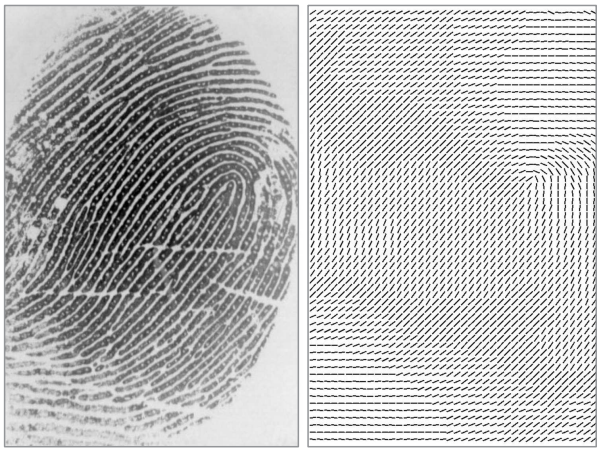


Рис. 4 Результирующая матрица потоков

$$\mathrm{Im}_h^{(0)}(x,y) = \sum_{(u,v)\in A_h(x,y,1)} \mu_h^{(0)}(u,v)\sin\left(2\delta_h^{(4)}(u,v)\right),$$

где $\mu_h^{(0)}(u,v)$ и $\delta_h^{(4)}(u,v)$ — когерентность и поток по (9) и (10) в отсчете (u,v) апертуры $A_h(x,y,1)$ по (4). На рис. 4 показана матрица потоков $\Lambda_h^{(5)}$.

3.3 Сглаживание

Сглаживающий фильтр устраняет микроразрывы и микрозалипания линий. Перед сглаживанием вычисляют матрицу регулярности потока $IR_h^{(0)}$, $h = 3$, используя следующую формулу [1]:

$$IR_h^{(0)} = \left[ir_h^{(0)}(x,y)\right] = \left[1 - \frac{\left\|\sum\limits_{m=-1}^{1}\sum\limits_{n=-1}^{1} d(x+n,y+m)\right\|}{\sum\limits_{m=-1}^{1}\sum\limits_{n=-1}^{1} \| d(x+n,y+m) \|}\right],$$

где $d(x,y) = \lfloor \cos 2\delta_h^{(5)}(x,y),\ \sin 2\delta_h^{(5)}(x,y)\rfloor$; $\| x \|$ — норма L_2.

Затем в основании каждого сегмента $S_h(x,y)$ величины отсчетов сглаживают по формуле:

$$F_0^{(1)} = \begin{cases} f_0^{(1)}(x,y) = \mathbf{H}_1 * \Xi_0^{(\alpha)}(x,y), \\ \qquad\qquad\qquad \text{если } ir_h^{(0)}(x,y) > t; \\ f_0^{(1)}(x,y) = f_0^{(0)}(x,y), \text{ иначе}, \end{cases}$$

где $\mathbf{H}_1$ — ядро одномерной свертки; t — некоторый порог (в реализации 0.3); набор $\Xi_0^{(\alpha)}(x,y) = \left\{\xi_0^{(\alpha)}(u,v)\right\}$ состоит из элементов, выбираемых из $F_0^{(0)}$ прямолинейной щелевой апертурой по (3) в виде

$$\left\{\xi_0^{(\alpha)}(u,v)\right\} = \left\{f_0^{(0)}(u,v) | (u,v) \in A_0^-(x,y,\alpha,w) \cup (x,y)\right\};$$

$\alpha = \delta_h^{(5)}(x,y) \in \Lambda_h^{(5)}$ — направление апертуры, одинаковое для всех отсчетов основания сегмента $S_h(x,y)$; w — размер апертуры.

Перенумеруем упорядоченные отсчеты набора $\Xi_0^{(\alpha)}(x,y) = \left\{\xi_0^{(\alpha)}(u,v)\right\}$, сгенерированного щелевой апертурой по (3), в виде $k \mapsto (u_k, v_k)$, где $k \in 0,\ldots,N$; $N = 2w+1$. Тогда ядро свертки $\mathbf{H}_1$ рассчитывают в виде:

$$\mathbf{H}_1 = \exp\left(-\frac{(w-k)^2}{2\sigma^2}\right),$$

где σ — среднеквадратичное отклонение, определяющее крутизну гауссианы [1] (2–4 в реализации); $k \equiv w$ — отсчет центра окна, здесь равный размеру апертуры w. Сглаживающий фильтр, по сути, является выделенным первым сомножителем разделимого фильтра Габора с числом отсчетов, меньшим в $\pi w/2$ раз, что позволяет повысить производительность обработки. Результат сглаживания представлен на рис. 5.

(*а*) (*б*)

Рис. 5 Исходное (*а*) и сглаженное (*б*) изображения

3.4 Построение матрицы периодов

Это базовый этап обработки, влияющий на точность распознавания частных признаков. Он выполняется на той же иерархии $h = 3$ и состоит из двух последовательно выполняемых процедур обработки ДИ.

Измерение матрицы локальных периодов линий

Определение 1. Под периодом линий понимается величина $t = w/n$, обратно пропорциональная среднему числу n линий, умещающихся в окрестности размером w на прямой, проведенной перпендикулярно линиям [1].

Зададим отрезок $C(x,y) = A_0^-(x,y,\alpha,w) \cup (x,y)$, сгенерированный щелевой апертурой по (3), и перенумеруем отсчеты $(u,v) \in C(x,y)$ в виде $k \mapsto \mapsto (u_k, v_k)$, где $k \in (0,\ldots,N; N = 2w+1$. В отрезке $C(x,y)$ с центром $k \in \{w\}$ в отсчете $(x,y) \in X \times Y$ собираются упорядоченные по k величины $f_0^{(1)}(k)$ яркости изображения. Ориентация щелевой апертуры определяется углом α, выбираемым перпендикулярно потоку по формуле

$$\alpha = \frac{\pi}{2} + \delta_h^{(5)}(x,y)$$

при $(x,y) \in X_h \times Y_h$, а ее длина определяется окрестностью размером w для отсчета $(x,y) \in X \times Y$ (16 в реализации). Для отрезка $C(x,y)$, центрированного в отсчете $(x,y) \in \hat{X}_{h0} \times \hat{Y}_{h0}$ по (2) на сегменте $S_h(x,y)$, введем автокорреляционную функцию в виде:

$$r(i) = \frac{1}{N} \sum_{k=0}^{N-i-1} \overset{\frown}{f}{}_0^{(1)}(k) \overset{\frown}{f}{}_0^{(1)}(k+i), \qquad (11)$$

где $\overset{\frown}{f}{}_0^{(1)}(k) = f_0^{(1)}(k) - \overline{f}$ и $\overline{f} = (1/N) \sum\limits_{k=0}^{N-1} f_0^{(1)}(k)$; N — число отсчетов щелевой апертуры. Определим $\Delta r(i) = r(i+1) - r(i)$. Тогда элементы матрицы локального периода линий $T_h^{(0)} = \left\{ t_h^{(0)}(x,y) \right\}$ вычисляются по формуле:

$$t_h^{(0)}(x,y) = \\ = \arg\min_j \{ (\Delta r(0), \ldots, \Delta r(j)) \,|\, \Delta r(j-1) > 0 \wedge \\ \wedge \Delta r(j) \le 0 \} . \qquad (12)$$

Фактически для каждого сегмента иерархии $h = 3$ выделяют его центр, через который проводят отрезок перпендикулярно потоку. На рис. 6 величины яркости изображения собираются в отсчетах забеленного отрезка. Для них по (12) оценивается локальный период линий на основе автокорреляционной функции по (11), график которой показан на рис. 6. Выбор иерархии $h = 3$ сокращает число оценок в 64 раза.

Отметим, что формула (12) определяет такой локальный период линий, который соответствует экстремуму отсчетов для положительных величин автокорреляционной функции во второй положительной полуволне. На рис. 6 период линий равен 9.

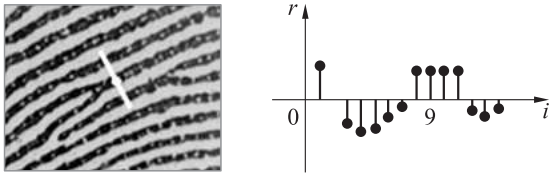


Рис. 6 Определение периода

Выбор экстремума «центрирует» маску фильтра, применяемого для фильтрации ДИ. Однако предположение о том, что оценка периода линий $t_h^{(0)}(x,y)$ может быть смещена, оставляет пространство для маневрирования параметрами фильтрации. На ровном фоне изображения элементы матрицы $T_h^{(0)}$ нулевые.

Анализ и коррекция матрицы периодов линий

Имея в виду то, что отношение значения в точке первого максимума $r(t_h^{(0)}(x,y))$ к постоянной составляющей автокорреляционной функции $r(0)$ для функции синус равно 1, определим матрицу достоверности периода в виде

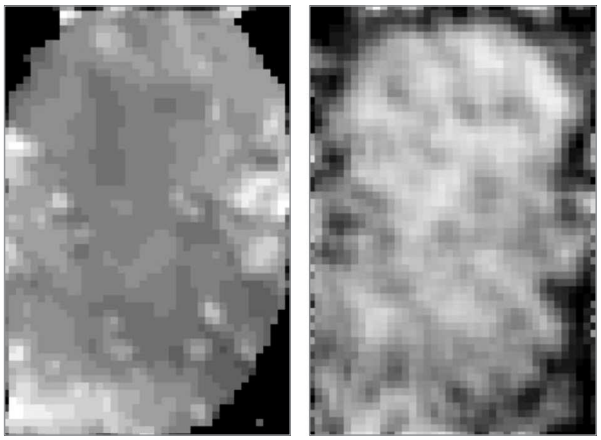
$$Z_h^{(0)} = \left\{ \zeta_h^{(0)}(x,y) \equiv \frac{r(t_h^{(0)}(x,y)}{r(0)} \right\} . \qquad (13)$$

Для ДИ с разрешением 500 dpi $4 \le t_h^{(0)}(x,y) \le \le 17$ [1]. Это позволяет фильтровать ошибки распознавания локального периода, задавая $t_h^{(0)}(x,y) = 0$.

Суть процедуры сводится к расчету матрицы периодов линий $T_h^{(1)} = \left\{ t_h^{(1)}(x,y) \right\}$, которая в начальной итерации с номером $j = 0$ инициализируется: $T_h^{(1)} = T_h^{(0)}$. Далее номер j итерации инкрементируется. В первой итерации для $t_n^{(1)}(x,y) \notin \{0\}$ период линий сглаживается по формуле:

$$t_{h,j}^{(1)}(x,y) = \frac{\sum\limits_R t_{h,j-1}^{(1)}(u,v) \zeta_h^{(0)}(u,v)}{\sum\limits_R \zeta_h^{(0)}(u,v)} , \qquad (14)$$

где условие суммирования $R = t_{h,j-1}^{(1)}(u,v) > 0$ элементов апертуры $(u,v) \in A_h(x,y,1)$ определяется по (4); $n = \sum\limits_R 1$ — количество ненулевых элементов в апертуре. В последующих итерациях для каждого отсчета с кодом пропуска $t_n^{(1)}(x,y) \in \{0\}$ и для смежных с ним ненулевых элементов количеством n, если $n > 4$, период линий прогнозируется по (14). Число итераций ограничивают величиной 2–3. Если ограничение снять, то бóльшая часть элементов из $T_h^{(1)}$ определится.



(*а*) (*б*)

Рис. 7 Результирующая матрица периодов (*а*) и матрица их достоверностей (*б*)

Таким образом, ошибки измерений фильтруются, периоды линий сглаживаются и в финале прогнозируются. Результат коррекции матрицы локальных периодов линий представлен на рис. 7. Нулевые значения периодов показаны черным цветом. Большие значения периодов окрашены светлее.

3.5 Сегментация

Сегментация необходима для отделения информативных областей ДИ от неинформативных. Она выполняется на той же иерархии $h = 3$ и заключается в расчете матрицы меток $C_h^{(0)} = \left\{c_h^{(0)}(x,y)\right\}$ по формуле:

$$c_h^{(0)}(x,y) = \\ = \begin{cases} 1\,, & \text{если } k_1\zeta_h^{(1)}(x,y) + k_2\mu_h^{(1)}(x,y) > \kappa_0\,; \\ 0 & \text{иначе}\,. \end{cases}$$

где $\zeta_h^{(1)}(x,y)$ — величина достоверности периода по (13), сглаженная с помощью маски размером 3×3; $\mu_h^{(1)}(x,y)$ — когерентность потоков по (9), сглаженная с помощью маски размером 3×3; k_0, k_1 и k_2 — обучаемые коэффициенты.

Фактически при выделении информативных областей опираются на два признака: корреляционную функцию и когерентность потоков. Эти признаки сами по себе комплексные. Их сочетание позволяет повысить точность сегментации.

При сегментации могут образовываться островки «разнородных» областей. Их можно дополнительно классифицировать операциями морфологической обработки изображения [4, 5]. Однако из-за ограничения по времени это нежелательно.

3.6 Бинаризация

Бинаризация опирается на ранее вычисленные данные, а именно: на матрицу периодов $T_h^{(1)}$ и матрицу меток $C_h^{(0)}$. Каждый элемент сегмента $S_h(x,y)$ изображения $f_0^{(1)}(x,y)$ с меткой $c_h^{(0)}(x,y) \in \in \{1\}$ бинаризуют следующим образом. Если значение элемента $f_0^{(1)}(x,y)$ меньше среднего значения m элементов в круговой апертуре $A_h(x,y,w)$, умноженного на некоторый коэффициент k (в реализации 0.98), то элементу присваивается значение 1, иначе — 0, т. е.

$$f_0^{(2)}(x,y) = \begin{cases} 1\,, & \text{если } f_0^{(1)}(x,y) < km\,; \\ 0 & \text{иначе}\,. \end{cases}$$

Рис. 8 Бинаризованное изображение

Среднее значение рассчитывается с помощью интегральных изображений по формулам (6) и (7), что позволяет заметно ускорить процесс бинаризации. Размер апертуры w для каждой точки изображения берется из матрицы периодов $T_h^{(1)}$. Результат бинаризации представлен на рис. 8.

3.7 Скелетизация и распознавание частных признаков

Линии бинарного ДИ утончаются до скелетных (рис. 9). Введем некоторые определения.

(*а*) (*б*) (*в*)

Рис. 9 Исходное изображение (*а*), частные признаки (*б*) и скелет изображения (*в*)

Определение 2. Скелетом линии называется простая цепь $\langle u, v\rangle$ с вершинами u и v в 8-смежности, проходящая вблизи геометрического центра линии, причем для каждой вершины $p_1 \in \langle u, v\rangle$ существует ровно две смежные с ней вершины p_2 и p_3, при этом вершины p_2 и p_3 несмежные.

Определение 3. Окончанием называется такая вершина p_1 скелета, что для вершины p_1 существует ровно одна смежная с ней вершина p_2.

Определение 4. Разветвлением называется такая вершина p_1 скелета, что для вершины p_1 существуют ровно три смежные с ней вершины p_2, p_3 и p_4, при этом любые две вершины из множества $\{p_2, p_3, p_4\}$ попарно несмежные.

Скелетизация опирается на раскрашивание точек линий $f_0^{(2)}(x, y) \in \{0\}$ по правилам $P(\xi(x, y))$, определяемым в специальной табличной форме на основе идентификатора окрестности точки в виде

$$\xi(x, y) = \sum_{i \in I} f(i) \cdot 2^i,$$

где $f(i)$ принимает значение 1 для линии и 0 в противном случае; $i \in I = 0, \ldots, 7$ — номер сектора апертуры 3×3 по (4). Величина $\xi(x, y) \in \in 0, \ldots, 255$ определяет ячейку в табличной форме. Согласно [6], итерационное применение правил из $P(\xi(x, y))$ позволяет вычислить скелет линий, показанный на рис. 9. С вершин скелета как графа [7] считываются окончания и разветвления, располагающиеся в информативной области изображения на достаточном расстоянии от ее границы, и помещаются в список (5). Затем применяется структурная постобработка скелета [1]. Частные признаки и их направления показаны на рис. 9, причем окончания окрашены черным цветом, а разветвления — серым.

4 Заключение

В статье предложена группа взаимосвязанных методов, обеспечивающая приемлемое качество распознавания частных признаков при жестких ограничениях на время обработки ДИ. К ним относятся: измерение и коррекция матриц потоков, сглаживание изображения, измерение и коррекция матриц периодов линий, сегментация, бинаризация, скелетизация и считывание с вершин скелета частных признаков. Построение матриц потоков основано на тензорном анализе простых окрестностей, а матриц периодов линий — на автокорреляционной функции. Общее время обработки составляет приблизительно 100 мс (изображение 320×480 пикселей, процессор Intel Pentium 4 CPU 3.0 ГГц).

Обработка основана на операции свертки, что с учетом временны́х характеристик позволяет перенести ее на целевые платы TMS или процессоры DSP [8] и использовать встроенные в них операции свертки. Последнее позволяет реализовать простые портативные биометрические системы, удовлетво-

ряющие достаточно жестким требованиям к производительности.

Аналогично процедуре сглаживания в качестве фильтра может быть использован одномерный фильтр Габора. Это повысит качество обработки, но ухудшит производительность. Кроме того, полученное бинарное изображение можно сгладить, что обычно повышает качество обработки. Дальнейшее направление развития быстрой обработки видится в улучшении метода сегментации изображения, т. е. в более качественном определении информативных зон изображения.

Литература

1. *Maltoni D., Maio D., Jain A. K., Prabhakar S.* Handbook of fingerprint recognition. — New York: Springer-Verlag, 2009. 494 p.
2. *Bolle R. M., Connel J. Y., Pankanti S., Ratha N. K.* Guide to biometrics. — New York: Springer-Verlag, 2004. 368 p.
3. *Гудков В. Ю.* Методы первой обработки дактилоскопических изображений. — Миасс: Геотур, 2008. 127 с.
4. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений / Пер. с англ. — М.: Техносфера, 2006. 1072 с.
5. *Яне Б.* Цифровая обработка изображений / Пер. с англ. А. М. Измайловой. — М.: Техносфера, 2007. 584 с.
6. *Гудков В. Ю., Коляда А. А. , Чернявский А. В.* Новая технология формирования скелетов дактилоскопических изображений // Методы, алгоритмы и программное обеспечение гибких информационных технологий для автоматизированных идентификационных систем. — Минск: БГУ, 1999. С. 71–82.
7. *Новиков Ф. А.* Дискретная математика для программистов: Учебник для вузов. — 3-е изд. — СПб.: Питер, 2008. 384 с.
8. *Сергиенко А. Б.* Цифровая обработка сигналов. — СПб.: Питер, 2002. 608 с.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 108–113

ОБУЧЕНИЕ АЛГОРИТМОВ ВЫДЕЛЕНИЯ КОЖИ НА ЦВЕТНЫХ ИЗОБРАЖЕНИЯХ ЛИЦ*

Ю. В. Визильтер[1], В. С. Горбацевич[2], С. Л. Каратеев[3], Н. А. Костромов[4]

Аннотация: Рассмотрены два способа обучения алгоритмов выделения кожи на цветных изображениях лиц — на основе самоорганизующейся нейронной сети типа «растущий нейронный газ» и морфологической классификации путем построения минимальных разрезов графов соседства на обучающей выборке. В качестве рабочего цветового пространства использовалось пространство CIE Lab. Показана эффективность обоих использованных методов, исследованы различия полученных результатов обучения.

Ключевые слова: биометрия; обнаружение кожи; самоорганизующиеся нейронные сети; морфологическая классификация

1 Введение

Задача выделения человеческих лиц на цифровых изображениях получила широкое распространение в связи с бурным развитием информационных сетей и охранных систем.

Существует много алгоритмов и методов выделения человеческих лиц на изображениях, но наиболее широко применяемым является известный алгоритм Виолы–Джонса, основанный на использовании процедуры обучения типа AdaBoost и хааро-подобных признаков [1]. Недостатком этого алгоритма является необходимость практически попиксельного сканирования изображения окнами различных размеров, что приводит к заметной потере производительности при обработке изображений большого размера.

Для преодоления описанных выше трудностей применяются алгоритмы предобработки, позволяющие сузить область поиска и тем самым повысить производительность. При этом широкое распространение получили методы, основанные на цветовой сегментации изображений по признаку принадлежности человеческой коже.

В работе рассматриваются два различных способа построения подобных классификаторов — на основе самоорганизующейся нейронной сети типа «растущий нейронный газ» и морфологическое обучение методом минимального разрезания графа соседства для обучающей выборки.

2 Морфологическое обучение методом минимального разрезания графа соседства для обучающей выборки

Морфологический подход к синтезу классификаторов основан на рассмотрении задачи синтеза метрического классификатора как задачи оптимальной сегментации (optimal labeling) конечной выборки точек метрического пространства. При этом «форма» и «сложность» классификаторов трактуются в терминах «формы» и «сложности» изображений (образованных метками классов на точках выборки), т. е. в терминах математической морфологии [2–6].

Для алгоритмической реализации процедур синтеза метрических классификаторов используется техника построения минимальных разрезов графов [7–11], применяемая к графам соседства элементов обучающей выборки.

Рассмотрим задачу обучения с учителем. Пусть дано пространство объектов $\mathcal{A}$, конечное множество классов $C = \{c_1, \ldots, c_l\}$ и известно разбиение объектов по классам: $c_{\mathcal{A}}(a) : a \in \mathcal{A} \mapsto c \in C$. Обозначение $c_{\mathcal{A}}$ указывает на то, что функция определена на $\mathcal{A}$.

Производится описание объектов из $\mathcal{A}$ дескрипторами из пространства описаний (признаков) $\mathcal{X}$: $x_{\mathcal{A}}(a) : a \in \mathcal{A} \mapsto x \in \mathcal{X}$. Случайным образом

* Работа выполнена при поддержке РФФИ, гранты № 11-08-01114-а, № 11-08-01039-а.

[1] Государственный научно-исследовательский институт авиационных систем, viz@gosniias.ru
[2] Государственный научно-исследовательский институт авиационных систем, gvs@gosniias.ru
[3] Государственный научно-исследовательский институт авиационных систем, goga@gosniias.ru
[4] Государственный научно-исследовательский институт авиационных систем

формируется конечная выборка объектов $A \subseteq \mathcal{A}$, $\|A\| < +\infty$ и соответствующая выборка описаний $X \subseteq \mathcal{X}$, $\|X\| < +\infty$. Каждому значению x ставится в соответствие класс c породившего его объекта a:

$$c_X(x) : x_{\mathcal{A}}(a) \in X \mapsto c_{\mathcal{A}}(a) \in C\,.$$

По обучающей выборке c_X требуется построить такой распознающий алгоритм, или классификатор,

$$f_{\mathcal{X}}(x) :\ x \in \mathcal{X} \mapsto c \in C\,,$$

который обеспечивает наилучшее разбиение $\mathcal{X}$ на классы из C. «Наилучшее разбиение» формализуется при помощи тестовой выборки

$$c'_Y(x) :\ x \in Y \mapsto c \in C\,,$$
$$Y \subseteq \mathcal{X}\,,\ Y \cap X = \varnothing\,, \|Y\| < +\infty\,,$$

и критерия эмпирического риска на выборке Y:

$$J_Y(f_{\mathcal{X}}) = \frac{d_H(f_Y, c'_Y)}{\|Y\|}\,;$$
$$d_H(f_Y, c'_Y) = \sum_{x \in Y} 1(f(x) \neq c'(x))\,,$$

где 1(true) = 1, 1(false) = 0, $\|Y\| = \sum\limits_{x \in Y} 1$. Здесь расстояние Хэмминга d_{H} имеет смысл числа ошибок классификации на тестовой выборке Y. Отсюда критерий среднего ожидаемого эмпирического риска имеет вид:

$$J_{\mathcal{X}}(f_{\mathcal{X}}) = E_{Y \subseteq \mathcal{X}}\{J_Y(f_{\mathcal{X}})\}\,,$$

где $E_{Y \subseteq \mathcal{X}}\{\cdot\}$ — математическое ожидание по всем возможным выборкам $Y \subseteq \mathcal{X}$.

