ISSN 2222-8772

Математические Структуры и Моделирование

№1(29)
2014

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Ф.М. ДОСТОЕВСКОГО»

МАТЕМАТИЧЕСКИЕ СТРУКТУРЫ И МОДЕЛИРОВАНИЕ

№ 1(29)

Омск
2014

Математические структуры и моделирование. — Омск : Омский государственный университет, 2014. – № 1(29). – 65 с.

ISSN 2222-8772 (print)
ISSN 2222-8799 (online)

Учредитель

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Омский государственный университет имени Ф. М. Достоевского»

Свидетельство о регистрации средства массовой информации
ПИ № ФС77-57908 от 28 апреля 2014 г.

Адрес научной редакции

Россия, 644077, Омск, пр. Мира 55А
Омский государственный университет им. Ф. М. Достоевского
факультет компьютерных наук
E-mail: lavrov@omsu.ru

МАТЕМАТИЧЕСКИЕ СТРУКТУРЫ и МОДЕЛИРОВАНИЕ

В журнале публикуются статьи, в которых излагаются результаты исследований по фундаментальной и прикладной математике, теоретической физике и размышления, касающиеся окружающей нас природы и общества.

Публикуются также статьи по информационным технологиям, компьютерным наукам, защите информации, философии и истории математики.

Объекты исследования должны быть представлены в форме некоторых математических структур и моделей.

Журнал является реферируемым. Рефераты статей публикуются в «Реферативном журнале» и в журналах «Zentralblat für Mathematik» (Германия) и «Mathematical Reviews» (США).

Электронная версия журнала представлена в сети Интернет по адресам:

http://msm.univer.omsk.su
http://msm.omsu.ru

Журнал издаётся на коммерческие средства факультета компьютерных наук Омского государственного университета.

Электронная почта главного редактора:

lavrov@omsu.ru

Подробную информацию можно найти на Web-серверах:

http://msm.univer.omsk.su
http://msm.omsu.ru

СОДЕРЖАНИЕ

Фундаментальная математика и физика

Прикладная математика и моделирование

Информация редколлегии

С текущего номера из-за допущенной неточности при интерпретации издательского стандарта редколлегией принято решение перейти на двойную нумерацию с валовым номером. Валовой номер, который по определению является номером периодического издания со дня его основания, будет указываться в круглых скобках после текущего номера. Кроме того, принято решение о выходе журнала 4 раза в год (поквартально). Таким образом, ближайшие номера на 2014 год — это № 1(29), № 2(30), № 3(31), № 4(32). Благодарим специалистов ВИНИТИ, указавших на допущенную неточность в нумерации.

Со следующего номера в должность главного редактора журнала возвращается основатель журнала, профессор, доктор физико-математических наук Александр Константинович Гуц. Вводится должность выпускающего редактора, которую займёт Дмитрий Николаевич Лавров.

Редколлегия журнала
«Математические структуры и моделирование»

Математические структуры и моделирование
2014. №1(29). С. 4–12

УДК 519.214.5

УСЛОВИЯ СЛАБОЙ ЗАВИСИМОСТИ В ПРЕДЕЛЬНЫХ ТЕОРЕМАХ ДЛЯ ОБОБЩЁННЫХ СУММ

А.Г. Гринь, профессор, д.ф.-м.н., e-mail: griniran@gmail.com

Омский государственный университет им. Ф.М. Достоевского

Аннотация. Приводятся «общеупотребительные» условия регулярности, обеспечивающие выполнение минимальных условий слабой зависимости в предельных теоремах для обобщенных сумм.

Ключевые слова: обобщённое суммирование, минимальные условия слабой зависимости, λ-перемешивание, абсолютная регулярность.

В работе [1] введён класс операций, названный там обобщённым суммированием, доказаны предельные теоремы для обобщённых сумм независимых случайных величин, описан класс предельных распределений и получены минимальные условия слабой зависимости в предельных теоремах для обобщённых сумм. В настоящей статье приводятся условия слабой зависимости, обеспечивающие выполнение этих минимальных условий.

Пусть $x \oplus y$ - бинарная операция на $\mathbb{D} \subseteq \mathbb{R}$, о которой мы будем предполагать, что она удовлетворяет условиям $A_1 - A_4$ (условия $(\mathbf{A})$):

A_1. Ассоциативность: $x \oplus (y \oplus z) = (x \oplus y) \oplus z,\ x, y, z \in \mathbb{D}$;

A_2. Коммутативность: $x \oplus y = y \oplus x,\ x, y \in \mathbb{D}$;

A_3. $x \oplus 0 = x,\ x \in \mathbb{D}$;

A_4. Равномерная непрерывность в следующем смысле:
Для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что из $|y| < \delta$ следует $|x \oplus y - x| < \varepsilon$, $\forall x \in \mathbb{D}$;

Этим условиям удовлетворяют, например, $x \oplus y = x + y,\ \mathbb{D} = \mathbb{R}$, $x \vee y = \max\{x, y\}, \mathbb{D} = \mathbb{R}_+ = [0, +\infty)$, $x \wedge y = \min\{x, y\},\ \mathbb{D} = \mathbb{R}_- = (-\infty, 0]$, а не удовлетворяют, скажем, $x \oplus y = xy$, и $x \oplus y = x + y\ (mod\ d),\ d > 0,\ \mathbb{D} = \mathbb{R}$ (не выполняются A_3 и A_4).

Если бинарная операция $x \otimes y$ удовлетворяет условиям $(\mathbf{A})$, а $f(x)$ возрастающая выпуклая (вниз) функция такая, что $f(0) = 0,\ f(\mathbb{D}) \subseteq \mathbb{D}$, то бинарная операция $x \oplus y = f^{-1}(f(x) \otimes f(y))$ также удовлетворяет условиям $(\mathbf{A})$ [1]. К примеру, бинарные операции $x \oplus y = \sqrt{x^2 + y^2}$, $x \oplus y = \ln(e^x + e^y - 1),\ \mathbb{D} = \mathbb{R}_+$ и т. п. удовлетворяют условиям $(\mathbf{A})$.

Пусть $\{\xi_n\}$ – последовательность случайных величин. Обозначим

$$X_{k,m}(b) = \left(\frac{\xi_k}{b}\right) \oplus ... \oplus \left(\frac{\xi_m}{b}\right), \quad X_n(b) = X_{1,n}(b),$$

$$X_n = X_n(1), \quad \overline{X}_n(b) = \max_{1\le k\le n} |X_k(b)|, \ k, m, n \in \mathbb{N}, \ b > 0,$$

Будем говорить, что выполнено условие A_5, если при любых $x > 0$, $y_i \in \mathbb{D}$, $i = 1, ...n$, $n \ge 2$

$$(xy_1) \oplus ... \oplus (xy_n) = x(y_1 \oplus ... \oplus y_n). \tag{1}$$

Например, $x \oplus y = x + y$, $\mathbb{D} = \mathbb{R}$, $x \oplus y = (x^p + y^p)^{1/p}$, $p \ge 1$, $x \oplus y = x \vee y$, $\mathbb{D} = \mathbb{R}_+$, удовлетворяют условию A_5.

Если $\mathbf{P}\{|\xi_1| \ge x\}$ является правильно меняющейся функцией порядка $-\rho$ и

$$\frac{\mathbf{P}\{\xi_1 \ge x\}}{\mathbf{P}\{|\xi_1| \ge x\}} \to a, \quad \frac{\mathbf{P}\{\xi_1 < -x\}}{\mathbf{P}\{|\xi_1| \ge x)\}} \to 1 - a, \ x \to +\infty, \ 0 \le a \le 1,$$

то говорят, что хвосты распределения ξ_1 имеют согласованное правильное изменение порядка $-\rho$. В этом случае

$$a_n = \sup\{x : n\mathbf{P}\{|\xi_1| \ge x\} \ge 1\}$$

является правильно меняющейся последовательностью порядка $1/\rho$ [2, стр. 111],

$$n\mathbf{P}\{\xi_1 \ge xa_n\} \to \frac{a}{x^\rho}, \quad n\mathbf{P}\{\xi_1 < -xa_n\} \to \frac{1-a}{x^\rho}, \quad n \to \infty \tag{2}$$

[2, стр. 94] и $\mathbf{E}|\xi_1|^p < \infty$, $0 < p < \rho$ [2, стр. 103].

Пусть

$$F_\rho(x) = \begin{cases} 1 - \dfrac{a}{x^\rho}, & x \ge 1 \\ 1 - a, & -1 < x < 1 \\ \dfrac{1-a}{|x|^\rho}, & x \le -1 \end{cases}.$$

Если ξ и η - независимые случайные величины с функциями распределения F_ξ и F_η, то будем обозначать $F_\xi * F_\eta = F_{\xi\oplus\eta}$.

Предположим, что при каждом $x \in \mathbb{R}$ существует

$$H_\rho(x) = \lim_{k\to\infty} F_\rho^{*k}\left(k^{1/\rho}x\right). \tag{3}$$

Пусть при любых $n \in \mathbb{N}$ и $z > 0$ $\mathbf{E}|X_n(z)|^p < \infty$. Положим

$$b_n(p) = \inf\left\{z > 0 : \max_{1\le k\le n} \mathbf{E}|X_k(z)|^p \le 1\right\}.$$

Сформулируем основные результаты из [1] (теоремы 1 и 2).

Теорема 1. *Пусть операция $\oplus$ удовлетворяет условиям $A_1 - A_5$ на $\mathbb{D} = \mathbb{R}$. Для того, чтобы*

$$\lim_{n\to\infty} \mathbf{P}\{X_n(a_n) < x\} = H_\rho(x), \ x \in \mathbb{R}, \tag{4}$$

где $H_\rho(x)$ удовлетворяет

$$\lim_{k\to\infty} k\left(1-H_\rho\left(k^{1/\rho}x\right)\right)=\frac{a}{x^\rho},\quad \lim_{k\to\infty} kH_\rho\left(-k^{1/\rho}x\right)=\frac{1-a}{x^\rho},\ x>0, \tag{5}$$

необходимо и достаточно, чтобы хвосты распределения ξ_1 имели согласованное правильное изменение порядка ρ и при любых $0<p<\rho$ и при некотором $\varepsilon>0$ выполнялось

$$\liminf_{n\to\infty} n\mathbf{P}\{|\xi_1|\geq \varepsilon b_n(p)\}>0. \tag{6}$$

Будем писать $\xi \stackrel{d}{=} \eta$, $\xi_n \stackrel{d}{\to} \eta$ и $\xi_n \stackrel{d}{\sim} \eta_n$ в случаях, когда, соответственно, распределения ξ и η совпадают, $\{\xi_n\}$ сходится к η по распределению и когда последовательности $\{\xi_n\}$ и $\{\eta_n\}$ слабо эквивалентны (см., например, [4, § 28.1]). Через $\hat{\xi}_1,...,\hat{\xi}_n$ будем обозначать *независимые* случайные величины такие, что $\hat{\xi}_k \stackrel{d}{=} \xi_k,\ k=1,2,...,n$.

Теорема 2. *Пусть $\{\xi_n,\ n=1,2,...\}$ – стационарная последовательность, у которой хвосты распределения ξ_1 имеют согласованное правильное изменение порядка $-\rho$, а операция $\oplus$ удовлетворяет условиям A_1-A_5 на $\mathbb{D}=\mathbb{R}$. Для того, чтобы выполнялось (4), где $H_\rho(x)$ удовлетворяет (5), необходимо и достаточно, чтобы выполнялись следующие утверждения*

а)

$$X_{n+m}(a_{n+m}) \stackrel{d}{\sim} \widehat{X}_n(a_{n+m}) \oplus \widehat{X}_m(a_{n+m}),\quad n+m\to\infty \tag{R_1}$$

(здесь символ $n+n\to\infty$ означает, что (R_1) выполняется при $n\to\infty$ и при любой последовательности $m=m(n)$);

б) при любом $x>0$ и при любой достаточно медленно растущей последовательности $k=k(n)\to\infty,\ n\to\infty$

$$\mathbf{P}\{\pm X_n(a_{kn})\geq x\}\sim n\mathbf{P}\{\pm\xi_1\geq xa_{kn}\},\ n\to\infty. \tag{R_2}$$

Теорему 2 можно интерпретировать так: условия (R_1) и (R_2) являются минимальными условиями слабой зависимости, при которых выполняется (4).

Далее в настоящей работе приводятся условия слабой зависимости, обеспечивающие выполнение (R_1) и (R_2).

Пусть $\mathcal{F}_{\leq n}$ и $\mathcal{F}_{\geq n}$ – σ -алгебры, порождённые семействами $\{\xi_i : i\leq n\}$ и $\{\xi_i : i\geq n\}$. Если для некоторой функции $\lambda(x)>0$ такой, что $\lambda(x)\to 0,\ x\to 0$

$$\sup\left\{\frac{\mathbf{P}(AB)}{\mathbf{P}(A)\lambda(\mathbf{P}(B))} : A\in\mathcal{F}_{\leq 0}, B\in\mathcal{F}_{\geq 1} \text{ или } A\in\mathcal{F}_{\geq 1}, B\in\mathcal{F}_{\leq 0}\right\}<1,$$

то говорят, что последовательность $\{\xi_n\}$ удовлетворяет *условию λ-перемешивания* (см. [5]).

Обозначим

$$\delta_n=\lambda\left(\max_{1\leq k\leq n}\mathbf{P}\{|X_k(c_n)|\geq\delta\}\right).$$

Лемма 1. *Для любого* $\varepsilon > 0$ *найдётся* $\delta > 0$ *такое, что при* $x > 0,\ m \le n$

$$\mathbf{P}\{\overline{X}_{m-1}(c_n) \ge x + \varepsilon\} \le (1 - \delta_n)^{-1}\mathbf{P}\{|X_m(c_n)| \ge x\}.$$

Доказательство. В силу свойства A_4 для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что при любом $x > 0$

$$\{\xi \ge x + \varepsilon,\ |\eta| < \delta\} \subseteq \{\xi \oplus \eta \ge x\},$$

$$\{|\xi| \ge x + \varepsilon,\ |\eta| < \delta\} \subseteq\ \{|\xi \oplus \eta| \ge x\}. \tag{7}$$

Аналогично из свойств $A_1 - A_4$ выводится

$$\{\xi \ge x + \varepsilon,\ |\eta| < \delta,\ |\zeta| < \delta\} \subseteq \{\xi \otimes \eta \oplus \zeta \ge x\},$$

$$\{|\xi| \ge x + \varepsilon,\ |\eta| < \delta,\ |\zeta| < \delta\} \subseteq \{|\xi \otimes \eta \oplus \zeta| \ge x\}. \tag{8}$$

Пусть $E_k = \{\overline{X}_{k-1}(c_n) < x + \varepsilon \le |X_k(c_n)|\},\ k = 1, ..., m$. Тогда $E_iE_j = \varnothing,\ i \ne j,\ \bigcup\limits_{k=1}^{m-1} E_k = \{\overline{X}_{m-1}(c_n) \ge x + \varepsilon\}$, а в силу (7) найдётся $\delta > 0$ такое, что

$$\{|X_k(c_n)| \ge x + \varepsilon,\ |X_{k+1,m}(c_n)| < \delta\} \subseteq \{|X_m(c_n)| \ge x\},$$

то есть

$$\{|X_m(c_n)| < x\} \subseteq \{|X_k(c_n)| < x + \varepsilon\} \cup \{|X_{k+1,m}(c_n)| \ge \delta\},$$

откуда

$$\{|X_m(c_n)| < x,\ E_k\} \subseteq \{|X_{k+1,m}(c_n)| \ge \delta,\ E_k\},\ k = 1, ..., m-1. \tag{9}$$

С помощью (9) и условия λ-перемешивания получаем

$$\mathbf{P}\{\overline{X}_{m-1}(c_n) \ge x + \varepsilon\} \le \mathbf{P}\{|X_m(c_n)| \ge x\} + \sum_{k=1}^{m-1}\mathbf{P}\{|X_m(c_n)| < x,\ E_k\} \le$$

$$\le \mathbf{P}\{|X_m(c_n)| \ge x\} + \sum_{k=1}^{m-1}\mathbf{P}\{|X_{k+1,m}(c_n)| \ge \delta,\ E_k\} \le$$

$$\le \mathbf{P}\{|X_m(c_n)| \ge x\} + \lambda\left(\max_{1\le k\le n}\mathbf{P}\{|X_k(c_n)| \ge \delta\}\right)\sum_{k=1}^{m-1}\mathbf{P}\{E_k\} =$$

$$= \mathbf{P}\{|X_m(c_n)| \ge x\} + \delta_n \cdot \mathbf{P}\{\overline{X}_{m-1}(c_n) \ge x + \varepsilon\},$$

откуда следует утверждение леммы. ■

Лемма 2. *Для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что при любом $x > 0$*

$$\mathbf{P}\{\pm X_n(c_n) \geq x\} \geq n\mathbf{P}\{\pm\xi_1 \geq (x+\varepsilon)c_n\}(1-3\delta_n)).$$

Неравенство с плюсом доказывается, когда $\mathbb{R}_+ \subseteq \mathbb{D}$, а с минусом – когда $\mathbb{R}_- \subseteq \mathbb{D}$.

Доказательство. Пусть $A_0 = \varnothing,\ A_n = \{\overline{X}_{n-1}(c_n) < \delta,\ \xi_n \geq (x+\varepsilon)c_n\}$

$$A_k = \{\overline{X}_{k-1}(c_n) < 2\delta,\ \xi_k \geq (x+\varepsilon)c_n,\ |X_{k+1,n}(c_n)| < \delta\},\ 1 \leq k \leq n-1.$$

В силу (8) и (9) для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что

$$\mathbf{P}\{X_n(c_n) \geq x\} \geq \mathbf{P}\left\{\bigcup_{k=1}^{n} A_k\right\} = \sum_{k=1}^{n} \mathbf{P}\{\overline{A}_1 \cdot \ldots \cdot \overline{A}_{k-1}A_k\} =$$

$$= \sum_{k=1}^{n} \mathbf{P}\{A_k\} - \sum_{k=1}^{n} \mathbf{P}\left\{A_k \cdot \bigcup_{j=1}^{k-1} A_j\right\}. \tag{10}$$

При $1 \leq k \leq n-1$ с помощью λ-перемешивания получаем

$$\mathbf{P}\{A_k\} = \mathbf{P}\left\{\xi_k \geq (x+\varepsilon)c_n\right\} -$$

$$-\mathbf{P}\left\{(\xi_k \geq (x+\varepsilon)c_n) \cdot \left(\left\{\overline{X}_{k-1}(c_n) \geq 2\delta\right\} \cup \left\{|X_{k+1,n}(c_n)| \geq \delta\right\}\right)\right\} \geq$$

$$\geq \mathbf{P}\left\{\xi_k \geq (x+\varepsilon)c_n\right\}\left(1 - \lambda\left(\mathbf{P}\{|X_{k+1,n}(c_n)| \geq \delta\}\right) - \lambda\left(\mathbf{P}\{\overline{X}_{k-1}(c_n) \geq 2\delta\}\right)\right). \tag{11}$$

$\mathbf{P}\{A_n\}$ оценивается аналогично. Без ограничения общности $\delta > 0$ можно считать таким, что

$$\{|X_{j-1}(c_n)| < 2\delta,\ \xi_j \geq (x+\varepsilon)c_n\} \subseteq \{|X_{j-1}(c_n)| < 2\delta,\ X_j(c_n) \geq x\}$$

так что если $2\delta < x$, то $\mathbf{P}\left\{A_k \cdot \bigcup\limits_{j=1}^{k-1} A_j\right\} \leq$

$$\leq \mathbf{P}\left\{\xi_k \geq (x+\varepsilon)c_n,\ \bigcup_{j=1}^{k-1}\left(\overline{X}_{j-1}(c_n) < 2\delta, \xi_j \geq (x+\varepsilon)c_n\right)\right\} \leq$$

$$\leq \mathbf{P}\left\{\xi_k \geq (x+\varepsilon)c_n\right\}\lambda\left(\mathbf{P}\{\overline{X}_{k-1}(c_n) \geq 2\delta\}\right). \tag{12}$$

и тогда из (10), (11) и (12) и леммы 1 следует

$$\mathbf{P}\{X_n(c_n) \geq x\} \geq \sum_{k=1}^{n} \mathbf{P}\left\{\xi_k \geq (x+\varepsilon)c_n\right\}(1-3\delta_n).$$

Вероятность $\mathbf{P}\{-X_n(c_n) \geq x\}$ оценивается аналогично.

