

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

8(180)
2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Ходашинский И. А., Синьков Д. С. Идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц 2
Масликов В. И. Универсальная система категорических суждений силлогистики . 6
Садыков С. С., Савичева С. В. Идентификация реальных плоских объектов на основе единственного признака точек их внешних контуров 13

КОМПЬЮТЕРНАЯ ГРАФИКА

- Бобков В. А., Роньшин Ю. И. Реализация алгоритма построения карт глубин на графических процессорах 17
Гулаков В. К., Трубаков А. О. Эффективный алгоритм преобразования полноцветного изображения к палитре для систем поиска по содержанию. 22

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Беляков С. Л., Белякова М. Л., Самойлов Д. С. Геоинформационный сервис ситуационного центра 29
Струченков В. И., Козлов А. Н., Егунов А. С. Динамическое программирование в проектировании трасс линейных сооружений. 33
Кобзаренко Д. Н. Особенности применения некоторых алгоритмов вычислительной геометрии в геоинформационных технологиях 37

СЕТИ И СИСТЕМЫ СВЯЗИ

- Львович Я. Е., Львович И. Я., Преображенский А. П., Головинов С. О. Исследование метода трассировки лучей при проектировании беспроводных систем связи 40
Мощевикин А. П., Галов А. С., Волков А. С. Локация в беспроводных сетях датчиков стандарта nanoLOC (IEEE 802.15.4a) 43

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Гильманова Н. А. Мультиметодный алгоритм трехмерной упаковки с регулируемым провисанием объектов в контейнере. 48
Ковалевич А. А., Якимов А. И., Албкеират Д. М. Исследование стохастических алгоритмов оптимизации для применения в имитационном моделировании систем . . 55
Пашенко В. Н., Мартынюк В. А., Монинов А. С., Пашенко В. В. Исследование влияния типа конечных элементов на продолжительность и точность расчета температурных напряжений в среде ANSYSWorkbench 61
Асадов Х. Г., Алиева Е. Н. О возможности применения метода многоуровневой оптимизации для оптимальной информационной оценки разнообразия видов в экологических системах 64

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

- Кузнецов А. А. Системная связь между процессами ритма сердца и динамики систолического потенциала 69
Галкин В. А., Белый Ю. А., Кучеров А. А. Информационное и программное обеспечение антимикробной фотодинамической терапии 74
Contents 78
Приложение. Королёв В. И., Новиков А. А., Шарков А. Е. Электронное правительство: анализ, концептуальное представление, функциональные приложения

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ФОМИЧЕВ В. А.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.82

И. А. Ходашинский, д-р техн. наук, проф.,
e-mail: hodashn@rambler.ru,
Д. С. Синьков, аспирант,
e-mail: express@sibmail.com,
Томский государственный университет
систем управления и радиоэлектроники

Идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц

Рассмотрены усовершенствования стандартного алгоритма роящихся частиц. Описана идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц. Приведены результаты имитационного эксперимента.

Ключевые слова: идентификация нечетких систем, метаэвристики, адаптивный алгоритм роящихся частиц

Введение

Под нечеткой системой будем понимать отображение из заданной области определения в область значений, представленное в виде процедур фаззификации, нечеткого вывода и дефаззификации.

Нечеткое моделирование включает два основных этапа: идентификацию структуры и идентификацию параметров нечеткой системы. Идентификация структуры — определение таких характеристик нечеткой системы, как число нечетких правил и число лингвистических термов, которыми представлены входные и выходные переменные. Идентификация параметров — это определение неизвестных параметров antecedентов и consequентов нечетких правил путем оптимизации работы нечеткой системы по заданному критерию. Для оптимизации параметров нечетких систем используют две группы методов. Первая группа — классические методы оптимизации, основанные на производных: метод наименьших квадратов, градиентный метод, фильтр Калмана, метод Левенберга—Марквардта. Эти методы дают точные результаты, но они имеют тенденцию сходиться к локальным оптимумам. Трудности применения классических методов оптимизации, в частности проблема локального экстремума и "проклятие

размерности", заставляют обратиться ко второй группе методов — метаэвристических, таких как алгоритмы муравьиной колонии, роящихся частиц, имитации отжига, генетические алгоритмы [1].