Таким образом, может быть сформулирована задача построения оператора оптимального синтеза θ, доставляющего минимум критерию $J_{\mathcal{X}}(f_{\mathcal{X}}) = J_{\mathcal{X}}(\theta_{c_X})$:

$$\left.\begin{array}{l} \Theta :\ c_X \in \Omega_X \mapsto f_{\mathcal{X}} \in \Omega_{\mathcal{X}}\,; \\ \theta = \arg\min\limits_{\theta'}\{J_{\mathcal{X}}(\theta' cX\}\,. \end{array}\right\} \qquad (1)$$

Здесь Ω_X и $\Omega_{\mathcal{X}}$ — множества всех возможных разбиений выборки X и пространства $\mathcal{X}$ по классам из C.

В большинстве известных подходов от задачи синтеза (1) сразу переходят к задаче обучения классификаторов заданного класса при помощи обучающего правила известного типа:

$$\left.\begin{array}{l} \theta \in \Theta :\ c_X \in \Omega_X \mapsto F_{\mathcal{X}} \subseteq \Omega_{\mathcal{X}}\,; \\ \theta = \arg\min\limits_{\theta' \in \Theta}\{J_Y(\theta' c_X)\}\,, \end{array}\right\} \qquad (2)$$

где $F_{\mathcal{X}}$ — класс классификаторов; Θ — класс алгоритмов обучения классификаторов из $F_{\mathcal{X}}$ на выборках $X \subseteq \mathcal{X}$.

Кроме того, вместо недоступного критерия $J_{\mathcal{X}}(f_{\mathcal{X}})$ на практике используется критерий наблюдаемого эмпирического риска $J_X(\theta c_X)$, который имеет глобальный минимум в точке $f_X \equiv c_X$, заведомо не пригодный для неизвестной тестовой выборки Y. Этой проблеме посвящена теория оценки и контроля переобучения, созданная Вапником и Червоненкисом [12]. Здесь эмпирический риск оценивается по обучающей выборке, но сложность решающего правила искусственно ограничивается. Для этого вводится понятие сложности классификатора $Q(f_{\mathcal{X}})$, а точнее сложности класса классификаторов $Q(F_{\mathcal{X}})$. Соответственно, вместо задачи (2) решается задача минимизации наблюдаемого риска с регуляризацией по сложности класса обучаемого классификатора:

$$\theta \in \Theta :\ c_X \in \Omega_X \mapsto f_{\mathcal{X}} \in F_{\mathcal{X}} \subseteq \Omega_{\mathcal{X}}\,;$$
$$\theta = \arg\min_{\theta' \in \Theta}\{J_X(\theta' c_X) + \alpha Q(F_{\mathcal{X}})\}\,,$$

где $\alpha \geq 0$ — параметр регуляризации, определяющий компромисс между точностью на обучающей выборке X и сложностью решающего правила, от которой зависит поведение $f_{\mathcal{X}}$ на тестовой выборке Y из (2).

Морфологический подход к машинному обучению направлен непосредственно на решение задачи (1) и позволяет исключить из рассмотрения априорно заданные классы классификаторов и алгоритмов обучения. При этом решение задачи (1) отыскивается в виде композиции решений подзадач

$$\theta_\alpha = \delta_\alpha \psi_\alpha\,, \qquad (3)$$

где ψ_α — оператор (процедура) синтеза оптимального отклика классификатора на обучающей выборке X с учетом его сложности (локальной некомпактности),

$$\left.\begin{array}{l} \psi_\alpha :\ c_X \in \Omega_X \mapsto f_X \in \Omega_X\,; \\ \psi_\alpha = \arg\min\limits_{\psi'}\{J_X(\psi' c_X) + \alpha Q_X(\psi' c_X)\}\,; \end{array}\right\} \qquad (4)$$

δ_α — оператор (процедура) оптимальной корректной интерполяции (расширения) классификатора f_X на $\mathcal{X}$ с учетом сложности получаемого классификатора $f_{\mathcal{X}}$,

$$\delta_\alpha :\ f_X \in \Omega_X \mapsto f_{\mathcal{X}} \in \Omega_{\mathcal{X}}\,;$$
$$\delta_\alpha = \arg\min_{\delta'}\{J_{NN}(\delta' f_X) + \beta Q(\delta' f_X)\}\,.$$

Здесь

$$J_{NN}(\delta f_X) = \\ = \begin{cases} +\infty\,, & \text{если } \exists x \in X:\ \delta f_X(x) \neq f_X(x)\,; \\ d_{\text{H}}\left(\delta f_X(x),\, \delta^{NN} f_X(x)\right) & \text{в противном случае}\,; \end{cases}$$

d_{H} — расстояние Хэмминга; δ^{NN} — простейший оператор интерполяции классификатора, соответствующий правилу ближайшего соседа (nearest neighbor).

Поскольку в задаче (1) функционал J имеет вид расстояния Хэмминга, из утверждения 1 следует, что оператор θ_α (3) является алгебраическим проектором:

$$\theta_\alpha^2 = \theta_\alpha \Rightarrow \forall x \in X:\ \theta_\alpha f_X(x) = f_X(x)\,.$$

Кроме того, на основе θ_α образуется система вложенных классов решающих правил, монотонная относительно α:

$$\forall \alpha \geq \beta \Rightarrow F_{\mathcal{X}}^{\alpha} \subseteq F_{\mathcal{X}}^{\beta}:\ Q(F_{\mathcal{X}}^{\alpha} \leq Q(F_{\mathcal{X}}^{\beta})\,,$$

где $F_{\mathcal{X}}^{\alpha} = \{f_X:\ \theta_\alpha\ f_X = f_X\}$ — множество классификаторов (разбиений), стабильное относительно проектора θ_α. В морфологиях изображений такая система вложенных проективных классов рассматривается как множество пытьевских «форм» нарастающей сложности. В задаче синтеза классификаторов последовательность «форм» используется для решения проблемы переобучения методом минимизации структурного риска.

Определим такой критерий $Q_X(f_X)$, который отдает предпочтение решающим правилам f_X, более компактным на выборке X. Для этого определим систему вложенных окрестностей $O_k(x) \subseteq X$, $x \in X \subseteq \mathcal{X}$, $k = 1, \ldots, \|X\| - 1$, состоящих из k ближайших соседей. Введем локальную меру некомпактности f_X в окрестности $O_k(x)$:

$$Q_k(x, f_X) = \frac{q_{\text{H}}(O_k(x))}{\|O_k(x)\|}\,;$$
$$q_{\text{H}}(O_k(x)) = \sum_{y \in O_k(x)} 1(f_X(x) \neq f_X(y))\,;$$
$$\|O_k(x)\| = \sum_{y \in O_k(x)} 1\,.$$

Тогда глобальная мера k-некомпактности имеет вид:

$$\left.\begin{aligned} Q_X^k(f_X) &= \frac{Q_{\text{H}}(X, f_X)}{\|X\|}\,; \\ Q_{\text{H}}(X, f_X) &= \sum_{x \in X} Q_k(x, f_X)\,. \end{aligned}\right\} \qquad (5)$$

Значение $Q_X^k(f_X)$ характеризует эмпирическую оценку вероятности того, что один из k ближайших соседей в разбиении $f_X(x)$ будет отнесен к другому классу. При любых фиксированных k и X усложнению классификатора f_X соответствует нарастание меры k-некомпактности $Q_X^k(f_X)$. С другой стороны, при увеличении параметра k в (5) преимущество получают более простые и «гладкие» разделяющие поверхности [8].

С учетом критерия (5) задача (4) сводится к хорошо известной задаче оптимальной разметки графа на основе скрытой марковской модели [9], для которой существует эффективное приближенное решение методом минимального разреза графа, вычислимого за низко полиномиальное время относительно числа узлов графа (объектов в выборке) при любом конечном числе классов.

Более того, для случая двух классов метод разреза графов может давать точное глобально оптимальное решение.

Алгоритм нахождения минимального разреза на графе с двумя терминальными вершинами позволяет находить минимум функционала энергии вида

$$E(T) = E_0 + \sum_{i=1,\ldots,N} E_i(t_i) + \sum_{(i,j) \in V} E_{ij}(t_i, t_j)\,, \quad (6)$$

где N — число нетерминальных вершин графа; $T = \langle t_1, \ldots, t_N \rangle$, $t_1, \ldots, t_N \in \{0, 1\}$ — метки ассоциирования каждой нетерминальной вершины с одной из терминальных; $E_i(0)$, $E_i(1) \in \{0, 1\}$ — унарные потенциалы; $E_{ij}(t_i, t_j)$ — парные потенциалы, задаваемые четверкой действительных коэффициентов $E_{ij}(0, 0)$, $E_{ij}(0, 1)$, $E_{ij}(1, 0)$, $E_{ij}(1, 1)$; V — подмножество пар индексов переменных, задающее систему соседства на T.

Энергия (6) считается субмодулярной [11], если

$$\forall (i, j) \in V:\ E_{ij}(0, 0) + E_{ij}(1, 1) \leq \\ \leq E_{ij}(0, 1) + E_{ij}(1, 0)\,. \quad (7)$$

Для субмодулярной энергии (6)–(7) метод построения минимального разреза графа [10, 11] гарантирует нахождение точного минимума [10].

Для реализации задачи синтеза двухклассового классификатора (4), (5) примем: $C\{0, 1\}$, $N = \|X\|$, $T = \langle t_1, \ldots, t_N \rangle$, $t_1 = f_X(x_1), \ldots, t_N = f_X(x_N)$; $E_i(x) = 1(f_X(x) \neq c_X(x))$, $E_{ij}(t_i, t_j) = 1(f_X(x_i) \neq \neq f_X(x_j))$; $V = \{(i, j) : j \in O_k(x_i)\}$.

Легко убедиться, что соответствующая энергия (6) будет субмодулярной, а значит, метод минимального разреза графа k-соседства для выборки c_X действительно оптимален и порождает α-семейства проекторов.

3 Самоорганизующаяся нейронная сеть — растущий нейронный газ

В качестве второго способа цветовой сегментации рассматривается алгоритм кластеризации цветового пространства, основанный на аппроксимации цветового пространства изображения самоорганизующейся сетью, обучаемой по алгоритму растущего нейронного газа [13, 14]. Главным преимуществом этого алгоритма является осуществление так называемой «адаптивной» кластеризации входных данных, т. е. пространство не только разделяется на кластеры, но и определяется необходимое их количество исходя из топологии распределения самих данных. Начиная всего с двух нейронов, алгоритм последовательно изменяет (по большей части увеличивает) их число, одновременно создавая набор связей между нейронами, наилучшим образом отвечающий распределению входных векторов, используя подход соревновательного хеббовского обучения. Каждый нейрон характеризуется так называемой «локальной ошибкой». Соединения между узлами характеризуются «возрастом». Пример такой структуры показан на рис. 1.

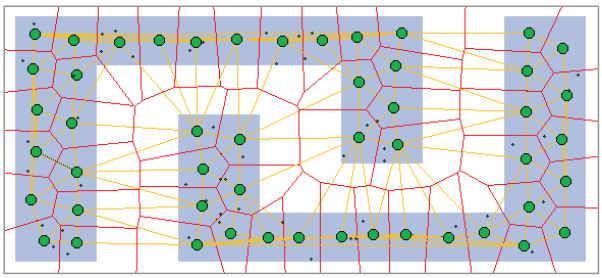


Рис. 1 Структура нейронного газа: распределение кластеров (зеленый), связей (оранжевый) и топология данных (синий), конкретные сигналы показаны в виде отдельных точек

Алгоритм работы растущего нейронного газа кратко можно описать следующим образом:

1. Инициализация: создать два узла с векторами весов, разрешенными распределением входных векторов, и нулевыми значениями локальных ошибок; соединить узлы связью, установив ее возраст равным 0.
2. Подать на вход нейросети вектор x.
3. Найти два нейрона s и t, ближайших к x, т. е. узлы с векторами весов w_s и w_t такими, что $\|w_s - x\|^2$ — минимальное, а $\|w_t - x\|^2$ — второе минимальное значение расстояния среди всех узлов.
4. Обновить локальную ошибку нейрона-победителя s путем добавления к ней квадрата расстояния между векторами w_s и x: $E_s \leftarrow E_s + \|w_s - x\|^2$.
5. Сместить нейрон-победитель s и всех его топологических соседей (т. е. все нейроны, имеющие соединение с победителем) в сторону входного вектора x на расстояния, равные долям ε_w и ε_n от полного:
$$w_s \leftarrow w_s + \varepsilon_w(w_s - x)\,;$$
$$w_n \leftarrow w_n + \varepsilon_n(w_n - x)\,.$$
6. Увеличить на 1 возраст всех соединений, исходящих от победителя s.
7. Если два лучших нейрона s и t соединены, обнулить возраст их связи. В противном случае создать связь между ними.
8. Удалить все соединения, возраст которых превышает $\mathrm{age}_{\max}$. Если после этого остаются нейроны, не имеющие связей с другими узлами, удалить эти нейроны.
9. Если номер текущей итерации кратен λ и предельный размер сети не достигнут, создать новый нейрон r по следующим правилам:
   - найти нейрон u с наибольшей локальной ошибкой;
   - среди соседей u найти нейрон v с максимальной ошибкой;
   - создать узел r «посредине» между u и v:
   $$w_r = \frac{w_u + w_v}{2}\,;$$
   - заменить связь между u и v на связи u и r, v и r;
   - уменьшить ошибки нейронов u и v, установить значение ошибки нейрона r:
   $$E_u \leftarrow E_u\alpha\,;\; E_v \leftarrow E_v\alpha\,;\; E_r \leftarrow E_u\,;$$
   - уменьшить ошибки всех нейронов j на долю β:
   $$E_j \leftarrow E_j - E_j\beta\,.$$
10. Если критерий останова не выполнен, перейти к шагу 2.

4 Результаты экспериментального исследования

Тестирование проводилось на базе изображений людей, снятых при различных условиях съемки. Изображения были предварительно размечены вручную на области принадлежности

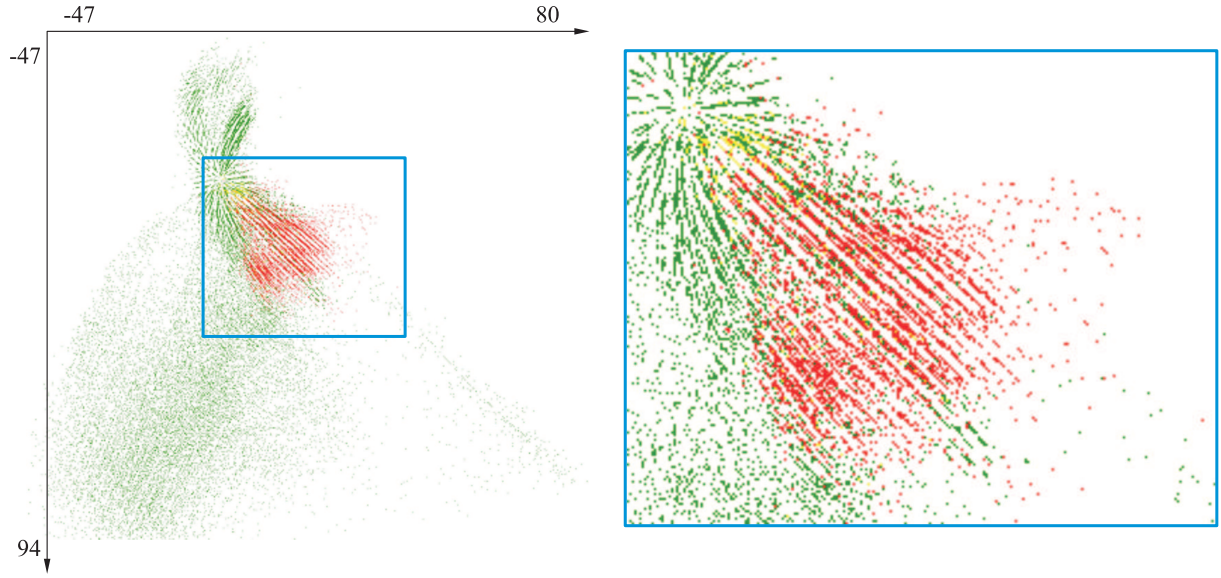
пикселов классу кожи. Выборка точек для обработки получена путем перевода тестовых изображений в цветовое пространство CIE Lab с целью отделения цветовых компонент от яркости. Это повышает компактность представления, так как кожа имеет характерный цвет, а не яркостную составляющую. Обучение производилось на 10% точек от общего объема выборки в 800 000 точек.

При построении графа соседства использовался алгоритм триангуляции Делоне с динамическим кэшированием треугольников [15]. Нахождение оптимальных разрезов графов осуществлялось с использованием библиотеки [16].

Полученная вероятность правильной классификации цвета пиксела (кожа / не кожа) на тестовой выборке — 0,937.

Для самоорганизующейся нейронной сети были рассмотрены результаты при 32, 128 и 256 кластерах, полученных после кластеризации обучающей выборки. Полученная вероятность правильной классификации цвета пиксела (кожа / не кожа) на тестовой выборке — 0,925.

Как видно, численные значения результатов обучения, полученные двумя описанными методами в задаче цветовой сегментации кожи на изображениях лиц, оказались достаточно близки.



(*а*) (*б*)

Рис. 2 Обучающая выборка (*а*) и ее увеличенный фрагмент (*б*), содержащий точки «кожи» (красные) и «не кожи» (зеленые); желтым показаны точки, имеющие обе метки

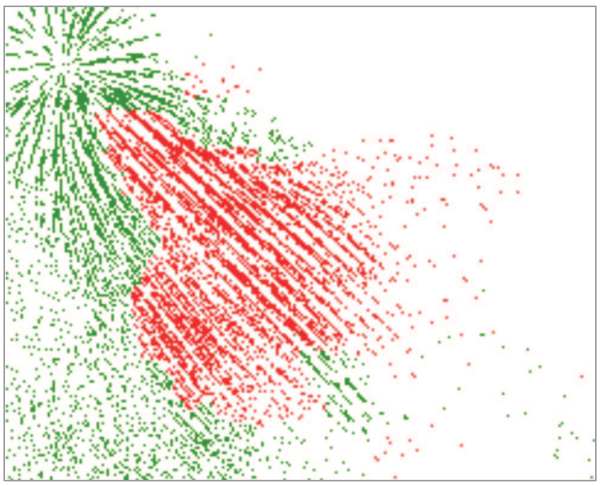


Рис. 3 Результат переразметки обучающей выборки после морфологического обучения на основе разреза графа

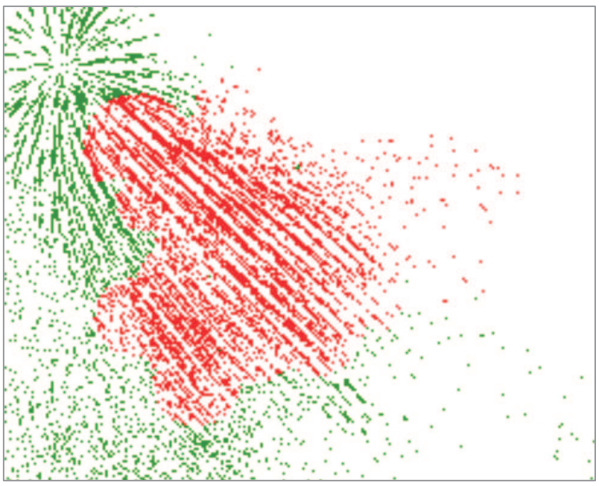


Рис. 4 Результат переразметки обучающей выборки после обучения на основе «растущего нейронного газа»

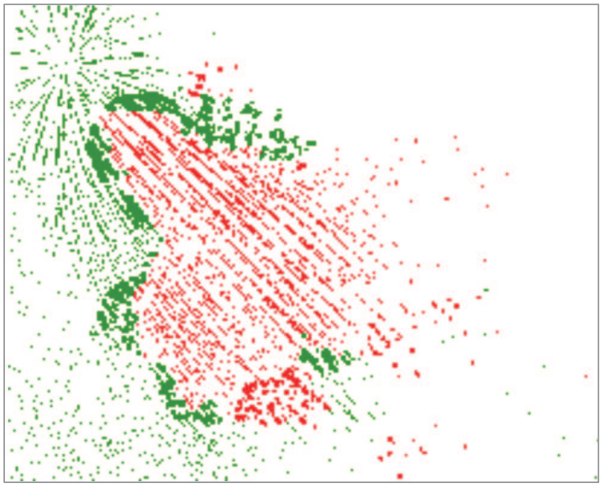


Рис. 5 Различия в результатах обучения (см. рис. 3 и 4). Жирным выделены точки, классифицируемые поразному

Однако более подробное рассмотрение выделенных кластеров демонстрирует существенные различия в их форме. На рис. 2 показана обучающая выборка в плоскости ab цветового пространства CIE Lab. Красными точками помечены пикселы кожи, зелеными — других классов.

На рис. 3 приведен результат переразметки обучающей выборки после применения процедуры обучения на основе разреза графа соседства, а на рис. 4 — после обучения на основе «растущего нейронного газа». Рисунок 5 демонстрирует различия в форме кластеров, полученных двумя способами обучения.

Как видно, значительные отличия в форме полученных кластеров цвета кожи указывают на существенно различную природу этих процедур обучения, что позволяет в дальнейшем рассматривать возможность их комплексирования с целью повышения вероятности правильной классификации.

Литература

1. *Viola P., Jones M.* Robust real-time object detection // IEEE Workshop on Statistical and Computational Theories of Vision Proceedings. — Vancouver, CA, 2001.
2. *Serra J.* Image analysis and mathematical morphology. — London: Academic Press, 1982.
3. *Пытьев Ю. П.* Морфологический анализ изображений // Доклады АН СССР, 1983. Т. 269. № 5. С. 1061–1064.
4. *Pavel M.* Fundamentals of pattern recognition. — New York: Marcel Dekker, Inc., 1989.
5. *Визильтер Ю. В.* Обобщенная проективная морфология // Компьютерная оптика, 2008. Т. 32. № 4. С. 384–399.
6. *Пытьев Ю. П., Чуличков А. И.* Методы морфологического анализа изображений. — М.: Физматлит, 2010. 336 с.
7. *Ford L., Fulkerson D.* Flows in networks. — Princeton University Press, 1962.
8. *Greig D., Porteous B., Seheult A.* Exact maximum *a posteriori* estimation for binary images // J. Roy. Statistical Soc., 1989. Vol. 51. No. 2. P. 271–279.
9. *Boykov Y., Kolmogorov V.* Computing geodesics and minimal surfaces via graph cuts // IEEE Conference (International) Computer Vision (ICCV) Proceedings, 2003. P. 26–33.
10. *Kolmogorov V., Zabih R.* What energy functions can be minimized via graph cuts? // IEEE Trans. Pattern Anal. Machine Intelligence (PAMI), 2004. Vol. 26. No. 2. P. 147–159.
11. *Boykov Y., Kolmogorov V.* An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision // IEEE Trans. Pattern Anal. Machine Intelligence (PAMI), 2004. Vol. 26. No. 9. P. 1124–1137.
12. *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
13. *Fritzke B.* Fast learning with incremental RBF networks // Neural Processing Lett., 1994. Vol. 1. No. 1. P. 2–5.
14. *Fritzke B.* A growing neural gas network learns topologies // Advances in neural information processing systems 7 / Eds. G. Tesauro, D. S. Touretzky, T. K. Leen. — Cambridge MA: MIT Press, 1995. P. 625–632.
15. *Скворцов А. В.* Обзор алгоритмов построения триангуляции Делоне // Вычислительные методы и программирование, 2002. Т. 3. С. 14–39.
16. *Boykov Y., Kolmogorov V.* MAXFLOW — software for computing mincut/maxflow in a graph. V. 3.01. http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 114–121

РАСПОЗНАВАНИЕ ЖЕСТОВ ЛАДОНИ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ ПЛОСКИХ И ПРОСТРАНСТВЕННЫХ СКЕЛЕТНЫХ МОДЕЛЕЙ*

А. В. Куракин[1]

Аннотация: Рассмотрена задача распознавания жестов ладони и определения координат частей ладони в пространстве на основе анализа формы силуэта ладони. Для ее решения разработан метод определения координат кончиков пальцев на бинарном изображении ладони посредством анализа его скелетного представления. Предложен алгоритм определения координат пальцев и центра ладони в пространстве за счет анализа стереопары изображений силуэтов ладони. Разработанные методы работают в реальном времени, что позволяет использовать их в прикладных системах распознавания жестов.