Лемма доказана. ■

Следующее предложение – это модификация леммы 3.1 из [8].

Лемма 3. *Для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что при достаточно больших n*

$$\mathbf{P}\{\pm X_n(c_n) \geq x + \varepsilon\} \leq (1 - \delta_n)^{-1}\delta_n \mathbf{P}\{|X_n(c_n)| \geq \delta\} + n\mathbf{P}\{\pm\xi_1 \geq xc_n\}.$$

(Предположения об области $\mathbb{D}$ те же, что и в лемме 2.)

Доказательство. Пусть $E_k = \{\overline{X}_{k-1}(c_n) < 2\delta \leq |X_k(c_n)|\}$, $k = 1, ..., n$. Тогда $E_i E_j = \varnothing$, $i \neq j$, $\bigcup\limits_{k=1}^{n-1} E_k = \{\overline{X}_{n-1}(c_n) \geq 2\delta\}$. В силу (9) для любого $\varepsilon > 0$ найдётся $\delta > 0$ такое, что при $1 \leq k \leq n-1$

$$\{|X_{k+1,n}(c_n)| < \delta,\ E_k,\ \max_{1\leq k\leq n} \xi_k < xc_n\} \subseteq$$

$$\subseteq \{X_n(c_n) < x + \varepsilon,\ E_k,\ \max_{1\leq k\leq n} \xi_k < xc_n\},$$

откуда

$$\{X_n(c_n) \geq x + \varepsilon,\ E_k,\ \max_{1\leq k\leq n} \xi_k < xc_n\} \subseteq \{E_k,\ |X_{k+1,n}(c_n)| \geq \delta\}. \tag{13}$$

Аналогично выводится

$$\{X_n(c_n) \geq x + \varepsilon, \max_{1\leq k\leq n} \xi_k < xc_n\} \subseteq \{\overline{X}_{n-1}(c_n) \geq 2\delta,\ \max_{1\leq k\leq n} \xi_k < xc_n\}.$$

Отсюда

$$\{X_n(c_n) \geq x + \varepsilon,\ \max_{1\leq k\leq n} \xi_k < xc_n\} =$$

$$= \{X_n(c_n) \geq x + \varepsilon,\ \overline{X}_{n-1}(c_n) \geq 2\delta,\ \max_{1\leq k\leq n} \xi_k < xc_n\}. \tag{14}$$

С помощью (13) и (14) получаем $\mathbf{P}\{X_n(c_n) \geq x + \varepsilon\} \leq$

$$\leq \mathbf{P}\{X_n(c_n) \geq x + \varepsilon, \max_{1\leq k\leq n} \xi_k < xc_n\} + \mathbf{P}\{\max_{1\leq k\leq n} \xi_k \geq xc_n\} =$$

$$= \mathbf{P}\{X_n(c_n) \geq x + \varepsilon, \overline{X}_{n-1}(c_n) \geq 2\delta, \max_{1\leq k\leq n} \xi_k < xc_n\} +$$

$$+\mathbf{P}\{\max_{1\leq k\leq n} \xi_k \geq xc_n\} = \sum_{k=1}^{n-1} \mathbf{P}\{X_n(c_n) \geq x + \varepsilon, E_k, \max_{1\leq k\leq n} \xi_k < xc_n\} +$$

$$+\mathbf{P}\{\max_{1\leq k\leq n} \xi_k \geq xc_n\} \leq n\mathbf{P}\{\xi_1 \geq xc_n\} + \sum_{k=1}^{n-1} \mathbf{P}\{|X_{k+1,n}(c_n)| \geq \delta, E_k\} \leq$$

$$\leq n\mathbf{P}\{\xi_1 \geq xc_n\} + \lambda\left(\max_{1\leq k\leq n} \mathbf{P}\{|X_k(c_n)| \geq \delta\}\right) \sum_{k=1}^{n-1} \mathbf{P}\{E_k\} =$$

$$= \delta_n \mathbf{P}\{\overline{X}_{n-1}(c_n) \geq 2\delta\} + n\mathbf{P}\{\xi_1 \geq xc_n\}.$$

Отсюда с помощью леммы 1 выводится оценка для $\mathbf{P}\{X_n(c_n) \geq x + \varepsilon\}$ в формулировке леммы. Оценка для $\mathbf{P}\{-X_n(c_n) \geq x + \varepsilon\}$ доказывается аналогично. ■

Следствие 1. *Из лемм 2 и 3 вытекает следующее утверждение: если последовательность положительных чисел* $\{c_n\}$ *такова, что при любом* $\delta > 0$

$$\delta_n = \lambda\left(\max_{1\le k\le n}\mathbf{P}\{|X_k(c_n)|\ge\delta\}\right)\to 0,\ n\to\infty$$

и при любых $x>0,\ \delta>0$ *выполняется следующее предположение:*

$$\delta_n\mathbf{P}\{|X_n(c_n)|\ge\delta\} = o\left(n\mathbf{P}\{\pm\xi_1\ge xc_n\}\right),\ n\to\infty, \tag{15}$$

то

$$\mathbf{P}\{\pm X_n(c_n)\ge x\}\sim n\mathbf{P}\{\pm\xi_1\ge xc_n\},\ n\to\infty. \tag{16}$$

Пусть хвосты распределения ξ_1 имеют согласованное правильное изменение порядка $-\rho,\ \rho>0$. Тогда $\mathbf{P}\{|\xi_1|\ge x\}$ правильно меняющаяся функция порядка $-\rho$, а $\{a_n\}$ – правильно меняющаяся последовательность порядка $1/\rho$. Если выполняется (6), то $b_n(p)=O(a_n),\ n\to\infty$. Пусть $k=k(n)\to\infty$. Если $k(n)$ растёт достаточно медленно, то

$$\max_{1\le m\le n}\mathbf{P}\{|X_m(a_{nk})|\ge\delta\}\le\frac{\max\limits_{1\le m\le n}\mathbf{E}|X_m|^p}{\delta^p a_{nk}^p}=O(a_n^p a_{nk}^{-p})=O\left(k^{-p/\rho}\right).$$

Отсюда

$$\max_{1\le m\le n}\mathbf{P}\{|X_m(a_{nk})|\ge\delta\}\mathbf{P}\{|X_n(a_{nk})|\ge\delta\}=O\left(k^{-2p/\rho}\right),$$

и так как в силу (2) $n\mathbf{P}\{|\xi_1|\ge xa_{nk}\}\sim(kx^\rho)^{-1}$, то при $p>\rho/2$ выполняется условие (15) и из (16) получаем

$$\mathbf{P}\{\pm X_n(a_{nk})\ge x\}\sim n\mathbf{P}\{\pm\xi_1\ge xa_{nk}\}.$$

Таким образом, мы доказали, что, если последовательность $\{\xi_n\}$ удовлетворяет условию λ-перемешивания, хвосты распределения ξ_1 имеют согласованное правильное изменение порядка $-\rho,\ \rho>0$ и выполнено (6), то имеет место R_2.

Будем говорить, что последовательность $\{\xi_n\}$ удовлетворяет условию абсолютной регулярности, если

$$\beta(n)=\mathbf{E}\sup_{A\in\mathcal{F}_{\ge n}}|\mathbf{P}\left(A|\mathcal{F}_{\ge n}\right)-\mathbf{P}(A)|\to 0,\quad n\to\infty.$$

(см. [3, 7]).

Доказывается [3, с.167], что

$$\beta(n)=\frac{1}{2}\sup\sum_{i,j}\left[\mathbf{P}(A_iB_j)-\mathbf{P}(A_i)\mathbf{P}(B_j)\right], \tag{17}$$

где sup берётся по всевозможным конечным разбиениям пространства элементарных исходов Ω на непересекающиеся события $(A_1,...,A_k)$, $(B_1,...,B_l)$ такие, что $A_i\in\mathcal{F}_{\le 0},\ B_j\in\mathcal{F}_{\ge n},\ i=1,...,k,\ j=1,...,l$. Отсюда нетрудно вывести, что

$$\beta(n)\le\varphi(n)=\sup_{A\in\mathcal{F}_{\le n},B\in\mathcal{F}_{\ge n}}|\mathbf{P}(A|B)-\mathbf{P}(A)|,$$

то есть стационарные последовательности, удовлетворяющие условию равномерно сильного перемешивания ($\varphi(n) \to 0,\ n \to \infty$), являются абсолютно регулярными.

Пусть ξ измерима относительно $\mathcal{F}_{\geq 0},\ \eta$ – относительно $\mathcal{F}_{\geq n},\ \mathbf{P}_\xi,\ \mathbf{P}_\eta$ и $\mathbf{P}_{\xi,\eta}$– распределения $\xi,\ \eta$, и (ξ,η) соответственно. Из (17) следует

$$\sum_{i,j} [\mathbf{P}_{\xi,\eta}(C_i \times D_j) - \mathbf{P}_\xi(C_i)\mathbf{P}_\eta(D_j)] \leq 2\beta(n), \tag{18}$$

где $(C_1, ..., C_k)$ и $(D_1, ..., D_l)$ – разбиения $\mathbb{R}$ на непересекающиеся борелевские множества. Так как $\mathbf{P}_{\xi,\eta} \ll \mathbf{P}_\xi \times \mathbf{P}_\eta$, из (18) следует

$$\int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} \left| \frac{d\,\mathbf{P}_{\xi,\eta}}{d\,\mathbf{P}_\xi \times \mathbf{P}_\eta}(x,y) - 1 \right| \mathbf{P}_\xi \times \mathbf{P}_\eta(dxdy) \leq 2\beta(n). \tag{19}$$

Докажем, что в предположениях теоремы 2 из условия абсолютной регулярности следует (R_1).

Прежде всего отметим, что $a_n \to \infty$ как правильно меняющаяся функция положительного порядка [6, с.24], так что $\xi_1/a_n \to 0$ по вероятности и из свойства A_4 следует, что $X_k(a_n) \to 0$ по вероятности при любом натуральном k и, следовательно, при достаточно медленно растущих $k = k(n) \to \infty$. В силу стационарности последовательности $\{\xi_n\}$ $X_{n+1,n+k}(a_{n+m}) \stackrel{d}{=} X_{n+m+1,n+m+k}(a_{n+m}) \stackrel{d}{=} \stackrel{d}{=} X_k(a_{n+m})$, так что все эти величины также стремятся к нулю по вероятности при $n \to \infty$, если $k = k(n) \to \infty$ достаточно медленно.

Пусть последовательность $\delta_n \to 0$ такова, что $\mathbf{P}\{|X_{n+1,n+k}(a_{n+m})| \geq \delta_n\} \to 0$, $\mathbf{P}\{|X_{n+m+1,n+m+k}(a_{n+m})| \geq \delta_n\} \to 0,\ n \to \infty$. Если $|X_{n+1,n+k}(a_{n+m})| < \delta_n$, $|X_{n+m+1,n+m+k}(a_{n+m})| < \delta_n$ то из свойства A_4 следует $|X_{n+m}(a_{n+m}) - X_n(a_{n+m}) \oplus$ $\oplus\, X_{n+k,n+m+k}(a_{n+m})| < \varepsilon_n$, где $\varepsilon_n \to 0$. Тогда

$$|\mathbf{E}\exp\{itX_{n+m}(a_{n+m})\} - \mathbf{E}\exp\{itX_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m})\}| \leq$$

$$\leq \mathbf{E}\,|\exp\{it\,(X_{n+m}(a_{n+m}) - X_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m}))\} - 1| \leq$$

$$\leq |t|\varepsilon_n + 2\mathbf{P}\{|X_{n+1,n+k}(a_{n+m})| \geq \delta_n\} + 2\mathbf{P}\{|X_{n+m+1,n+m+k}(a_{n+m})| \geq \delta_n\} \to 0,$$

$n \to \infty$. Это означает, что

$$X_{n+m}(a_{n+m}) \overset{d}{\sim} X_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m}),\ n \to \infty \tag{20}$$

[4, с.393]. Обозначим через $\mathbf{P}_n,\ \mathbf{P}_m$ и $\mathbf{P}_{n,m}$ распределения величин $X_n(a_{n+m})$, $X_{n+k,n+m+k}(a_{n+m})$ и $X_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m})$ соответственно. $X_n(a_{n+m})$ измерима относительно $\mathcal{F}_{\leq n}$, а $X_{n+k,n+m+k}(a_{n+m})$ – относительно $\mathcal{F}_{\geq n+k}$, так что в силу (19)

$$|\mathbf{E}\exp\{itX_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m})\} -$$

$$-\mathbf{E}\exp\{itX_n(a_{n+m})\}\,\mathbf{E}\exp\{itX_{n+k,n+m+k}(a_{n+m})\}| =$$

$$= \left| \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} e^{it(x+y)} \mathbf{P}_{n,m}(dxdy) - \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} e^{it(x+y)} \mathbf{P}_n \times \mathbf{P}_m(dxdy) \right| \leq$$

$$\leq \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} \left| \frac{d\,\mathbf{P}_{n,m}}{d\,\mathbf{P}_n \times \mathbf{P}_m}(x,y) - 1 \right| \mathbf{P}_n \times \mathbf{P}_m(dxdy) \leq 2\beta(k) \to 0,$$

$n \to \infty$. Это означает, что

$$X_n(a_{n+m}) \oplus X_{n+k,n+m+k}(a_{n+m}) \overset{d}{\sim} \widehat{X}_n(a_{n+m}) \oplus \widehat{X}_{n+k,n+m+k}(a_{n+m}),\ n \to \infty, \quad (21)$$

где $\widehat{X}_n(a_{n+m})$ и $\widehat{X}_{n+k,n+m+k}(a_{n+m})$ – независимые величины, распределения которых совпадают с распределениями величин $X_n(a_{n+m})$ и $X_{n+k,n+m+k}(a_{n+m})$. Поскольку $\widehat{X}_{n+k,n+m+k}(a_{n+m}) \overset{d}{=} \widehat{X}_m(a_{n+m})$, из (20) и (21) следует (R_1).

Таким образом, если бинарная операция $x \oplus y$ удовлетворяет условиям $A_1 - A_5$, а последовательность $\{\xi_n\}$ абсолютно регулярна, то выполняется (R_1).

Литература

1. Гринь А.Г. Минимальные условия слабой зависимости в схеме обобщенного суммирования // Теория вероятн. и ее примен. (В печати)
2. Ибрагимов И.А., Линник Ю.В. Независимые и стационарно связанные величины. М. : Наука, 1965. 524 с.
3. Ибрагимов И.А., Розанов Ю.А. Гауссовские случайные процессы. М. : Наука, 1970. 384 с.
4. Лоэв М. Теория вероятностей. М. : ИЛ, 1962. 719 с.
5. Гринь А.Г. Области притяжения для последовательностей с перемешиванием // Сибирский математический журнал. 1990. Т. 31, № 1. С. 53–63.
6. Сенета Е. Правильно меняющиеся функции. М. : Наука, 1985. 141 с.
7. Bradley R. Basic properties of strong mixing conditions // Dependence in Probability and Statistics (Ser. Progress in Probability and Statistics). Boston – Basel – Stuttgart : Birkhäuser, 1986. V. 11. P. 165–192.
8. Peligrad M. An invariance principle for φ– mixing sequences //Ann. Probab. 1985. V. 13, N. 4. P. 1304–1313.

CONDITIONS OF THE WEAK DEPENDENCE IN THE LIMIT THEOREMS FOR GENERALIZED SUMS

A.G. Grin, professor, Doctor of Mathematics, e-mail: griniran@gmail.com

Omsk State University n.a. F.M. Dostoevskiy

Abstract. "Commonly used" regularity conditions ensuring the implementation of the minimum conditions of weak dependence in limit theorems for generalized sums are given in this article.

Keywords: generalized sums, minimum conditions of weak dependence, λ-mixing, absolute regularity.

Mathematical
Structures and Modeling
2014. N. 1(29). PP. 13–30

UDC 510.5

SPACE-TIME ASSUMPTIONS BEHIND NP-HARDNESS OF PROPOSITIONAL SATISFIABILITY

O. Kosheleva, Ph.D. (Math.), Associate Professor, e-mail: olgak@utep.edu
V. Kreinovich, Ph.D. (Math.), Professor, e-mail: vladik@utep.edu

University of Texas at El Paso, El Paso, TX 79968, USA

Abstract. For some problems, we know feasible algorithms for solving them. Other computational problems (such as propositional satisfiability) are known to be NP-hard, which means that, unless P=NP (which most computer scientists believe to be impossible), no feasible algorithm is possible for solving all possible instances of the corresponding problem. Most usual proofs of NP-hardness, however, use Turing machine – a very simplified version of a computer – as a computation model. While Turing machine has been convincingly shown to be adequate to describe what can be computed *in principle*, it is much less intuitive that these oversimplified machines are adequate for describing what can be computed *effectively*; while the corresponding adequacy results are known, they are not easy to prove and are, thus, not usually included in the textbooks. To make the NP-hardness result more intuitive and more convincing, we provide a new proof in which, instead of a Turing machine, we use a generic computational device. This proof explicitly shows the assumptions about space-time physics that underlie NP-hardness: that all velocities are bounded by the speed of light, and that the volume of a sphere grows no more than polynomially with radius. If one of these assumptions is violated, the proof no longer applies; moreover, in such space-times we can potentially solve the satisfiability problem in polynomial time.

Keywords: NP-hard problems, space-time models, computation in curved space-time.

1. Formulation of the Problem

General problem. Which problems can be solved in feasible time and which cannot? To answer this question, it is necessary to formally describe which algorithms are feasible, what is a problem, and how can we know that a problem cannot be solved by a feasible algorithm. Let us recall how this is done in theory of computation; for details, see, e.g., [2, 4, 8].

Feasibility: a brief reminder. Many algorithms are feasible; for example, most algorithms whose computation time is bounded by a square or a cube of the bit size n of the input are usually feasible.

However, some algorithms require, even for inputs of reasonable length, computation time which exceeds the lifetime of the Universe. For example, for problems for which we know that the bit size of the solution y does not exceed the bit size $\mathrm{len}(x)$ of the input x, we can find a solution by using exhaustive search, i.e., by trying all possible words y of size $\mathrm{len}(y) \le n \stackrel{\mathrm{def}}{=} \mathrm{len}(x)$. However, even for words in a binary 0-1 alphabet, this would require, in the worst case, trying $1 + 2 + \ldots + 2^n = 2^{n+1} - 1$ possible words.

Even for reasonable-size inputs, of size $n \approx 1000$, this would require $2^{1000} \approx$ $\approx 10^{300}$ computation steps. Even if each of $\approx 10^{90}$ elementary particles which form the Universe serves as one of the parallel processors, each of these processors would still need to perform 10^{200} computation steps: and even if we divide the lifetime of the Universe to the smallest possible time quantum (the time during which light passes through an elementary particle), we would still get no more than $\approx 10^{40}$ computation steps. Thus, such exponential-time algorithms are usually considered to be infeasible.

This observation prompts the usual definition of feasibility. For each algorithm A, let $t_A(x)$ denote the number of computation steps on input x. The worst-case number of computation steps $t^w_A(n) \stackrel{\mathrm{def}}{=} \max\{t_A(x) : \mathrm{len}(x) = n\}$ on all inputs x of size (length) n is known as the (worst-case) *computational complexity* of the algorithm A. In these terms, an algorithm is called *feasible* if and only if it is *polynomial-time*, i.e., if there exists a polynomial $P(n)$ for which $t^w_A(n) \le P(n)$ for all n.