Алгоритм оптимизации на основе роящихся частиц благодаря своей простоте нашел широкое применение в решении научных и технических задач [2]. Использовался этот алгоритм и в идентификации нечетких систем [3, 4]. Однако несмотря на достаточную эффективность у алгоритма выявились и некоторые недостатки, главным из которых является преждевременная сходимость. Было предпринято много попыток устранить указанный недостаток. Условно эти попытки можно разделить на несколько категорий. Первая категория связана с изменением параметров самого алгоритма в процессе его работы [5–7] или топологии взаимодействия частиц [8], вторая — с включением в алгоритм дополнительных операторов типа кроссовера и мутации [9], смеси взвешенных функций Гаусса [10, 11] или функций Вейбулла [12]. Третья категория направлена на предотвращение образования группировок частиц путем введения в рой лучшей, но "хищной" частицы или отрицательной энтропии [13].

Постановка задачи

Рассматривается нечеткая система типа сингтон, i -е правило в которой имеет следующий вид:

$$\text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \\ \dots \text{ AND } x_n = A_{ni} \text{ THEN } y = r_i,$$

где A_{ij} — лингвистический терм, которым оценивается переменная x_j ; r_i — действительное число, которым оценивается выход y .

Нечеткая система осуществляет отображение $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$:

$$f(\mathbf{x}) = \sum_{i=1}^R \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \dots \times \\ \times \mu_{A_{ni}}(x_n) \cdot r_i / \sum_{i=1}^R \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \dots \mu_{A_{ni}}(x_n),$$

где \mathbf{x} — входной вектор; R — число правил; n — число входных переменных; $\mu_{A_{ij}}$ — функция принадлежности, определяемая набором своих параметров, например, треугольная — тремя параметрами, трапециевидная — четырьмя, гауссова и параболическая — двумя.

Нечеткая система может быть представлена как

$$y = f(\mathbf{x}, \boldsymbol{\theta}),$$

где y — скалярный выход системы, $\boldsymbol{\theta} = \|\theta_1, \dots, \theta_N\|$ — вектор параметров, $N = nab$, a — число параметров, описывающих одну функцию принадлежности, b — число термов, описывающих одну входную переменную.

Пусть дано множество обучающих данных (таблица наблюдений) $\{(\mathbf{x}_p; t_p), p = 1, \dots, m\}$, тогда средняя квадратическая функция ошибки, являющаяся численным критерием адекватности модели, вычисляется по следующей формуле:

$$E(\boldsymbol{\theta}) = \frac{1}{m} \sqrt{\sum_{p=1}^m (t_p - f(\mathbf{x}_p, \boldsymbol{\theta}))^2}. \quad (1)$$

Проблема идентификации сводится к проблеме поиска минимума заданной функции в многомерном пространстве, координаты которого соответствуют параметрам $\boldsymbol{\theta}$ нечеткой системы. Для решения указанной проблемы в нашей работе предлагается использовать адаптивный алгоритм роящихся частиц.

Стандартный алгоритм роящихся частиц

Алгоритм роящихся частиц — это стохастический метод поиска, основанный на итеративном взаимодействии частиц, образующих рой. Перемещение частицы в пространстве поиска определяют три фактора: инерция, память, сотрудничество. Инерция подразумевает, что частица не может мгновенно изменить свое направление движения. Каждая частица имеет память и хранит свою лучшую позицию в пространстве поиска. Известна частице и лучшая позиция роя. Зная эти две позиции, частица динамически изменяет скорость согласно ее собственному опыту и опыту полета других частиц. Таким образом, движение каждой частицы задается ее лучшей позицией, ее текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое. Рой прекращает движение при выполнении хотя бы одного из следующих условий: рой достиг состояния равновесия; найдено оптимальное решение (ошибка меньше заданной); выполнено заданное число итераций.