Ключевые слова: непрерывный скелет; анализ формы; распознавание жестов; стереозрение

1 Введение

О важности задачи распознавания жестов говорит огромное число работ в данной области. Потенциальные приложения технологий распознавания жестов включают человеко-машинное взаимодействие [1], приложения виртуальной реальности [2], распознавание языка жестов [3] и др. [4, 5].

В данной работе рассматривается задача распознавания жестов ладони по видеопоследовательностям изображений, полученным с одной или двух веб-камер. Для решения поставленной задачи в работе предлагается метод, работающий на основе анализа формы ладони. Использование только формы, без текстурных признаков, позволяет применять метод даже на изображениях низкого качества, полученных с веб-камер.

Анализ формы выполняется на основе использования непрерывного скелета — множества центров максимальных вписанных в силуэт исходной фигуры кругов. Непрерывный скелет является весьма информативным дескриптором формы, позволяет анализировать топологию объекта и измерять такие его признаки, как ширина объекта в произвольной точке скелета. Понятие непрерывного скелета описано в разд. 3.

В работе предлагается метод выделения кончиков пальцев, вводится понятие центра ладони и предлагается метод его вычисления. Эти данные используются как признаки для классификации жестов. Рассматриваемый в разд. 4 алгоритм позволяет найти двухмерные координаты кончиков пальцев и центра ладони на основе сегментации скелета силуэта ладони. Предлагается простой классификатор жестов рук по вычисленным характеристикам. Более того, алгоритм допускает обобщение на случай трех измерений за счет использования стереопары силуэтов ладони в качестве входных данных (см. разд. 5).

Для апробации методов разработаны аппаратно-программные комплексы, состоящие из одной или двух веб-камер и программного обеспечения для распознавания жестов. Они описаны в разд. 6.

Рассмотренные методы работают в реальном времени и допускают применение в практических системах распознавания жестов.

2 Обзор литературы

В литературе рассмотрено большое количество подходов к решению задачи распознавания жестов. Эти подходы можно классифицировать по типу используемых входных данных и сенсоров для восприятия руки. Во-первых, это методы, использующие специализированные невизуальные сенсоры, такие как роботизированные перчатки [2]. Во-вторых, методы, использующие визуальную информацию, но требующие от пользователя размещать на руке какие-либо маркеры, облегчающие трекинг и классификацию позы ладони [6]. В-третьих, методы, подобные рассмотренному в данной статье, которые работают исключительно с визуальной информацией и не предъявляют специальных требований к оснащению пользователя дополнительным оборудованием [5].

Визуальные методы распознавания жестов можно разделить на три большие категории. К первой

* Работа выполнена при финансовой поддержке РФФИ, проекты № 11-01-00783 и № 11-07-00462.

[1] Московский физико-технический институт (государственный университет), alekseyvk@yandex.ru

относятся методы, которые основаны на восстановлении полной модели кисти с 27 степенями свободы по входному изображению [7]. Теоретически это наиболее перспективные методы, так как они подразумевают полное оценивание позы и динамики руки. Основными ограничениями подобных методов являются большая вычислительная сложность и ограниченная точность восстановления модели руки из-за наличия окклюзий, что делает невозможным их применение на практике.

Ко второй категории относятся методы, которые вместо восстановления полной модели руки предлагают построение признакового описания входного изображения и дальнейшую классификацию жестов именно по этому описанию. В работах [1, 3] в качестве такого описания используются макрохарактеристики силуэта ладони (размеры, положение, инварианты Ху [8]).

К третьей категории относятся метрические методы распознавания жестов. Подобные методы предполагают построение некоторой метрики на множестве входных изображений и выполнение классификации за счет сравнения входного изображения с набором эталонов. Например, в [9] предлагается метрика, характеризующая степень сходства скелетов силуэтов ладони, и выполняется классификация жестов с помощью метода ближайшего соседа.

Данная работа относится ко второй категории визуальных методов: по изображению генерируются признаки, на основе которых выполняется классификация жестов. В качестве признаков используются координаты кончиков пальцев и центра ладони.

3 Непрерывный скелет

Рассматриваемые в статье методы анализа формы базируются на понятии непрерывного скелета фигуры. Определение непрерывного скелета вместе со способом его получения кратко описано в этом разделе. Более детальную информацию можно найти в книге [10].

Для многоугольной фигуры F *максимальным пустым кругом* будем называть всякий круг B, полностью содержащийся внутри фигуры F, такой что любой другой круг B', содержащийся внутри фигуры F, не содержит в себе B, т. е. $\forall B' \subset F$, $B' \neq B : B \not\subset B'$.

Используя понятие максимального пустого круга, определим скелет следующим образом:

Определение 1. *Скелетом многоугольной фигуры F является множество центров ее максимальных пустых кругов.*

На скелете определена *радиальная функция* $R(x, y)$, которая ставит в соответствие каждой точке скелета (x, y) значение радиуса максимального пустого круга с центром в этой точке.

Можно доказать [11], что скелет многоугольной фигуры состоит из объединения конечного числа отрезков и дуг парабол. Таким образом, скелет многоугольной фигуры можно рассматривать как планарный граф, вершины которого — это точки соединения отрезков и дуг парабол, а ребра — это собственно отрезки и дуги парабол, составляющие скелет. Степень любой вершины в таком графе будет равна 1, 2 или 3.

Далее в статье будут использоваться свойства скелета и как графа, и как объединения кривых.

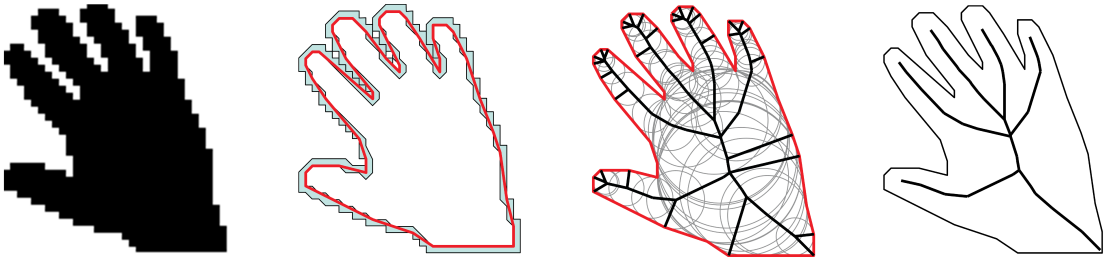
На практике при анализе формы зачастую приходится иметь дело не с многоугольниками, а с растровыми бинарными изображениями, на которых один из двух цветов обозначает принадлежность соответствующего пиксела объекту. Соответственно, для построения непрерывного скелета необходимо в первую очередь построить многоугольную аппроксимацию исходной фигуры. Далее для полученного многоугольника можно строить скелет. Существующие эффективные алгоритмы позволяют выполнять построение скелета за время $O(N \log N)$, где N — число вершин в многоугольнике. После построения скелета обычно выполняется его дополнительная обработка, называемая «стрижкой», с целью удаления малозначимых и шумовых ветвей. На рис. 1 продемонстрирован процесс построения скелета на примере изображения ладони низкого разрешения.

4 Анализ формы ладони и распознавание жестов с помощью скелета

Распознавание жестов выполняется путем анализа бинарного изображения силуэта ладони. Сначала для бинарного изображения ладони выполняется построение скелета и его регуляризация («стрижка»). Затем выполняется поиск координат кончиков пальцев и центра ладони на силуэте. Далее выполняется распознавание жестов за счет анализа числа видимых пальцев и динамики их взаимного перемещения.

Определение 2. *Центром ладони будем считать центр вписанного в силуэт ладони круга, имеющего максимальный радиус среди всех вписанных кругов.*

Центр ладони определяется по скелету как точка, в которой радиальная функция принимает максимальное значение. Алгоритм поиска пальцев



(*а*) (*б*) (*в*) (*г*)

Рис. 1 Процесс построения скелета: исходная бинарная картинка (*а*); многоугольная аппроксимация границы объекта (*б*); скелет многоугольника (*в*); скелет после стрижки (*г*)

описан ниже в подразд. 4.2, а в подразд. 4.1 вводятся необходимые определения.

Состав рассматриваемых жестов и их назначение в разработанном программном комплексе (см. разд. 6) следующие:

(1) виден один палец (см. рис. 4, *б*). Палец и рука движутся как целое. Этот жест используется для перемещения курсора и перетаскивания объектов;

(2) видны два пальца, и их кончики двигаются вдоль прямой (см. рис. 5). Этот жест используется для изменения размера объектов;

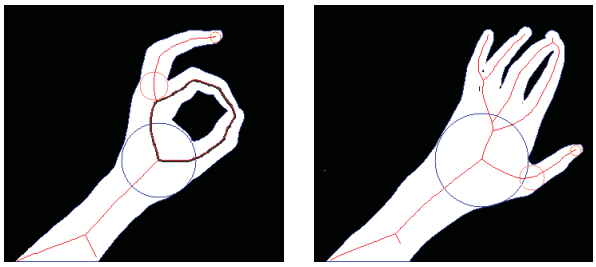
(3) видны три пальца, и они двигаются так, чтобы длины сторон образованного ими треугольника менялись незначительно во времени (см. рис. 4, *а*). Этот жест используется для вращения объектов;

(4) большой и указательный пальцы образуют кольцо на небольшой фиксированный промежуток времени (см. рис. 2, *а*). Этот жест используется для захвата объектов;

(5) все пять пальцев видны и широко расставлены на небольшой фиксированный промежуток времени (см. рис. 1). Этот жест используется для освобождения объектов.

Жесты с 1, 2, 3 и 5 пальцами легко распознаются путем подсчета числа видимых пальцев. Для жестов из двух и трех пальцев начальное и конечное положение пальцев используется для определения количественных характеристик жеста, таких как коэффициент масштабирования при изменении размера объекта и угол поворота при вращении.

Жест «кольцо» распознается путем поиска циклов в скелетном графе. Следует заметить, что только циклы, проходящие вблизи центра ладони, рассматриваются как подходящие (рис. 2), потому что только такие циклы образованы кольцом между большим и указательным пальцами. Такая



(*а*) (*б*)

Рис. 2 Циклы в скелете: цикл (*а*) соответствует жесту-кольцу, а цикл (*б*) жестом-кольцом не является

классификация возможна благодаря тому, что известны координаты всех вершин скелета. Более того, для обнаружения кольца нет необходимости анализировать контуры ладони, а достаточно лишь использования скелетного графа.

Жест «кольцо» и жест из пяти пальцев распознаются как динамические жесты, т. е. измеряется время наблюдения жеста и он засчитывается, только если это время превышает заданный порог. Динамическое распознавание жестов стало возможным благодаря высокой скорости обработки кадров.

4.1 Ветвь скелета и ее свойства

Анализ формы ладони производится на основе анализа ветвей скелета. Ветвь скелета — это просто часть скелета, рассмотренная как непрерывная кривая. Формальное определение дано ниже.

Определение 3. *Пусть* $\vec{s}(\bullet) : s(l) = \{x(l), y(l)\}, l \in \in [0, L]$, *— непрерывная кусочно-гладкая кривая без самопересечений и* l *является естественной параметризацией кривой (т. е. длиной дуги кривой). Пусть каждая точка кривой* $\vec{s}(\bullet)$ *является одновременно и точкой скелета. В таком случае кривую* $\vec{s}(\bullet)$*, соединяющую точки скелета* $r(0)$ *и* $r(L)$*, будем называть* ветвью скелета.

Для каждой точки скелета с координатами (x, y) известно значение радиальной функции $R(x, y)$, равное радиусу максимального пустого круга с центром в этой точке. Дополнительно для произвольной ветви скелета $\vec{s}(\bullet)$ будем рассматривать *радиальную функцию вдоль ветви* $R_s(l) = R(\vec{s}(l)), l \in [0, L]$.

Из-за наличия в скелете дуг парабол работа с радиальной функцией вдоль ветви сопряжена с вычислительными сложностями. Но у скелетов реальных изображений дуги парабол очень короткие, имеют малую кривизну и приближенно могут быть рассмотрены как отрезки. Таким образом, заменив все дуги-параболы на отрезки, получим скелет, для которого радиальная функция вдоль любой ветви будет кусочно-линейной. Будем называть ее *аппроксимированной радиальной функцией вдоль ветви*, а вычислять описанным ниже способом.

Рассмотрим две вершины исходного скелета A и B и простой путь P между ними в скелетном графе. Путь P однозначно определяет ветвь скелета $\vec{s}(\bullet)$.

Обозначим вершины скелетного графа, входящие в путь P (в порядке прохождения пути), как $V_0 = A, V_1, \ldots, V_{n-1}, V_n = B$.

Обозначим значения радиальной функции в этих вершинах как $R(V_i) = R_i$.

Пусть L_i — длина пути между вершинами A и V_i в предположении, что все дуги скелета являются отрезками, т. е. $L_i = \sum\limits_{k=0}^{i-1} |V_k V_{k+1}|$.

В данных обозначениях аппроксимированная радиальная функция $\tilde{R}_{\tilde{s}}(l)$ вдоль ветви $\vec{s}(\bullet)$ может быть вычислена следующим образом:

$$\tilde{R}_{\tilde{s}}(l) = \begin{cases} R_i & \text{при } l = L_i; \\ \alpha R_i + (1-\alpha) R_{i+1} & \text{при } L_i < l < L_{i+1}, \\ & \alpha = \dfrac{l - L_i}{L_{i+1} - L_i}. \end{cases}$$

4.2 Распознавание пальцев

Поиск пальцев на силуэте ладони производится путем анализа ветвей скелета, а также радиальной функции вдоль них.

Анализ силуэтов ладоней разных людей по собранной базе примеров показал, что ветви, соответствующие пальцам, имеют набор сходных особенностей.

Во-первых, все ветви, соответствующие пальцам, оканчиваются в висячих вершинах скелета, поэтому далее рассматриваем только ветви, соединяющие висячие вершины с вершинами степени 3.

Во-вторых, каждую ветвь, соответствующую пальцу, можно условно разбить на две части: собственно палец и пясть. При этом для пальца радиальная функция колеблется в небольших пределах около значения, являющегося полушириной пальца, а для пясти она значительно растет.

График радиальной функции для пальца приведен на рис. 3, *а*. Заметим, что для всех ветвей, соответствующих пальцам, радиальная функция выглядит сходным образом; отличия касаются только конкретного значения ширины пальца, а также положения точки, где заканчивается палец и начинается пясть.

В свою очередь, для ветвей, не являющихся пальцами, радиальная функция может иметь различный вид, но он всегда отличается от вида радиальной функции для пальца. Один из примеров радиальной функции для ветви, не являющейся пальцем, приведен на рис. 3, *б*.

С учетом этих особенностей предлагается следующий эвристический алгоритм для распознавания пальцев:

1. Анализируем все ветви скелета, соединяющие висячие вершины скелета с ближайшими вершинами степени 3.

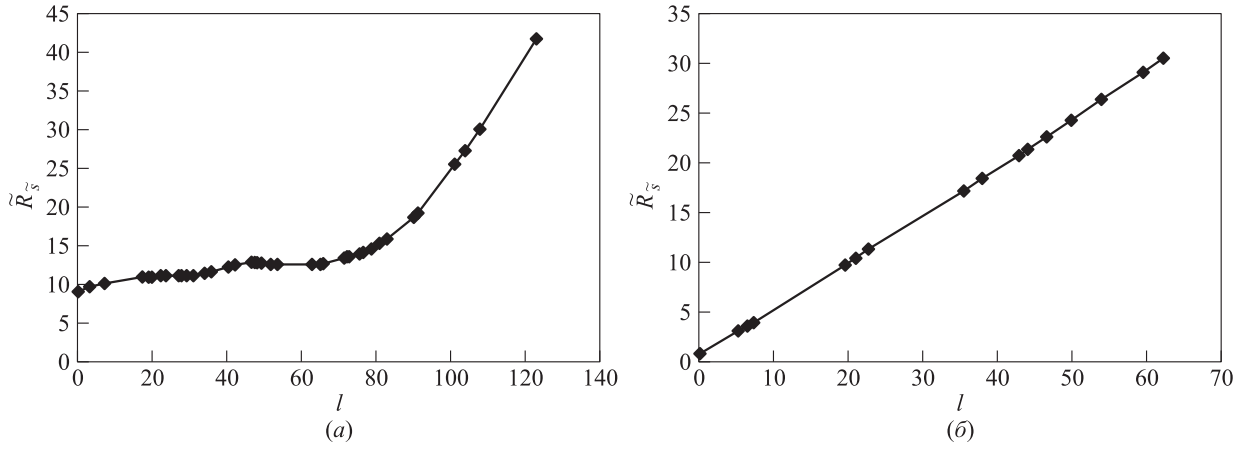


Рис. 3 Пример радиальной функции для пальца (*а*) и для ветви, не являющейся пальцем (*б*)

2. Для каждой такой ветви выполняем поиск точки C, которая является наиболее вероятным местом сочленения пальца и пясти.

3. Когда определена точка C, проверяем набор критериев на геометрические размеры (длину, толщину) ветви, чтобы отсечь те ветви, которые не являются пальцами.

Будем использовать обозначения V_i, R_i и L_i из предыдущего подраздела и введем дополнительно величину D_i как дискретную производную R по L. Положим $D_0 = 0$, $D_n = +\infty$, а в остальных точках D_i будем вычислять по формуле:

$$D_i = \frac{R_{i+1} - R_{i-1}}{L_{i+1} - L_{i-1}}, \qquad i = 1, \ldots, n-1\,.$$

Поиск точки C выполняется из тех соображений, что в момент, когда заканчивается палец и начинается пясть, происходит выполнение одного из следующих условий:

– R увеличивается более чем в 2–2,5 раза по сравнению с началом пальца;

– радиус начинает резко расти, т. е. частные производные D_i превосходят наперед заданный порог (использовалось значение порога, равное 0,4–0,6).

После того как для сегмента AB найдена точка C вероятного сочленения пальца и ладони, выполняется вычисление длины сегментов AC и AB и толщины пальца (как радиуса максимальной вписанной окружности в определенной точке скелета либо как среднего значения радиуса вдоль AC). Сегмент AB классифицируется как палец, если выполняются все следующие условия:

– $|AC|/|AB| \geq 0{,}35$;

– толщина ветви AC находится в заданных пределах;

– длина $|AB|$ превышает величину порога (т. е. ветвь, классифицируемая как палец, должна быть достаточно длинной).

Конкретные значения параметров алгоритма получены эмпирическим путем по собранной базе изображений ладони.

На рис. 4 приведен пример результата работы алгоритма детектирования пальцев. На нем изображены самые большие круги, вписанные в ладонь, и маленькие круги, соответствующие кончикам пальцев и местам сочленения пальцев и ладони.

5 Анализ стереопары ладоней и слежение за рукой в пространстве

Алгоритм анализа формы ладони, описанный в предыдущем разделе, можно расширить для определения трехмерных координат руки и кончиков пальцев за счет использования стереопары силуэтов ладоней.

Рука человека может быть приближенно описана в виде циркулярной модели — пространственного графа, с каждой точкой которого связана сфера. Если рука наблюдается без окклюзий, т. е. разные точки руки проецируются в разные точки ее изображения, то скелет силуэта проекции руки приближенно совпадает с проекцией пространственного графа, порождающего руку [12]. Благодаря этому свойству можно выполнять восстановление модели руки по стереопаре силуэтов [13, 14]. Однако метод не будет работать при наличии окклюзий.

С другой стороны, для практических задач восстановление полной модели руки зачастую не нужно. В ситуации, когда присутствуют окклюзии, можно выполнять частичное восстановление модели руки по ключевым точкам. В качестве таких ключевых точек могут выступать, например, кончики пальцев и центр ладони. Идея состоит в том, что, выполнив поиск этих точек на обоих изображениях стереопары силуэтов руки и найдя соответствие между точками, можно вычислить их координаты в пространстве. Детально алгоритм определения пространственного положения кончиков пальцев и центра ладони описан ниже.

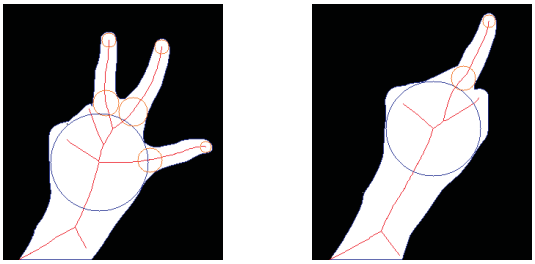
Пусть имеется пара откалиброванных камер P_1 и P_2, наблюдающих искомую модель руки. Обозначим через S_1 и S_2 силуэты искомой модели в проекции на плоскости камер P_1 и P_2.

Построим скелеты M_1 и M_2 для силуэтов S_1 и S_2.

Для каждого из полученных скелетов выполним детектирование кончиков пальцев и центра ладони, описанное в подразд. 4.2. Множества найденных точек $\{A_i^1 = (x_i^1, y_i^1), i = 1, \ldots, N_1\}$ и $\{A_i^2 = (x_i^2, y_i^2), i = 1, \ldots, N_2\}$ для изображений, полученных с первой и второй камер соответственно, являются ключевыми точками, используемыми далее для определения положения частей искомой модели в пространстве.

Для найденных ключевых точек выполним определение соответствия между ними. Алгоритм определения соответствия следующий:

– с помощью эпиполярной геометрии [15] выполняется сопоставление центров ладоней на паре кадров (если в кадре присутствует несколько ладоней);



(*а*) (*б*)

Рис. 4 Пример детектирования пальцев

– для каждой пары соответствующих проекций ладоней выполняется определение соответствия между кончиками пальцев этой ладони. При этом используется как эпиполярная геометрия, так и тот факт, что ориентация пальцев относительно центра ладони должна быть одинаковой на обоих кадрах.

В результате получается набор стереопар $\{A^1_{f(t)}, A^2_{g(t)}\}$, где $t = 1, \ldots, N$, а $f(t)$ и $g(t)$ — целочисленные функции, определяющие соответствие между точками. При этом точки, для которых не нашлось пары на другом изображении, не участвуют в дальнейшем рассмотрении.

Для всех найденных стереопар выполним стереотриангуляцию [15] и вычислим трехмерные координаты их прообразов. В результате получим пространственные координаты кончиков пальцев и центра ладони.