This definition is not perfect:

- an algorithm with computational complexity $t^w_A(n) = 10^{1000} \cdot n$ is polynomial-time, but clearly not feasible;
- on the other hand, an algorithm with computational complexity $t^w_A(n) = \exp(10^{-9} \cdot n)$ is practically feasible for all inputs of size $\le 10^9$, but is is not polynomial time.

However, the above definition is the best we have.

What is a problem. In a precisely formulated problem, it may be difficult to solve a problem, but it should be feasible to check whether a proposed candidate for a solution is indeed a solution.

For example, in mathematics, the main problem is: given a statement x, produce a detailed proof y of either the statement x or of its negation. Coming up with a proof is often very difficult, but once a detailed step-by-step proof is produced, it is easy to check step-by-step whether each step is correct – even a computer can do it provided that the proof is detailed enough. In this case, the problem is: given x, find y such that $C(x, y)$ holds, where $C(x, y)$ is a feasibly computable predicate describing that y is a proof of x or of $\neg x$.

Of course, to be able to check the proof in reasonable time, we must also require that length of this proof is feasible. Similarly to feasible time, it is reasonable to

formalize this requirement by requesting that there exists a polynomial $P_\ell(n)$ such that $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$. Thus, a problem takes the following form: given a word x, find a word y such that $C(x, y)$ and $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$ – or produce a message that such a proof y is not possible.

Similarly, in physics, the main problem is: given the observation data x, find a law y that fits all this data. Once a formula y is found, it is easy to check, observation-by-observation, that all the observations x satisfy this formula; however, coming up with an appropriate formula is often very difficult. In this example, the limitation on the size of y is even more severe: namely, the length of y must not exceed the length of x – if we do not make this requirement, then we can simply take the listing of all the observations as the desired formula. In this case, $\mathrm{len}(y) \le \mathrm{len}(x)$, i.e., $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$ for $P_\ell(n) = n$.

In engineering, we are given specifications x, e.g., about a bridge, and we need to find a design y which satisfies all the specifications. Modern software enables us to feasibly check whether a given design satisfies the desired specifications, but finding such a design is often difficult. The design must be feasible to implement, which means that we must have $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$ for some polynomial $P_\ell(n)$.

In all these cases, we have a feasible algorithm $C(x, y)$ and a polynomial $P_\ell(n)$, and our task is: given a word x, find a word y for which $C(x, y)$ and $\mathrm{len}(y) \le \le P_\ell(\mathrm{len}(x))$ – or produce a message that such y is not possible. This will be our general definition of a problem.

In this definition, once we have a guess y, it is feasible (i.e., requires polynomial time) to check whether this guess is a correct solution. In theoretical computer science, computations with guesses are called *non-deterministic*. Because of this, such problems are called *non-deterministic polynomial-time*, or NP, for short.

All problems from the class NP are algorithmically solvable: e.g., by exhaustive search. For each input x, the length of possible solution y is bounded. Thus, we can, in principle, find the solution y by applying *exhaustive search*, i.e., by testing all possible words y of length $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$.

Exhaustive search is not feasible. The problem with the exhaustive search algorithm is that the corresponding computation time is proportional to the number of possible words of a given length, and this number grows exponentially with the length of the input, as $S_A^{P_\ell(\mathrm{len}(x))}$, where S_A is the number of possible symbols. We already know that such exponential-time algorithms are not practically feasible.

Are feasible algorithms possible? Is P equal to NP? For some problems from the class NP, there exists a feasible (polynomial-time) algorithm for solving the corresponding problem. The class of such feasibly solvable problems is denoted by P.

It is not known whether all the problems from the class NP can be thus solved, i.e., whether P=NP. This is a long-standing open problem. Most computer scientists believe that P≠NP.

The notion of NP-hardness. While it is not known whether P is equal to NP, it is known that some problems from the class NP are the hardest. This "hardness" is described by the notion of *reduction*: if a problem A can be reduced to problem A', this means that the problem A' is at least as hard as the problem A.

The notion of reduction can illustrated on the following simple example. A usual way to solve an equation of the type $a \cdot x^4 + b \cdot x^2 + c = 0$ is to reduce it to the problem A' of solving the quadratic equation. For this reduction, we introduce a new variable $y = x^2$; in terms of this new variable, the original equation takes the form $a \cdot y^2 + b \cdot y + c = 0$. We know how to solve the corresponding quadratic equation; once we find its solution, we can find x as $\pm\sqrt{y}$. Thus, to solve a particular case of the original problem A, we:

- form the corresponding particular case of the problem A'; we will denote the corresponding algorithm by U_1;
- solve this new particular case;
- use the solution to compute the solution to the original problem; we will denote the corresponding algorithm by U_3.

Both algorithms U_1 and U_3 can be multiple-valued.

It is also important to make sure that in this manner, we can find *all* solutions to the original problem, i.e., that for every solution of the original problem, there is a solution to the problem A' from which this solution can be obtained (in our case, $y = x^2$); we will denote the corresponding algorithm by U_2.

In general, when we have two problems from the class NP, a problem A described by a feasible property $C(x, y)$ and a problem A' described by a feasible property $C'(x', y')$, then we say that A is *reducible* to A' if these exists three feasible algorithms U_1, U_2, and U_3 with the following properties:

- if $C'(U_1(x), y')$, then $C(x, U_3(y'))$;
- if $C(x, y)$, then $C'(U_1(x), U_2(y))$ and $U_3(U_2(y)) = y$ (in the multiple-valued case, $y \in U_3(U_2(y))$).

The first property means that if we start with an instance x of the problem A, build the corresponding instance $x' = U_1(x)$ of the problem A', and a solution y' to this new instance, then, by applying the algorithm U_3 to this solution y', we get a solution $y = U_3(y')$ to the original problem.

The second property means that if y is any solution to the original problem, then it can be obtained by applying the above procedure, when we use an appropriate solution $y' = U_2(y)$ to the corresponding instance $x' = U_1(x)$ of the problem A'.

We say that a problem P is *NP-hard* if it is as hard or harder than every problem from the class NP, i.e., in precise terms, if every problem from the class NP can be reduced to the problem P.

Propositional satisfiability: historically first example of an NP-hard problem. The first problem for which NP-hardness was proven was the problem of *propositional satisfiability*. In this problem, the input x is a *propositional formula*, i.e., a formula which can be obtained by Boolean ("true"-"false") variables $z_1, \ldots, z_v$ by using propositional operations "or" ($\vee$), "and" ($\&$), and "not" ($\neg$). An example of a propositional formula is $(z_1 \vee \neg z_3 \neg z_3) \,\&\, (\neg z_1 \vee z_3)$. The objective is to find a tuple $y = (z_1, \ldots, z_v)$ of Boolean values for which the given formula is true.

One can easily see that this is a problem from the class NP: if we have a formula x and a Boolean tuple y, then checking whether x is true for these values of z_i takes linear (thus polynomial) time – hence the corresponding property $C(x, y)$ is feasible. The length of the tuple does not exceed the length of the original formula, so here $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$ for a simple polynomial $P_\ell(n) = n$.

NP-hardness has actually been proven for a special class of propositional formulas in *Conjunctive Normal Form* (CNF), i.e., formulas of the type $C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_m$, where each *clause* has the form $a \vee \ldots \vee b$, and a, $\ldots$, b, are *literals*, i.e., variables z_i or their negations $\neg z_i$.

Most textbook proofs of satisfiability's NP-hardness are based on Turing machines. NP-hardness of satisfiability means that we can reduce every problem from the class NP to the satisfiability (and even to CNF-SAT). This is how NP-hardness of satisfiability is usually proven: by taking a general problem from the class NP and showing that this problem can be reduced to CNF-SAT.

These proofs are usually reasonably simple and straightforward, so at first glance, the proofs seem to be intuitively clear. However, a more detailed look shows that these proofs are not as intuitive as they may seem.

Indeed, by definition, a problem from the class NP means that we have a feasible (polynomial-time) algorithm $C(x, y)$, and the problem is: given x, find y for which the property $C(x, y)$ is satisfied. In the existing proofs, *polynomial-time* is understood as polynomial-time on a Turing machine.

Again, at first glance, this may seem reasonable. A Turing machine is what we would now call a simplified computer. A Turing machine consists of a tape (which is potentially infinite) which consists of cells. Each cell can be either empty or contain a symbol from the given list (e.g., 0 or 1). There is also a *head* which, at any given moment of time, is located near one of the cells. The head can be in one of the states from a given list. It starts at a special *start* state, with the input x written on a tape.

At each moment of time, depending on the current state h of the head and on the symbol s in the corresponding cell, the machine can do three things:

- overwrite the symbol s with a new symbol $s' = f(h, s)$ depending on h and s;
- change its state h to a new state $h' = g(h, s)$ depending on h and s; and
- depending on h and s, either stay at the same cell, or move one step to the left, or move one step to the right.

The machine stops when it reaches a special *halt* state. Once the Turing machine stops, what is written on the tape is considered to be the result of the computations. In other words, we say that a Turing Machine computes a function $y = F(x)$ if, every time we start it with the input x, it eventually halts and produces $y = F(x)$.

Why Turing machines are used in theory of computation. While Turing machine is a very primitive device, more like an old-fashioned tape recorder than a computer, it is known to be a universal computational device – in the sense that whatever complex computer can compute, a Turing machine can compute as well. This explains why Turing machines are used in theory of computation: they are much simpler than actual computers and, at the same time, they describe the exact same class of computable functions as more complex computers.

Because of this, if we want to prove that a function is not computable, there is no need to consider more complex devices: it is sufficient to prove that this function cannot be computed on a Turing machine.

Why the use of Turing machines in NP-hardness proofs is not fully satisfactory. As we have mentioned, Turing machines are perfect in describing what can be, in principle, computed. Of course, from the practical viewpoint, it makes no sense to build and use Turing machines: they are often very slow in comparison with the actual computers.

For example, if we are looking for an element e in a sorted array $a_1 \le \ldots \le a_n$, then on a real computer, we can use bisection and find the location i of the element e (i.e., the index for which $a_i = e$) in logarithmic time $t \le \log_2(n)$. In the beginning, we know that i is in the interval $\left[\underline{i}, \overline{i}\,\right]$, with $\underline{i} = 1$ and $\overline{i} = n$. On each iteration, once we know such an interval, we compute the midpoint $m = \lfloor (\underline{i} + \overline{i})/2 \rfloor$ and compare e with a_m.

- if $e = a_m$, the problem is solved, we found the index, it is m;
- if $e < a_m$, this means that $i < m$, so we replace the original interval $\left[\underline{i}, \overline{i}\,\right]$ with the half-size interval $[\underline{i}, m-1]$;
- if $e > a_m$, this means that $i > m$, so we replace the original interval $[\underline{i}, \overline{i}\,]$ with the half-size interval $\left[m+1, \overline{i}\,\right]$.

In both cases, we get an interval which is at least twice narrower than the original one. After k iterations, the interval's width is decreased by a factor of 2^k. So, after $k = \log_n(n)$ iterations, the original width $n-1$ is decreased at least by a factor of $2^k = n$. The resulting interval of width ≤ 1 cannot contain two different integers and thus, consists of a single integer i.

On a Turing machine, however, we start with the head located before a_1. When the desired value is located as $i = n$ (i.e., when $e = a_n$), the only way to find this location is to read the word a_n and to compare it with e. This means that the machine must move from a_1 all the way to a_n, thus passing by at least n cells. But a Turing machine can move at most one cell at a time. Thus, on a Turing machine,

search requires at least n computational steps – and for large n, the amount n is much larger than $\log_2(n)$:

- for $n = 10^3$, we have $\log_2(n) \approx 10$;
- for $n = 10^6$, we have $\log_2(n) \approx 20$;
- for $n = 10^9$, we have $\log_2(n) \approx 30$;
- etc.

So, when we require that $C(x, y)$ is computable in polynomial time on such a super-slow device as a Turing machine, we are unnecessarily limiting ourselves, since what we really want is properties $C(x, y)$ which can be computed in feasible time on a real computer.

Mathematically, it is OK to use Turing machines, but intuitively, it is desirable to consider more realistic computational devices. From the purely mathematical viewpoint, the situation is not as bad as it may seem: it turns out that, while Turing machines are indeed slower, they preserve computability in polynomial time. Many results show that if a function can be computed in polynomial time on a more complex computational device, then it can be also computed in polynomial time on a Turing machine.

With these additional results in mind, we can conclude that even if we understand feasible time as polynomial time on a realistic complex computer, every problem from the corresponding class NP can still be reduced to CNF-SAT. However, these additional results – that polynomial time on a computer translated into polynomial time on a Turing machine – results without which we do not get the desired reduction, are much more complex and less intuitive than the textbook proofs of SAT's NP-hardness. These additional results are therefore not included in the usual textbook analysis of NP-hardness – and so, the easiness of the usual proof kind of hides the fact that the actual proof of the desired result is much less intuitive than it seems at first glance.

What we do in this paper. To make the NP-hardness proof more convincing, we provide a new proof, a proof in which instead of a Turing machine we use a generic computational device.

This proof makes it clear what assumptions about space-time are needed in this derivation. We also show that these assumptions are necessary: if one of these assumptions is violated, then we can potentially solve satisfiability problems in polynomial time.

Comment. The main results of this paper were first announced in [3].

2. A New Proof that Satisfiability Is NP-Hard – Which Makes Space-Time Assumptions Behind This Result Explicit

What we start with. We have a problem from a class NP, we want to show how to reduce this problem to CNF-SAT. By definition, a problem from the class NP can be formulated as follows:

- we have a feasible predicate $C(x, y)$ (i.e., a feasible algorithm that always returns "true" or "false"),
- we have a polynomial $P_\ell(n)$, and
- we have a word x.

The problem is to find a word y for which $C(x, y) =$"true" and whose length $\mathrm{len}(y)$ is bounded by the polynomial of the length $\mathrm{len}(x)$ of the input word x, i.e., $\mathrm{len}(y) \le P_\ell(\mathrm{len}(x))$.

The algorithm $C(x, y)$ checks, in polynomial time, whether a given "guess" y is indeed a solution to the problem with the given x.

By definition, the algorithm $C(x, y)$ is feasible, i.e., on some computational device, its running time $t_C(x, y)$ is bounded by a polynomial of the length of its input: $t_C(x, y) \le P_C(\mathrm{len}(x) + \mathrm{len}(y))$ for an appopriate polynomial $P_C(n)$.

Computational device: component cells and their states. Let us analyze a computational device on which this algorithm $C(x, y)$ runs. A typical computational device consists of discrete *cells*. For example, each memory bit can be viewed as an elementary cell, a piece of wire that connects several elements on a chip can be viewed as a cell, etc.

Cells can be of different volume. Let us denote the smallest volume of a cell by ΔV.

Each cell can be in different *states*. For example, a memory bit can be in two states: 0 and 1. A wire can be in three states: not sending any signal, sending 0, and sending 1; etc. In principle, a physical object can be in infinitely many different states, but since all measurements are not accurate, we can only distinguish between finitely many states.

Different cells can have different number of possible states. Let us denote the largest number of possible states by S.

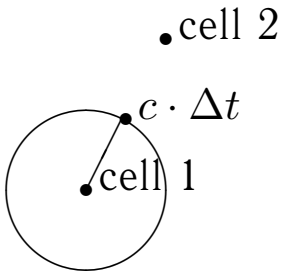
We will assume that the *time quantum* for this computational device is equal to Δt; this means that we can only consider the state of the computer at times 0, Δt, $2\Delta t$, etc.

First physical assumption: $v \le c$. We will take into consideration the fact that, according to modern physics, the speed of every process is limited by the speed of light c.

Dynamics of states. Let us use the above physical assumption to describe how a state of each cell changes with time.

Since the speed of communication is bounded by the speed of light, the state of the cell in the next moment of time can only be influenced by the states of the cells that are at a distance $\le r = c \cdot \Delta t$ from this desired cell: indeed, if a cell is further away, then during the time quantum, its influence will not be able to reach the original cell.

So, the state of the cell at the next moment of time $t + 1$ is determined only by the states of the cells inside the sphere of radius r, which is the "sphere of influence" of a given cell. We will call cells that can influence a given cell its *neighbors*.



Let us estimate the number N_{neigh} of neighbors.

By definition, ΔV is the smallest volume of a cell. This means that each cell occupies the volume that is greater than or equal to ΔV. Thus, N_{neigh} cells occupy the volume $\ge N_{\text{neigh}} \cdot \Delta V$. On the other hand, all these cells are located inside the sphere of radius r. The total volume inside the sphere is $\frac{4}{3} \cdot \pi \cdot r^3$; therefore, $N_{\text{neigh}} \cdot \Delta V \le \frac{4}{3} \cdot \pi \cdot r^3$, and hence,

$$N_{neigh} \le \frac{\frac{4}{3} \cdot \pi \cdot r^3}{\Delta V}.$$

Let us define the state of a cell i at moment t by $S_{i,t}$. Then, we can describe the evolution of the states as follows:

$$S_{i,t+1} = f_{i,t}(S_{i,t}, S_{j,t}, \ldots, S_{k,t}), \tag{1}$$

where the number of neighboring cells $(S_{j,t}, \ldots, S_{k,t})$ is $\le N_{\text{neigh}}$.

Towards reduction to propositional satisfiability: making all variables Boolean. We want to reduce our problem to propositional satisfiability. In propositional satisfiability, all the variables are Boolean. To get closer to this problem, let us represent each state by a sequence of Boolean (0-1) values.

To do that, we will enumerate all the states of each cell, describe each state by its ordinal number, and represent this ordinal number in the same manner as this number is represented in the computer, i.e., by a sequence of its binary digits.

Since the largest possible number of states of a cell is S, we can represent these states by integers from 0 to $S-1$. Let us denote by B the total number of binary digits in the binary representation of $S-1$. Then, all numbers smaller than $S-1$ require the same or smaller number of digits. Hence, we need B bits to describe each state.

By using k bits, we can describe 2^k different numbers; thus, to represent S different states by B bits, we must have $2^B \geq S$, i.e., $B \geq \log_2(S)$. Therefore, we can take, as B, the smallest integer for which this inequality is true, i.e., $B = \lceil \log_2(S) \rceil$.

Thus, each state $S_{i,t}$ can be represented as a sequence of B bits $s_{i,1,t}, s_{i,2,t}, \ldots, s_{i,b,t}, \ldots, s_{i,B,t}$. Here, the bit number b takes values $b = 1, \ldots, B$. From the equation (1), we can now conclude that the value of each of these variables at time $t+1$ depends on the values of the variables that describe neighboring cells at the time t:

$$s_{i,b,t+1} = f_{i,b,t}(s_{i,1,t}, \ldots, s_{i,B,t}, \ldots, s_{j,1,t}, \ldots, s_{j,B,t}, \ldots, s_{k,1,t}, \ldots, s_{k,B,t}). \qquad (2)$$

The total number of variables in the right-hand side is bounded by $\leq N_{\text{neigh}} \cdot B$.

Transforming the conditions into propositional form. All the variables in the expression (2) are Boolean, but the relation between these variables is not yet Boolean. To make it Boolean, let us express each formula (2) in Conjunctive Normal Form (CNF).

This can be done if we first translate a general formula F into a Disjunctive Normal Form, i.e., form of the type $D_1 \vee \ldots \vee D_m$, where each disjunction D_j is of the type $a \,\&\, \ldots \,\&\, b$, with literals $a, \ldots, b$. For that, we form a truth table for the formula F, i.e., describe its value (true or false) for all 2^k possible combinations of truth values of its k variables. A formula is true if and only if the inputs coincide with one of the tuples for which F is true. For example, if the formula F is true when x_1 and x_2 are both true and when x_1 and x_2 are both false, then F is equivalent to $(x_1 \,\&\, x_2) \vee (\neg x_1 \,\&\, \neg x_2)$.