В нашем случае пространство поиска $\boldsymbol{\theta} \subset \mathbb{R}^N$, рой состоит из M частиц. Позиция i -й частицы определяется вектором

$$\mathbf{x}_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iN}) \in \boldsymbol{\theta}.$$

Лучшая позиция, которую занимала i -я частица, определяется вектором

$$\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iN}) \in \boldsymbol{\theta}.$$

Скорость частицы определяется также N -местным вектором

$$\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iN}) \in \boldsymbol{\theta}.$$

Скорость и положение i -й частицы на $k + 1$ шаге определяют следующие уравнения:

$$\mathbf{v}_i(k + 1) = w\mathbf{v}_i(k) + c_1 \text{rand}(\mathbf{p}_i(k) - \mathbf{x}_i(k)) + c_2 \text{Rand}(\mathbf{p}_g(k) - \mathbf{x}_i(k)); \quad (2)$$

$$\mathbf{x}_i(k + 1) = \mathbf{x}_i(k) + \mathbf{v}_i(k + 1), \quad (3)$$

где $i = 1, 2, \dots, M$; $\mathbf{v}_i(k)$ — вектор скорости частицы i на итерации k ; $\mathbf{x}_i(k)$ — координаты частицы i на итерации k ; c_1, c_2 — положительные коэффициенты ускорения; $\mathbf{p}_i(k)$ — лучшая позиция частицы i на первых k итерациях; $\mathbf{p}_g(k)$ — глобально лучшая позиция частицы в рое на первых k итерациях (задается индексом g); w — эмпирический коэффициент инерции; rand , Rand — случайные числа из интервала $[0, 1]$.

Коэффициент c_1 является когнитивным (познавательным) параметром, отражающим доверие частицы к ее собственному прошлому опыту, этот коэффициент ответственен за обнаружение новых областей в пространстве поиска. Коэффициент c_2 является социальным параметром, показывающим насколько частица доверяет рое, этот коэффициент ответственен за исследование окрестностей ранее найденной перспективной области. Коэффициент инерции w ответственен за изменение скорости и управляет обнаружением новых областей и поиском в окрестностях перспективной области.

Собственно алгоритм идентификации параметров нечетких систем приведен ниже.

Вход. Параметры алгоритма: $c_1, c_2, w, M; k_{\max}$ — максимальное число итераций или E_{\min} — минимальное значение ошибки.

Выход. Вектор оптимальных параметров $\boldsymbol{\theta}$.

1. Создание случайной начальной популяции частиц, $\mathbf{v}_i(0) = 0; k = 1$.
2. Оценка каждой частицы роя путем вычисления ошибки по формуле (1).
3. Если достигнуто условие окончания работы (k_{\max}, E_{\min}), то ОСТАНОВ.
4. Для каждой частицы определение ее вектора скорости $\mathbf{v}_i(k + 1)$ и новых координат $\mathbf{x}_i(k + 1)$ по формулам (2) и (3), соответственно.
5. Расчет новых лучших позиций каждой частицы $\mathbf{p}_i(k + 1)$.
6. Определение частицы с лучшей позицией в рое $\mathbf{p}_g(k + 1)$;
7. $k = k + 1$;
8. Переход к шагу 2.

Алгоритм роящихся частиц применялся нами и для оптимизации консеквентов правил. Для модели типа сингтон в этом случае в качестве координат частиц выступают значения консеквентов базы правил.

Адаптивный алгоритм роящихся частиц

Хорошая настройка популяционного алгоритма означает поиск компромисса между сходимостью и разнообразием. Высокое разнообразие означает рассеивание частиц по всему пространству поиска. Высокая сходимость предполагает расположение частиц в небольшой области, что позволяет интенсивно искать и находить оптимум. Лучшая стратегия заключается в том, чтобы сосредоточиться на разнообразии в начале поиска и на сходимости на заключительной стадии. В адаптивный алгоритм роящихся частиц внесено изменение параметров самого алгоритма в процессе поиска. К таким параметрам относятся коэффициенты ускорения и коэффициент инерции.

Ниже приведены три возможных способа изменения коэффициента инерции.