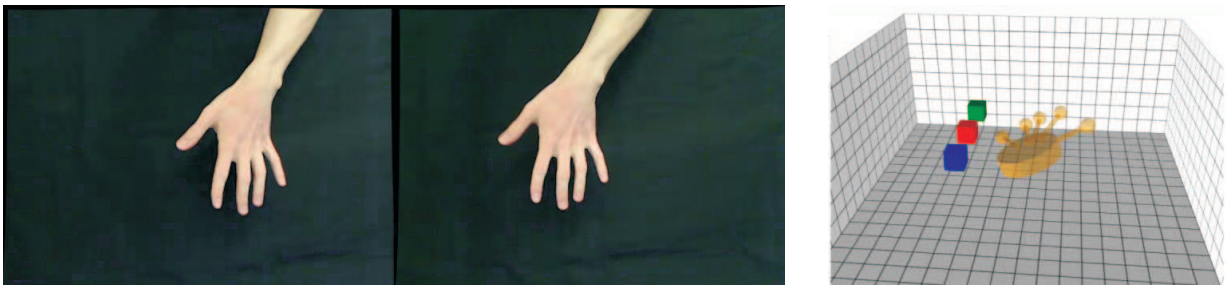
6 Аппаратно-программный комплекс

Для экспериментов и демонстрации вышеописанных методов разработаны два аппаратно-программных комплекса. Один из них выполняет распознавание жестов руки посредством анализа изображения, полученного с одной веб-камеры. Второй выполняет анализ изображений с двух веб-камер и определяет положение руки и кончиков пальцев в трехмерном пространстве.

Для двухмерного распознавания жестов используется следующий комплекс. Обычная веб-камера (Logitech 9000) располагается над однородной темной поверхностью. Пользователь двигает рукой перед поверхностью, и изображение снимается камерой. С использованием точечных методов детектирования кожи [16, 17] выделяется силуэт ладони. Он анализируется алгоритмами из разд. 4, в

(*а*) (*б*)

Рис. 5 Иллюстрация работы комплекса для распознавания жестов в двух измерениях: изображен жест, используемый для масштабирования



(*а*) (*б*)

Рис. 6 Иллюстрация работы комплекса для слежения за рукой в трехмерном пространстве: стереопара изображений, полученных с веб-камер (*а*); модель руки, визуализированная в виртуальном трехмерном пространстве (*б*)

результате чего осуществляется распознавание жестов. Обнаруженные жесты используются для перемещения, вращения и масштабирования объектов на экране компьютера. На рис. 5 приведены снимки экрана данной программы.

В аппаратно-программном комплексе для восстановления положения ладони и пальцев в трехмерном пространстве используется пара откалиброванных веб-камер. В остальном он повторяет двухмерный вариант. На паре изображений, полученных с камер, выделяются силуэты руки и производится их обработка методом, описанным в разд. 5. На рис. 6 приведен пример его работы.

Эффективные алгоритмы построения и стрижки скелета позволяют использовать системы для слежения за рукой в реальном времени. Например, для однопоточной реализации двухмерного алгоритмы распознавания жестов (включая сегментацию руки, построение и стрижку скелета, распознавание жестов и рисование результата) требуется около 22 мс на обработку одного кадра на компьютере 2,4 ГГц Intel Core 2 Quad CPU.

7 Заключение

В работе рассмотрен метод распознавания жестов ладони. Предложен алгоритм определения положения кончиков пальцев и центра ладони с помощью анализа скелета силуэта ладони. Координаты кончиков пальцев и центра ладони могут быть найдены как на плоскости, при наличии одного изображения, так и в пространстве, при наличии стереопары силуэтов. Найденный набор координат используется как признаковое описание для классификации жестов.

Среди достоинств метода следует отметить:

- использование исключительно силуэта, что обеспечивает независимость от текстуры руки и освещения;
- возможность применения недорогих веб-камер в качестве сенсоров;
- возможность обработки видео в реальном времени.

Среди дальнейших направлений исследований можно отметить следующие:

(1) усложнение набора распознаваемых жестов. В частности, можно реализовать распознавание сложных динамических жестов с помощью скрытых марковских моделей;

(2) расширение спектра возможных применений метода, например адаптация метода для распознавания поз и жестов тела человека по силуэтам фигуры;

(3) разработка методов сегментации руки на произвольном фоне.

Литература

1. *Dhawale P., Masoodian M., Rogers B.* Bare-hand 3d gesture input to interactive systems // CHINZ'06: 7th ACM SIGCHI New Zealand Chapter's Conference (International) on Computer–Human Interaction: Design Centered HCI Proceedings. — New York, NY, USA: ACM, 2006. P. 25–32.
2. *Aguiar R., Pereira J. M., Braz J.* Gadevi — game development integrating tracking and visualization devices into virtools // GRAPP 2009: 4th Conference (International) on Computer Graphics Theory and Applications Proceedings. — INSTICC Press, 2009. P. 313–321.
3. *Burger T., Urankar A., Aran O., Akarun L., Caplier A.* Cued speech hand shape recognition — belief functions as a formalism to fuse svms and expert systems // VISAPP 2007: 2nd Conference (International) on Computer Vision Theory and Applications Proceedings. — INSTICC Press, 2007. Vol. 2. P. 5–12.
4. *Mitra S., Acharya T.* Gesture recognition: A survey // IEEE Trans. Syst. Man Cybernetics, Part C, 2007. Vol. 37. No. 3. P. 311–324.
5. *Garg P., Aggarwal N., Sofat S.* Vision based hand gesture recognition // World Academy Sci. Engng. Technol., 2009. P. 972–977.
6. *Wang R. Y., Popovič J.* Real-time hand-tracking with a color glove // ACM Trans. Graphics, 2009. Vol. 28. No. 3.
7. *Liu T., Liang W., Jia Y.* 3d articulated hand tracking by nonparametric belief propagation on feasible configuration space // VISAPP 2008: 3rd Conference (International) on Computer Vision Theory and Applications Proceedings. — INSTICC Press, 2008. Vol. 2. P. 508–513.
8. *Hu M.-K.* Visual pattern recognition by moment invariants // IRE Trans. Information Theory, 1962. Vol. 8. No. 2. P. 179–187.
9. *Beristain A., Grana M.* A stable skeletonization for tabletop gesture recognition // Computational science and its applications — ICCSA 2010 / Eds. D. Taniar, O. Gervasi, B. Murgante, E. Pardede, B. Apduhan. — Berlin/Heidelberg: Springer, 2010. Lecture notes in computer science ser. Vol. 6016. P. 610–621.
10. *Местецкий Л. М.* Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. — М.: Физматлит, 2009.
11. *Mestetskiy L.* Skeleton representation based on compound Bezier curves // VISAPP 2010: 5th Conference (International) on Computer Vision Theory and Applications Proceedings. — INSTICC Press, 2010. Vol. 1.
12. *Pillow N., Utcke S., Zisserman A.* Viewpoint-invariant representation of generalized cylinders using the symmetry set // Conference on British Machine Vision

Proceedings. — Surrey, UK: BMVA Press, 1994. Vol. 2. P. 539–548.

13. *Mestetskiy L., Tsiskaridze A.* Spatial reconstruction of locally symmetric objects based on stereo mate images // VISAPP 2009: 4th Conference (International) on Computer Vision Theory and Applications Proceedings. — INSTICC Press, 2009. Vol. 1. P. 443–448.
14. *Цискаридзе А. К.* Математическая модель и метод восстановления позы человека по стереопаре силуэтных изображений // Информатика и её применения, 2010. Т. 4. Вып. 4. С. 26–32.
15. *Hartley R. I., Zisserman A.* Multiple view geometry in computer vision. — 2nd ed. — Cambridge University Press, 2004.
16. *Vezhnevets V., Sazonov V., Andreeva A.* A survey on pixel-based skin color detection techniques // GraphiCon Proceedings, 2003. P. 85–92.
17. *Phung S. L., Bouzerdoum A., Chai D.* Skin segmentation using color pixel classification: Analysis and comparison // IEEE Trans. Pattern Anal. Mach. Intell, 2005. Vol. 27. P. 148–154.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 122–131

КОМБИНИРОВАННЫЙ ПОДХОД К ЛОКАЛИЗАЦИИ РАЗЛИЧИЙ МНОГОМОДАЛЬНЫХ ИЗОБРАЖЕНИЙ*

Д. М. Мурашов[1]

Аннотация: Предложен подход к решению задачи локализации различий изображений, полученных в разных спектральных диапазонах. Подход основан на применении специфических для конкретной прикладной задачи детекторов объектов на исследуемых изображениях и теоретико-информационных мер различия изображений. В качестве локальной меры различия изображений используется условная энтропия. Рассмотрено применение предложенного подхода к решению задачи локализации областей с нарушенным авторским лакокрасочным слоем на изображениях произведений живописи в видимом и ультрафиолетовом (УФ) диапазоне.

Ключевые слова: многомодальные изображения; мера различия изображений; теоретико-информационная мера; условная энтропия; изображения произведений живописи

1 Введение

Рассматривается задача анализа изображений, полученных в различных спектральных диапазонах.

В ряде задач, например поиска дефектов на изображениях негативов, зафиксированных в технике тройной цветной фотографии начала XX в. [1, 2], или выявления нарушений авторской живописи на произведениях изобразительного искусства, необходимо найти объекты, видимые только на одном из анализируемых изображений.

Многомодальные изображения широко применяются в музейной практике для целей реставрации и атрибуции. Важным аспектом исследований таких изображений является поиск невидимой для человеческого глаза, но важной для специалистов информации с использованием комбинирования изображений, зафиксированных в УФ, инфракрасном (ИК), рентгеновском и видимом спектральных диапазонах [3]. Исследования с ИК излучением проявляют углеродсодержащие красители (наброски, сделанные углем, чернилами). Ультрафиолетовое излучение позволяет увидеть участки, ранее подвергавшиеся реставрации (рис. 1), подлаковые загрязнения и ряд других дефектов красочного слоя. Тяжелые металлы не пропускают рентгеновские лучи, что позволяет видеть красочные слои, выполненные, например, свинцовыми белилами [4].

Для формирования задания на реставрационные работы необходимо выделить области с нарушенной авторской живописью на изображении в УФ диапазоне и обозначить контуры выделенных областей на цифровой фотографии в видимом

(*а*) (*б*)

Рис. 1 Изображения фрагментов портрета в видимом (*а*) и УФ (*б*) спектральных диапазонах. Картина хранится в Государственном историческом музее, г. Москва

спектральном диапазоне. Такая задача относится к классу задач выявления различий на изображениях.

Для сравнения пары изображений необходимо иметь меру, которая позволила бы получить количественную характеристику различия сравниваемых изображений. Особенности многомодальных изображений исключают возможность использования меры на основе попиксельной разности значений серого тона.

Для достижения цели работы предлагается использовать следующий подход: (а) вводится мера различия изображений; (б) с помощью детекторов, ориентированных на определенные классы объек-

* Работа выполнена при финансовой поддержке РФФИ, проект № 09-07-00368.

[1] Вычислительный центр им. А. А. Дородницына Российской академии наук, d_murashov@mail.ru

тов, на одном из изображений, где проявляются искомые объекты, выделяются области интереса; (в) по величине меры различия из найденных на этапе (б) областей выбираются области, соответствующие решаемой задаче.

Задача поиска различий на изображениях может быть сформулирована следующим образом. Предполагается, что имеются два зафиксированных в разных диапазонах спектра изображения сцены U и V размером $m \times n$ с K и L градациями полутонов соответственно. Предполагается, что изображения совмещены. Пусть на U имеется K объектов O_k^U, $k = 1, \ldots, K$, а на изображении V имеется L объектов O_l^V, $l = 1, \ldots, L$. Объекты O_k^U и O_l^V являются связными множествами пикселов, характеризуемых k-м и l-м уровнем полутонов соответственно. При наложении изображений U и V J объектов совпадают: $O_k^U \cap O_l^V = O_k^U = O_l^V$ для J пар k и l, $J \leq K$, $J \leq L$. Требуется локализовать объекты O_i^U, видимые на изображении U и отсутствующие на V и удовлетворяющие условию $\mu(x, y) > t$, где $\mu(x, y)$ — величина, характеризующая различия изображений U и V в точке $(x, y) \in O_i^U$, t — константа.

2 Современное состояние проблемы

В литературе имеется ряд публикаций по методам поиска различий на сериях изображений при решении различных задач.

В работе [5] представлен метод автоматизированного поиска скрытой информации по фотографии картины и ее рентгенограмме. Выявление невидимых глазу объектов на рентгеновском снимке осуществляется сравнением описаний пары изображений, построенных с помощью двух типов иерархических моделей: модели изображений и модели соответствия деталей изображений. На верхнем уровне изображения сегментируются на области, однородные по яркости. На нижнем уровне каждая из выделенных областей раскладывается на текстурную компоненту и компоненту среднего значения яркости. Анализируются перепады яркости на границах однородных областей, текстурные признаки. Соответствия между деталями изображений описываются линейной регрессионной моделью для текстурной составляющей и результатами сравнения компонент среднего значения яркости. Метод обладает высокой вычислительной сложностью. Выполняется сегментация и анализ выделенных объектов и краев каждого из двух сравниваемых изображений.

В работе [2] при локализации дефектов на изображениях негативов, полученных в технике тройной цветной фотографии, применяется процедура на основе логических операций над бинарными изображениями. При формировании масок дефектов отбрасываются те найденные детекторами области, которые при наложении масок трех компонент негатива дают непустое пересечение. Такой метод требует значительных вычислительных затрат на больших изображениях (поскольку приходится обнаруживать объекты на трех компонентах негатива) и не обеспечивает надежного обнаружения искомых объектов.

В работе [6] представлена программная система для сравнения изображений видимых и скрытых слоев живописи. Система предназначена для визуального анализа комбинированных ИК-рефлектограмм и цветных цифровых фотографий только в интерактивном режиме.

В работе [7] предложен метод оценивания визуальных различий изображений на основе модели зрительной системы человека. Мерой различия в точке с заданными координатами является вероятность обнаружения несовпадений двух изображений в этой точке.

На базе меры [7] в [8] вводятся модификации меры визуальных различий для последовательности многоспектральных изображений. Постановка задачи оценивания визуальных различий изображений не в полной мере соответствует решаемой в данной работе задаче, где требуется выделить объекты, видимые только на одном из предъявляемых изображений.

Ряд работ посвящен задаче сравнения изображений для оценивания визуальной различительной способности целей относительно фона [9]. Один из подходов основан на теоретико-информационных мерах, которые будут рассмотрены в следующем разделе.

3 Теоретико-информационные меры различия изображений

В ряде работ для оценивания сходства и различия изображений применяются теоретико-информационные подходы и методы. Особенностью теоретико-информационных методов является то, что они не требуют предобработки, сегментации изображений и анализа выделенных компонент. Используются непосредственно значения уровней яркости в пикселах изображения.

В работах [10, 11] и других для решения задачи совмещения изображений предложена теоретико-

информационная мера сходства модельного и преобразованного входного изображения в виде значения взаимной информации, вычисленного на этих изображениях. В работе [12] введена мера различия изображений в виде суммы их условных энтропий $H(X|Y) + H(Y|X)$, где X и Y — случайные переменные, характеризующие значения яркости в пикселах изображений. Условная энтропия $H(X|Y)$ интерпретируется как средняя информация, которая требуется для того, чтобы определить X, если известна Y. В работе [13] для оценивания качественных показателей результата комбинирования последовательностей многоспектральных изображений использовалась мера стабильности комбинирования на основе взаимной информации.

В работе [9] для решения задачи измерения визуальной различительной способности целей относительно фона разработан метод оценивания информации о цели по двум изображениям: фона и цели на том же самом фоне. Показано, что мерой различия изображений является дивергенция Кульбака–Лейблера, однако использование этой меры затруднено применительно к многомодальным изображениям, так как требуются сильные ограничения на распределения значений полутонов сравниваемых изображений.

Для использования теоретико-информационного подхода необходима вероятностная модель связи между изображениями. Пусть значения яркости на сравниваемых изображениях в точке с координатами $(x,\, y)$ описываются дискретными случайными переменными $U(x,\, y)$ и $V(x,\, y)$ со значениями u и v, квантованными на конечное число уровней K и L соответственно. Поскольку изображения U и V отображают одну и ту же сцену, то существует связь между переменными $U(x,\, y)$ и $V(x,\, y)$. Будет использоваться модель, аналогичная предложенной в [11]:

$$U\left(\mathrm{Tr}\,(x,\, y)\right) = F(V(x,\, y)) + \eta(x,\, y)\,, \qquad (1)$$

где Tr — преобразование координат (для совмещенных изображений $U\left(\mathrm{Tr}\,(x,\, y)\right) = U(x,\, y)$); F — функция преобразования яркости, моделирующая связь между двумя изображениями объекта в двух спектральных диапазонах; $\eta(x,\, y)$ — случайная переменная, моделирующая артефакты (например, нарушения лакокрасочного слоя, видимые при УФ освещении). Модель (1) можно рассматривать как модель дискретной стохастической информационной системы с входом V и выходом U.

В отличие от задачи совмещения изображений в решаемой задаче требуется мера, позволяющая выделить объекты на изображении U, отсутствующие на V. Это означает, что мера должна включать составляющие, обусловленные только объектами изображения U, которых нет на V, и не учитывать составляющие тех объектов на V, которых нет на U. Желательно, чтобы мера вычислялась на основе двумерных распределений, что позволит эффективнее использовать взаимосвязь анализируемой пары изображений.

Предлагается характеризовать отличия изображения U от V значениями условных энтропий $H(U|V)$ и $H(V|U)$. В работе [14] условная энтропия для дискретной системы определяется следующим образом:

$$H(U|V) = -\sum_{k=1}^{K}\sum_{l=1}^{L} p(u_k, v_l)\log\left[p(u_k|v_l)\right] = \\ = -\sum_{k=1}^{K}\sum_{l=1}^{L} p(u_k, v_l)\log\left[\frac{p(u_k, v_l)}{p(v_l)}\right]\,; \quad (2)$$

$$H(V|U) = -\sum_{k=1}^{K}\sum_{l=1}^{L} p(u_k, v_l)\log\left[p(v_l|u_k)\right] = \\ = -\sum_{k=1}^{K}\sum_{l=1}^{L} p(u_k, v_l)\log\left[\frac{p(u_k, v_l)}{p(u_k)}\right]\,, \quad (3)$$

где $p(u_k)$, $p(v_l)$ и $p(u_k, v_l)$ — вероятности появления уровней v_l и u_k на входе и выходе системы и их совместная вероятность; $p(u_k|v_l)$ и $p(v_l|u_k)$ — соответствующие условные вероятности.

Условная энтропия неотрицательна и аддитивна. Дополнительные условия, при которых $H(U|V)$ и $H(V|U)$ выполняют функции меры различия, дают сформулированные далее утверждения.

Утверждение 1. Условная энтропия $H(U|V)$ является мерой отличия изображения U от изображения V в тех случаях, когда

$$p(v_l) = p(u_k, v_l)\,, \quad k = 1, \ldots, K\,;\; l = 1, \ldots, L\,, \quad (4)$$

или

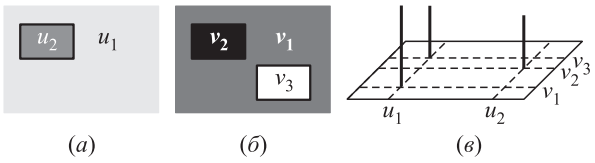
$$p(v_l) > p(u_k, v_l)\,; \quad p(u_k) = p(u_k, v_1)\,, \\ k = 1, \ldots, K\,;\; l = 1, \ldots, L\,, \quad (5)$$

где $p(u_k)$, $p(v_l)$ и $p(u_k, v_l)$ — вероятности появления уровней полутонов u_k и v_l на изображениях U и V.

Доказательство утверждения следует из выражения (2). Если имеет место условие (4), то

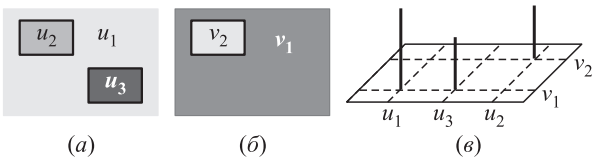
$$H(U|V) = -\sum_{k=1}^{K}\sum_{l=1}^{L} p(u_k, v_l)\log\left[\frac{p(u_k, v_l)}{p(v_l)}\right] = 0\,.$$

Это означает, что на U имеется M объектов O_k^U, каждый из которых характеризуется k-м уровнем серого, $k = 1, \ldots, M$, а на V имеется L объектов O_l^V



(*а*) (*б*) (*в*)

Рис. 2 Изображения, для которых $H(U|V) = 0$: (*а*) изображение U; (*б*) изображение V; (*в*) совместная гистограмма изображений U и V



(*а*) (*б*) (*в*)

Рис. 3 Изображения U (*а*) и V (*б*), для которых $H(U|V) > 0$; (*в*) совместная гистограмма изображений U и V

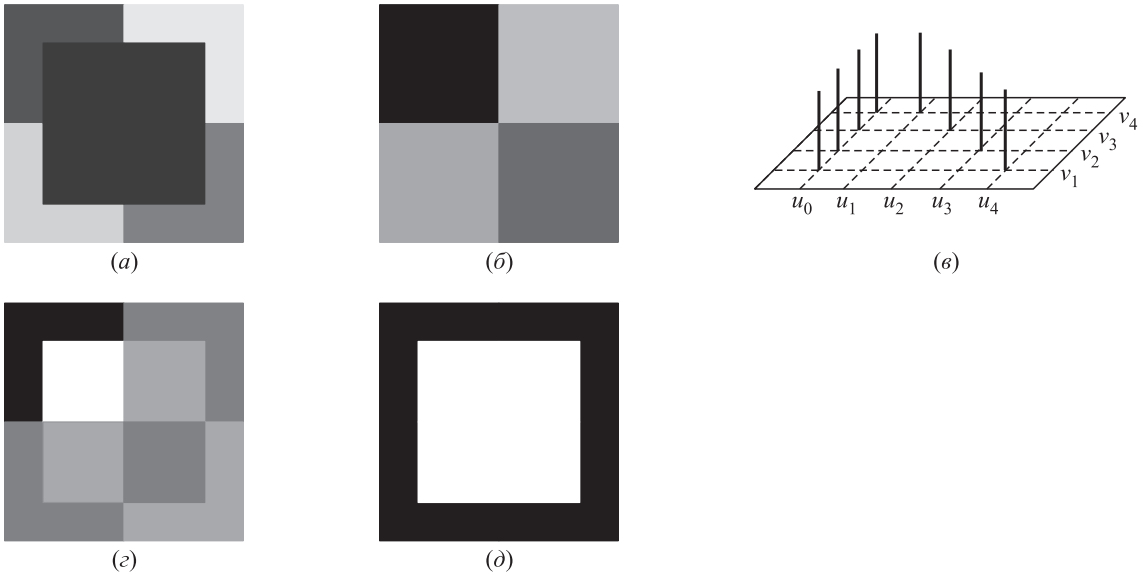
l-го уровня серого, $M \le K \le L$. Границы M объектов на U при наложении совпадают с границами M объектов на V, т. е. $O_k^U = O_l^V$, $k = l = 1, \ldots, M$. Имеются также объекты O_l^V, $O_k^U \cap O_l^V = O_l^V < O_k^U$ для некоторых $k > M$, $l > M$. Пример таких изображений показан на рис. 2, *а* и *б*, совместная гистограмма изображений U и V показана на рис. 2, *в*.

Если выполняется (5), то $H(U|V) > 0$, а величина $H(U|V)$ зависит от соотношения величин $p(u_k, v_l)$ и $p(v_l)$. Это означает, что на U имеется M объектов O_k^U, каждый из которых характеризуется k-м уровнем серого, $k = 1, \ldots, M$, а на V имеется L объектов O_l^V l-го уровня серого, $M \le L \le K$. Границы M объектов на U при наложении совпадают с границами M объектов на V, т. е. $O_k^U = O_l^V$, $k = l = 1, \ldots, M$. Кроме того, имеются объекты O_k^U такие, что $O_k^U \cap O_l^V = O_k^U < O_l^V$ для некоторых $k > M$, $l > M$. Этот случай проиллюстрирован на рис. 3.

Условие $p(u_k) = p(u_k, v_l)$ в (5) исключает составляющие объектов O_l^V на V в $H(U|V)$. Случай, когда условие $p(u_k) = p(u_k, v_l)$ в (5) не выполняется при $k = 0$, показан на рис. 4.