To translate a formula F into CNF, we transform $\neg F$ into DNF, and then apply de Morgan rules $\neg(A \vee B) \equiv \neg A \,\&\, \neg B$, $\neg(A \,\&\, B) \equiv \neg A \vee \neg B$, and $\neg(\neg A) \equiv A$ to transform the negation of the DNF into a CNF. For example, if $\neg F \equiv (x_1 \,\&\, x_2) \vee \vee (\neg x_1 \,\&\, \neg x_2)$, then

$$F \equiv \neg((x_1 \,\&\, x_2) \vee (\neg x_1 \,\&\, \neg x_2)) \equiv \neg(x_1 \,\&\, x_2) \,\&\, \neg(\neg x_1 \,\&\, \neg x_2) \equiv$$

$$\equiv (\neg x_1 \vee \neg x_2) \,\&\, (\neg(\neg x_1) \vee \neg(\neg x_2)) \equiv (\neg x_1 \vee \neg x_2) \,\&\, (x_1 \vee x_2)).$$

This translation requires 2^k computational steps, where k is the number of variables. In our case, k is bounded by a constant $\leq N_{\text{neigh}} \cdot B$ which does not depend on the size of the input. Thus, 2^k is also bounded by a constant: $2^k \leq 2^{N_{\text{neigh}} \cdot B}$.

The translation gives us a propositional formula $F_{i,b,t}$ which describes the evolution of the b-th bit $s_{i,b,t+1}$ in the description of the i-th state.

Combining these formulas by "and", we can now describe the entire computation of $C(x, y)$ by a single formula. Indeed, given algorithm $C(x, y)$ and input x, it is necessary to describe that:

- The device operates correctly, i.e., all the states are changed accordingly. This is described by the following long formula:

$$F_{1,1,1} \,\&\, F_{1,1,2} \,\&\, \ldots \,\&\, F_{i,b,t} \,\&\, \ldots \,\&\, F_{N_{\text{cells}},B,T},$$

  where $1 \le i \le N_{\text{cells}}$, $1 \le b \le B$, $1 \le t \le T$, and T is the computation time (= total number of computational steps) in computing $C(x, y)$.

- We also need to describe that the input is the given one $x = x_1 x_2 \ldots$:

$$s_{i_1,b_1,1} = x_1 \,\&\, s_{i_2,b_2,1} = x_2 \,\&\, \ldots,$$

  where i_k is the cell that contains the k-th bit of the input x.

- Finally, we need to describe that the result of the computation is "true" in the "final" cell i_r: $s_{i_r,b_r,T} =$"true".

So, we use "and" to combine these formulas into a "long formula" F.

This is indeed a reduction to satisfiability. We have designed the algorithm U_1 that transforms each instance x of the original NP-problem into a propositional formula $x' = F$. This long formula describes the fact that:

- we started with given input x and some y,
- we performed the computation of the property $C(x, y)$, and
- we got $C(x, y)$ to be true.

Once we have a satisfying tuple y' for this formula, we read y from the bits describing the inputs y at moment 1. This is our algorithm U_3.

If we know the solution y to the original problem, then we can run a feasible algorithm for checking $C(x, y)$ and record all the values of all the bits of all the states at all moments of time. This is our algorithm U_2. One can easily check that this is indeed the desired reduction:

- if the tuple y' makes the propositional formula $C'(U_1(x), y')$ true, this means that for the input x and for the $y = U_3(y')$ which corresponds to y', the value $C(x, y)$ is also true, i.e., that y is indeed a solution to the original problem;
- vice versa, if y is a solution to the original problem, then for the Boolean tuple $y' = U_2(y)$ which describes the process of computing $C(x, y)$, the long Boolean formula $F = x' = U_1(x)$ holds, i.e., we have $C'(x', y')$.

The reduiction is feasible. To complete our proof, let us show that the designed algorithms U_i are indeed *feasible*, i.e., that their computation time is bounded by a polynomial of the input x.

This is clear for the algorithm U_3, in which we simply pick some bit values. Let us prove feasibility of the main reduction algorithm U_1. In this algorithm, we apply a constant number of computation steps to each of N_{cells} cells, to each of B bits, and to each of T moments of time. Thus, the computation time of this algorithm is proportional to the product $N_{\text{cells}} \cdot B \cdot T$. The number of bits B is a constant that does not depend on the length of the input at all.

Since $C(x, y)$ is a feasible algorithm, its computation time T is bounded by the polynomial of the length of its input. Each polynomial can be bounded, from above, by a simple polynomial $A \cdot n^k$: indeed, for all natural numbers n, we get

$$a_0 + a_1 \cdot n + \ldots + a_k \cdot n^k \le |a_0| \cdot n^k + |a_1| \cdot n^k + \ldots + |a_k| \cdot n^k = (|a_0| + |a_1| + \ldots + |a_k|) \cdot n^k.$$

Thus, we can always conclude that $T \le A \cdot (\text{len}(x) + \text{len}(y))^k$ for some A and k.

The length of y is limited by a polynomial $\text{len}(y) \le P_\ell(\text{len}(x))$. We can similarly conclude that $\text{len}(y) \le A' \cdot (\text{len}(x))^{k'}$. Thus,

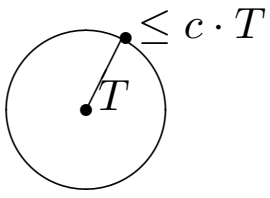
$$T \le A \cdot (\text{len}(x) + A' \cdot (\text{len}(x))^{k'})^k,$$

i.e., $T \le P(\text{len}(x))$, where we denoted $P(n) \stackrel{\text{def}}{=} A \cdot (n + A' \cdot n^{k'})^k$. So, the computation time T is indeed bounded by a polynomial of the length of the original input x.

Let us estimate the total number of cells N_{cells} that participate in this computation.

In principle, many cells could be computing, but only those cells can influence the final result which are not too far away, because if the cell is at a distance $> c \cdot T$ from the final monitor, then, even if it is sending all its information with the largest possible speed – the speed of light – the final cell will still not be able to receive this information before the computations are over.

Thus, it is sufficient to consider only the cells that are located within a distance $\le c \cdot T$ from the final cell, i.e., within a sphere of radius $R = c \cdot T$:



The volume of this sphere V is $\frac{4}{3} \cdot \pi \cdot (c \cdot T)^3$. Therefore, the total number of cells N_{cells} in this sphere is bounded by the ratio $\frac{V}{\Delta V}$, i.e.,

$$N_{\text{cells}} \le \frac{\frac{4}{3} \cdot \pi \cdot (c \cdot T)^3}{\Delta V} = \frac{4\pi \cdot c^3}{3 \cdot \Delta V} \cdot T^3.$$

Since T is bounded by a polynomial $T \le P(n)$, we conclude that

$$N_{\text{cells}} \le \frac{4\pi \cdot c^3}{3 \cdot \Delta V} \cdot (P(\text{len}(x)))^3.$$

The cube of a polynomial is also a polynomial; thus, the number of cells is bounded by the polynomial of $\text{len}(x)$.

Hence, the time $t_{U_1}(x)$ needed to compute the formula F is bounded by the product of three polynomials, and hence, also by a polynomial: $t_{U_1}(x) \le P_1(\text{len}(x))$ for some polynomial $P_1(n)$.

Similarly, the algorithm U_2 finishes in polynomial time. The reduction is feasible, so NP-hardness is proven.

3. Example

Description of a toy problem. To make the above construction clearer, let us illustrate it on the example of the following toy problem. In this problem, the input x is one bit, the output y is one bit, and the condition $C(x, y)$ that we want to achieve is $x = y$.

In other words, in this toy problem, we are given a bit x, and we want to find a bit y which satisfies the property $x = y$.

Computational device for checking the desired property. In accordance with the above proof, we need to start with a computational device that, given x and y, checks whether $x = y$. In the beginning, we have two cells: an x-cell that contains the input bit x and a y-cell which contains the bit y.

We also need a wire to transmit the information. We will thus send the content of the y-cell to the x-cell, and then use the x-cell to compare its original content with what is send by wire. Once the y-signal is sent, we no longer need it, so we can simply erase it (i.e., replace it with 0).

The whole computation process takes 3 moments of time:

- at moment $t = 1$, the x-cell contains x, the y-cell contains y, and the wire is inactive;
- at moment $t = 2$, the x-cell still contains x, the y-cell now contains 0, and the wire transmits the y signal;
- at moment $t = 3$, the x-cell contains 1 if $x = y$ and 0 otherwise, the y-cell contains 0, and the wire is again inactive.

In this computations process, we have 3 cells: the x-cell, the y-cell, and the wire. The x-cell has 2 possible states: 0 and 1, so one bit is sufficient to describe its state. According to the general notation, we will denote the state of this bit at moment t by $s_{1,1,t}$. Similarly, to describe the state of the y-cell, we need one but $s_{2,1,t}$.

The wire can be in 3 possible states: inactive, sending 0, and sending 1. Thus, to describe the state of the wire, we will need 2 bits. Let the first bit describe whether the wire is active or not, and the second bit describe the signal sent via an active wire. So, the state S_3 of the wire is either 00 (inactive), or 10 (sending 0), or 11 (sending 1).

In this case, $S = 3$, and the number of bits B needed to describe the state of each of the cells is $B = 2$.

Corresponding dynamics of states. Let us describe the above computations in terms of changing states.

At the first moment of time, the wire is inactive: $s_{3,1,1} = s_{3,2,1} = 0$.

At the second moment of time, the first cell retains its state, i.e., $s_{1,1,2} = s_{1,1,1}$. The second cell becomes 0: $s_{2,1,2} = 0$. The wire becomes active: $s_{3,1,2} = 1$, and the signal transmits exactly the bit originally stored in the y-cell: $s_{3,2,2} = s_{2,1,1}$.

At the third moment of time, the x-cell gets the value 1 if the value that was previously stored in this cell coincides with what was sent through the wire: $s_{1,1,3} = 1 \Leftrightarrow s_{1,1,2} = s_{3,2,2}$. The y-cell still contains 0: $s_{2,1,3} = 0$, and the wire is again inactive: $s_{3,1,3} = s_{3,2,3} = 0$.

Describing the dynamics in CNF terms. To describe the above formulas in the CNF terms, we need to translate the following formulas into CNF: $a = 0$, $a = 1$, $a = b$, and $a = 1 \Leftrightarrow b = c$. Let us use the above algorithm to translate these formulas into CNF one by one.

Translating $a = 0$ into CNF. For the formula $a = 0$, the truth tables for formula F itself and for its negation $\neg F$ take the form

a	F	$\neg F$
0	1	0
1	0	1

The formula $\neg F$ is true only when $a = 1$, so its DNF form is a. Thus, its CNF form is $\neg a$. This means, e.g., that the formula $s_{3,1,1} = 0$ becomes $\neg s_{3,1,1}$.

Translating $a = 1$ into CNF. For the formula $a = 1$, the truth tables for formula F itself and for its negation $\neg F$ take the form

a	F	$\neg F$
0	0	1
1	1	0

The formula $\neg F$ is true only when $a = 0$, so its DNF form is $\neg a$. Thus, its CNF form is a. This means, e.g., that the formula $s_{3,1,2} = 1$ becomes $s_{3,1,2}$.

Translating $a = b$ into CNF. For the formula $a = b$, the truth tables for formula F itself and for its negation $\neg F$ take the form

a	b	F	$\neg F$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

The formula $\neg F$ is true either when $a = 0$ and $b = 1$, or when $a = 1$ and $b = 0$. So, its DNF form is $(\neg a \,\&\, b) \vee (a \,\&\neg b)$. According to de Morgan laws, to get a negation F, we need to change all conjunctions to disjunctions, all disjunctions to conjunctions, and each literal by its negation. Thus, the CNF form is $(a \vee \vee \neg b) \,\&\, (\neg a \,\&\, b)$. This means, e.g., that the formula $s_{1,1,2} = s_{1,1,1}$ becomes $(s_{1,1,2} \vee \vee \neg s_{1,1,1}) \,\&\, (\neg s_{1,1,2} \vee s_{1,1,1})$.

Translating $a = 1 \Leftrightarrow b = c$ into CNF. Finally, for the formula $a = 1 \Leftrightarrow b = c$, the truth tables for formula F itself and for its negation $\neg F$ take the form

a	b	c	F	$\neg F$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

The corresponding DNF form for $\neg F$ is

$$(\neg a \,\&\, \neg b \,\&\, \neg c) \vee (\neg a \,\&\, b \,\&\, c) \vee (a \,\&\, \neg b \,\&\, c) \vee (a \,\&\, b \,\&\, \neg c),$$

so its negation F takes the CNF form

$$(a \vee b \vee c) \,\&\, (a \vee \neg b \vee \neg c) \,\&\, (\neg a \vee b \vee \neg c) \,\&\, (\neg a \vee \neg b \vee c).$$

This means that the formula $s_{1,1,3} = 1 \Leftrightarrow s_{1,1,2} = s_{3,2,2}$ takes the form

$$(s_{1,1,3} \vee s_{1,1,2} \vee s_{3,2,2}) \,\&\, (s_{1,1,3} \vee \neg s_{1,1,2} \vee \neg s_{3,2,2}) \,\&$$

$$\&\, (\neg s_{1,1,3} \vee s_{1,1,2} \vee \neg s_{3,2,2}) \,\&\, (\neg s_{1,1,3} \vee \neg s_{1,1,2} \vee s_{3,2,2}).$$

The resulting long formula. The resulting formula should include:

- the CNF forms of all the formulas describing the state's dynamics,
- the fact that the initial value x is given; for example, for $x = 0$, it should be $s_{1,1,1} = 0$, i.e., $\neg s_{1,1,1}$; and
- the fact that the result of checking the property $C(x, y)$ is "true"; according to our computation scheme, this result is stored in the x-cell at moment 3, so this requirement takes the form $s_{1,1,3} = 1$, i.e., $s_{1,1,3}$.

Thus, the corresponding long formula takes the following form:

$$\neg s_{3,1,1} \,\&\, \neg s_{3,2,1} \,\&$$

$$\&\, (s_{1,1,2} \vee \neg s_{1,1,1}) \,\&\, (\neg s_{1,1,2} \vee s_{1,1,1}) \,\&$$

$$\&\, \neg s_{2,1,2} \,\&\, s_{3,1,2} \,\&$$

$$\&\, (s_{3,2,2} \vee \neg s_{2,1,1}) \,\&\, (\neg s_{3,2,2} \vee s_{2,1,1}) \,\&$$

$$\&\, (s_{1,1,3} \vee s_{1,1,2} \vee s_{3,2,2}) \,\&\, (s_{1,1,3} \vee \neg s_{1,1,2} \vee \neg s_{3,2,2}) \,\&\, (\neg s_{1,1,3} \vee s_{1,1,2} \vee \neg s_{3,2,2}) \,\&$$

$$\&\, (\neg s_{1,1,3} \vee \neg s_{1,1,2} \vee s_{3,2,2}) \,\&$$

$$\&\, \neg s_{2,1,3} \,\&\, \neg s_{3,1,3} \,\&\, \neg s_{3,2,3} \,\&$$

$$\&\, \neg s_{1,1,1} \,\&\, s_{1,1,3}.$$

This formula says that for given $x = 0$ and for some y, we performed the checking of the property $C(x, y) \equiv x = y$ and concluded that the result of checking is "true". Once the formula is satisfied, we can find y as the original value of the y-cell, i.e., as $y = s_{2,1,1}$.

4. Space-Time Physics Behind the NP-Hardness Result

Space-time assumptions behind the proof. The above proof used two main assumptions about space-time:

- that there is a limitation on the communication speeds, and
- that the volume of a sphere of radius R is bounded by a polynomial of R.

Both space-time assumptions are crucial for the NP-hardness result. Let us show that both space-time assumptions are necessary not just for our *proof* of NP-hardness, but also for the NP-hardness *result* itself.

Indeed, if we do not have any limitations on the communication speed, if we can set up any communication speed with want, then we can exponentially increase communication speed with the increase in the input size, and thus, transform the exponential number of computation steps for an exhaustive-search solution to any NP problem into computations which require a constant time.

Similarly, if the volume of the sphere grows exponentially with the radius r, as $\exp(k \cdot r)$, then we can place exponentially many processors into a sphere, make each processor test one of the exponentially many possible solutions y, and let the processor which finds a solution report to the center. For example, for satisfiability, we have 2^v possible combinations $y = (z_1, \ldots, z_v)$, so to fit 2^v processor, we need a radius R for which $\dfrac{\exp(k \cdot r)}{\Delta V} = 2^n$, i.e., for which $r = a \cdot v + b$. The resulting time is composed of linear time for testing whether y is a solution, and linear time r/c to communicate the results – so we can solve satisfiability in linear time.

Comments.

- It is worth mentioning that in some physically reasonable models of space-time, we do have such an exponential dependence of the volume on radius, so in these models, we can potentially solve NP-hard problems in polynomial time; see, e.g., [1, 4–7].
- If the volume of the sphere grows slower than exponentially but faster than polynomially with the radius r, then, by parallelizing exhaustive search, we get an algorithm which is not polynomial, but it is still faster than all parallel algorithms corresponding to Euclidean geometry (in which the volume grows as r^3).

Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

The authors are thankful to all the students from the University of Texas at El Paso graduate Theory of Computation classes, especially to Monica Nogueira, and to all participants of the 2011 International Sun Conference on Teaching and Learning, for valuable suggestions.

References

1. Aaaronson S. NP-complete problems and physical reality // ACM SIGACT News, 2005. V. 36. P. 30–52.
2. Garey M.G. and Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, California : Freeman, 1979.
3. Kosheleva O. and Kreinovich V. NP-hardness proofs with realistic computers instead of Turing machines: Towards making Theory of Computation course more understandable and relevant // Abstracts of the 2011 International Sun Conference on Teaching and Learning. El Paso, Texas, March 10–11, 2011. P. 19.

4. Kreinovich V., Lakeyev A., Rohn J., and Kahl P. Computational Complexity and Feasibility of Data Processing and Interval Computations. Dordrecht : Kluwer, 1997.
5. Kreinovich V. and Margenstern M. In some curved spaces, one can solve NP-hard problems in polynomial time // Notes of Mathematical Seminars of St. Petersburg Department of Steklov Institute of Mathematics. 2008. V. 358. P. 224–250; reprinted in Journal of Mathematical Sciences. 2009, V. 158, N. 5, P. 727–740.
6. Margenstern M. and Morita K. NP problems are tractable in the space of cellular automata in the hyperbolic plane // Theoretical Computer Science. 2001. V. 259, N. 1–2. P. 99–128.
7. Morgenstein D. and Kreinovich V. Which algorithms are feasible and which are not depends on the geometry of space-time // Geombinatorics. 1995. V. 4, N. 3. P. 80–97.
8. Papadimitriou C. Computational Complexity, Reading. Massachusetts : Addison-Wesley, 1994.

ПРОСТРАНСТВЕННО-ВРЕМЕННЫЕ ПРЕДПОЛОЖЕНИЯ ДЛЯ ДОКАЗАТЕЛЬСТВА NP-ТРУДНОСТИ ЗАДАЧИ ВЫПОЛНИМОСТИ БУЛЕВЫХ ФОРМУЛ

О. Кошелева, к.ф.-м.н, доцент, e-mail: olgak@utep.edu
В. Крейнович, к.ф.-м.н, профессор, e-mail: vladik@utep.edu

Техасский университет в Эль Пасо, El Paso, TX 79968, США

Аннотация. Для решения некоторых задач известны вычислительно осуществимые алгоритмы. Другие задачи (например, о выполнимости булевых формул), известны как NP-трудные, то есть, при невыполнении условия P = NP (что по мнению большинства исследователей соответствует действительности), не существует вычислительно осуществимого алгоритма для решения произвольного случая соответствующей задачи. Обычно NP-трудность доказывается путем моделирования вычислений на машине Тьюринга — очень упрощённой версии компьютера. Хотя было убедительно показано, что машина Тьюринга является адекватным средством моделирования того, что может быть вычислено в принципе, гораздо менее очевидной является адекватность ее применения для моделирования эффективно вычислимых задач. Доказательства последнего факта существуют, но они достаточно сложны, и потому обычно не включаются в учебные пособия. Здесь мы приводим более понятное и убедительное доказательство NP-трудности, в котором вместо машины Тьюринга используется обобщенное вычислительное устройство. Это доказательство явно описывает пространственно-временные физические предположения, лежащие в основе понятия NP-трудности: что все скорости ограничены скоростью света, и что объем сферы растет не быстрее чем полиномиально в зависимости от ее радиуса. Если одно из этих предположений нарушается, доказательство становится неприменимым. Более того, в таких пространствах-временах задача выполнимости булевых формул может быть потенциально решена за полиномиальное время.

Ключевые слова: NP-трудные задачи, пространственно-временная модель, вычисления в искривлённом пространстве-времени.

Mathematical
Structures and Modeling
2014. N. 1(29). PP. 31–37

UDC 514.8

KNOWLEDGE GEOMETRY IS SIMILAR TO GENERAL RELATIVITY: BOTH MASS AND KNOWLEDGE CURVE THE CORRESPONDING SPACES

F. Zapata[1], Research Assistant Professor, Ph.D. (Computer Science),
e-mail: fazg74@gmail.com
V. Kreinovich[2], Ph.D. (Math.), Professor, e-mail: vladik@utep.edu

[1]Research Institute for Manufacturing and Engineering Systems (RIMES), USA
[2]Department of Computer Science, University of Texas at El Paso, El Paso,
TX 79968, USA

Abstract. In this paper, we explain and explore the idea that knowledge is similar to mass in physics: similarly to how mass curves space-time, knowledge curves the corresponding knowledge space.

Keywords: knowledge geometry, curved space-time.

1. Knowledge Geometry

How to detect that two objects are different? Let us start with the following sample situation. To observe wild parrots, an ornithologist sets up a feeder. Every morning, a parrot appears, and every evening, a parrot appears. These two parrots look similar. A natural question is: is the same parrot coming in the mornings and in the evenings, or these are two different parrots?

Natural idea: observe properties. A natural way to answer the above question is to observe various properties of these two parrots. For example, if the morning parrot has a red spot, and the evening parrot does not, this means that they are different birds. If the wing span of evening parrot is smaller than the wing span of the morning parrot, they are different birds.

Without losing generality, we can consider binary properties. In general, properties can be of binary (yes-no) type, e.g., "has a red spot". We can also consider numerical properties like the wing span. In the computer, whatever information we have can be represented in terms of binary digits (bits), i.e., in terms 0s and 1s:

- on the one hand, the property of having or not having a red spot is represented by a single bit;
- on the other hand, the numerical value of the wingspan is represented by several bits.

Instead of considering all these different types of properties, let us simply consider all the information as the sequence of bits. From this viewpoint, measuring the wing span means determining the values of several binary properties:

- the first of these binary properties is the value of the 1st bit in the binary expansion of the measured value,
- the second of these binary properties is the value of the 2nd bit in the binary expansion of the measured value,
- etc.

Representing knowledge about each object. Let N be the total number of binary properties. Let us denote these properties by $P_1, \ldots, P_N$. For each object a and for each property P_i, we have the following three possibilities:

- the first possibility is that we know that the property P_i holds for the object a, i.e., that the value $P_i(a)$ is "true"; in the computers, the value "true" is usually represented by 1;
- the second possibility is that we know that the property P_i does not hold for the object a, i.e., that the value $P_i(a)$ is "false"; in the computers, the value "false" is usually represented by 0;
- the third possibility is that we do not know whether the object a satisfies the property P_i; let us denote this case by $P_i(a) = *$.

Gauging difference between the two objects. In the first approximation, it is reasonable to gauge the difference between the two objects by the number of properties in which these two objects differ. In other words, if the object a is characterized by the values $P_1(a), \ldots, P_N(a)$, and the object b is characterized by the values $P_1(b), \ldots, P_N(b)$, then we take $D(a,b) \stackrel{\text{def}}{=} \sum_{i=1}^{N} d(P_i(a), P_i(b))$, where we define:

- $d(v, v') = 1$ if we know that the values v and v' are different, i.e., when either $v = 0$ and $v' = 1$, or $v = 1$ and $v' = 0$; and
- $d(v, v') = 0$ for all other pairs v and v'.

Some properties may be more important, some less important. To take the difference in importance into account, we can assign weights $w_i > 0$ to different properties P_i, so that differences in the more important properties will be added with more weight. In other words, it makes sense to consider the following formula for the distance between the two objects:

$$D(a,b) = \sum_{i=1}^{N} w_i \cdot d(P_i(a), P_i(b)). \tag{1}$$

Comment. The idea of a reasonable knowledge-based distance between objects is not new; it has been described, e.g., in [8].

2. Both Mass and Knowledge Curve the Corresponding Spaces: An Idea

Observation: additional knowledge increases distances. Let us analyze what happens to thus defined knowledge-based distance between objects if we gain additional knowledge. Gaining additional knowledge means that for some properties and for some objects:

- where we previously did not know whether this property is satisfied or not (i.e., we had the unknown value $*$),
- now we know that this property is satisfied or that it is not satisfied.

How does this change in knowledge affect the distance $D(a,b)$, i.e., a weighted sum of the distances $d(P_i(a),P_i(b))$?

If for some property P_i, we had $d(P_i(a),P_i(b))=1$, this means that one of the values $P_i(a)$ and $P_i(b)$ was equal to 0 ("false") and another to 1 ("true"). In other words, if $d(P_i(a),P_i(b))=1$, this means that we already know both truth values $P_i(a)$ and $P_i(b)$. For this property, the additional knowledge will not change these truth values and thus, the distance $d(P_i(a),P_i(b))=1$ will remain unchanged. So:

- values $d(P_i(a),P_i(b))=1$ remain unchanged, while
- the values $d(P_i(a),P_i(b))=0$ may increase to 1: e.g., if some truth values were unknown $P_i(a)=P_i(b)=*$, and we found out that the property P_i is false for a and true for b.

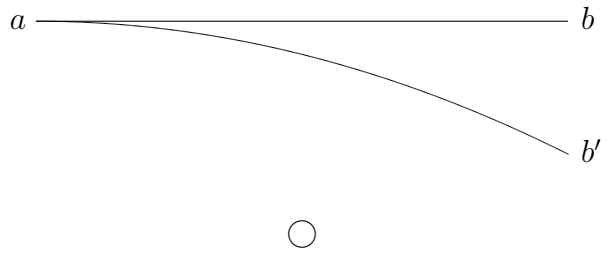
In both cases, the value $d(P_i(a),P_i(b))$ either remains the same or increases – and, as a result, the distance $D(a,b)$ between the two objects either remains the same (if we did not learn any new information about their difference) or increases. In short, in general, *additional knowledge increases distance*.

Conclusion: shortest paths change. In the vicinity of the object c for which we gained the new knowledge, distance increases. As a result, if originally, the shortest path between some objects a and b passed through c (or near c), its length increases – and an alternative path which does not pass near c becomes now the shortest.

Example. The changing of the shortest path can be illustrated on the example of traffic. Let us measure the distance $d(a,b)$ between the two points by the time that it takes to travel between the locations a and b. In the absence of heavy traffic (e.g., at night), the shortest path, e.g., between a location to the South of downtown and a location to the North of it goes through downtown.

However, during the rush hours, the traffic in downtown is usually congested. As a result, the path through downtown becomes much longer. In this case, a different path will be the shortest: the one which goes around downtown.

This is similar to curving of space-time. This phenomenon is similar to the geometric interpretation of gravity in General Relativity; see, e.g., [4, 6]. In the absence of masses, a body follows the straight line – which happens to be the shortest path between the initial position a and the final position b. In the presence of a heavy mass (e.g., the Earth or the Sun), the bodies start falling on this mass.



Shortest paths are a specific example of *geodesics*, i.e., paths on which the length is stationary (e.g., the smallest or the largest). In General Relativity, particles follow the geodesics in space-time. Because of the curving of space-time (largely time), spatial projections of geodesics are not straight lines – i.e., the actual path of a body in a curved space is different from the shortest path as measured by the spatial distance.

3. Both Mass and Knowledge Curve the Corresponding Spaces: Towards a Quantitative Description

Towards a quantitative description. In the ideal case, we know the exact values $v_i(a)$ of all the quantities describing an object a. In such an ideal situation, it is easy to check whether two objects a and b are identical or not:

- if for all quantities, we get $v_i(a) = v_i(b)$, then the objects a and b are indistinguishable – i.e., in effect, a and b are the same object;
- if for some quantity i, we have $v_i(a) \neq v_i(b)$, this means that the objects a and b are different from each other.

In practice, the values come from measurements, and measurements are never 100% exact; there is always a measurement error due to which the measured value $\widetilde{v}_i(a)$ is, in general, somewhat different from the actual (unknown) value $v_i(a)$ of this quantity; see, e.g., [7]. In many cases, we know the probability distribution of the measurement error $\Delta v_i(a) \stackrel{\text{def}}{=} \widetilde{v}_i(a) - v_i(a)$. Often, this distribution is Gaussian; this is in line with the fact that usually, many different phenomena contribute to the

measurement error, and, according to the Central Limit Theorem, the distribution of the sum of a large number of small independent random variables is close to Gaussian; see, e.g., [10]. It is usually assumed that the bias (mean error) has been compensated, so the mean value of the measurement error is 0.

Because of the measurement errors, even if a and b are the same object, i.e., even if the actual values of the corresponding quantities coincide $v_i(a) = v_i(b)$, the measured values $\widetilde{v}_i(a) = v_i(a) + \Delta v_i(a)$ and $\widetilde{v}_i(b) = v_i(b) + \Delta_i(b)$ will be, in general, slightly different. Vice versa, when the objects differ and $v_i(a) \neq v_i(b)$ for some i, it is possible that we will have $\widetilde{v}_i(a) = v_i(a) + \Delta v_i(a) = v_i(b) + \Delta_i(b) = \widetilde{v}_i(b)$, i.e., the observed values will be the same.

So, based on the measurement results, we can never know for sure whether the two objects a and b are identical or not, we can only make this conclusion with a certain probability. Specifically, based on the known probability distribution of the measurement error, we can estimate what is the probability that the observed values $\widetilde{v}_i(a)$ and $\widetilde{v}_i(b)$ come from the same object.

- When this probability is high, we conclude that the objects are most probably the same.
- When this probability is low, we conclude that the objects are different.

It is therefore reasonable to define the distance between the objects a and b in such a way that the larger the distance, the smaller the corresponding probability. Namely:

- we assume that we observe the exact values of the corresponding quantities $v_i(a)$ and $v_i(b)$, and
- we compute the probability that the difference between these values can be explained by the measurement errors.

Gaussian distributions correspond to Riemannian geometry, more general distributions to more general (Finsler) geometry. Let us assume that we observe the values $v_i(a) \neq v_i(b)$. Under the hypothesis that the difference between these values can be explained by the measurement error, i.e., that $v_i(a) = v_i + \Delta v_i(a)$ and $v_i(b) = v_i + \Delta v_i(b)$ for some values v_i, we conclude that $\Delta v_i(a) - \Delta v_i(b) = d_i \stackrel{\text{def}}{=} v_i(a) - v_i(b)$.

Measurement errors $\Delta v_i(a)$ and $\Delta v_i(b)$ corresponding to different measurements are usually independent. So, when the measurement errors Δh_j are normally distributed (with 0 mean), the difference $d_i = \Delta v_i(a) - \Delta v_i(b)$ is also normally distributed (and also with 0 mean). The corresponding probability density function has the form $\text{const} \cdot \exp(-c_i \cdot (d_i)^2)$ for an appropriate value c_i.

Since we assumed that measurement errors corresponding to different measurements are independent, we conclude that the overall probability that the objects a and b are different is equal to the product of the corresponding probabilities, i.e., to the value $\prod_i \text{const} \cdot \exp(-c_i \cdot (d_i)^2) = \text{const} \cdot (-s)$, where $s \stackrel{\text{def}}{=} \sum_i c_i \cdot (d_i)^2$.

So, the probability is uniquely determined by the weighted sum s. In general, each object a can be characterized by the values of the corresponding parameters $a_1, \ldots, a_n$, and the quantities $v_i(a)$ smoothly depend on these parameters. As a result, for close objects $a = (a_1, \ldots, a_n)$ and

$$b = a + \Delta a = (a_1 + \Delta a_1, \ldots, a_n + \Delta a_n),$$

we get

$$d_i = v_i(a + \Delta a) - v_i(a) = \sum_{j=1}^{n} D_{ij} \cdot \Delta a_j + o(\Delta a),$$

where $D_{ij} \stackrel{\text{def}}{=} \dfrac{\partial v_i}{\partial a_j}$, and thus,

$$s = \sum_i c_i \cdot (d_i)^2 = \sum_i c_i \cdot \left(\sum_{j=1}^{n} D_{ij} \cdot \Delta a_j \right)^2 + o(\Delta a).$$

Therefore, s is a quadratic function of Δa_j, $s = \sum\limits_{j=1}^{n} \sum\limits_{k=1}^{n} g_{jk} \cdot \Delta a_j \cdot \Delta a_k$ for some g_{jk}. Thus, the value s is naturally related to the Riemannian distance

$$d(a, a + \Delta a) = \sqrt{\sum_{j=1}^{n} \sum_{k=1}^{n} g_{jk} \cdot \Delta a_j \cdot \Delta a_k}.$$

For more general probability distributions, we get a more general formula for the corresponding metric – i.e., in the smooth case, a general Finsler space [1–3, 5, 9] instead of a specific Riemannian space.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

The authors are thankful to Luis Rocha and Ron Yager for stimulating discussions.

References

1. Anotonelli P.L. et al. (eds.) Handbook of Finsler geometry. Vols. 1, 2. Boston : Kluwer Academic Publishers, 2003.
2. Bao D., Chern S.S. and Shen Z. An Introduction to Riemann-Finsler Geometry. Berlin : Springer-Verlag, 2000.
3. Chern S. Finsler geometry is just Riemannian geometry without the quadratic restriction // Notices of the American Mathematical Society. 1996. V. 43. P. 959–963.
4. Feynman R., Leighton R. and Sands M. The Feynman lectures on physics. Boston, Massachusetts : Addison Wesley, 2005.

5. Finsler P. Über Kurven und Flächen in allgemeinen Räumen. PhD Dissertation. Göttingen University, 1918; reprinted by Birkhäuser, 1951.
6. Misner C.W., Thorne K.S. and Wheeler J.A. Gravitation. W. H. Freeman : New York, 1973.
7. Rabinovich S. Measurement Errors and Uncertainties: Theory and Practice. New York : American Institute of Physics, 2005.
8. Rocha L.M. Combination of evidence in recommendatuon systems characterized by distance functions // Proceedings of the 2002 International IEEE Conference on Fuzzy Sets and Systems FUZZ-IEEE'2002. Honolulu, Hawaii, May 2002. P. 203–208.
9. Shen Z. Lectures on Finsler Geometry. Singapore : World Scientific Publishers, 2001.
10. Sheskin D.J. Handbook of Parametric and Nonparametric Statistical Procedures. Boca Raton, Florida : Chapman and Hall/CRC Press, 2011.

ГЕОМЕТРИЯ ИНФОРМАЦИОННОГО ПРОСТРАНСТВА ПОДОБНА ОБЩЕРЕЛЯТИВИСТСКОЙ ГЕОМЕТРИИ: И МАССА, И ИНФОРМАЦИЯ ИСКРИВЛЯЮТ СООТВЕТСТВУЮЩИЕ ПРОСТРАНСТВА

Ф. Запата[1], научный ассистент, Ph.D. (Computer Science), e-mail: fazg74@gmail.com
В. Крейнович[2], профессор, к.ф.-м.н., e-mail: vladik@utep.edu

[1]Научно-исследовательский институт производственных и инженерных систем (RIMES), Техасский университет в Эль-Пасо, Эль-Пасо, США
[2]Факультет компьютерных наук, Техасский университет в Эль-Пасо, Эль-Пасо, США.

Аннотация. В этой статье рассматривается идея о том, что информация подобна физической массе: подобно тому, как масса искривляет пространство-время, информация искривляет информационное пространство

Ключевые слова: геометрия знаний, искривлённое пространство-время.

Mathematical
Structures and Modeling
2014. N. 1(29). PP. 38–42

UDC 514.8

ON HYPERBOLIC MOTION IN TWO HOMOGENEOUS SPACE TIMES (RESEARCH ANNOUNCEMENT)

A.V. Levichev[1], Doctor of Mathamatics, e-mail: levit@bu.edu
O. Simpson[2], Student, e-mail: osimpson@bu.edu
B. Vadala-Roth[2], Student, e-mail: benLvr@bu.edu

[1]Sobolev Institute of Mathematics, Novosibirsk, Russia

[2]Department of Mathematics and Statistics, Boston University, USA

Abstract. In 1960 W. Rindler generalized the concept of hyperbolic motion to an arbitrary Lorentzian manifold and studied this motion in the case of de Sitter space-time. We specify Rindlers (non-linear) system of differential equations in the case of the Segals compact cosmos D (which is locally isometric to the Einstein static universe), and in the case of the Heraclitian space-time F. This F is the real Lie group U(1,1) with a certain bi-invariant metric on it whereas D is U(2) with a bi-invariant metric on it. In each case, we present a particular solution to the Rindlers system.

Keywords: Lie group, hyperbolic motion, Lorentzian manifold.

1. Motivation and Introduction

This work is partly motivated by publication [7] where W. Rindler says (p.2082) that "in the special theory of relativity the term 'hyperbolic motion' is commonly applied to the motion of a test particle moving with constant proper acceleration along a straight line in a suitable Galilean frame of reference. (Proper acceleration is the acceleration relative to the instantaneous Galilean rest frame.) Hyperbolic motion was first noted by Minkowski [4] and was further studied by Born [2], who also coined its name. This name derives from the fact that the plot of distance against time is a rectangular hyperbola (see equation (9) of [7]). By the same terminology the classical motion with constant acceleration is 'parabolic'."

Let us notice that (as it is obvious from [7, p.2083] calculations) a Galilean frame (from above) is another name for an inertial frame in special relativity theory. W. Rindler generalizes the concept of hyperbolic motion to a general space-time ([7, p.2083]) by which (as it becomes clear from [7, p.2084]) Rindler understands a manifold with Lorentzian metric on it. In his article he only solves the proposed equations (that is, our (1.4) below) for the particular case of de Sitter space-time ([7, p.2085]).

Again, the equations mentioned above are deduced as the ones which describe the motion of a uniformly accelerated (point-like) particle. In order to present these equations let us first describe our notation(s). Let a particle (in a portion of

space-time with coordinates x^0, x^1, x^2, x^3) have world line

$$x^m = x^m(p) \tag{1.1}$$

where parameter p is the arc length. Velocity U and acceleration A are defined ([7, p.2084]) as follows:

$$U^m = \frac{dx^m}{dp}, \tag{1.2}$$

$$A^m = \frac{DU^m}{dp} \tag{1.3}$$

where the uppercase D is an indication of covariant differentiation. The Rindlers equations ([7, (16) on p.2084]) read as follows:

$$\frac{DA^m}{dp} = a^2 U^m \tag{1.4}$$

where a (positive) constant a is the magnitude of the acceleration A given by (1.3).W. Rindler only analyzed these equations for the de Sitter world (which is a well-known example of a homogeneous space-time).

We introduce below two homogeneous space-times, D and F. Each of them can be viewed as a four-dimensional real Lie group equipped with certain bi-invariant metric of Lorentzian signature. In both cases we specify equations (1.4) and present particular solutions.