При решении практических задач в процессе анализа пар изображений возникает необходимость визуализации различий. В качестве индикатора различий целесообразно использовать условные энтропии $H(U|V)$ и $H(V|U)$, вычисляемые в некоторой окрестности каждого пиксела анализируемых изображений с необходимым числом уровней квантования полутонов [15]. Размеры окрестности и число уровней квантования выбираются таким образом, чтобы: (а) получить корректные оценки вероятностей; (б) обеспечить выполнение условий (4) и (5); (в) обеспечить приемлемую точность локализации различий.

Если на изображениях имеются текстурные области с большой дисперсией уровня серого тона, то не всегда удается выбором размеров окна обеспе-



(*а*) (*б*) (*в*)

(*г*) (*д*)

Рис. 4 Пример применения условной энтропии для выявления различий изображений: (*а*) и (*б*) соответственно изображения U и V; (*в*) совместная гистограмма значений полутонов; (*г*) и (*д*) визуализированные локальные значения условных энтропий $H(U|V)$ и $H(V|U)$ соответственно

(*а*) (*б*) (*в*) (*г*)

Рис. 5 Локализация различий в синем и красном каналах цветного изображения: (*а*) цветное изображение с внесенными в канал B объектами в виде темных дисков; (*б*) канал R; (*в*) канал B; (*г*) визуализированные локальные значения $H(U|V)$

Таблица 1 Характеристики областей изображения на рис. 5

Фрагмент	Среднее значение серого		Среднеквадратическое отклонение		Энтропия	
	Канал B	Канал R	Канал B	Канал R	Канал B	Канал R
Небо	198	161	14,70	11,08	5,63	5,35
Лес	29	69	18,71	29,26	5,81	6.79
Поле	28	108	28,98	39,54	5,83	7,31

чить выполнение условий (4) и (5). Так, на рис. 5 на фрагменте «поле» с помощью индикатора $H(U|V)$, где U и V — синий и красный каналы цветного изображения, не удается локализовать объекты, внесенные в синий канал. Статистические характеристики фрагментов изображения, представленного на рис. 5, приведены в табл. 1. В этом случае в качестве меры различия предлагается использовать условную энтропию $H(V|U)$. Условия, при которых $H(V|U)$ можно использовать в качестве меры различия, задаются утверждением 2.

Утверждение 2. Условная энтропия $H(V|U)$ является мерой отличия изображения U от изображения V в тех случаях, когда выполняются условия:

$$p(u_k) = p(u_k, v_l)\,, \quad k = 1, \ldots, K\,;\; l = 1, \ldots, L\,, \quad (6)$$

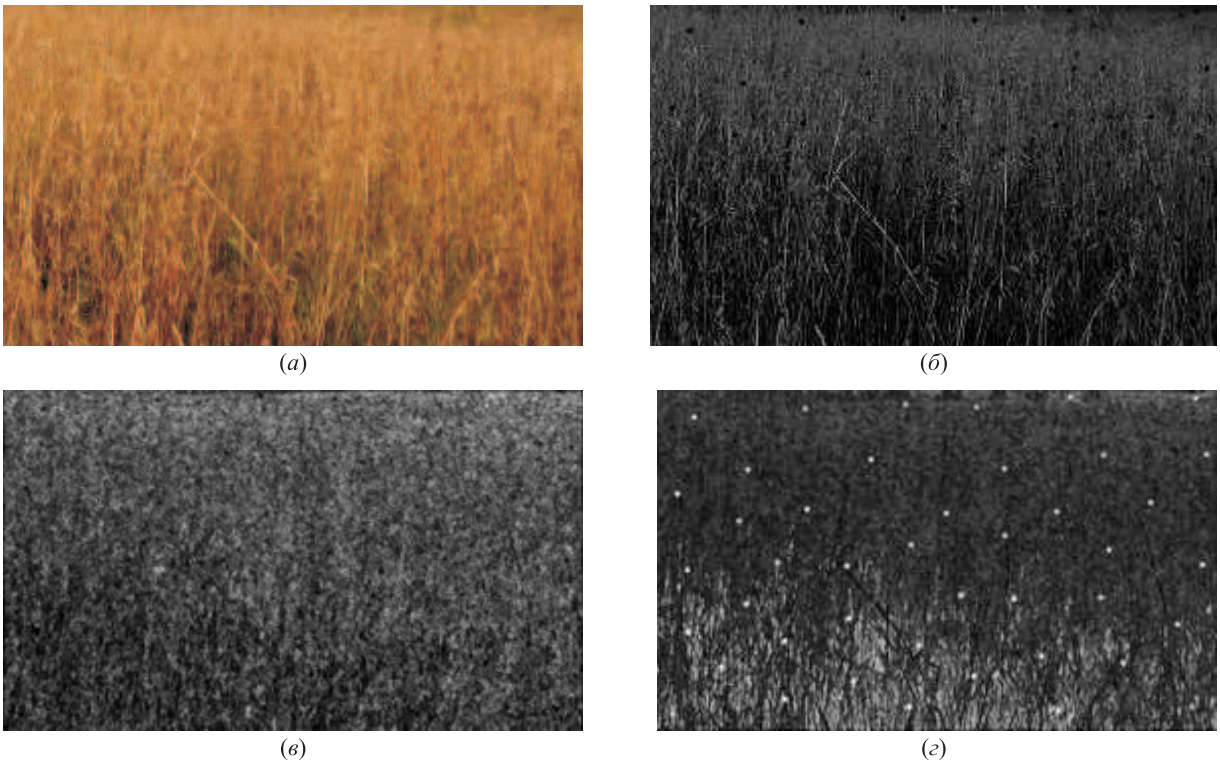
или (5) и

$$\exists u_0 :\; p(u_0) > p(u_0, v_l)\,,\; p(u_0) = \sum_{l=1}^{J} p(u_0, v_l)\,,\; l = 1, \ldots, J\,;\; J \le L\,. \quad (7)$$

Доказательство следует из выражения (3). Если имеет место условие (6), то $H(V|U) = 0$, что соответствует случаю, когда на U имеются только объекты, которые есть и на V, границы объектов при наложении изображений совпадают ($O_k^U = O_l^V$, $k = l = 1, \ldots, K$).

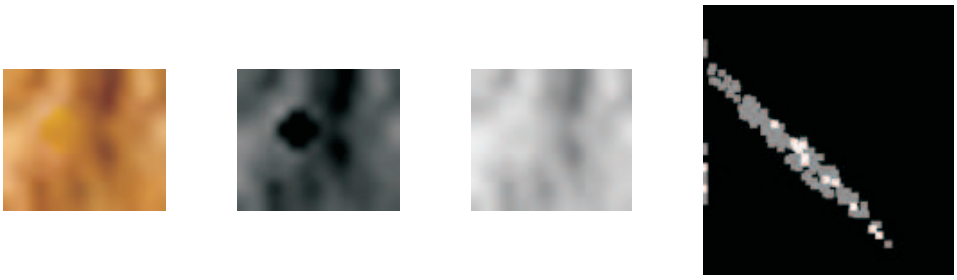
Если выполняются условия (5) и (7), то $H(V|U) > 0$. Это соответствует случаю, когда на U имеется объект O_0^U такой, что $O_0^U \cap O_l^V \neq \varnothing\; \forall\, l$, $l = 1, \ldots, K$. Условие $p(u_k) = p(u_k, v_l)$ в (5) исключает появление в $H(V|U)$ составляющей, вносимой объектами O_k^U на U.

Пример пары изображений, для которых не применима мера $H(U|V)$, но выполняются условия (6), (5) и (7) и применима мера $H(V|U)$, показан на рис. 4, *а* и *б*. Совместная гистограмма изображений показана на рис. 4, *в*. Визуализация локальных значений $H(U|V)$ и $H(V|U)$ приведена на рис. 4, *г* и *д*. Здесь оценки вероятностей получены по всем точкам изображений, а энтропии вычислялись в каждой точке отдельно.



(*а*) (*б*) (*в*) (*г*)

Рис. 6 Пример выявления артефактов в синем канале фрагмента цветной фотографии, показанной на рис. 5: (*а*) фрагмент цветного изображения с артефактами в синем канале; (*б*) синий канал фрагмента; (*в*) и (*г*) визуализированные локальные значения условных энтропий $H(U|V)$ и $H(V|U)$ соответственно



(*а*) (*б*) (*в*) (*г*)

Рис. 7 Увеличенная область с артефактом изображения, показанного на рис. 6: (*а*) цветное изображение; (*б*) и (*в*) соответственно синий и красный канал фрагмента; (*г*) совместная гистограмма синей и красной компоненты для выделенного фрагмента при 256 уровнях квантования

Пример выявления артефактов в виде дисков диаметром 8 пикселов в синем канале фрагмента цветной фотографии, показанной на рис. 5, *а*, с помощью меры различия в виде условной энтропии $H(V|U)$ показан на рис. 6.

Вероятности оценивались в окне размером 11×11 пикселов при 256 уровнях полутонов, значения условных энтропий вычислялись в окне размером 5×5. Увеличенная область, содержащая артефакт, ее цветовые компоненты и совместная гистограмма для 256 уровней полутонов показаны на рис. 7.

4 Тестирование

Для проверки эффективности применения используемых теоретико-информационных мер различия для обнаружения артефактов на реальных изображениях проведен вычислительный эксперимент.

Использовались цветные фотографии размером 988×1631 пиксел (см. рис. 5) с однородными и текстурными областями (небо, лес, поле). Характеристики областей используемого изображения приведены в табл. 1. В один из цветовых каналов изоб-

ражения внедрялось 140 объектов в виде размытых дисков диаметром от 4 до 8 пикселов или сегментов кривых толщиной от 1 до 7 пикселов.

Для поиска внедренных объектов использовались меры $H(U|V)$ и $H(V|U)$. Оценивание вероятностей производилось в окнах размером от 7×7 до 11×11, а вычисление локальных значений условных энтропий — в окнах от 3×3 до 5×5 пикселов. Использовалось 8 уровней квантования для фрагмента «небо» и 256 для фрагментов «лес» и «поле». Найдено 100% внедренных объектов в виде дисков диаметром $d = 8$ пикселов. Результаты обнаружения дисков диаметром $d = 4$ пиксела, внедренных в синий канал, представлены в табл. 2. Найдено 100% внедренных линейных объектов на однородных фрагментах и 82%–93% — на фрагментах с текстурой. Наибольшую трудность для обнаружения представляют объекты малого размера на текстурных областях с сильными перепадами яркости («лес», «поле»).

Таблица 2 Результаты эксперимента при $d = 4$

Фрагмент	Число объектов	Найдено	
Небо	68	66	97,1%
Лес	28	27	96,4%
Поле	44	40	91,0%
Всего	140	133	95,0%

5 Применение теоретико-информационной меры различия в задаче поиска записей на изображениях произведений живописи в различных спектральных диапазонах

Рассматривается задача выявления нарушений авторской живописи (записей) на произведениях изобразительного искусства (см. рис. 1): необходимо найти объекты, видимые только на одном из анализируемых изображений.

Рассматриваются изображения в формате JPEG размером 1640×1950 пикселов глубиной 8 или 24 бита, полученные с помощью фотокамеры с CCD-матрицей. Исследуемые изображения обладают рядом особенностей, оказывающих влияние на решение задачи. Во-первых, неравномерное освещение при съемке. Во-вторых, объекты интереса — области реставрации и записи авторской живописи, области подлакового загрязнения — имеют различные яркостные профили и контрастность. В-третьих, размеры объектов могут составлять от нескольких десятков до нескольких сотен и тысяч пикселов. Форма объектов может быть самой разнообразной. В-четвертых, искомые области существенно неоднородны по яркости. Таким образом, области повреждений и вмешательства в авторский красочный слой могут быть разделены на классы по характеру проявления на изображениях и могут потребоваться разные методы для их локализации.

В соответствии с предложенным во введении подходом далее будут представлены алгоритмы детекторов для нахождения областей, соответствующих по характеристикам решаемой задаче, и с помощью описанной выше меры различия отобраны только те объекты, которые видны на УФ-изображении, но отсутствуют на фотографии в видимой части спектра.

5.1 Детекторы областей интереса

При решении рассматриваемой задачи будут использованы модели изображений и детекторов, соответствующих проявлениям записей авторской живописи.

Пусть функции $U^k = u^k(x,y)$, $(x,y) \in X$, $U^k : X \to Z^+$, $k = 1, \ldots, K$, определены в некоторой области $X \subset Z^2$ и описывают полутоновый рельеф на K изображениях сцены, зафиксированных в разных диапазонах спектра при отсутствии дефектов или результатов вмешательства в авторский красочный слой. Все анализируемые изображения предварительно совмещены, и скомпенсирована неравномерность освещенности при съемке. Пусть функции $\xi_{ij}^k = v_{ij}^k(x,y)$, $\xi_{ij}^k : X \to Z$, $i = 1, \ldots, N_j$, $j = 1, \ldots, J$, описывают рельеф дефектной области $D_{ij}^k \subset X$. Здесь i — номер дефекта, j — номер класса дефектов. Пусть отображение $\varphi_j : X \times Z \to [0, 1]$ описывает детектор дефектов класса j:

$$\varphi_j(u^k(x,y)) = 0 \,\forall\, (x,y) \in X\,, \quad (x,y) \notin D_{ij}^k\,;$$

$$\varphi_j\left(u^k(x,y) + \sum_i \sum_j \xi_{ij}^k\right) = 1 \,\forall\, (x,y) \in D_{ij}^k\,.$$

Для поиска областей интереса применяются два детектора. Первый предназначен для выделения крупных объектов и основан на операциях геодезической реконструкции полутоновых изображений [16] и понятии «бассейна» яркостного рельефа глубиной h. Детектор включает операции выделения областей, в которых яркость пикселов меньше яркости внешних относительно бассейна пикселов на величину не более чем h, и получения бинарной маски выделенных областей.

Детектор 1. Бинарная маска формируется следующим образом:

$$M_1^U = T\left(U_{\text{bas}} - U_{\text{dom}}\right), \tag{8}$$

где $T(\cdot)$ — операция пороговой бинаризации, U_{dom} — изображение найденных ярких областей («куполов» яркостного рельефа) на U:

$$U_{\text{dom}} = U - R_U^{\delta}\left(U - g\right). \tag{9}$$

Здесь $R_U^{\delta}\left(U-g\right)$ — результат операции реконструкции геодезической дилатацией маски U из маркера $U-g$, где g — наибольшая относительная высота выделяемых «куполов».

«Бассейны» с относительной глубиной h на изображении U в ультрафиолетовом спектральном диапазоне находятся как

$$U_{\text{bas}} = R_U^{\delta}\left(U+h\right) - U, \tag{10}$$

где $R_U^{\delta}\left(U+h\right)$ — операция реконструкции геодезической эрозией маски U из изображения-маркера $U+h$, которое получено из изображения U увеличением яркости на h. Операция поэлементного вычитания $U_{\text{bas}} - U_{\text{dom}}$ в (8) выполняется для повышения контрастности изображения U_{bas} и повышения точности пороговой бинаризации. Бинарные маски областей интереса на УФ-изображении, полученные с помощью детектора 1 (8)–(10), показаны на рис. 8, *а*.

Второй детектор предназначен для выделения небольших фрагментов, отличающихся по уровню серого тона от окружающих областей, и основан на алгоритме локальной адаптивной пороговой бинаризации [17].

Детектор 2. Функция детектора строится следующим образом. Пусть $\overline{u}^k$ — среднее значение функции $u^k(x,y)$ в некоторой области $W \subset X$, $u_m < \overline{u}^k(x,y) < u_M$ для $(x,y) \in W$. Тогда функция детектора имеет вид:

$$\varphi(x,y) = \begin{cases} 1, & u(x,y) \geq u_M; \\ 0, & u(x,y) < u_M. \end{cases} \tag{11}$$

Здесь u_M задается в виде $u_M = \overline{u}^k(x,y) + q\sigma$, где σ — среднеквадратическое отклонение яркости, вычисленное в скользящем окне W; q — коэффициент. Изображение маски дефектов формируется в виде

$$M_2^U(x,y) = \varphi(x,y). \tag{12}$$

Полученные с помощью детектора 2, заданного выражениями (11) и (12), бинарные маски областей интереса на УФ-изображении показаны на рис. 8, *б*.

Однако не все выделенные объекты на бинарной маске соответствуют искомым объектам. Необходимо отобрать только те объекты, которые видны на УФ-изображении и не видны на фотографии в видимом диапазоне. Для выявления указанных выше различий будут использоваться меры различия, описанные в предыдущих разделах.

(*а*) (*б*)

Рис. 8 Изображения бинарных масок M_1^U (*а*) и M_2^U (*б*) областей интереса на УФ-изображении

5.2 Отбор объектов

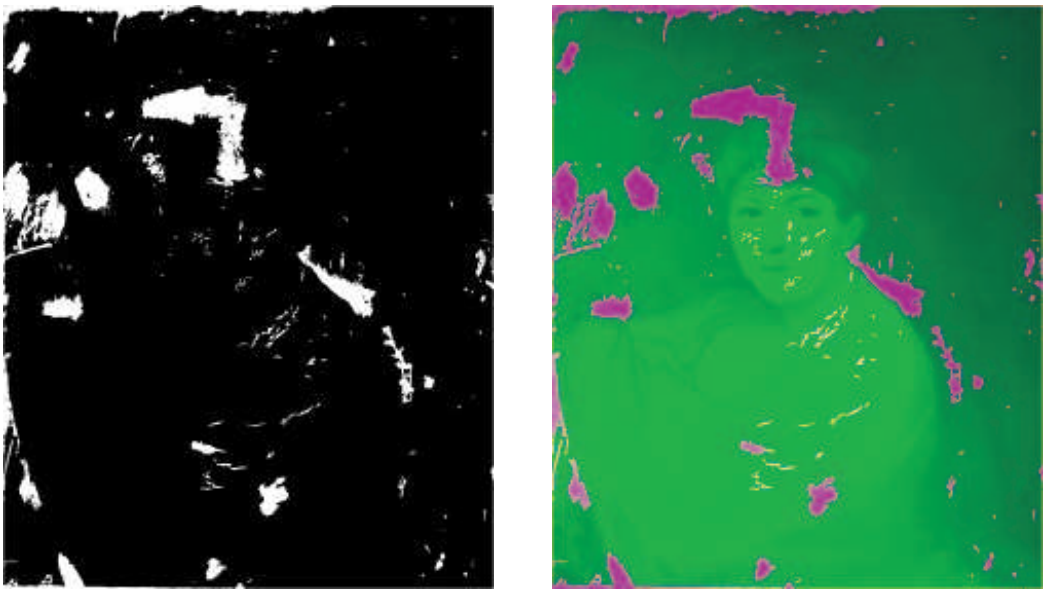
В данном подразделе по значениям используемой меры различия с изображений, показанных на рис. 8, будут отобраны объекты, соответствующие решаемой задаче.

Изображения величин условных энтропий (2) и (3) для полутоновых изображений, полученных из изображений рис. 1, показаны на рис. 9. Вероятности оценивались в окне размером 11×11 при 32 уровнях квантования, а значения $H(U|V)$ и $H(V|U)$ вычислялись в окрестности 3×3. На изображении $H(U|V)$ видны контуры объектов, отсутствующих на изображении V.

Для получения маркеров искомых объектов выполняется следующая операция:

(*а*) (*б*)

Рис. 9 Изображения, построенные по локальным значениям условных энтропий $H(U|V)$ (*а*) и $H(V|U)$ (*б*), вычисленных для полутоновых версий изображений с рис. 1



(*а*) (*б*)

Рис. 10 Результирующая маска дефектов, видимых на УФ-изображении, (*а*) и комбинация маски и цифровой фотографии картины (*б*)

$$M^{U|V}(x,y) = T\left(H\left(U|V\right) \bullet H(U) - H(V|U)\right) ,$$

где $T(\cdot)$ — операция пороговой бинаризации; $H(U)$ — изображение энтропии выхода системы (1); «$\bullet$» — операция поэлементного умножения изображений. Операции поэлементного умножения и вычитания применяются для повышения контраста и улучшения качества пороговой бинаризации.

Тогда искомое изображение дефектов, проявляемых в УФ диапазоне, будет найдено с помощью морфологической реконструкции комбинации бинарных масок (8) и (12) (см. рис. 8):

$$M(U,V) = R^{\delta}_{M^U}\left(M^{U|V}\right) , \qquad (13)$$

где $M(U) = M_1^U \vee M_2^U$, «$\vee$» — операция поэлементного логического «ИЛИ». Изображения результирующей бинарной маски (13) искомых объектов и маски, наложенной на изображение в видимом диапазоне, показаны на рис. 10.

6 Выводы

Предложен комбинированный подход к локализации различий многомодальных изображений. Подход заключается в использовании детекторов, соответствующих специфике решаемой задачи, и выборе из множества найденных объектов подмножества, удовлетворяющего заданным значениям меры различия. Предложено использовать в качестве меры различий условную энтропию, вычисляемую по значениям уровней полутонов сравниваемых изображений. Сформулированы условия, при которых значения условной энтропии характеризуют локальные отличия изображений. Проведенный вычислительный эксперимент показал эффективность используемой теоретико-информационной меры.

Разработанный подход использован для локализации записей авторского лакокрасочного слоя на изображениях произведений живописи. Объекты, соответствующие по величине яркости участкам вмешательства в авторскую живопись, находятся детекторами на УФ-изображении на основе операций полутоновой морфологической реконструкции и пороговой бинаризации с определением порога по локальным характеристикам. На основе предложенной меры производится отбор найденных бинарных объектов.

Предложенный подход позволил выполнять детектирование объектов только на одном из анализируемых изображений, в отличие от подходов, представленных в [2, 5].