2. Space-Times D and F

The Lie groups U(2) and U(1,1) can each be defined as the totality of all 2 by 2 matrices Z (with complex entries allowed) which satisfy

$$Z^* s Z = s \tag{2.1}$$

where s is the unit matrix in case of U(2), and in case of U(1,1) s is the diagonal two by two matrix with entries 1,-1. To make U(2) and U(1,1) space-times, we only need to supply them with metric tensors of Lorentzian signatures: see Section 3.1 (respectively, 3.2) of [3] for these (and more) details on U(2) = D (respectively, U(1,1) = F). At this point in our article, we only need to know that a left-invariant orthonormal basis of vector fields X_0, X_1, X_2, X_3 is chosen on D, and of H_0, H_1, H_2, H_3 on F. In each case, the choice of signature is $+$, $-$, $-$, $-$. The corresponding metric is bi-invariant (on each of the two groups). The space-time F is known as a tachyonic fluid, [3, Theorem 10] (see more details in that Theorem 10 which justify the $Heraclitian$ world name).

To specify equations (1.4), we will use the following result from [5, Section 8]: in terms of such a basis of vector fields, each Christoffel symbol G^k_{ij} is nothing but one-half of the structure constant C^k_{ij}.

Recall that the structure constants C^k_{ij} are detected from the commutation relations

$$[X_i, X_j] = C_{ij}^k X_k \tag{2.2}$$

(where summation in k goes from k = 0 to k = 3). The Christoffel symbols G_{ij}^k are coefficients in the decomposition of the covariant derivative $D_i(X_j)$ with respect to this very basis of four basic vector fields. Here $D_i(X_j)$ denotes covariant derivative of vector field X_j w.r.t. vector field X_i. Clearly, we have to use vector fields H_i (rather than X_i) when we deal with space-time F. The commutation tables for the two sets of basic vector fields are given in Section 8 of [3].

We can now think of the vector field U from (1.2) as a vector field on the entire space-time, D or F. Each curve of constant acceleration is thus an integral curve of U. By U_0, U_1, U_2, U_3 we now understand coordinates of U with respect to the (above chosen) basis on D (or F). By f′ , f″, etc. below we understand the corresponding (ordinary) derivative of a function f(p) on an integral curve.

Proposition 1. *In the case of D, the vector field U from (1.4) is a solution of the system*

$U_0'' = a^2 U_0$,
$U_1'' + U_2' U_3 - U_2 U_3' = a^2 U_1$,
$U_2'' + U_1 U_3' - U_1' U_3 = a^2 U_2$,
$U_3'' + U_1' U_2 - U_2' U_1 = a^2 U_3$.

The 'F-system' reads as follows:

$U_0'' + U_1' U_2 - U_2' U_1 = a^2 U_0$,
$U_1'' + U_0' U_2 - U_2' U_0 = a^2 U_1$,
$U_2'' + U_1' U_0 - U_0' U_1 = a^2 U_2$,
$U_3'' = a^2 U_3$.

The **proof** is omitted: it is a straightforward application of Milnors result [5, Section 8] to Rindlers system (1.4).

In this article we only discuss the following two solutions. From time to time, we use the abbreviations $C = \cosh(ap), S = \sinh(ap)$.

Proposition 2. *The vector field*

$$U = \{\cosh(ap), \sinh(ap), 0, 0\} = CX_0 + SX_1 \tag{2.3}$$

satisfies the D-system, whereas

$$U = \{\cosh(ap), 0, 0, 0\} = CH_0 \tag{2.4}$$

satisfies the F-system.

The **proof** is an easy verification.

Our next goal is to present the corresponding D-curve which (when $p = 0$) passes through the neutral element of D = U(2). To do so, we use ('excessive') coordinates u_{-1}, u_0, u_1, u_2, u_3, u_4 on D (see [6, p.92]) rather than deal with matrices defined by (2.1).

Theorem 1. *The curve*

$$Z(p) = \left\{\cos\left[\frac{S}{a}\right], \sin\left[\frac{S}{a}\right], \sin\left[\frac{(C-1)}{a}\right], 0, 0, \cos\left[\frac{(C-1)}{a}\right]\right\} \tag{2.5}$$

is an integral curve of vector field (2.3). It passes (when p = 0) through the neutral element of D = U(2).

The **proof** amounts to the direct calculation, and to the careful application of [6, p.92 and p.95] data.

Remark 1. The curve (2.5) is a subset of the 2-dimensional torus T in D: T is defined by equations $u_{-1}^2 + u_0^2 = 1$, $u_1^2 + u_4^2 = 1$, $u_2 = u_3 = 0$.

To deal with the F-system, let us recall the conformal embedding *E* of F into D. This *E* is (indirectly) defined in the proof of [3, Theorem 6]. From that last proof, it follows that an inverse map *G* is defined on the orbit of the four-dimensional group which is generated by vector fields H_0, H_1, H_2, H_3 (which are now viewed as vector fields on D). Here we have the orbit of the U(2) neutral element in mind. A directly defined analogue of *G* is given by formula (3.4) of [1].

Once again, we can use coordinates u_{-1}, u_0, u_1, u_2, u_3, u_4.

Theorem 2. *The image of the curve*

$$Z(p) = \left\{\cos\left[\frac{S}{a}\right], \sin\left[\frac{S}{a}\right], 0, 0, 0, 1\right\} \tag{2.6}$$

under G is an integral curve of vector field (2.4).

The **proof** is based on how the vector field $U = CH_0$ is expressed in terms of fields X_0, X_1, X_2, X_3 (see section 3.2 of [3]) and on the conformal embedding *E* of F into D. Once again, it involves application of [6, p.92 and p.95] data and direct calculation.

More details of the Theorem 2 proof are to be presented elsewhere.

Remark 2. It is of interest to consider more details on the F-system within the space-time F itself. To do so, one can start with an explicit formula for the conformal mapping *G* from above.

Remark 3. The curve (2.6) is a geodesic in the space-time D but its image under *G* is not a geodesic in F otherwise it would not be a curve of constant nonzero acceleration in F. In this regard, John Stachel suggested studying how curves of constant acceleration transform when a conformal transformation is applied. Such (and other) questions arise naturally in the scope of his *Unimodular Conformal Projective Relativity* (UCPR), see [8].

3. Acknowledgments

In regards to the second author, this project was funded (in part) by Boston University UROP. The work of both the second and the third authors was funded (in part) by the BU GUTS 2012-2013 grant.

References

1. Akopyan A.A., Levichev A.V. The Sviderskiy formula and a contribution to Segals chronometry // Mathematical Structures and Modeling. V. 25 (2012). P. 44–51.
2. Born M. Ann. Physik 30, 1 (1909), Sec. 5.
3. Levichev A.V. Pseudo-Hermitian realization of the Minkowski world through the DLF-theory // Physica Scripta. 83 (2011), N. 1. P. 1–9.
4. Minkowski H. Physik. Z. 10, 104 (1909).
5. Milnor J. Curvatures of Left Invariant Metrics on Lie Groups // Advances in math. V. 21 (1976), N. 3. P. 293–329.
6. Paneitz S.M. and Segal I.E. Analysis in space-time bundles I: General considerations and the scalar bundle // Journal of Functional Analysis. V. 47 (1982). P.78–142.
7. Rindler W. Hyperbolic Motion in Curved Space Time // Phys. Rev. 1960. V. 119, Issue 6. P. 2082–2089.
8. Stachel J. Conformal and projective structures in general relativity, Gen. Relativ. &Gravit. (2011) 43: 3399-34-09.

О ГИПЕРБОЛИЧЕСКОМ ДВИЖЕНИИ В ДВУХ ОДНОРОДНЫХ ПРОСТРАНСТВАХ-ВРЕМЕНАХ (АНОНС ИССЛЕДОВАНИЯ)

А.В. Левичев[1], с.н.с., д.ф.-м.н., профессор, e-mail: levit@bu.edu
О. Симпсон[2], студент, e-mail: osimpson@bu.edu
В. Вадала-Рот[2], студент, e-mail: benLvr@bu.edu

[1]Институт Математики им. С.Л.Соболева СО РАН, Новосибирск
[2]Факультет математики и статистики, Бостонский университет, США

Аннотация. В 1960 году В. Риндлер обобщил понятие гиперболического движения для произвольного лоренцева многообразия и изучил это движение в случае пространства-времени де Ситтера. Мы определяем систему (нелинейных) дифференциальных уравнений Риндлера в случае Сигаловского компактного пространства-времени D (которое локально изометрично статической Вселенной Эйнштейна) и в случае Гераклитианского пространства-времени F. При этом пространство-время F является вещественной группой Ли U(1,1) с определённой на ней биинвариантной метрикой, в то время как мир D представляет собой группу Ли U(2) с также определённой на ней биинвариантной метрикой. Для каждого случая представлено частное решение системы Риндлера.

Ключевые слова: группа Ли, гиперболическое движение, лоренцево многообразие.

Математические структуры и моделирование
2014. №1(29). С. 43–64

УДК 004.93

ОБЗОР МЕТОДОВ ИДЕНТИФИКАЦИИ ЧЕЛОВЕКА ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА

Н.П. Гришенкова[1], инженер-программист, e-mail: natalia.grishenkova@gmail.com
Д.Н. Лавров[2], доцент, к.т.н., e-mail: Dmitry.Lavrov72@gmail.com

[1]Омский научно-исследовательский институт приборостроения

[2]Омский государственный университет им. Ф.М. Достоевского

Аннотация. В статье представлен обзор современных методов идентификации и верификации человека по радужной оболочке глаза. Вначале излагаются хорошо известные факты о строении глаза и его радужной оболочки. Далее описываются основные принципы работы биометрических систем и показатели качества их работы. Следующий раздел описывает известные методы: выделения зрачка, выделения радужной оболочки, преобразования к эталону, определения границ радужки. Затем описываются алгоритмы определения ключевых точек на основе вейвлет-преобразований. В частности, описывается применение фильтра Габора. Описаны методы сравнения с эталоном на основе расстояния Хэмминга и метода проекционной фазовой корреляции. Обсуждаются достоинства и недостатки описанных подходов.

Ключевые слова: радужка, преобразование Габора, выделение границ, верификация, идентификация, распознавание.

Введение

В современном мире чрезвычайно остро стоит проблема защиты информации. На сегодняшний день использование парольной системы идентификации уже не удовлетворяет требованиям безопасности. Чтобы обеспечить достаточный уровень безопасности, пароль должен быть сложным. Сложность пароля обеспечивается совместным использованием букв (как строчных, так и прописных), цифр и знаков и его длиной. Причём для каждого информационного ресурса рекомендуется создавать свой пароль. Очень часто это приводит к тому, что пользователь попросту забывает такой пароль, и для его восстановления необходимо идентифицировать человека, обратившегося в службу поддержки. Также парольные системы идентификации никак не защищены от использования паролей третьими лицами для несанкционированного доступа. Чтобы удовлетворить все растущие потребности в повышении уровня безопасности информации, все чаще для идентификации личности используются методы биометрии. При биометрической аутентификации используются уникальные характеристики отдельно взятого человека. Это могут быть как врождённые

признаки (отпечатки пальцев, радужная оболочка глаза), так и приобретённые характеристики (почерк, голос или походка).

В данной работе рассматриваются методы идентификации человека по радужной оболочке глаза. Такой выбор не случаен. Идентификация по радужной оболочке глаза является одним из наиболее точных и надёжных способов биометрической идентификации. Это связано с тем, что радужная оболочка глаза имеет особую структуру, которая уникальна для каждого человека. При этом методы идентификации по радужной оболочке глаза являются бесконтактными.

Ещё одним достоинством подобных методов является то, что ношение контактных линз, даже цветных, не является проблемой. В процессе идентификации никак не учитывается информация о цвете глаз. Это делает применение подобных систем идентификации и верификации ещё более привлекательным.

Цель: изучить методы идентификации и верификации человека по радужной оболочке глаза.

Задачи:

1. Изучить характеристики биометрических систем;
2. Изучить методы определения границ зрачка и радужной оболочки глаза;
3. Изучить методы извлечения индивидуальной информации из изображения для идентификации и верификации человека;
4. Изучить способы принятия решения в процессе идентификации;
5. Изучить достоинства и недостатки систем идентификации и верификации человека по радужной оболочке глаза.

1. Физиология

1.1. Строение глаза

Глазное яблоко имеет форму, близкую к шаровидной. В нем различают передний и задний полюсы; прямая линия, соединяющая их, называется осью глазного яблока. Глазное яблоко состоит из капсулы, которая окружает его снаружи, и ядра. Капсула построена из трёх оболочек: наружной (фиброзной), средней (сосудистой) и внутренней (сетчатки). В состав ядра входят проводящие и преломляющие свет среды: водянистая влага, хрусталик и стекловидное тело (рисунки 1 и 2[1]) [3].

В наружной, или фиброзной, оболочке глазного яблока различают два отдела: роговицу и склеру.

Средняя, или сосудистая, оболочка глазного яблока содержит большое количество сосудов и пигмент. В ней принято различать три части: собственно сосудистую оболочку, ресничное тело и радужку.

[1] `http://www.sportmedicine.ru/eye.php`

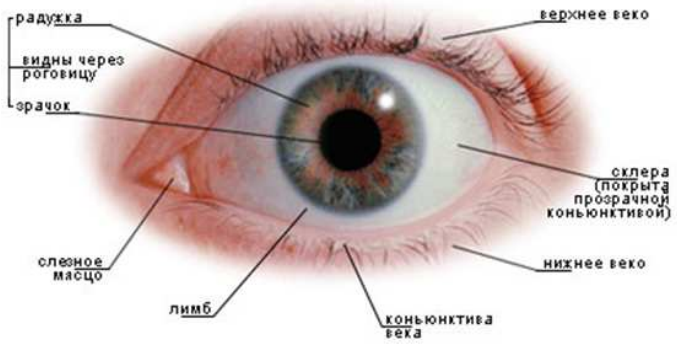


Рис. 1. Веки и глазная щель

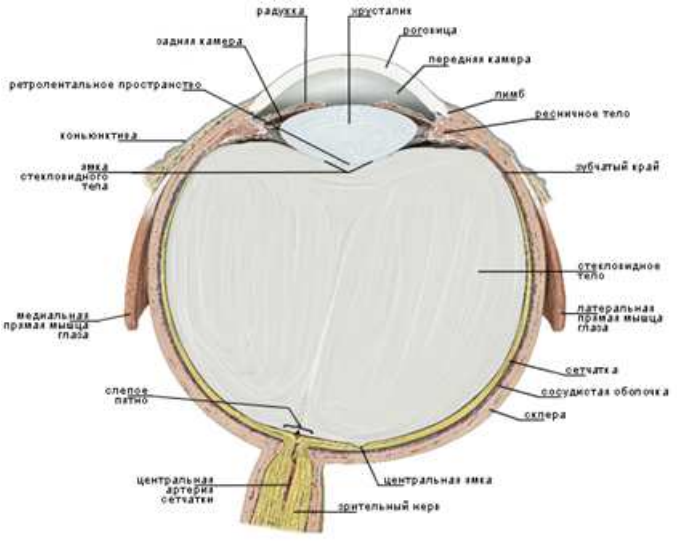


Рис. 2. Глазное яблоко, правое. Горизонтальный срез

Внутренняя оболочка глазного яблока, или сетчатка, является наиболее важной из оболочек глаза, так как в ней происходит восприятие зрительных раздражений. Она непосредственно связана со зрительным нервом.

Задняя часть сетчатки имеет сложное строение. Именно здесь расположены периферические отделы зрительного анализатора: свето- и цветочувствительные элементы (фоторецепторные клетки) — палочки и колбочки. Поэтому задний отдел сетчатки называют её зрительной частью. Местом наибольшей чувствительности сетчатки является её центральная ямка, в области которой сконцентрирована большая часть фоторецепторных клеток.

Все образования, составляющие ядро глазного яблока (хрусталик, водянистая влага, которая заполняет переднюю и заднюю камеры глазного яблока, и стекловидное тело), в норме совершенно прозрачны и обладают способностью преломлять свет. Поэтому их, как и роговицу, относят к преломляющим средам глаза. Благодаря преломлению лучи света фокусируются в наиболее чувствительной зоне сетчатки — в центральной ямке.

Хрусталик имеет вид двояковыпуклого тела. Своей передней поверхностью он прилежит к радужке, а позади него находится стекловидное тело. Посредством тонких прочных нитей хрусталик связан с ресничной мышцей, расположенной циркулярно в цилиарном теле. Благодаря сокращению или расслаблению ресничной мышцы хрусталик изменяет свою кривизну. Это приспосабливание глаза к наилучшему видению на близком и далёком расстояниях носит название аккомодации [22].

1.2. Строение радужной оболочки глаза

Радужка составляет переднюю часть сосудистой оболочки. При осмотре передней поверхности радужной оболочки она выглядит тонкой почти округлой пластинкой, лишь слегка эллиптической формы. У края зрачка на всем его протяжении отмечается чёрная зубчатая оторочка, окаймляющая его на всем протяжении и представляющая выворот заднего пигментного листка радужной оболочки.

Радужная оболочка своей зрачковой зоной прилежит к хрусталику, опирается на него и свободно скользит по его поверхности при движениях зрачка. Зрачковая зона радужной оболочки оттесняется несколько кпереди прилежащей к ней сзади выпуклой передней поверхностью хрусталика вследствие чего, радужная оболочка в целом имеет форму усечённого конуса.

Основными свойствами радужной оболочки, обусловленными анатомическими особенностями её строения, являются рисунок, рельеф, цвет, расположение относительно соседних структур глаза и состояние зрачкового отверстия.

Параллельно зрачковому краю, концентрически к нему расположен невысокий зубчатый валик — круг Краузе или брыжжи, где радужная оболочка имеет наибольшую толщину. По направлению к зрачку радужная оболочка становится тоньше, но наиболее тонкий её участок соответствует корню радужной оболочки.

Соответственно кругу Краузе в строме радужной оболочки, также концен-

трически к зрачку, располагается сплетение сосудов — малый круг кровообращения радужной оболочки. Кругом Краузе пользуются для выделения двух топографических зон этой оболочки: внутренней, более узкой, зрачковой и наружной, более широкой, цилиарной. На передней поверхности радужной оболочки отмечается радиарная исчерченность, хорошо выраженная в её цилиарной зоне. Она обусловлена радиальным расположением сосудов, вдоль которых ориентирована и строма радужной оболочки. По обе стороны круга Краузе на поверхности радужной оболочки видны щелевидные углубления, глубоко проникающие в неё — крипты или лакуны. Такие же крипты, но меньшего размера, располагаются и вдоль корня радужной оболочки.

В наружном отделе цилиарной зоны заметны складки радужной оболочки, идущие концентрически к её корню, — контракционные бороздки, или бороздки сокращения. Они представляют обычно лишь отрезок дуги, но не захватывают всей окружности радужной оболочки. При сокращении зрачка они сглаживаются, при расширении — наиболее выражены.

Все перечисленные образования на поверхности радужной оболочки и обусловливают как её рисунок, так и рельеф.

В радужной оболочке различают два листка:

1) передний, мезодермальный, увеальный, составляющий продолжение сосудистого тракта;

2) задний, эктодермальный, ретинальный, составляющий продолжение эмбриональной сетчатки, в стадии вторичного глазного пузыря, или глазного бокала.

Передний мезодермальный листок состоит из переднего пограничного слоя и сосудистого слоя радужной оболочки. Задний эктодермальный листок представлен дилататором с его задней пограничной пластинкой и пигментированным эпителием. К нему же принадлежит и сфинктер, сместившийся в строму радужки по ходу ее эмбрионального развития. Передний пограничный слой мезодермального листка состоит из густого скопления клеток, расположенных тесно друг к другу, параллельно поверхности радужной оболочки. Передний пограничный слой у края крипт прерывается (рисунок 3[2]).