Литература

1. *Wagner J.* Die additive Dreifarbenfotografie nach A. Miethe — Untersuchung des Verfahrens und Wege zur Wiedergabe von Dreifarbendiapositiven, Diplomarbeit. — TU München, 2006.
2. *Minakhin V., Murashov D., Davidov Yu., Dimentman D.* Compensation for local defects in an image created using a triple-color photo technique // Pattern Recognition Image Analysis: Advances Math. Theory Applications, 2009. Vol. 19. No. 1. P. 137–158.
3. *Kirsh A., Levenson R. S.* Seeing through paintings: Physical examination in art historical studies. — Yale: Yale U. Press, 2000.
4. *Иванова Е. Ю., Постернак О. П.* Техника реставрации станковой масляной живописи. — М.: ИНДРИК, 2005.
5. *Heitz F., Maitre H., de Couessin C.* Event detection in multisource imaging: Application to fine arts painting analysis // IEEE Trans. Acoustics Speech Signal Processing, 1990. Vol. 38. No. 1. P. 695–704.
6. *Kammerer P., Hanbury A., Zolda E.* A visualization tool for comparing paintings and their underdrawings // Conference on Electronic Imaging and the Visual Arts (EVA 2004) Proceedings. — Florence, Italy, 2004. P. 148–153.
7. *Daly S.* The visible differences predictor: An algorithm for the assessment of image fidelity. Digital images and human vision. — Cambridge: MIT Press, 1993.
8. *Petrovic V., Xydeas C.* Evaluation of image fusion performance with visible differences // ECCV'2004, LNCS, 2004. Vol. 3023. P. 380–391.
9. *Garcia J. A., Fdez-Valdivia J., Fdez-Vidal X. R., Rodriguez-Sanchez R.* Information theoretic measure for visual target distinctness // IEEE Trans. Pattern Analysis Machine Intelligence, 2001. Vol. 23. No. 4. P. 362–383.
10. *Viola P.* Alignment by maximization of mutual information. Ph.D. Thesis. — Cambridge, MA: MIT, 1995.
11. *Escolano F., Suau P., Bonev B.* Information theory in computer vision and pattern recognition. — London: Springer-Verlag, 2009.
12. *Zhang J., Rangarajan A.* Affine image registration using a new information metric // IEEE Computer Vision and Pattern Recognition (CVPR), 2004. Vol. 1. P. 848–855.
13. *Rockinger O., Fechner T.* Pixel-level image fusion: The case of image sequences // SPIE Proceedings, 1998. Vol. 3374. P. 378–388.
14. *Gallager R. G.* Information theory and reliable communication. — New York: J. Wiley Inc., 1968.
15. *Rajwade A., Banerjee A., Rangarajan A.* Continuous image representations avoid the histogram binning problem in mutual information based image registration // IEEE Symposium (International) on Biomedical Imaging (ISBI) Proceedings, 2006. P. 840–843.
16. *Soille P.* Morphological image analysis: Principles and applications. — Berlin: Springer-Verlag, 2004.
17. *Niblack W.* An introduction to digital image processing. — Englewood Cliffs, NJ: Prentice Hall, 1986.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2012. Т. 6. Вып. 1. С. 132–140

АЛГОРИТМЫ ЗАЩИЩЕННОЙ БИОМЕТРИЧЕСКОЙ ВЕРИФИКАЦИИ НА ОСНОВЕ БИНАРНОГО ПРЕДСТАВЛЕНИЯ ТОПОЛОГИИ ОТПЕЧАТКОВ ПАЛЬЦЕВ*

О. С. Ушмаев[1], В. В. Кузнецов[2]

Аннотация: Рассмотрена задача, относящаяся к проблеме совмещения биометрической верификации по отпечаткам пальцев и криптографических конструкций. Основой для такого совмещения является алгоритм извлечения достаточно длинной устойчивой бинарной строки из изображения отпечатка пальца. Предложен алгоритм извлечения бинарной строки из отпечатков пальцев на основе топологической связанности контрольных точек отпечатка. Каждая контрольная точка характеризуется ближайшими папиллярными линиями. При прослеживании папиллярной линии встречаются «события»: контрольные точки или их проекции. Эти события индексируются, что позволяет описывать окрестность любой точки бинарным вектором длиной 50–100 бит. Для извлечения более длинных векторов предложено два метода. Первый метод не требует взаимного выравнивания отпечатков, в то время как второй использует для выравнивания открытый хелпер. Таким образом, удается добиться построения векторов длиной 384–756 бит. Полученные векторы содержат примерно 20% ошибок, для исправления которых используется двухслойное кодирование (Боуза–Чоудхури–Хоквингема (БЧХ) и репликация). Эксперименты с использованием базы FVC2002 DB1 показали, что возможно построение вектора с энтропией 20–40 бит с 90-процентной вероятностью успешной идентификации.

Ключевые слова: защищенная биометрическая верификация; нечеткий экстрактор; отпечатки пальцев

1 Введение

Одной из важных теоретических проблем защиты информации является совмещение криптографических методов и биометрической верификации личности. При этом такое совмещение имеет два основных приложения: инкорпорирование механизмов верификации в криптографические конструкции (биометрическая криптография) или защита процесса верификации (защищенная верификация).

Первая работа по этой тематике была опубликована в 1991 г. [1], и с тех пор данное направление активно развивается. На сегодняшний день в рамках этой задачи можно определить несколько универсальных подходов, которые могут с той или иной степенью успешности применяться ко многим биометрическим данным.

Первый подход, называемый биохешированием [2], заключается в том, что для защиты биометрических шаблонов используются специальные необратимые функции, переводящие шаблоны одного пользователя в идентичные образы (биохеши). При этом на общей совокупности пользователей функции близки к случайному шуму. Верификация личности производится путем сравнения биохешей.

Второй распространенный подход — протоколы с нулевым разглашением (*zero-knowledge protocols*) [3]. В них используются свойства гомоморфных алгоритмов шифрования для кодирования биометрических шаблонов и их последующего сравнения в зашифрованном виде. Эти протоколы применимы только для биометрических характеристик, представимых в виде бинарных векторов, которые можно сравнивать в метрике Хемминга, как, например, биометрия лица или радужной оболочки глаза.

Третий, наиболее распространенный, подход — «нечеткие экстракторы» [4]. Он является более слабой версией первых двух подходов. При реализации нечетких экстракторов защита биометрических данных пользователя достигается за счет того, что биометрический шаблон шифруется на этапе регистрации и все последующие операции с ним проводятся без промежуточной расшифровки. Отличительной особенностью метода является использование дополнительной открытой информации, называемой открытым хелпером (*public helper*). Нечеткие экстракторы обычно строятся для био-

*Работы выполнена при поддержке программы «Инфотекс Академия 2011» и гранта Президента РФ МД.72-2011.9.

[1]Институт проблем информатики Российской академии наук, oushmaev@ipiran.ru

[2]Институт проблем информатики Российской академии наук, k.v.net@rambler.ru

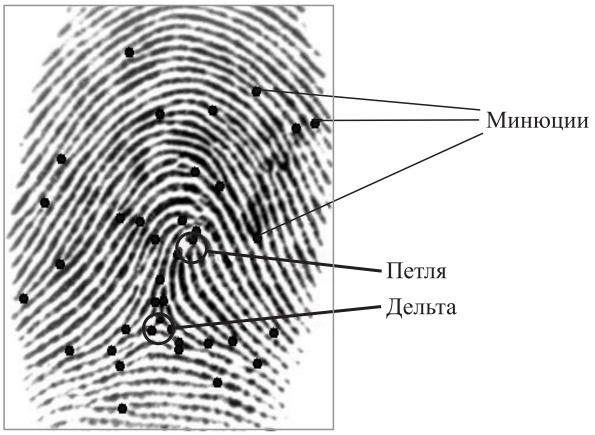


Рис. 1 Контрольные точки папиллярного рисунка

метрических характеристик, естественным образом представимых в виде бинарной строки. К таковым относятся радужная оболочка глаза, голос, лицо. Для биометрии отпечатков пальцев этот подход используется редко.

Несмотря на существование универсальных методов защищенной биометрической верификации [3, 4], в большинстве из опубликованных на данный момент эффективных решений используется специфика конкретных биометрических характеристик. В настоящей статье будет рассматриваться биометрия отпечатков пальцев.

Отпечатки пальцев чаще других биометрических характеристик используются для точной идентификации или верификации личности. Однако общепринятое представление шаблона отпечатка в виде неупорядоченного набора контрольных точек (разветвлений и окончаний папиллярных линий, рис. 1) является неудобным для задач защищенной верификации. Сдвиги и повороты изображения, исчезновение и мутации (из окончания в разветвление и обратно) контрольных точек — все это делает реализацию защищенной верификации с использованием универсальных методов крайне неэффективной.

В литературе представлено несколько специализированных методов защищенной верификации по отпечаткам пальцев. Наиболее известен метод, называемый «нечетким хранилищем» (*fuzzy vault*) [5]. Координаты контрольных точек рассматриваются как точки, лежащие на графике полинома. Его коэффициенты формируют секретный ключ, которым «закрывается» хранилище. В хранилище помещается бинарный ключ, закодированный с использованием помехоустойчивого кодирования. На этапе верификации закрытый в хранилище ключ может быть восстановлен однозначно, если предъявленный набор минюций существенным образом перекрывается с исходным. В частности, ключ длиной 69 бит успешно восстанавливался с вероятностью ошибки первого рода в 30%. Также следует отметить несколько работ [6–12], описывающих построение нечетких экстракторов для отпечатков пальцев.

В настоящей работе рассматривается метод защищенной биометрической верификации, основанный на детальном исследовании топологии потока папиллярных линий. Основная идея, лежащая в основе метода, состоит в кодировании топологических отношений между минюциями (как, например, соседство двух минюций на одной линии) в виде бинарного характеристического вектора. Затем этот характеристический вектор используется для реализации нечеткого экстрактора.

2 Схема защищенной идентификации

В настоящей работе предлагается использовать алгоритм защищенной биометрической верификации, основанный на статье [13]. Алгоритм верификации можно разделить на две функции: регистрацию пользовательской биометрии и верификацию.

Допустим, что имеется возможность извлекать из образцов отпечатков пальцев бинарные строки (называемые характеристическими векторами) достаточно большой длины (несколько сотен бит). Тогда на этапе регистрации пользователю выдается случайный ключ k, который является случайным бинарным вектором. Затем этот вектор с использованием помехоустойчивого кодирования преобразуется в строку $K = \mathrm{Encode}\,(k)$ такой же длины, как и характеристический вектор. Эта строка посредством исключающего ИЛИ складывается с характеристическим вектором DH_{ref}, извлеченным из предъявленного при регистрации отпечатка пальца:

$$T = DH_{\mathrm{ref}} \oplus K\,.$$

Бинарный вектор T, а также $H(k)$ — хеш-функция ключа k — сохраняются в качестве открытого шаблона (хелпера). Сразу следует отметить, что биометрические данные являются достаточно шумным объектом. Поэтому характеристические векторы отличаются в различных предъявлениях в достаточно большом числе битов.

На этапе верификации восстанавливается ключ k с использованием открытого хелпера и предъявленного отпечатка пальца. В соответствии с используемым алгоритмом строится соответствующий характеристический вектор DH_{sam} для

предъявленного отпечатка пальца, который затем исключающим ИЛИ складывается со строкой T из открытого хелпера:

$$K' = GH_{\text{sam}} \oplus T = K \oplus \text{err}\,, \qquad (1)$$

где err — ошибка, т. е. разница между DH_{ref} и DH_{sam}. Она будет относительно мала при предъявлении своего и велика при предъявлении чужого отпечатка пальца.

Теперь можно попытаться восстановить ключевую информацию посредством декодирования:

$$k' = \text{Decode}\,(K')\ .$$

Если хеш $H(k')$ декодированного ключа совпадает с вычисленным при регистрации хешем $H(k)$, то делается вывод, что ключ восстановлен корректно (соответственно, личность верифицирована), иначе делается заключение о некорректности ключа k' (соответственно, личность не верифицирована).

Для реализации такой схемы требуется, во-первых, синтезировать алгоритм извлечения достаточно длинного бинарного вектора из изображения отпечатка пальца таким образом, чтобы расстояние Хемминга для разных предъявлений одного отпечатка было статистически меньше расстояния Хемминга для разных отпечатков пальцев. Во-вторых, для конкретной реализации алгоритма извлечения бинарного вектора необходимо подобрать помехоустойчивые коды.

Далее статья организована следующим образом. В разд. 3 описывается топология отпечатка и метод ее кодирования. Раздел 4 содержит алгоритмы преобразования отпечатка пальца в бинарный вектор без использования дополнительной информации. В разд. 5 предложен алгоритм извлечения бинарного вектора с опубликованием дополнительной информации. Эксплуатационные характеристики предложенных алгоритмов представлены в разд. 6.

3 Построение топологических дескрипторов

3.1 Топологическая модель отпечатка пальца

Изображение отпечатка пальца представляет собой поток папиллярных линий. Ветвления и окончания папиллярных линий называются контрольными точками, или минуциями. Набор минуций с атрибутами (разветвление или окончание, направление потока папиллярных линий) образует стандартизированный шаблон отпечатка пальца. Набор может иметь разные представления, такие как, например, множество или граф. В данной работе рассматривается топологическое представление. Базовая идея, лежащая в основе использованной топологии, состоит в следующем. Описание каждой минуции можно дополнить топологическими дескрипторами — бинарными векторами, описывающими связь с другими минуциями через соседние папиллярные линии.

Формально это можно записать в виде отображения t, переводящего изображение I в шаблон отпечатка b:

$$t:\ I \to b\,;$$
$$b = \{x_i, y_i, D_i\}_{i=1}^{m}\ ,$$

где m — число минуций; x_i, y_i — их координаты; $D_i \in [0,\,1]^{n_d}$ — топологический дескриптор длиной n_d. Для того чтобы построить топологический дескриптор, введем понятия топологической связи и топологического события. Выберем произвольную контрольную точку. Из нее проведем отрезок перпендикулярно к потоку папиллярных линий. Точка пересечения этого отрезка с каждой из папиллярных линий делит их на два луча (рис. 2). Эти лучи называются *топологическими связями*, а количество лучей, которые пересекает отрезок с каждой стороны, — *глубиной прослеживания*.

Рис. 2 Построение топологического дескриптора

Рассмотрим каждую из связей. Найдем на связи ближайшую минуцию или проекцию минуции на соседнюю папиллярную линию; назовем их «*топологическим событием*». В патенте М. Спэрроу [14] различается восемь топологических событий, однако в работах В. Ю. Гудкова [15, 16] этот набор был расширен до четырнадцати событий, кодируемых четырьмя битами (рис. 3).

Топологические связи могут быть упорядочены однозначным образом (рис. 4). Если обозначить глубину прослеживания как d, то тогда для каждой минуции можно построить топологический дескриптор длиной $n_d = 16d + 4$ бит. Например,

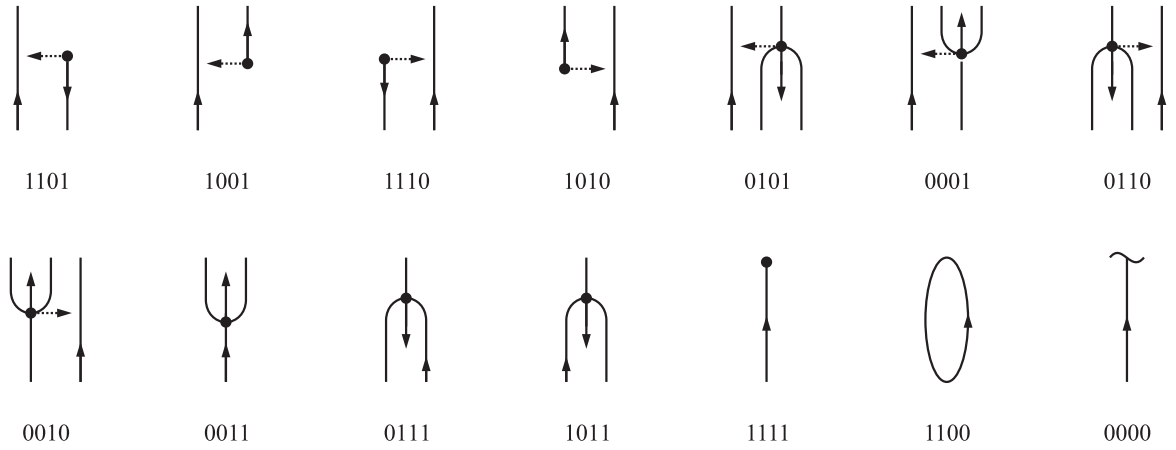


Рис. 3 Топологические события

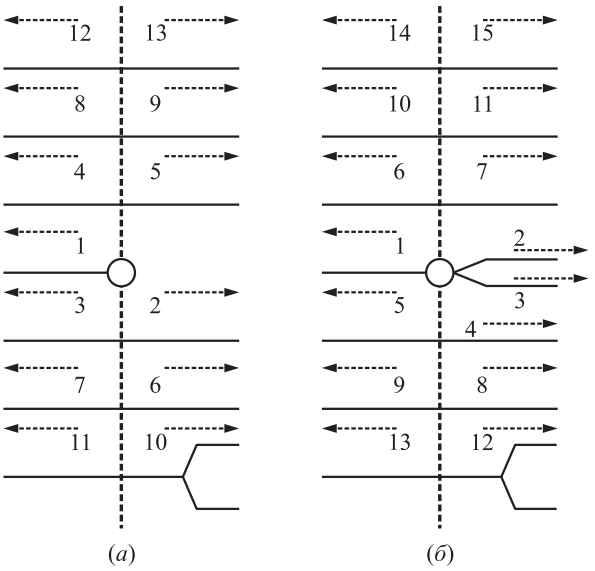


Рис. 4 Способы нумерации топологических связей

для глубины прослеживания $d = 6$ получаем 100-битное описание минюции. В среднем, отпечаток пальца содержит от 30 до 60 минюций [17], т. е. топологические дескрипторы могут иметь суммарный объем до 6 Кбит. Таким образом, можно заключить, что рассмотренная выше топология позволяет извлекать достаточно много бинарной информации.

3.2 Статистические свойства дескрипторов

Перечислим желаемые свойства дескрипторов. Во-первых, на генеральной совокупности они должны иметь равновероятное распределение битов. Во-вторых, если соответственные дескрипторы взяты из различных предъявлений одного и того же пальца (т. е. принадлежат одному человеку), то расстояние Хемминга между ними должно быть как можно ближе к 0. Если дескрипторы принадлежат разным людям, то требуется, чтобы их разница была близка к случайному шуму. Поскольку не исключается существование корреляций внутри дескрипторов, то важнейшим их параметром является энтропия. Статистические исследования настоящего раздела проводились на базах отпечатков пальцев FVC2002 DB1a [18] и собственной базе отпечатков (полученных с использованием оптического сканера с разрешением 500 dpi).

Для оценки вероятностей распределений битов анализировалось 20 000 топологических дескрипторов с глубиной прослеживания $d = 4$. Результаты приведены на рис. 5. Как видно, единичные и нулевые биты имеют практически одинаковую вероятность, равную 0,5.

Для исследования внутриклассовой вариации был проведен следующий эксперимент. Для двух шаблонов «своими» дескрипторами называются те

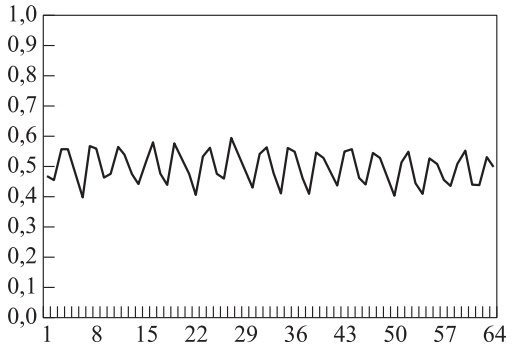


Рис. 5 Вероятности битов для каждой позиции в топологическом дескрипторе

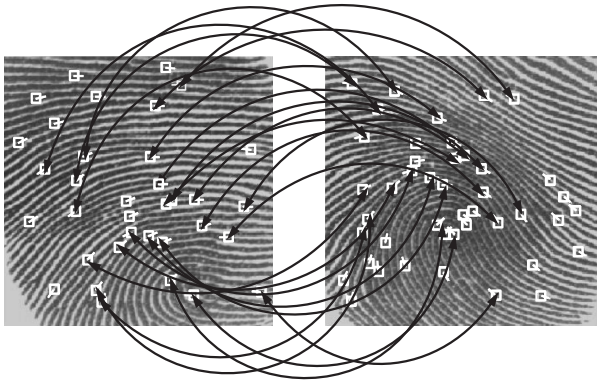


Рис. 6 Пример соответствия контрольных точек

дескрипторы, которые отвечают парам соответствующих друг другу минуций в двух предъявлениях одного отпечатка пальца. Иначе дескрипторы называются «чужими».

Примеры соответственных минуций приведены на рис. 6.

На рис. 7 приведены гистограммы расстояний для базы FVC2002 DB1.

Из представленных гистограмм видно, что для пар «своих» дескрипторов расстояние чаще всего оказывается ближе к нулю, а для «чужих» распределение достаточно близко соответствует случайному шуму.

Подробный анализ распределения расстояний в начальных 20 битах дескриптора (соответствующих глубине прослеживания $d = 1$) показывает, что оно является биномиальным распределением с 18 степенями свободы, т. е. 20-битные дескрипторы практически случайны. Для больших глубин прослеживания ($d > 2$) наблюдаются существенные корреляции и энтропия дескриптора начинает сильно отличаться от его длины. В частности, по результатам исследования FVC2002 DB1 распределение имеет 24 степени свободы для дескрипторов длиной 36 бит ($d = 2$), 30 степеней — для 52 бит ($d = 3$), 33 — для 68 ($d = 4$) и 35 — для 84 ($d = 5$).

В среднем ошибка наблюдается в 21% битов для отпечатков в FVC2002 DB1 и 20% — для собственной базы данных.

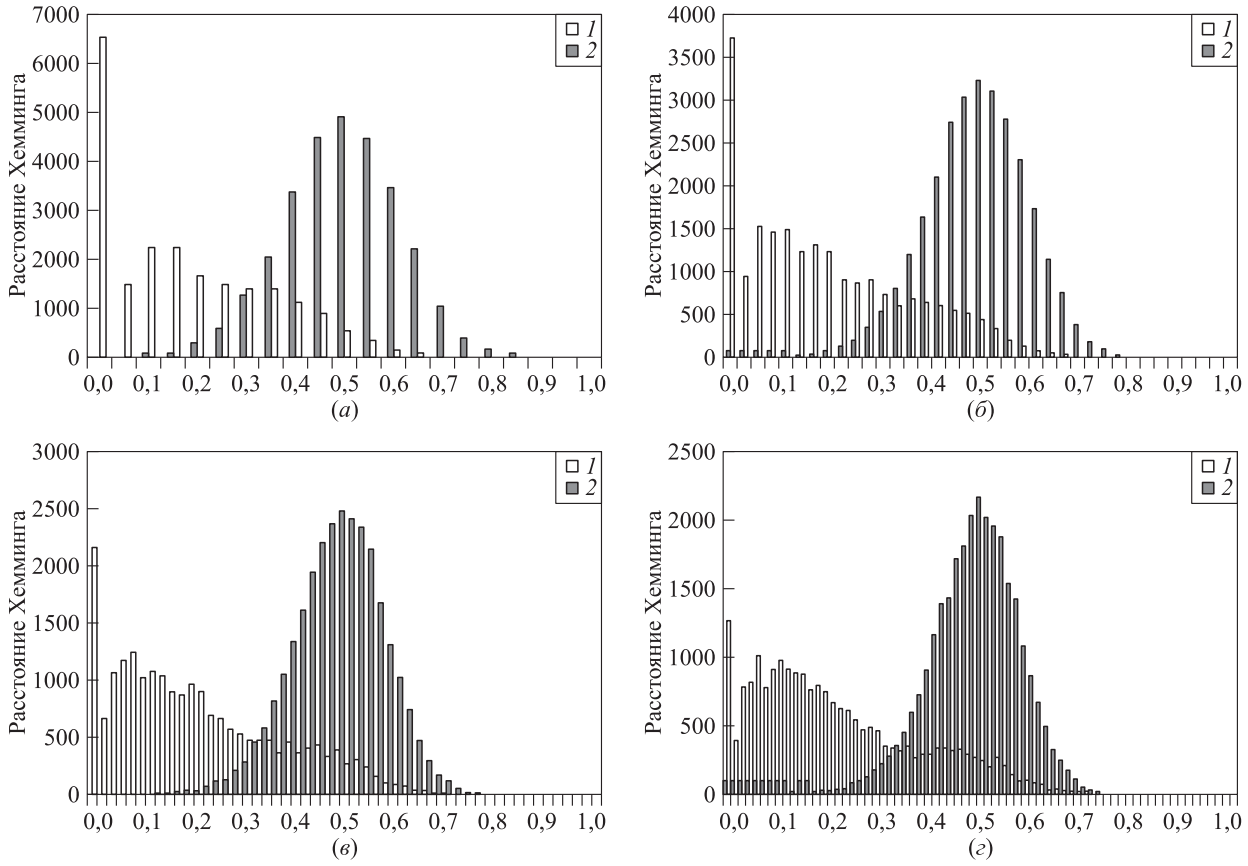


Рис. 7 Расстояния Хемминга для топологических дескрипторов соответственных (*1*) и несоответственных (*2*) минуций: (*а*) $d = 1$; (*б*) 2; (*в*) 3; (*г*) $d = 4$

4 Извлечение бинарного вектора без вспомогательной информации

В рамках данной работы были разработаны два метода извлечения бинарного характеристического вектора. Первый из них вводит внутренний порядок на множестве дескрипторов и не требует публикации дополнительной информации. Второй вводит внешний порядок за счет публикации дополнительной информации в составе открытого хелпера. Формально эти методы можно представить в виде отображений f, переводящих шаблон отпечатка пальца b в бинарный вектор DH длиной n. Рассмотрим построение первого алгоритма.