Из наружного слоя заднего пигментного листка в период эмбрионального развития формируются две мышцы радужной оболочки: сфинктер, сужающий зрачок, и дилататор, обусловливающий его расширение. В процессе развития сфинктер перемещается из толщи заднего пигментного листка в строму радужной оболочки, в её глубокие слои, и располагается у зрачкового края, окружая зрачок в виде кольца. Волокна его проходят параллельно зрачковому краю, примыкая непосредственно к его пигментной кайме. Цилиарный край мышцы несколько смыт, от него кзади в косом направлении отходят мышечные волокна к дилататору. По соседству со сфинктером, в строме радужной оболочки

[2]http://zrenue.com/anatomija-glaza/40-raduzhka/345-raduzhnaja-obolochka-glaza-raduzhka-stroenie.html

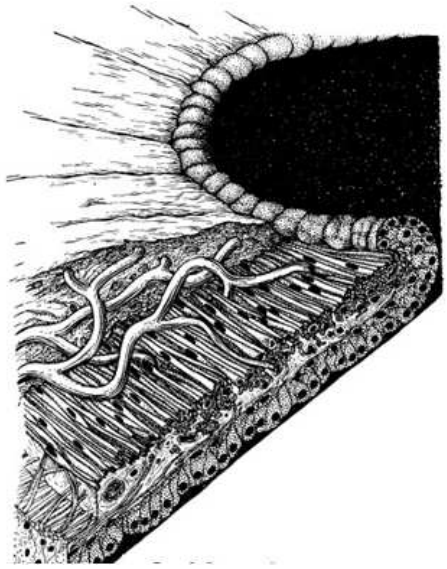


Рис. 3. Строение радужки

в большом количестве разбросаны крупные, округлые, густо пигментированные клетки, лишенные отростков, — «глыбистые клетки», возникшие также в результате смещения в строму пигментированных клеток из наружного пигментного листка.

За счёт наружного слоя заднего пигментного листка развивается дилататор — мышца, расширяющая зрачок. В отличие от сфинктера, сместившегося в строму радужной оболочки, дилататор остаётся на месте своего образования, в составе заднего пигментного листка, в его наружном слое.

Дилататор имеет вид тонкой пластинки, расположенной между цилиарной частью сфинктера и корнем радужной оболочки. Клетки дилататора располагаются в один слой, радиально по отношению к зрачку. Основания клеток дилататора, содержащие миофибриллы, обращены к строме радужной оболочки, лишены пигмента и в совокупности составляют заднюю пограничную пластинку. Сокращение дилататора осуществляется за счёт миофибрилл, причём изменяется как величина, так и форма его клеток.

В результате взаимодействия двух антагонистов — сфинктера и дилататора — радужная оболочка получает возможность путём рефлекторного сужения и расширения зрачка регулировать поток проникающих внутрь глаза световых лучей, причём диаметр зрачка может изменяться от 2 до 8 мм [21].

2. Биометрия

2.1. Основные понятия

Биометрия — это наука, представляющая совокупность математических методов, применяемых в биологии и заимствованных главным образом из области математической статистики и теории вероятностей, но имеющая свою

Рис. 4. Структура радужной оболочки [10]

специфику [13].

Биометрические технологии основаны на биометрии, измерении уникальных характеристик отдельно взятого человека. Это могут быть как уникальные признаки, полученные им с рождения, например: отпечатки пальцев, радужная оболочка глаза; так и характеристики, приобретённые со временем или же способные меняться с возрастом или под внешним воздействием. Например: почерк, голос или походка.

Основные определения, используемые в сфере биометрических приборов [37]:

Универсальность — каждый человек должен обладать измеряемой характеристикой.

Уникальность — это насколько хорошо человек отделяется от другого с биометрической точки зрения.

Постоянство — мера того, в какой степени выбранные биометрические черты остаются неизменными во времени, например в процессе старения.

Взыскания — простота осуществления измерения.

Производительность — точность, скорость и надёжность используемых технологий.

Приемлемость — степень достоверности технологии.

Устранение — простота использования замены.

В последнее время термин «Биометрия» претерпевает изменения и уточняется. Так в монографии [6] приведено следующее определение.

Биометрия — это наука об идентификации или верификации личности по физиологическим или поведенческим отличительным характеристикам [6].

Итак, биометрическая система может работать в двух режимах:

— *верификация* — сравнение один к одному с биометрическим шаблоном. Проверяет, что человек тот, за кого он себя выдаёт. Верификация может быть осуществлена по смарт-карте, имени пользователя или идентификационному номеру;
— *идентификация* — сравнение один ко многим: после «захвата» биометрических данных идёт соединение с биометрической базой данных для определения личности. Идентификация личности проходит успешно, если биометрический образец уже есть в базе данных.

2.2. Статические и динамические методы

Обычно при классификации биометрических технологий выделяют две группы систем по типу используемых биометрических параметров.

Первая группа систем использует статические биометрические параметры: отпечатки пальцев, геометрия руки, сетчатка глаза и т. п.

Вторая группа систем использует для идентификации динамические параметры: динамика воспроизведения подписи или рукописного ключевого слова, голос и т. п. [13]

2.3. Характеристики биометрических систем

Описанные ниже параметры используется как показатели эффективности биометрических систем [37].

Коэффициент ложного приёма (FAR) и коэффициент ложного совпадения (FMR).

FAR (False Acceptance Rate, коэффициент ложного пропуска, вероятность ложной идентификации) — вероятность того, что система биоидентификации по ошибке признает подлинность пользователя, не зарегистрированного в системе.

FMR (коэффициент ложного совпадения) — вероятность, что система неверно сравнивает входной образец с несоответствующим шаблоном в базе данных.

Коэффициент ложного отклонения (FRR) и коэффициент ложного несовпадения (FNMR).

FRR (False Rejection Rate, коэффициент ложного отказа доступа) — вероятность того, что система биоидентификации не признает подлинность шаблона зарегистрированного в ней пользователя.

FNMR (коэффициент ложного несовпадения) — вероятность того, что система ошибётся в определении совпадений между входным образцом и соответствующим шаблоном из базы данных. Система измеряет процент верных входных данных, которые были приняты неправильно.

Рабочая характеристика системы или относительная рабочая характеристика (ROC).

График ROC — это визуализация компромисса между характеристиками FAR и FRR. В общем случае сравнивающий алгоритм принимает решение на основании порога, который определяет, насколько близко должен быть входной

образец к шаблону, чтобы считать это совпадением. Если порог был уменьшен, то будет меньше ложных несовпадений, но больше ложных приёмов. Соответственно, высокий порог уменьшит FAR, но увеличит FRR. Линейный график свидетельствует о различиях для высокой производительности (меньше ошибок — реже возникают ошибки). Равный уровень ошибок (коэффициент EER) или коэффициент переходных ошибок (CER) — это коэффициенты, при которых обе ошибки (ошибка приёма и ошибка отклонения) эквивалентны. Значение EER может быть с лёгкостью получено из кривой ROC. EER — это быстрый способ сравнить точность приборов с различными кривыми ROC. В основном, устройства с низким EER наиболее точны. Чем меньше EER, тем более точной будет система.

Коэффициент отказа в регистрации (FTE или FER) и коэффициент ошибочного удержания (FTC).

Коэффициент отказа в регистрации (FTE или FER) — коэффициент, при котором попытки создать шаблон из входных данных безуспешны. Чаще всего это вызвано низким качеством входных данных.

Коэффициент ошибочного удержания (FTC) — в автоматизированных системах это вероятность того, что система не способна определить биометрические входные данные, когда они представлены корректно.

Ёмкость шаблона — максимальное количество наборов данных, которые могут храниться в системе [37].

2.4. Схема работы

Все биометрические системы работают практически по одинаковой схеме. Сначала система запоминает образец биометрической характеристики (это и называется процессом записи или регистрацией). Во время записи некоторые биометрические системы могут попросить сделать несколько образцов для того, чтобы составить наиболее точное изображение биометрической характеристики. Затем полученная информация обрабатывается и преобразовывается в математический код. Кроме того, система может попросить произвести ещё некоторые действия для того, чтобы связать биометрический образец с определённым человеком. Например, ввести персональный идентификационный номер (PIN), либо вставить в считывающее устройство смарт-карту, содержащую образец.

Идентификация по любой биометрической системе проходит четыре стадии [4]:

запись — физический или поведенческий образец запоминается системой;

выделение — уникальная информация выносится из образца и составляется биометрический образец;

сравнение — сохранённый образец сравнивается с представленным;

совпадение/несовпадение — система решает, совпадают ли биометрические образцы, и выносит решение.

Большинство современных систем хранят в специальной базе данных цифровой код, который связывается с конкретным человеком, имеющим право доступа. Сканер или любое другое устройство, используемое в системе, считыва-

ет определённый биологический параметр человека. Далее полученные данные обрабатываются путём преобразования их в цифровой код. Именно этот ключ и сравнивается с содержимым специальной базы данных для идентификации личности [5].

Для получения ключа из записанного биологического параметра часто используются шаблоны. Элементы биометрического измерения, которые не используются в сравнительном алгоритме, не сохраняются в шаблоне, чтобы уменьшить размер файла и защитить личность регистрируемого, сделав невозможным воссоздание исходных данных по информации из образца.

3. Процесс идентификации

3.1. Получение изображения

Практически все исследования ведутся на основе изображений, взятых из баз CASIA (Chinese Academy of Sciences Institute of Automation).

Институтом проделана огромная работа по сбору обширных баз данных.

Базы CASIA содержат несколько разделов.

Самой часто используемой базой является CASIA-Iris-Interval. Изображения из этой базы получены в ближнем инфракрасном диапазоне с разрешением 320x280 пикселей. Спектр ближнего инфракрасного излучения выделяет особенности структуры радужки, облегчая последующие измерения в процессе идентификации.

Для изучения изменения структуры радужной оболочки при изменении размеров зрачка используется база CASIA-Iris-Lamp. Изображения из этой базы содержат снимки с включённой и выключенной лампой с разрешением 640x480 пикселей.

Для исследований индивидуальных особенностей строения радужки часто используется база CASIA-Iris-Twins. Она содержит изображения радужных оболочек более 100 пар однояйцовых близнецов различного пола и возраста.

База CASIA-Iris-Distance используется для разработки методов идентификации, работающих на значительных расстояниях, и для разработки многопараметрических методов биоидентификации. Изображения в этой базе получены с помощью камеры высокого разрешения с расстояния 3 м. Разрешение изображений 2352*1728.

База CASIA-Iris-Thousand содержит изображения радужных оболочек более 1000 человек. Эта база используется для изучения уникальных особенностей структуры радужки, проверки методов определения радужной оболочки и идентификации, а также для усовершенствования этих методов при условии наличия бликов, ношения очков и контактных линз.

База CASIA-Iris-Syn содержит синтезированные изображения радужной оболочки [40].

3.2. Выделение зрачка

3.2.1. Определение границ

На изображении глаза зрачок представляет собой очень отчётливый чёрный круг. Это позволяет легко найти его внешнюю границу. Кроме того, уровень границы может быть установлен очень высоким для того, чтобы опустить мелкие неконтрастные области границы, пока не будет занят весь периметр зрачка. Лучший алгоритм для определения границы зрачка — это алгоритм Канни. Этот алгоритм использует горизонтальные и вертикальные градиенты, чтобы определить границы на изображении. После обработки изображения с помощью алгоритма Канни, будет найдена окружность, точно определяющая границу зрачка [35].

3.2.2. Алгоритм Канни

Нахождение контуров сводится к обнаружению разрывов интенсивности при переходе от одной области изображения к другой. Большинство алгоритмов анализа изображения рассматривают ее просто как некоторую скалярную функцию от пространственных переменных, абстрагируясь от физического смысла. Т.е. под интенсивностью пикселя может подразумеваться, например, уровень красного, светлота, насыщенность, яркость и т.д.

Для обнаружения перепадов яркости будем применять дискретные аналоги первых производных.

Приближённые значения первых частных производных $G_x(i,j)$ и $G_y(i,j)$ в каждой точке (i,j) изображения f обычно определяются как свёртки 3x3-окрестности с матрицами дискретных дифференциальных операторов. Будем использовать дискретные дифференциальные операторы Собеля (Sobel), определяемые матрицами:

$$D_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} D_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Применение операторов к изображению определяется через оператор свёртки:

$$G_x = D_x * f,$$

$$G_y = D_y * f.$$

Поточечные формулы свёртки дают оценку по направлению x

$$\begin{aligned} G_x(i,j) &= (-1)\cdot f(i-1,j-1) + (-2)\cdot f(i-1,j) + (-1)\cdot f(i-1,j+1) + \\ &\quad +(0)\cdot f(i,j-1) + (0)\cdot f(i,j) + (0)\cdot f(i,j+1) + \\ &\quad +(1)\cdot f(i+1,j-1) + (2)\cdot f(i+1,j) + (1)\cdot f(i+1,j+1) = \\ &= -f(i-1,j-1) - 2\cdot f(i-1,j) - f(i-1,j+1) + \\ &\quad +f(i+1,j-1) + 2\cdot f(i+1,j) + f(i+1,j+1) \end{aligned}$$

и, аналогично, по направлению y

$$\begin{aligned} G_y(i,j) &= (-1)\cdot f(i-1,j-1)+(0)\cdot f(i-1,j)+(1)\cdot f(i-1,j+1)+ \\ &\quad +(-2)\cdot f(i,j-1)+(0)\cdot f(i,j)+(2)\cdot f(i,j+1)+ \\ &\quad +(-1)\cdot f(i+1,j-1)+(0)\cdot f(i+1,j)+(1)\cdot f(i+1,j+1)= \\ &= -f(i-1,j-1)-2\cdot f(i,j-1)-f(i+1,j-1)+ \\ &\quad +f(i-1,j+1)+2\cdot f(i,j+1)+f(i+1,j+1). \end{aligned}$$

Направление градиента $\nabla f = G = (G_x, G_y)^T$ определяется углом между его направлением и осью абсцисс

$$\vartheta = \operatorname{arctg}\left(\frac{G_y}{G_x}\right).$$

Величина градиента определяется, как правило, любой Гёльдеровой нормой

$$||x||_p = \sqrt[p]{\sum_i x_i^p}$$

или предельной нормой ($p \to \infty$)

$$||x||_\infty = \max_i |x_i|.$$

Чаще всего используют нормы при $p = 1$, $p = 1$ и $p = \infty$.

В нашем случае $||G||_1 = |G_x| + |G_y|$, $||G||_2 = \sqrt{G_x^2 + G_y^2}$ и $||G||_\infty = \max_i |G_x|, |G_y|$.

Длина вектора градиента в точке будет влиять на то, войдёт ли пиксель в состав границы. Угол используют для определения направления контура в точке. Этот угол используется для процедуры утончения границы — светлые пиксели результирующего изображения, которые не лежат в направлении пути по границе, подавляются.

Последний шаг алгоритма — двухпороговое отсечение с гистерезисом. Если выбрать слишком высокий порог, то возможны ошибки первого рода — потеря точек, которые лежат на границе. Если выбрать слишком низкий порог, то участятся ошибки второго рода — лишние точки принимаются за фрагменты границы. Поэтому используют два порога. Сначала применяют высокий порог, который выделяет пиксели, достоверно принадлежащие границе. Затем обходят границу и, используя информацию о направлениях, достраивают её, применяя низкий порог. Такой алгоритм отдаёт предпочтение непрерывным кривым в качестве контуров и игнорирует мелкие изменения интенсивности. Вся эта последовательность шагов называется алгоритмом Канни [28]. В результате из чёрно-белого изображения мы получаем бинарное изображение контуров (см. рис. 5).

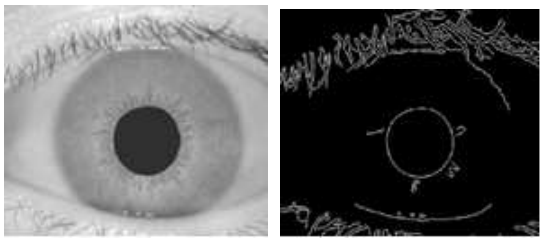


Рис. 5. Слева исходное изображение, справа — изображение контуров [35]

3.3. Очистка изображения

Для получения информации в месте границы возможно использование фильтров. Первый шаг очистки изображения — это расширение всех найденных границ. Увеличением размера линий вокруг найденных компонентов мы добиваемся объединения их в большие линейные сегменты. В конце концов, линии, не полностью определённые во время детектирования границ, приобретают форму. Это даёт бóльшую вероятность, что периметр зрачка примет форму замкнутой окружности.

Зная о том, что зрачок хорошо определён, можно использовать больше фильтров без боязни потери этой важной информации. Допуская, что изображение центрировано, фильтр может быть использован для заливки круга, внутри границ зрачка. Таким образом, мы точно определяем внутреннюю область зрачка. После этого фильтр, который просто отделяет область соединённых пикселей, может быть использован для маленьких несвязанных частей изображения, которые были найдены при определении границ. Наконец, какие-либо блики на зрачке, вызванные отражением или другими причинами, могут быть заполнены путём сравнения областей светлых пикселей с областью ниже порога. После этого процесса мы сохраняем изображение, на котором ярко выделена область зрачка, до тех пор, пока не очистим его от посторонних данных.

3.4. Определение параметров зрачка

На спектре изображения виден большой круг, площадь которого задаётся совокупностью пикселей. Поскольку зрачок — самый большой и яркий круг на всем изображении, то интенсивность спектра в области зрачка будет достигать пика. В области зрачка строго в центре значение интенсивности будет максимально. Это происходит потому, что центр — точка внутри круга, наиболее удалённая от всех его границ. Поэтому максимальное значение должно соответствовать центру зрачка, и, кроме того, расстояние от центра зрачка до границы должно быть равно радиусу зрачка [35].

4. Выделение радужки

4.1. Нахождение радужной оболочки глаза

Когда определена информация о зрачке, можно приступить к определению параметров радужки. Важно помнить, что зрачок и радужка являются практически концентрическими окружностями. Следовательно, зная центр и радиус зрачка, мы не можем определить из них эти же параметры для радужки. Однако информация о зрачке даёт хорошую отправную точку в виде центра зрачка.

Самые современные алгоритмы определения радужки используют произвольные окружности для определения параметров радужки. Начиная от зрачка, эти алгоритмы перебирают потенциальные значения центра и радиуса радужки.

Первым шагом в поиске реального радиуса радужки является нахождение приблизительного радиуса радужки. Это приближение может потом помочь найти реальные параметры. Для того чтобы найти это приближение, необходимо найти хотя бы одну точку границы радужки. Зная, что верхняя и нижняя части глаза часто могут быть закрыты веками и ресницами, лучшим вариантом является поиск незакрытой границы вдоль горизонтальной линии, проходящей через центр зрачка.

Предварительно следует использовать специальный сглаживающий фильтр, такой как медианный фильтр, к исходному изображению. Этот фильтр устраняет мелкие шумы, сохраняя контуры изображения. После использования медианного фильтра может потребоваться увеличение контрастности изображения.

Теперь, когда изображение подготовлено, можно приступить к определению границ. Интересующая нас область не является просто горизонтальной линией, проходящей через радужку, нас интересует часть этой линии правее зрачка. Увеличение яркости при переходе от радужки к склере является единственным крупным шагом.

Радужка должна представлять собой пошаговое изменение яркости в интересующей нас области. Следовательно, эта область изображения должна соответствовать компоненту с наивысшим значением на выходе из фильтра. Находя максимальное значение справа от зрачка, мы найдём границу радужки. Следует заметить, что т.к. радужка и зрачок могут не быть концентрическими окружностями, то расстояние от центра зрачка до этой границы может не соответствовать радиусу радужки.

4.2. Настройка контраста

Отметим, что второе и третье изображения рисунка 6 являются более контрастными, чем изображение 1. Контрастность этих рисунков была подстроена таким образом, чтобы увеличить разницу в яркости значений изображения радужки. Это численно упрощает анализ данных радужной оболочки. Подстройка выполняется при помощи построения гистограммы яркости изображения и растягиванием верхних и нижних границ гистограммы к делению всего разбиения по значениям яркости в диапазоне от 0 до 255. Рисунок 7 демонстрирует этот процесс [35].