В качестве исходных данных алгоритм использует множество топологических дескрипторов, найденных на изображении отпечатка пальца. Допустим, что в пространстве значений дескрипторов $[0,\,1]^{n_d}$ выбрано 2^l центров, обозначенных через $C = \{c_j\}_{j=1}^{2^l}$, $c_j \in [0,\,1]^{n_d}$. Для каждого топологического дескриптора D_i выберем k ближайших центров, используя в качестве метрики расстояние Хемминга. Таким образом, получим отображение неупорядоченного множества дескрипторов на множество центров:

$$f_{1A} : \{x_i, y_i, D_i\}_{i=1}^{m} \to \{c_i^w\}\,, \quad i = 1, \ldots, m\,, \quad w = 1, \ldots, k\,.$$

Теперь построим характеристический вектор DH длиной $d_{DH} = 2^l$. Для всех $j \in [1 \ldots 2^l]$ положим j-й бит вектора DY равным 1 при условии, что $c_j \in \{c_i^w\}$ (т. е. хотя бы один дескриптор близок к j-му центру) и 0 в противном случае. В векторе DH будет приблизительно mk ненулевых битов. Формально этот этап можно записать в виде отображения

$$f_{1B} : \{c_i^w\} \to DH \in [0,\,1]^{2^l}\,, \quad i = 1, \ldots, m\,, \quad w = 1, \ldots, k\,.$$

Соответственно, искомое отображение f_1 будет являться композицией f_{1A} и f_{1B}.

Ключевой особенностью алгоритма является разбиение $[0,\,1]^{n_d}$ на области, характеризуемые соответствующим им множеством центров C. Если C является разреженным множеством, т. е. расстояния Хемминга между элементами велики, результирующий алгоритм будет иметь высокий уровень FAR (false accept rate). Если C плотно, то тогда вырастет FRR (false reject rate). Одним из самых очевидных способов построения C является помехоустойчивое кодирование, а именно: центры могут быть получены как результаты кодирования l бит в n_d, и в результате образуется отображение $i \to c_i$. Соотношение между c и плотностью центров e, или корректирующая способность, может быть выведено из следующего неравенства:

$$2^l \le \frac{2^{n_d}}{\sum\limits_{i=0}^{e} C_{n_d}^{e}}\,.$$

Приближенное решение этого неравенства находится из предела Шеннона:

$$l \le n_d \left(1 + \frac{e}{n_d}\log_2\left(\frac{e}{n_d}\right)\right)\,.$$

Для исправления ошибок в характеристическом векторе использовались БЧХ-коды. Проведенное тестирование показало, что характеристические векторы, получаемые с помощью предложенного алгоритма, имеют низкую энтропию.

5 Извлечение бинарного вектора со вспомогательной информацией

Альтернативным способом получения более длинного вектора из множества дескрипторов является конкатенация нескольких дескрипторов. Для этого необходимо ввести некоторый порядок на подмножестве минуций. С этой целью на этапе регистрации из шаблона выбираются случайным образом l минуций $\{p_i\}_{i=1,\ldots,l}$, которые нумеруются, и их координаты сохраняются в составе открытого хелпера. Характеристический вектор получается конкатенацией дескрипторов и используется в схеме (1).

На этапе верификации происходит поиск поворота R, сдвига S и подмножества минуций $\{q_i\}_{i=1,\ldots,l}$ предъявленного отпечатка, которые бы лучше всего соответствовали (в терминах среднеквадратичной ошибки) точкам, опубликованным в составе открытого хелпера:

$$\sum_{i=1}^{l} \left(Rq_i + S - p_i\right)^2\,.$$

Таким образом, устанавливается соответствие между p_i и q_i, т. е. отыскиваются соответственные точки в двух предъявлениях. Для каждой из найденных точек q_i построим ее топологический дескриптор D_i. Они образуют упорядоченный набор и могут быть конкатенированы:

$$DH = \overline{D_1, D_2, \ldots, D_l}\,,\ \ DH \in [0,\,1]^{ln_d}\,.$$

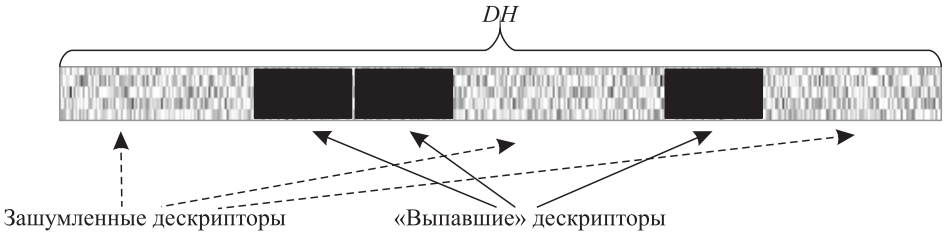


Рис. 8 Распределение ошибок в топологических векторах

Формально полученный алгоритм можно записать в виде отображения

$$f_2 : b \times \{p_i\} \to DH \in [0, 1]^{ln_d}, \quad 2 \le l \le m, \quad i = 1, \ldots, l.$$

Обозначим длину характеристического вектора как $n_{DH} = ln_d$. Тогда, к примеру, для глубины прослеживания $d = 6$ и количества точек в хелпере $l = 5$ алгоритм позволяет построить 500-битный характеристический вектор.

Стоит заметить, что данный алгоритм чувствителен к «выпадению» минуций — когда на предъявленном отпечатке пальца ключевая точка не была распознана правильно или ее предполагаемые координаты оказались за пределами информативной области отпечатка пальца. И если бороться с первым достаточно нетривиально, то для второго есть простая эвристика — уменьшить вероятность включения граничных точек в $\{p_i\}_{i=1,\ldots,l}$. Поэтому этап регистрации был доработан соответствующим образом. В случае если минуция все же оказывается выпавшей, ее топологический дескриптор заменяется нулевым.

Из формулы (1) очевидно следует, что ошибки в ключевой информации (из-за ассоциативности исключающего ИЛИ) есть в точности ошибки в векторе DH, поэтому имеется возможность проанализировать их подробнее. Ошибки в нем можно разделить на две категории: блочные (связанные с «выпадением» минуций при построении $\{q_i\}$) и шумовые. Таким образом, в результирующей бинарной строке могут быть блоки с вероятностью ошибки 20% (типичная шумовая ошибка дескрипторов, которые соответствуют верно сопоставленной паре минуций) и 50% (рис. 8). Гистограммы распределения шумовых ошибок приведены на рис. 7.

Для борьбы с подобными разнородными ошибками было применено двухслойное кодирование (рис. 9). На первом этапе ключ k длиной $|k|$ кодировался БЧХ-кодом в бинарную строку K_{BCH} длиной n_{BCH}. Реализация БЧХ, использованная в эксперименте, позволяет восстанавливать до 25% ошибок,

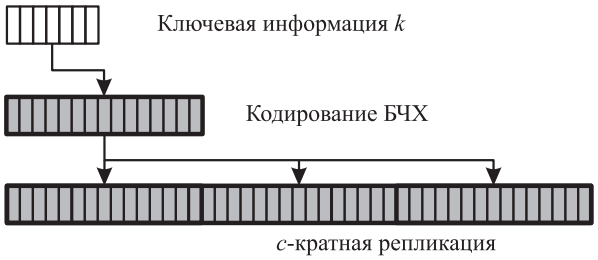


Рис. 9 Двухслойное кодирование

что само по себе недостаточно для полного восстановления ошибок с приемлемой вероятностью. Поэтому на втором этапе бинарная строка K получается путем c-кратной репликации строки K_{BCH}:

$$K = \underbrace{\overline{K_{\text{BCH}}, K_{\text{BCH}}, \ldots, K_{\text{BCH}}}}_{c}, \quad K \in [0, 1]^s, \quad s = cn_{\text{BCH}}.$$

Репликация помогает справиться с блочными ошибками. Очевидным требованием к параметрам кодирования является равенство длин K и DH, т. е. необходимо выполнение равенства $s = n_{DH}$. На этапе восстановления сначала выполняется восстановление K_{BCH}, а затем используется декодирование БЧХ-кодов.

6 Эксперименты

Эксперименты по извлечению характеристического вектора без публикации дополнительных данных проводились только для FVC2002 DB1. Эксперимент с публикацией дополнительных данных проводился на обеих базах данных.

Для каждого отпечатка проводился этап регистрации, после чего оставшиеся отпечатки с этого же пальца (по 2 для собственной базы данных и по 7 для FVC2002 DB1) использовались для верификации, что дало 5544 сравнений на TAR (true accept rate) для FVC2002 DB1 и 594 для собственной базы отпечатков. Для метода с публикацией дополнительной информации эксперимент проводился

Таблица 1 Результаты эксперимента (собственная база данных)

$\|k\|$	FAR	FRR	n_{BCH}	c	n_d	l	c
15	0,03%	12,09%	127	3	64	6	381
22	0,00%	18,35%	127	3	64	6	381
15	0,03%	11,11%	127	3	80	5	381
22	0,00%	14,08%	127	3	80	5	381
29	0,00%	16,97%	127	3	80	5	381
16	0,07%	4,67%	63	7	64	7	441
18	0,02%	5,38%	63	7	64	7	441
24	0,00%	11,54%	63	7	64	7	441
30	0,00%	14,29%	63	7	64	7	441
36	0,00%	17,97%	63	7	64	7	441
16	0,02%	3,25%	63	7	76	6	441
18	0,02%	5,42%	63	7	76	6	441
24	0,02%	12,64%	63	7	76	6	441
30	0,00%	15,16%	63	7	76	6	441
16	0,07%	3,13%	31	15	64	8	465
21	0,02%	5,88%	31	15	64	8	465
26	0,01%	10,38%	31	15	64	8	465
16	0,11%	2,89%	31	15	80	6	465
21	0,05%	5,78%	31	15	80	6	465
26	0,02%	11,19%	31	15	80	6	465
11	0,87%	1,32%	15	32	64	8	480
15	0,00%	4,91%	127	5	80	8	635
22	0,00%	7,59%	127	5	80	8	635
29	0,00%	9,82%	127	5	80	8	635
36	0,00%	17,41%	127	5	80	8	635
16	0,07%	1,34%	31	21	84	8	651
21	0,04%	2,23%	31	21	84	8	651
26	0,02%	5,80%	31	21	84	8	651
16	0,02%	1,34%	63	11	88	8	693
18	0,02%	2,23%	63	11	88	8	693
24	0,00%	4,46%	63	11	88	8	693
30	0,00%	6,70%	63	11	88	8	693
36	0,00%	7,14%	63	11	88	8	693
39	0,00%	8,93%	63	11	88	8	693
45	0,00%	11,61%	63	11	88	8	693

Таблица 2 Результаты эксперимента (FVC2002 DB1)

$\|k\|$	FAR	FRR	n_{BCH}	c	n_d	l	c
36	0,61%	10,30%	127	3	64	6	381
43	0,36%	12,60%	127	3	64	6	381
30	2,20%	2,80%	63	7	64	7	441
36	0,95%	5,20%	63	7	64	7	441
39	0,36%	9,00%	63	7	64	7	441
45	0,16%	15,70%	63	7	64	7	441

несколько раз, так как из-за случайности выбора точек имеется возможность провести несколько процедур регистрации для каждого отпечатка пальца. Для FVC2002 DB1 было выполнено два прохода, для собственной базы — десять. В табл. 1 и 2 и на рис. 10 и 11 приводятся усредненные результаты FRR и FAR.

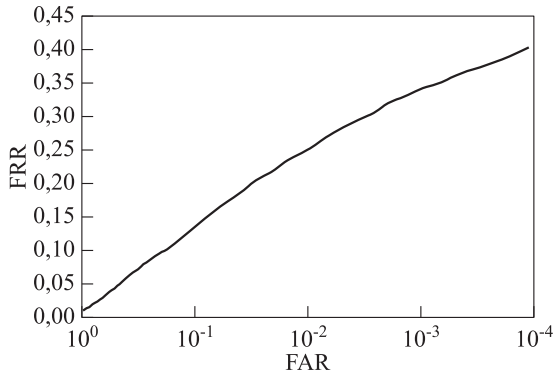


Рис. 10 Кривая DET (detection error tradeoff) для алгоритма без публикации дополнительной информации, длина БЧХ-кода 64 бита, $|k| = 16$

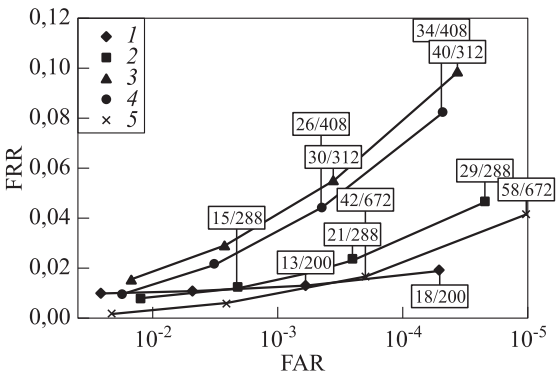


Рис. 11 Кривая DET для алгоритма с публикацией дополнительной информации для различной глубины и количества точек (на подписях к точкам указаны энтропия ключа/длина характеристического вектора): *1* — $d = 1$, $l = 10$; *2* — $d = 2$, $l = 8$; *3* — $d = 3$, $l = 6$; *4* — $d = 4$, $l = 6$; *5* — $d = 5$, $l = 8$

Как видно из графиков рис. 10 и 11, алгоритм с публикацией дополнительной информации в открытом хелпере позволяет реализовывать защищенную верификацию по отпечаткам пальцев с приемлемыми ошибками первого и второго рода.

7 Заключение

В данной статье предложены алгоритмы защищенной биометрической верификации по отпечатку пальца. Для их реализации был предложен метод извлечения повторяемого бинарного вектора из изображения отпечатка пальца. Главные преимущества реализованного метода — устойчивое кодирование окрестности контрольных точек и, как следствие, приемлемые значения ошибок алгоритмов верификации (TAR $> 0{,}9$, FAR $\leq 10^{-3}$).

Слабым местом алгоритма с частичной публикацией минуций является потенциальная утечка части пользовательского шаблона. Несмотря на то что идентифицировать личность пользователя по 5–8 случайным минуциям практически невозможно, угроза безопасности пользовательского шаблона является недостатком алгоритма. Предложенный алгоритм без публикации информации имеет значительные ошибки первого и второго рода (FRR $\geq$ $\geq$ 0,3 при FAR $= 0{,}1\%$).

В качестве направлений дальнейших исследований следует отметить совершенствование алгоритма с частичной публикацией минуций. В частности, перспективным подходом минимизации утечки представляется использование необратимого преобразования публикуемого набора минуций. Также можно увеличить энтропию топологических дескрипторов, если рассматривать не только топологические события, но и длину соответствующей топологической связи.

Литература

1. *Abraham D. G., Dolan G. M., Double G. P., Stevens J. V.* Transaction security system // IBM Syst. J., 1991. Vol. 30. No. 2. P. 206–229.
2. *Ratha N., Connell J., Bolle R.* Enhancing security and privacy in biometrics-based authentication systems // IBM Syst. J., 2001. Vol. 40. No. 3. P. 614–634.
3. *Bringer J., Chabanne H., Izabachene M., Pointcheval D., Tang Q., Zimmer S.* An application of the goldwasser-micali cryptosystem to biometric authentication // Conference (Australian) on Information Security and Privacy, ACISP 2007 Proceedings. — Berlin–Heidelberg: Springer, 2007. LNCS 4586. P. 96–106.
4. *Dodis Y., Ostrovsky R., Reyzin L., Smith A.* Fuzzy extractors: How to generate strong keys from biometrics and other noisy data // SIAM J. Computing, 2008. Vol. 38. No. 1. P. 97–139.
5. *Clancy T. C., Kiyavash N., Lin D. J.* Secure smartcard-based fingerprint authentication // ACM Workshop on Biometrics: Methods and Applications. — Berkeley, CA, Nov., 2003. P. 45–52.
6. *Tong V. V. T., Sibert H., Lecoeur J., Girault M.* Biometric fuzzy extractors made practical: A proposal based on fingercodes // Advances in biometrics. — Berlin–Heidelberg: Springer, 2007. LNCS 4642. P. 604–613.
7. *Arakala A., Jeffers J., Horadam K. J.* Fuzzy extractors for minutiae-based fingerprint authentication // Advances in biometrics. — Berlin–Heidelberg: Springer, 2007. LNCS 4642. P. 760–769.
8. *Draper S., Yedidia J., Khisti A., Martinian E., Vetro A.* Using distributed source coding to secure fingerprint biometrics // Conference (International) on Acoutics Speech Signal Processing Proceedings. — Honolulu, 2007. P. 129–132.
9. *Farooq F., Bolle R. M., Jea T.-Y., Ratha N. K.* Anonymous and revocable fingerprint recognition // IEEE Conference on Computer Vision and Pattern Recognition, CVPR-07, 2007. P. 1–7.
10. *Barni M., Bianchi T., Catalano D., Raimondo M., Labati R., Failla P., Fiore D., Lazzeretti R., Piuri V., Scotti F., Piva A.* Privacy-preserving fingercode authentication // 12th ACM Workshop on Multimedia and Security Proceedings, 2010. P. 231–240.
11. *Nagar A., Rane S. D., Vetro A.* Alignment and bit extraction for secure fingerprint biometrics // Proc. SPIE, 2010. Vol. 7541; doi:10.1117/12.839130.
12. *Bringer J., Despiegel V.* Binary feature vector fingerprint representation from minutiae vicinities // 4th IEEE Conference (International) on Biometrics: Theory, Applications and Systems (BTAS-10) Proceedings, 2010. P. 1–6.
13. *Hao F., Anderson R., Daugman J.* Combining cryptography with biometrics effectively: Technical Report UCAM-CL-TR-640. — Cambridge: University of Cambridge Computer Laboratory, 2005. 17 p.
14. *Sparrow M. K.* Vector based topological fingerprint matching. U.S. Patent 5631971. May 20, 1997.
15. *Гудков В. Ю.* Способ кодирования отпечатка папиллярного узора. Патент РФ № 2321057 от 04.12.2006.
16. *Гудков В. Ю.* Способ генерирования набора параметров ключа доступа и система для аутентификации человека по отпечаткам пальцев: Патент РФ № 2363048 от 11.10.2007.
17. *Pankanti S., Prabhakar S., Jain A. K.* On the individuality of fingerprints // IEEE Trans. PAMI, 2002. Vol. 24. No. 8. P. 1010– 1025.
18. *Maio D., Maltoni D., Cappelli R., Wayman J. L., Jain A. K.* FVC2002: Second fingerprint verification competition // 16th Conference (International) on Pattern Recognition (ICPR2002) Proceedings. — Quebec City, 2002. Vol. 3. P. 811–814.

ABSTRACTS

SKEW STUDENT DISTRIBUTIONS, VARIANCE-GAMMA DISTRIBUTIONS AND THEIR GENERALIZATIONS AS ASYMPTOTIC APPROXIMATIONS

V. Korolev[1] and I. Sokolov[2]

[1] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; IPI RAN, vkorolev@comtv.ru

[2] IPI RAN, isokolov@ipiran.ru

The paper demonstrates that skew Student distributions and (skew) variance-gamma distributions can appear as limit laws in rather simple limit theorems for regular statistics, in particular, in the scheme of random summation of random variables and, hence, can be regarded as asymptotic approximations to the distributions of many processes related to the evolution of complex systems.

Keywords: skew Student distribution; variance-gamma distribution; limit theorem; random sum; transfer theorem

MATHEMATICAL SUPPORT FOR NONLINEAR MULTICHANNEL CIRCULAR STOCHASTIC SYSTEMS ANALYSIS BASED ON DISTRIBUTION PARAMETRIZATION

I. N. Sinitsyn

IPI RAN, sinitsin@dol.ru

Theory and mathematical support for nonlinear multichannel circular stochastic systems (CStS) on distribution parametrization are given. The main topics are: circular orthogonal expansions (COE) for circular random variables and stochastic processes, stochastic equatioons for multichannel CStS, integrodifferential equations for one- and multidimensional densities, general COE method, methods of wrapped normal approximation, initial and central moments, software tools "CStS-ANALYSIS." The results are illustrated by examples.

Keywords: analytical modeling; circular random variable; circular stochastic process; coefficients of circular orthogonal expansion; "CStS-ANALYSIS;" MATLAB; nonlinear multichannel stochastic system; one- and multidimensional densities; circular orthogonal expansion; standard density; wrapped normal density

ANALYSIS AND OPTIMIZATION PROBLEMS FOR SOME USERS ACTIVITY MODEL. PART 2. INTERNAL RESOURCES OPTIMIZATION

A. V. Bosov

IPI RAN, AVBosov@ipiran.ru

The paper continues investigation of the mathematical model describing the activity of users sugessted by the author earlier. The problem of optimizing the distribution of information system internal resources based on the quadratic criterion of quality is formulated and solved. Suboptimal optimization algorithms are presented.

Keywords: information system; stochastic observation system; quadratic criterion

ON A VIRTUAL WAITING TIME IN THE QUEUEING SYSTEM WITH HEAD-OF-THE-LINE PRIORITY AND HYPEREXPONENTIAL INPUT STREAM

A. V. Ushakov

IPI RAN, grimgnau@rambler.ru

The single server queue with hyperexponential input stream, head-of-the-line priority discipline and two service disciplines, FIFO (first in, first out) and LIFO (last in, first out), for customers of one priority class is considered. The distributions of the virtual waiting time for each priority class are obtained.

Keywords: virtual waiting time; head-of-the-line priority; hyperexponential input stream

A REFINEMENT OF NONUNIFORM ESTIMATES OF THE RATE OF CONVERGENCE IN THE CENTRAL LIMIT THEOREM UNDER THE EXISTENCE OF MOMENTS OF ORDER NOT HIGHER THAN THE SECOND

S. V. Popov

Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, popovserg@yandex.ru

Nonuniform estimates of the rate of convergence in the central limit theorem for sums of independent random variables with the moments of order not higher than the second are specified.

Keywords: central limit theorem; convergence rate estimate; absolute constants

COMPUTER SYSTEM OPTIMIZATION USING SIMULATION MODEL AND ADAPTIVE ALGORITHMS

M. G. Konovalov

IPI RAN, mkonovalov@ipiran.ru

A problem of effective jobs execution control in computer system is considered. An approach based on the use of adaptive strategies in the simulation model is proposed. On an example of the specific computer system, an original methodology of simulation model construction is outlined. As adaptive strategies, the gradient algorithms are used, developed in the theory of partially observed Markov decision-making process. The results of computational experiment are presented.

Keywords: computer systems; simulation; adaptive algorithms

EXTRACTION OF IMPLICIT INFORMATION FROM THE TEXTS IN NATURAL LANGUAGE: PROBLEMS AND METHODS

I. P. Kuznetsov[1] and N. V. Somin[2]

[1]IPI RAN, igor-kuz@mtu-net.ru

[2]IPI RAN, somin@post.ru

A semantic-oriented linguistic processor, which provides deep analysis of text in natural language and forms knowledge structures, is considered. A significant direction in development of such processors is connected with the extraction from the texts the named entities, their properties, and links, presented in implicit forms. The methods of such extraction at different levels of text analysis (lexical-grammatical, syntactical-semantic and conceptual) are proposed.