Рис. 6. Изображения радужной оболочки глаза при разной контрастности [35]

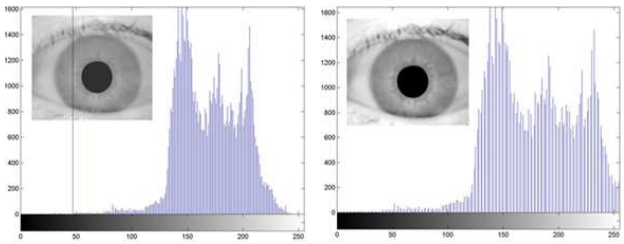


Рис. 7. Изменение контрастности изображения [35]

5. Определение границ радужки

5.1. Определение параметров радужки

Радужка должна представлять собой пошаговое изменение яркости в интересующей области. Следовательно, эта область изображения должна соответствовать компоненту с наивысшим значением на выходе фильтра выделения границ. Поиском максимального значения справа от зрачка находится граница радужки. Так как радужка и зрачок могут не быть концентрическими окружностями, то расстояние от центра зрачка до этой границы может не соответствовать радиусу радужки [26].

Для определения края (границы) радужки можно использовать несколько подходов. Опишем один из них, следуя [26]. Алгоритм выглядит следующим образом:

1. Ранее описанным алгоритмом находим центр зрачка и его радиус.

2. Определяем грубую оценку радиуса радужки. Вначале применяем медианный фильтр. Вычитанием из исходного изображения отфильтрованного изображения получаем грубую оценку границы. Это позволяет определить интересующую нас область вдоль горизонтальной линии, проведённой к границе от центра зрачка.

3. Затем анализируем детали дискретного вэйвлет-преобразования вдоль этой линии. Максимум в деталях ближе к грубой оценке уточняет радиус радужки. Так как радиус радужки, как правило, не совпадает с радиусом зрачка, то необходимо дополнительное уточнение.

4. Определение центра радужки основано на построении двух хорд, проходящих через центр зрачка (желательно под углом 90°). Центр определяется

пресечением перпендикуляров, проведённых через середины хорд. В качестве новой оценки радиуса берётся среднее значение длин хорд. Это не точная оценка, но вполне приемлемая для работы алгоритма [26].

Остаётся открытым вопрос выбора хорд с концами, лежащими на границе радужки, максимально перпендикулярными и максимально приближенными к векам. В работе [26] этот алгоритм чётко не прописан.

Далее, имея оценки радиуса и центра радужки, производится развёртка изображения: переход от полярных координат в декартовы с нормировкой по радиусу для компенсации линейного сжатия и растяжения радужки в следствие изменений размеров зрачка.

6. Нахождение ключевых точек

6.1. Композиция фильтров Гаусса и оператора Лапласа

Отличительные пространственные характеристики радужки человека проявляются различно в различных масштабах. Например, отличительный диапазон структур из общей формы радужной оболочки к распределению мелких крипт и детали текстуры. Для захвата этого диапазона пространственных деталей предпочтительно использовать разложение представления на несколько масштабов.

Система делает изотропное полосовое разложение, полученное от применения оператора Лапласа гауссовых фильтров [33, 36] к данным изображения. Эти фильтры могут быть определены как

$$-\frac{1}{\pi\sigma^4}\left(\frac{\rho^2}{2\sigma^2}e^{-\frac{\rho^2}{2\sigma^2}}\right),$$

где σ — среднеквадратичное отклонение, ρ — расстояние от центра фильтра до точки. На практике фильтрованное изображение реализуется в виде пирамиды Лапласа [31,36]. Это представление описывается процедурно в терминах каскада малых фильтров Гаусса. Так если $w = (1\ 4\ 6\ 4\ 1)^T/16$ является одномерной маской, то $W = ww^T$ является двумерной маской, которая получается как результат внешнего произведения. Для заданной интересующей нас области I построение пирамиды Лапласа начинается со свертки I с W, для того чтобы получить набор изображений g_k, полученных низкочастотной фильтрацией из g_{k-1} по формуле

$$g_k = (W * g_{k-1})_{\downarrow 2},$$

где $g_0 = I$ и $(\downarrow 2)$ обозначает уменьшение два раза изображения в каждом направлении. Очередной k-ый уровень пирамиды Лапласа l_k формируется как разница между g_k и g_{k+1}, расширенным перед вычитанием, таким образом, что он соответствует частоте дискретизации g_k. Расширение достигается путём увеличения частоты дискретизации и интерполяции

$$l_k = g_k - 4W * (g_{k+1})_{\uparrow 2},$$

где $(\uparrow 2)$ означает увеличение изображения в 2 раза путём добавления нулевых строк и столбцов между строками и столбцами исходного изображения. Генерируемое ядро W используется в качестве фильтра интерполяции, а деление на 4 необходимо потому, что 3/4 значений в изображении — это только что вставленные нули. Полученная пирамида Лапласа, состоящая из четырёх уровней, служит основой для последующей обработки. При построении пирамиды Лапласа следует напрямую создавать её согласно определённой процедуре.

Представление выводится непосредственно из фильтрованного изображения с размером порядка количества байт в области радужки первоначально полученного изображения. Система сохраняет больше имеющейся информации о радужке и могла бы быть способна сделать более тонкие различия между различными радужками [43].

6.2. Фильтр Габора

Фильтры, основанные на вейвлетах Габора, очень хороши для выделения шаблонов на изображении. Рассмотрим одномерный фильтр Габора с фиксированной частотой (fixed frequency 1D Gabor filter) для поиска шаблонов в развёрнутом изображении. Вейвлеты Габора состоят из двух компонент, комплексной синусоидальной несущей и гауссовой огибающей:

$$g(x, y) = s(x, y) * w_r(x, y).$$

Комплексная несущая имеет форму:

$$s(x, y) = e^{j*(2*\pi(u_0*x+v_0*y)+P)}.$$

Параметры u_0 и v_0 представляют собой частоты горизонтальной и вертикальной синусоид соответственно. P — произвольный сдвиг фазы. Второй компонент преобразования Габора представляет произвольный фазовый сдвиг. Итоговый вейвлет формируется из синусоидальной несущей и этой огибающей. Огибающая имеет гауссовский профиль и описывается следующим выражением:

$$w_r(x, y) = K * e^{-\pi((a^2)*(x-x_0)_r{}^2+(b^2)*(y-y_0)_r{}^2},$$

где:

$$\begin{aligned} (x - x_0)_r &= (x - x_0) * \cos\theta + (y - y_0) * \sin\theta, \\ (y - y_0)_r &= -(x - x_0) * \sin\theta + (y - y_0) * \cos\theta, \end{aligned}$$

K — постоянная масштабирования, (a, b) — постоянные масштабирования осей, θ — постоянная поворота, (x_0, y_0) пик огибающей. Чтобы получить вейвлет Габора, мы перемножаем $s(x, y)$ и $w_r(x, y)$. Полученный вейвлет изображён на рисунке 8.

Рассмотрим процесс выделения индивидуальных особенностей с помощью фильтра Габора. Сначала мы возьмём колонку шириной в 1 пиксель и выполним свёртку его с одномерным вейвлетом Габора. Т.к. фильтр Габора комплексный,

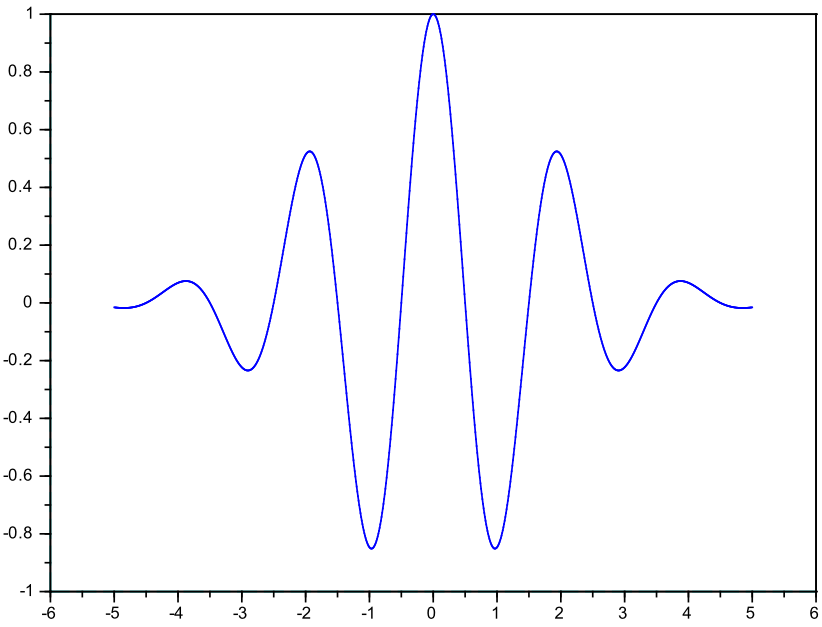


Рис. 8. Одномерный вейвлет Габора $G(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\cos(2\pi\vartheta x)$ при $\sigma = 3$ и $\vartheta = 0.5$

в результате мы получим отдельно действительную и мнимую части. Если полученное значение больше нуля, сохраняем 1, иначе — сохраняем 0. Когда все столбцы изображения обработаны, мы можем сформировать черно-белое изображение, составляя колонки друг к другу. Действительная и мнимая части изображения показаны на рисунке 9 [35].

Для формирования кода радужки необходимо сравнить значения мнимой и действительной части в каждой точке. Если хотя бы 1 из этих значений больше нуля, сохраняем 1, иначе — сохраняем 0.

(a) (b)

Рис. 9. (a) – действительная часть кода радужки; (b) – мнимая часть кода радужки [35]

Заключение

В работе был произведён обзор методов, применяемых при идентификации человека по радужной оболочке глаза.

При определении границ зрачка наиболее часто используется алгоритм Канни или метод Даугмана.

При определении границ радужной оболочки наиболее часто используемыми методами являются метод Даугмана, метод Хоу и метод Хаара. Интегродифференциальный метод Даугмана является каноническим в данной области. Но при наличии бликов метод Даугмана даёт неудовлетворительные результаты. Поэтому он часто используется совместно с методом Хоу.

Приведён также метод нахождения параметров радужки с помощью двух хорд. Этот метод, хотя и является приблизительным, даёт достаточно точную оценку и не требует больших вычислительных мощностей.

Для нахождения ключевых точек чаще всего используется фильтр Габора. Он сочетает в себе простоту реализации и точность, столь необходимые в данной области. Однако также может быть использована композиция фильтров Гаусса и оператора Лапласа.

Несмотря на некоторые недостатки, технология идентификации и верификации человека по радужной оболочке глаза является весьма перспективной. Особенно хороша она благодаря своей надёжности и хорошему соотношению ошибок первого и второго рода для систем доступа к различным объектам. Это подтверждается растущим с каждым годом интересом к этой области, а также увеличением доли рынка метода идентификации по радужке среди других биометрических технологий.

Литература

1. Абраменко Е.А., Минакова Н.Н., Третьяков И.Н., Петров И.В. Применение спектров текстурной картины для изучения и классификации неоднородных структур // Известия Алтайского государственного университета. 2010. № 1–2. С. 147–149.
2. Алгулиев Р.М.О., Имамвердиев Я.Н.О., Мусаев В.Я.О. Методы обнаружения живучести в биометрических системах // Вопросы защиты информации. 2009. Вып. 3. С. 16–21.
3. Биологический энциклопедический словарь / Гл. ред. М.С. Гиляров. М. : Советская энциклопедия, 1989. 864 с.
4. Биометрия. URL: `http://www.fond-ai.ru/art1/art228.html` (дата обращения: 10.02.2014).
5. Биометрия – Энциклопедия безопасности URL: `http://www.secuteck.ru/wiki/index.php?title=%D0%91%D0%B8%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F` (дата обращения: 10.02.2014).
6. Болл Р., Коннел Д., Панканти Ш., Ратха Н., Сеньор Э. Руководство по биометрии. М. : Техносфера, 2007. 368 с.
7. Буй Т.Ч., Спицын В.Г. Разложение изображений с помощью двумерного дискретного вейвлет-преобразования и быстрого преобразования Хаара // Проблемы информатики. 2011. № 2. С. 11–15.

8. Давлетханов М. Идентификация по радужке глаза. Часть 1. http://www.bre.ru/security/24514.html (дата обращения: 10.02.2014).
9. Давлетханов М. Идентификация по радужке глаза. Часть 2. http://www.bre.ru/security/24531.html (дата обращения: 10.02.2014).
10. Ковалевский Е.И. Офтальмология. Учебник. М.: Медицина, 1995. 480 с.
11. Корепанов А.О. Обнаружение границ радужной оболочки с использованием преобразования Хоу // Вестник Самарского государственного аэрокосмического университета им. Академика С.П. Королёва (Национального исследовательского университета). 2008. № 2. С. 235–239.
12. Кухарев Г.А. Биометрические системы: Методы и средства идентификации личности человека. СПб. : Политехника, 2001. 240 с.
13. Лакин Г.Ф. Биометрия. 4-ое изд. М. : Высшая школа, 1990. 392 с.
14. Минакова Н.Н., Петров И.В. Информационная система анализа структуры радужной оболочки глаза // Ползуновский вестник. Барнаул, 2012. № 3/2.
15. Павельева Е.А. Метод Проекционной Фазовой Корреляции в Ключевых Точках Радужной Оболочки Глаза // The 22nd International Conference on Computer Graphics and Vision. 2012. С. 128–132.
16. Павельева Е.А., Крылов А.С. Поиск и анализ ключевых точек радужной оболочки глаза методом преобразования Эрмита // Информатика и её применения. 2010. № 1. С. 79–82.
17. Павельева Е.А., Крылов А.С. Алгоритм сравнения изображений радужной оболочки глаза на основе ключевых точек // Информатика и её применения. 2011. Т. 5, Вып. 1. С. 68–72.
18. Павельева Е.А., Крылов А.С. Алгоритм сравнения изображений радужной оболочки глаза на основе ключевых точек // Информатика и её применения. 2011. № 1. С. 68–72.
19. Павельева Е.А., Крылов А.С. Определение локальных сдвигов изображений радужных оболочек глаз методом проекционной фазовой корреляции // Труды конференции GraphiCon'2011. Москва, 2011. С. 188–191.
20. Пустынский И.Н., Дементьев А.Н., Мищенко Н.И., Зайцева Е.В. Методы и средства формирования и обработки изображения переднего отдела глаза // Доклады Томского государственного университета систем управления и радиоэлектроники. 2012, 2-1. С. 121–128.
21. Радужная оболочка глаза (радужка), строение // Современная офтальмология. URL: http://zrenue.com/anatomija-glaza/40-raduzhka/345-raduzhnaja-obolochka-glaza-raduzhka-stroenie.html (дата обращения: 10.02.2014).
22. Синельников Р.Д. Атлас анатомии человека: в 3-х томах. 3-е изд. М. : «Медицина», 1967.
23. Третьяков И.Н., Минакова Н.Н. Параметризация структуры радужной оболочки глаза с использованием вейвлет-преобразования // Известия Алтайского государственного университета. 2009. № 1. С. 129–130.
24. Третьяков И.Н., Минакова Н.Н. Алгоритм разграничения доступа по радужной оболочке глаза для решения задач контроля доступа к информационным ресурсам // Доклады Томского государственного университета систем управления и радиоэлектроники. 2010, 1-1. С 100–102.

25. Федотов Н.Г. Методы стохастической геометрии в распознавании образов. М. : Радио и связь, 1990. 144 с.
26. Шокуров А.В., Михалев А.В. Оптимальное использование вейвлет-компонент // Успехи мат. наук. 2007. Т. 62, № 4. С. 171–172.
27. Юрьева Т.Н. Современные представления о структурно-функциональной организации иридоцилиарной системы // Медицинская визуализация. 2011. Вып. 2. С 46–50.
28. All things considered... – Горизонт завален! http://bik-top.livejournal.com/37060.html?thread=197828 (дата обращения: 10.02.2014).
29. Anuj B., Rashid A. Image compression using modified fast Haar wavelet transform // World Appl. Sci. J. 2009. V. 7, N. 5. P. 647–653.
30. Arvacheh Ehsan M. A Study of Segmentation and Normalization for Iris Recognition Systems // A thesis presented to the University of Waterloo in fullment of the thesis requirement for the degree of Master of Applied Science In Systems Design Engineering.
31. Burt P.J. and Adelson E. The Laplacian pyramid as a compact image code // IEEE Trans. Comput. 1983. V. 31, N. 4. P. 532–540.
32. Daugman J. How Iris Recognition Works http://www.cl.cam.ac.uk/\~jgd1000/irisrecog.pdf (дата обращения: 10.02.2014).
33. Horn B.K.P. Robot Vision. Cambridge, MA : MIT Press, 1986.
34. Huang J. Iris Model Based on Local Orientation Description // The First International Conference on Machine Learning and Cybernetics. 2002. P. 450–454.
35. Iris Recognition. http://cnx.org/content/col10256/1.1/ (дата обращения: 10.02.2014).
36. Jahne B. Digital Image Processing. 2nd ed. Berlin : Springer-Verlag, 1993.
37. Jain A.K., Ross A., Prabhakar S. An introduction to biometric recognition // IEEE Transactions on Circuits and Systems for Video Technology. January 2004. T. 14 (1). P. 4–20.
38. Krylov A., Korchagin D. Fast Hermite Projection Method // LNCS. 2006. V. 4141, P. 329–338.
39. Miyazawa K., Ito K., Aoki T., Kobayashi K., Nakajima H. A phase-based iris recognition algorithm // Proceedings of the International Conference on Advances on Biometrics (ICB '06). V. 3832 of Lecture Notes in Computer Science. P. 356–365. Springer, Hong Kong, January 2006.
40. Note on CASIA-IrisV4. http://www.idealtest.org/dbDetailForUser.do?id=4 (дата обращения: 10.02.2014).
41. Tisse C. Person Identification Technique using Human Iris Recognition // Proc. of Vision Interface. 2002. P. 294–299.
42. Tisse C., Martin L., Torres L., Robert M. Person identification technique using human iris recognition / Acoustics, Speech, and Signal Processing // Proceedings ICASSP '05. 2005. V. 2. P. 949–952.
43. Wildes R.P. Iris Recognition: An Emerging Biometric Technology // Proceedings of the IEEE. 1997. V. 85, N. 9.

A REVIEW OF PERSON IDENTIFICATION METHODS USING IRIS RECOGNITION

N.P. Grishenkova[1], Software engineer / Software developer,
e-mail: natalia.grishenkova@gmail.com
D.N. Lavrov[2], Ph.D. (Eng.), Associate Professor, e-mail: Dmitry.Lavrov72@gmail.com

[1]Omskiy Nauchno Issledovatelskiy Institut Priborostroeniya
[2]Omsk State University n.a. F.M. Dostoevskiy

Abstract. The article presents an overview of current methods of person identification and verification using human iris. At first we state some well-known facts about the structure of the eye and its iris. Then the basic principles and quality metrics of biometric systems are described. The next chapter describes the known methods: eye and iris localization, histogram normalization and iris boundaries determination. After that we consider keypoint detection algorithms based on wavelet transforms including those using Gabor filters. Methods of pattern matching based on the Hamming distance and projective phase correlation are described. Finally, we discuss the advantages and disadvantages of the considered approaches.

Keywords: iris, Gabor transform, edge detection, verification, identification, recognition.

Научный журнал

Математические структуры и моделирование

№1(29)

Главный редактор
Д.Н. Лавров

Корректоры:
И.Н. Баловнева
Е.А. Илюшечкин

Художественное оформление
Д.Н. Лавров

Адрес научной редакции
Россия, 644053, Омск-53, ул. Грозненская, 11
Омский государственный университет
факультет компьютерных наук
E-mail: lavrov@omsu.ru
Электронная версия журнала:
http://msm.univer.omsk.su
http://msm.omsu.ru

Подписано в печать 17.06.2014. Формат 60 × 84 1/8.
Усл. печ. л. 7,6. Тираж 125 экз. Заказ № 119.

Отпечатано на полиграфической базе издательства ОмГУ им. Ф.М. Достоевского
644077, г. Омск, пр. Мира, 55А

ISSN 2222-8772