Keywords: automatic text analysis; knowledge extraction; linguistic processor; implicit information

IDENTITY AND ACCESS MANAGEMENT OF THE USERS' RIGHTS IN HIGH AVAILABLE DATA CENTER

M. V. Benderina[1], S. V. Borokhov[2], V. I. Budzko[3], P. V. Stepanov[4], and A. P. Suchkov[5]

[1]IPI RAN, mbenderina@ipiran.ru
[2]IPI RAN, sborokhov@ipiran.ru
[3]IPI RAN, vbudzko@ipiran.ru
[4]IPI RAN, pvstepanov@ipiran.ru
[5]IPI RAN, asuchkov@ipiran.ru

The functional and organizational charts and principles of the identity and access management of the users' rights, developed for the two strategies to information protection, which are accepted by the organization or cloud computing community, are stated. The works organization procedure to create a centralized identity and access management system of the users' rights as a part of the information security maintenance system for high available collective data-processing centers is defined.

Keywords: high availability; information security; data-processing center

EXTENDING INFORMATION INTEGRATION TECHNOLOGIES FOR PROBLEM SOLVING OVER HETEROGENEOUS INFORMATION RESOURCES

L. A. Kalinichenko[1], S. A. Stupnikov[2], and V. N. Zakharov[3]

[1]IPI RAN, leonidk@synth.ipi.ac.ru
[2]IPI RAN, ssa@ipi.ac.ru
[3]IPI RAN, vzakharov@ipiran.ru

This position paper is an attempt to match up the emerging challenges for problem solving over heterogeneous distributed information resources. State-of-the-art in subject mediation technology reached at IPI RAN is presented. The technology is aimed at filling the widening gap between the users (applications) and heterogeneous resources of data, knowledge, and services. Also, the paper affects the semantic-based information integration technologies challenges including investigation of application-driven approach for problem solving in the subject mediator environment, a provision for support of executable declarative specifications of the applications over the mediator, enhancement of presence of knowledge-based facilities at the mediator level, and mediation of databases with nontraditional data models motivated by the need of large data support.

Keywords: subject mediation; heterogeneous information resources; scientific problem solving; information integration; application-driven approach; rule-based languages; nontraditional data models

THE MOTIF INFORMATION ANALYSIS BASED ON THE SOLVABILITY CRITERION FOR THE PROTEIN SECONDARY STRUCTURE RECOGNITION

K. V. Rudakov and I. Yu. Torshin[2]

[1]Dorodnicyn Computing Center of the Russian Academy of Sciences; Moscow Institute of Physics and Technology (State University), rudakov@ccas.ru
[2]Russian Center of the Trace Element Institute for UNESCO, tiy135@yahoo.com

The development of the formal description of recognition of protein secondary structure problem is presented. The key concepts of motif, informativity estimate of a motive, and the order of the motives, allowing to use the formalism for the analysis of actual precedent sets are introduced. The experiments on the solvability testing indicate a possibility of an efficient selection of the most informative motifs.

Keywords: algebraic approach; bioinformatics; locality; solvability; feature value classification

SPEAKER IDENTIFICATION SYSTEM FOR THE *NIST SRE 2010*

I. N. Belykh[1], A. I. Kapustin[2], A. V. Kozlov[3], A. I. Lohanova[4], Yu. N. Matveev[5], T. S. Pekhovsky[6], K. K. Simonchik[7], and A. K. Shulipa[8]

[1]Speech Technology Center, St.-Petersburg, belykh@speechpro.com
[2]Speech Technology Center, St.-Petersburg, kapustin@speechpro.com
[3]Speech Technology Center, St.-Petersburg, kozlov-a@speechpro.com
[4]Speech Technology Center, St.-Petersburg, lohanova@speechpro.com
[5]Speech Technology Center, St.-Petersburg, matveev@speechpro.com
[6]Speech Technology Center, St.-Petersburg, tim@speechpro.com
[7]Speech Technology Center, St.-Petersburg, simonchik@speechpro.com
[8]Speech Technology Center, St.-Petersburg, shulipa@speechpro.com

A description of a speaker identification system by voice is presented. This system was developed for submission on speaker recognition system evaluation at *NIST SRE 2010*.

Keywords: biometry; speaker identification; voice recognition; pitch; formants; GMM; SVM; NIST

FAST PROCESSING OF FINGERPRINT IMAGES

V. J. Gudkov[1] and M. V. Bokov[2]

[1]Chelyabinsk State University, Department of Applied Mathematics, diana@sonda.ru
[2]South Ural State University, Department of Applied Mathematics, guardian@mail.ru

A consistent method of identification of individual features from fingerprint image with the severe restrictions is briefly described. Individual features are stored in the image template. The templates are used for fingerprint identification.

Keywords: fingerprint; image processing; flow matrix; period matrix; minutiae

TEACHING OF SKIN EXTRACTION ALGORITHMS FOR HUMAN FACE COLOR IMAGES

Y. Vizilter[1], V. Gorbatcevich[2], S. Karateev[3], and N. Kostromov[4]

[1]State Research Institute of Aviation Systems (GosNIIAS), viz@gosniias.ru
[2]State Research Institute of Aviation Systems (GosNIIAS), gvs@gosniias.ru
[3]State Research Institute of Aviation Systems (GosNIIAS), goga@gosniias.ru
[4]State Research Institute of Aviation Systems (GosNIIAS)

Two methods for teaching of algorithms for the skin extraction in color images of human faces are proposed and discussed. The first method is based on self-organizing neural network called “growing neural gas.” The second one is based on morphological classification by minimal cutting of neighborhood graph for a training set in color space. The CIE Lab color space is applied for color description in both cases. The efficiency of both methods is demonstrated. The differences in selection results of the proposed methods are explored and demonstrated.

Keywords: biometrics; human skin extraction; self-organizing neural networks; morphological classification; graph cut

REAL-TIME HAND GESTURE RECOGNITION BY PLANAR AND SPATIAL SKELETAL MODELS

A. V. Kurakin

Moscow Institute of Physics and Technology (State University), alekseyvk@yandex.ru

The hand gesture recognition problem is considered. Algorithm to detect planar positions of fingertips by the image of hand silhouette is proposed. It operates with the analysis of continuous skeleton of the hand shape for silhouette segmentation. An extension of the algorithm to spatial case is presented. It uses stereo pair of hand silhouettes to estimate spatial positions of hand and fingertips. The developed methods work in real time, allowing their use in applied hand gesture recognition systems.

Keywords: continuous skeleton; shape analysis; gesture recognition; stereovision

COMBINED APPROACH TO LOCALIZATION OF DIFFERENCES FOR MULTIMODAL IMAGES

D. M. Murashov

Dorodnicyn Computing Center, Russian Academy of Sciences, d_murashov@mail.ru

An approach to the problem of localization of the differences in multimodal images is suggested. The approach is based on specific object detectors and local information-theoretical image difference measures implemented as conditional entropy of the analyzed image pair. The proposed approach is applied to the problem of detecting repainting areas of fine art paintings using the images acquired in visible and ultraviolet spectral ranges.

Keywords: multimodal images; measure of image difference; information-theoretical measure; conditional entropy; images of fine art paintings

SECURED BIOMETRIC VERIFICATION BASED ON FINGERPRINT TOPOLOGY BINARY REPRESENTATION

O. S. Ushmaev[1] and V. V. Kuznetsov[2]

[1]IPI RAN, oushmaev@ipiran.ru
[2]IPI RAN, k.v.net@rambler.ru

The paper deals with combination of cryptographic constructions and fingerprint identification. A technique of repeatable binary string extraction from fingerprint images is suggested. The binary features from topological relations between minutiae points are extracted. For an arbitrary minutiae point, the neighboring ridges are traced until the event is encountered: minutiae or projection of minutiae. Then, these events are encoded. Thus, 50–100-bit descriptions for each minutiae point are obtained. In order to extract longer binary string, two techniques are suggested. The first one is the self-aligned technique, while the second one requires public helper. Thus, 384–756-bit binary string is extracted. The strings have approximately 20% erroneous bits, which are corrected using two-layer Bose–Chaudhuri–Hocquenghem (BCH) major voting codes. The experiments on FVC2002 DB1 dataset show that 20–40-bit error-free binary string can be reproduced from genuine fingerprint with 90 percent success rate.

Keywords: secured biometric verification; fuzzy extractor; fingerprint

Об авторах

Белых Игорь Николаевич (р. 1959) — кандидат физико-математических наук, директор научно-исследовательского департамента Центра речевых технологий, г. Санкт-Петербург

Бендерина Мария Владимировна (р. 1986) — младший научный сотрудник ИПИ РАН

Боков Максим Владимирович (р. 1982) — аспирант Южно-Уральского государственного университета

Борохов Сергей Владимирович (р. 1972) — старший научный сотрудник ИПИ РАН

Босов Алексей Вячеславович (р. 1969) — доктор технических наук, заведующий сектором ИПИ РАН

Будзко Владимир Игоревич (р. 1944) — доктор технических наук, заместитель директора ИПИ РАН

Визильтер Юрий Валентинович (р. 1970) — доктор физико-математических наук, начальник лаборатории Государственного научно-исследовательского института авиационных систем (ГосНИИАС)

Горбацевич Владимир Сергеевич (р. 1985) — научный сотрудник Государственного научно-исследовательского института авиационных систем (ГосНИИАС)

Гудков Владимир Юльевич (р. 1959) — кандидат технических наук, доцент Челябинского государственного университета

Захаров Виктор Николаевич (р. 1948) — доктор технических наук, доцент, ученый секретарь ИПИ РАН

Калиниченко Леонид Андреевич (р. 1937) — доктор физико-математических наук, профессор, заслуженный деятель науки РФ, заведующий лабораторией ИПИ РАН

Капустин Алексей Игоревич (р. 1985) — научный сотрудник Центра речевых технологий, г. Санкт-Петербург

Каратеев Сергей Львович (р. 1956) — начальник сектора Государственного научно-исследовательского института авиационных систем (ГосНИИАС)

Козлов Александр Викторович (р. 1985) — программист Центра речевых технологий, г. Санкт-Петербург

Коновалов Михаил Григорьевич (р. 1950) — доктор технических наук, заведующий сектором ИПИ РАН

Королев Виктор Юрьевич (р. 1954) — доктор физико-математических наук, профессор кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; ведущий научный сотрудник ИПИ РАН

Костромов Никита Алексеевич (р. 1986) — аспирант Государственного научно-исследовательского института авиационных систем (ГосНИИАС)

Кузнецов Владислав Владимирович (р. 1988) — аспирант ИПИ РАН

Кузнецов Игорь Петрович (р. 1938) — доктор технических наук, профессор, главный научный сотрудник ИПИ РАН

Куракин Алексей Владимирович (р. 1986) — аспирант кафедры «Интеллектуальные системы» факультета управления и прикладной математики Московского физико-технического института

Лоханова Александра Ивановна (р. 1985) — научный сотрудник Центра речевых технологий, г. Санкт-Петербург

Матвеев Юрий Николаевич (р. 1955) — доктор технических наук, руководитель отдела Центра речевых технологий, г. Санкт-Петербург

Мурашов Дмитрий Михайлович (р. 1958) — кандидат технических наук, старший научный сотрудник Вычислительного центра им. А. А. Дородницына РАН

Пеховский Тимур Сахиевич (р. 1959) — кандидат физико-математических наук, ведущий научный сотрудник Центра речевых технологий, г. Санкт-Петербург

Попов Сергей Владимирович (р. 1986) — аспирант кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

Рудаков Константин Владимирович (р. 1954) — доктор физико-математических наук, член-корреспондент РАН, заведующий отделом вычислительных методов прогнозирования Вычислительного центра им. А. А. Дородницына РАН, заведующий кафедрой «Интеллектуальные системы» Московского физико-технического института

Симончик Константин Константинович (р. 1983) — научный сотрудник Центра речевых технологий, г. Санкт-Петербург

Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий отделом ИПИ РАН

Соколов Игорь Анатольевич (р. 1954) — академик (действительный член) Российской академии наук, доктор технических наук, директор ИПИ РАН

Сомин Николай Владимирович (р. 1947) — кандидат физико-математических наук, ведущий научный сотрудник ИПИ РАН

Степанов Павел Владимирович (р. 1957) — доктор технических наук, заместитель директора ИПИ РАН

Ступников Сергей Александрович (р. 1978) — кандидат технических наук, старший научный сотрудник ИПИ РАН

Сучков Александр Павлович (р. 1954) — доктор технических наук, ведущий научный сотрудник ИПИ РАН

Торшин Иван Юрьевич (р. 1972) — кандидат химических наук, ведущий научный сотрудник Российского отделения Института микроэлементов ЮНЕСКО; сотрудник Центра систем прогнозирования и распознавания

Ушаков Андрей Владимирович (р. 1987) — аспирант, ИПИ РАН

Ушмаев Олег Станиславович (р. 1981) — доктор технических наук, ведущий научный сотрудник ИПИ РАН

Шулипа Андрей Константинович (р. 1977) — научный сотрудник Центра речевых технологий, г. Санкт-Петербург

About Authors

Belykh Igor N. (b. 1959) — Candidate of Science (PhD) in physics and mathematics, Director of R&D Department, Speech Technology Center, St. Petersburg

Benderina Maria V. (b. 1986) — junior scientist, Institute for Informatics Problems, Russian Academy of Sciences

Bokov Maxim V. (b. 1982) — PhD student, South Ural State University

Borokhov Sergey V. (b. 1972) — senior scientist, Institute for Informatics Problems, Russian Academy of Sciences

Bosov Alexey V. (b. 1969) — Doctor of Science in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences

Budzko Vladimir I. (b. 1944) — Doctor of Science in technology, Deputy Director, Institute of Informatics Problems, Russian Academy of Sciences

Gorbatcevich Vladimir S. (b. 1985) — scientist, State Research Institute of Aviation Systems (GosNIIAS)

Gudkov Vladimir Yu. (b. 1959) — Candidate of Science (PhD) in technology, associate professor, Chelyabinsk State University

Kalinichenko Leonid A. (b. 1937) — Doctor of Science in physics and mathematics, professor, Honored scientist of RF, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences

Kapustin Alexey I. (b. 1985) — scientist, Speech Technology Center, St. Petersburg

Karateev Sergey L. (b. 1956) — Head of research group, State Research Institute of Aviation Systems (GosNIIAS)

Konovalov Mikhail G. (b. 1950) — Doctor of Science in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences

Korolev Victor Yu. (b. 1954) — Doctor of Science in physics and mathematics; professor, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Kostromov Nikita A. (b. 1986) — PhD student, State Research Institute of Aviation Systems (GosNIIAS)

Kozlov Alexander V. (b. 1985) — programmer, Speech Technology Center, St. Petersburg

Kurakin Alexey V. (b. 1986) — PhD student, Department of Intelligent Systems, Faculty of Control Management and Applied Mathematics, Moscow Institute of Physics and Technology (State University)

Kuznetsov Igor P. (b. 1938) — Doctor of Science in technology, professor, principal scientist, Institute of Informatics Problems, Russian Academy of Sciences

Kuznetsov Vladislav V. (b. 1988) — PhD student, Institute of Informatics Problems, Russian Academy of Sciences

Lohanova Alexandra I. (b. 1985) — scientist, Speech Technology Center, St. Petersburg

Matveev Yuri N. (b. 1955) — Doctor of Science in technology, Head of Speaker Verification and Identification Division, Speech Technology Center, St. Petersburg

Murashov Dmitry M. (b. 1958) — Candidate of Sciences (PhD) in technology, senior scientist, Dorodnicyn Computing Center, Russian Academy of Sciences

Pekhovsky Timur S. (b. 1959) — Candidate of Science (PhD) in physics and mathematics, leading scientist, Speech Technology Center, St. Petersburg

Popov Sergey V. (b. 1986) — PhD student, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Rudakov Konstantin V. (b. 1954) — Doctor of Science in physics and mathematics, Corresponding Member of the Russian Academy of Sciences, Head of Department of Computational Methods of Prediction, Dorodnicyn Computing Center of the Russian Academy of Sciences; Head of Department of Intelligent Systems, Moscow Institute of Physics and Technology (State University)

Shulipa Andrey K. (b. 1977) — scientist, Speech Technology Center, St. Petersburg

Simonchik Konstantin K. (b. 1983) — scientist, Speech Technology Center, St. Petersburg

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences

Sokolov Igor A. (b. 1954) — Academician of the Russian Academy of Sciences, Doctor of Science in technology, Director, Institute of Informatics Problems, Russian Academy of Sciences

Somin Nicolay V. (b. 1947) — Candidate of Science (PhD) in physics and mathematics, leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Stepanov Pavel V. (b. 1957) — Doctor of Science in technology, Deputy Director, Institute of Informatics Problems, Russian Academy of Sciences

Stupnikov Sergey A. (b. 1978) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Suchkov Alexander P. (b. 1954) — Doctor of Science in technology, principal scientist, Institute of Informatics Problems, Russian Academy of Sciences

Torshin Ivan Yu. (b. 1972) — Candidate of Science (PhD) in chemistry, leading scientist, Russian Center of the Trace Element Institute for UNESCO; researcher, Center of Forecasting Systems and Recognition

Ushakov Andrey V. (b. 1987) — PhD student, Institute of Informatics Problems, Russian Academy of Sciences

Ushmaev Oleg S. (b. 1981) — Doctor of Science in technology, leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Vizilter Yury V. (b. 1970) — Doctor of Science in physics and mathematics, Head of Laboratory, State Research Institute of Aviation Systems (GosNIIAS)

Zakharov Victor N. (b. 1948) — Doctor of Science in technology, associate professor; Scientific Secretary, Institute of Informatics Problems, Russian Academy of Sciences

Академик Сергей Константинович Коровин

24.05.1945–7.12.2011

Редакционная коллегия журнала «Информатика и её применения» с глубоким прискорбием извещает, что 7 декабря 2011 года на 67-м году жизни скоропостижно скончался выдающийся российский ученый в области теории управления сложными динамическими системами, член редколлегии журнала «Информатика и её применения» академик КОРОВИН Сергей Константинович.

Коровин Сергей Константинович окончил факультет радиотехники и кибернетики Московского физико-технического института в 1969 г. С 1969 г. по 1975 г. работал в Институте проблем управления АН СССР. Здесь же без отрыва от производства учился в аспирантуре (1971–1974), защитил диссертацию на степень кандидата технических наук по теме «Алгоритмы оптимизации на скользящих режимах» (1975). С 1975 по 2011 гг. работал в Институте системного анализа Российской академии наук в должностях от ведущего инженера до главного научного сотрудника, заведующего лабораторией. В 1985 г. защитил диссертацию на степень доктора технических наук по теме «Системы управления с автоматически регулируемыми связями», в 1990 г. ему присвоено ученое звание профессора.

С 1989 г. С. К. Коровин работал в МГУ им. М. В. Ломоносова, с 1996 г. являлся профессором кафедры нелинейных динамических систем и процессов управления факультета вычислительной математики и кибернетики.

В 1994 г. избран членом-корреспондентом РАН, в 2000 г. — действительным членом РАН (2000). Лауреат Государственной премии РФ (1994), премии Совета Министров СССР (1981), премии Правительства РФ (2009), премии РАН им. А. А. Андронова (2000), Ломоносовской премии МГУ I степени в области науки (2002). С. К. Коровин — автор 260 научных работ, в том числе 15 книг, 50 авторских свидетельств.

Сергей Константинович Коровин являлся членом редколлегии журнала «Информатика и её применения» с момента основания журнала и принимал активное участие в формировании редакционной политики журнала.

Правила подготовки рукописей статей для публикации в журнале «Информатика и её применения»

Журнал «Информатика и её применения» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информатики и ее приложений. Журнал издается на русском языке. По специальному решению редколлегии отдельные статьи, в виде исключения, могут печататься на английском языке. Тематика журнала охватывает следующие направления:

- теоретические основы информатики;
- математические методы исследования сложных систем и процессов;
- информационные системы и сети;
- информационные технологии;
- архитектура и программное обеспечение вычислительных комплексов и сетей.

1. В журнале печатаются результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях. Публикация не должна нарушать закон об авторских правах. Направляя свою рукопись в редакцию, авторы автоматически передают учредителям и редколлегии неисключительные права на издание данной статьи на русском языке и на ее распространение в России и за рубежом. При этом за авторами сохраняются все права как собственников данной рукописи. В связи с этим авторами должно быть представлено в редакцию письмо в следующей форме: Соглашение о передаче права на публикацию:

   «Мы, нижеподписавшиеся, авторы рукописи « », передаем учредителям и редколлегии журнала «Информатика и её применения» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторского права других лиц или организаций. Подписи авторов: (ф. и. о., дата, адрес)».

   Указанное соглашение может быть представлено как в бумажном виде, так и в виде отсканированной копии (с подписями авторов).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности опубликования представленной статьи в открытой печати.

2. Статья подписывается всеми авторами. На отдельном листе представляются данные автора (или всех авторов): фамилия, полные имя и отчество, телефон, факс, e-mail, почтовый адрес. Если работа выполнена несколькими авторами, указывается фамилия одного из них, ответственного за переписку с редакцией.

3. Редакция журнала осуществляет самостоятельную экспертизу присланных статей. Возвращение рукописи на доработку не означает, что статья уже принята к печати. Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редакционной коллегии о принятии статьи к печати или ее отклонении сообщается авторам. Редколлегия не обязуется направлять рецензию авторам отклоненной статьи.

5. Корректура статей высылается авторам для просмотра. Редакция просит авторов присылать свои замечания в кратчайшие сроки.

6. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки. Параметры страницы: формат — A4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху — 2, снизу — 2, от края до нижнего колонтитула — 1,3. Основной текст: стиль — «Обычный»: шрифт Times New Roman, размер 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине. Рекомендуемый объем рукописи — не свыше 25 страниц указанного формата. Ознакомиться с шаблонами, содержащими примеры оформления, можно по адресу в Интернете: http://www.ipiran.ru/journal/template.doc.

7. К рукописи, предоставляемой в 2-х экземплярах, обязательно прилагается электронная версия статьи (как правило, в форматах MS WORD (.doc) или LaTeX (.tex), а также — дополнительно — в формате .pdf) на дискете, лазерном диске или по электронной почте. Сокращения слов, кроме стандартных, не применяются. Все страницы рукописи должны быть пронумерованы.

8. Статья должна содержать следующую информацию на русском и английском языках: название, Ф.И.О. авторов, места работы авторов и их электронные адреса, подробные сведения об авторах, оформленные в соответствии с форматом, определяемым файлами

   http://www.ipiran.ru/journal/issues/2011_05_01/authors.asp и

   http://www.ipiran.ru/journal/issues/2011_01_eng/authors.asp,

   аннотация (не более 100 слов), ключевые слова. Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в порядке их первого упоминания. В списке литературы не должно быть позиций, на которые нет ссылки в тексте статьи. Все фамилии авторов, заглавия статей, названия книг, конференций и т. п. даются на языке оригинала, если этот язык использует кириллический или латинский алфавит.

9. Присланные в редакцию материалы авторам не возвращаются.

10. При отправке файлов по электронной почте просим придерживаться следующих правил:

    – указывать в поле subject (тема) название журнала и фамилию автора;

    – использовать attach (присоединение);

    – в случае больших объемов информации возможно использование общеизвестных архиваторов (ZIP, RAR);

    – в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.

11. Журнал «Информатика и её применения» является некоммерческим изданием. Плата за публикацию с авторов не взимается, гонорар авторам не выплачивается.

Адрес редакции: Москва 119333, ул. Вавилова, д. 44, корп. 2, ИПИ РАН
Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05 E-mail: rust@ipiran.ru

Технический редактор Л. Кокушкина
Выпускающий редактор Т. Торжкова
Художественный редактор М. Седакова
Сдано в набор 11.01.12. Подписано в печать 02.03.12. Формат 60 x 84 / 8
Бумага офсетная. Печать цифровая. Усл.-печ. л. 19,0. Уч.-изд. л. 24,2. Тираж 100 экз.

Заказ № 279

Издательство «ТОРУС ПРЕСС», Москва 119991, ул. Косыгина, д. 4
torus@torus-press.ru; http://www.torus-press.ru

Отпечатано в Академиздатцентре «Наука» РАН с готовых файлов
Москва 121099, Шубинский пер., д. 6