1. Линейное уменьшение от максимального до минимального значения:

$$w(k) = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}} \right) k,$$

такое изменение коэффициента инерции позволяет частицам далеко разлетаться на ранних стадиях поиска и собираться вместе на поздних стадиях поиска.

2. Случайное изменение, зависящее от предыдущего:

$$w(k+1) = w(k) + \alpha N(0,1),$$

где α — параметр обучения; $N(0,1)$ — случайное число с гауссовым распределением.

3. Случайное изменение, не зависящее от предыдущего,

$$w(k) = a + \text{rand}/2,$$

где a — параметр, принимающий значения в интервале $[0,1; 0,75]$; rand — равномерно распределенное случайное число в интервале $[0, 1]$.

Высокое разнообразие в начале и уменьшение его к концу поиска в пользу высокой конвергенции можно осуществить, меняя коэффициенты ускорения. Указанные величины могут быть изменены следующим образом: линейное уменьшение c_1 от максимального до минимального значения и увеличение c_2 от минимального до максимального значения:

$$c_1(k) = c_{1\max} - \left(\frac{c_{1\max} - c_{1\min}}{k_{\max}} \right) k;$$

$$c_2(k) = c_{2\min} + \left(\frac{c_{2\max} - c_{2\min}}{k_{\max}} \right) k.$$

Чтобы регулировать влияние лучшей частицы в рое и избежать движения всех частиц к этой лучшей частице, в работе [6] введен дополнительный

параметр — функция предпочтения $PF_i(k)$. С учетом этого параметра скорость i -й частицы на $(k+1)$ -м шаге определяется следующим образом:

$$\mathbf{v}_i(k+1) = w(k) \cdot \mathbf{v}_i(k) + c_1 \text{rand}(\mathbf{p}_i(k) - \mathbf{x}_i(k)) PF_i(k) + c_2 \text{Rand}(\mathbf{p}_g(k) - \mathbf{x}_i(k))(1 - PF_i(k)).$$

Процедура вычисления функции предпочтения приведена ниже.

1. Вычислить для каждой частицы расстояние от i -й частицы до лучшей позиции в рое:

$$d_i = \|\mathbf{p}_g(k) - \mathbf{x}_i(k)\|^2.$$

2. Сортировать по возрастанию $\{d_i | i = 1, 2, M\}$ и присвоить полученные индексы положения вектору $\mathbf{s} = \{s_1, s_2, \dots, s_M\}$.

3. Вычислить для каждой i -й частицы функцию предпочтения:

$$PF_i(k) = \frac{s_i}{M},$$

либо по формуле

$$PF_i(k) = \frac{d_i(k)}{\sum_{j=1}^M d_j(k)}.$$

4. Конец процедуры.

Исследование алгоритма

Исследование алгоритмов идентификации нечетких систем проводилось при решении задач аппроксимации. В качестве тестовых были выбраны следующие функции:

$$1) f(x_1, x_2) = x_1 \sin(x_2), -\pi/2 < x_1, x_2 < \pi/2;$$

$$2) f(x_1, x_2) = \sin(2x_1/\pi) \sin(2x_2/\pi), -5 < x_1, x_2 < 5;$$

$$3) f(x) = (1 + 10 \exp(-100(x - 0,7)^2)) \left(\frac{\sin\left(\frac{125}{x + 1,5}\right)}{x + 0,1} \right),$$

$$0 < x < 1;$$

$$4) f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1,5})^2, x_1, x_2 \in [1, 5].$$

На основе тестовой функции формировалась таблица наблюдений, по которой строилась нечеткая система, аппроксимирующая данную функцию. Критерием качества аппроксимации была средняя квадратическая ошибка вывода. Лингвистические переменные разбиты на пять термов. Так как алгоритм роящихся частиц является стохастическим алгоритмом, то проводилось по 10 испытаний с одними и теми же параметрами.

Эксперименты с изменяющимися коэффициентами ускорения и инерции показали, что ошибка

вывода уменьшается при увеличении числа итераций и числа особей в популяции. Для первоначальной настройки целесообразно использовать алгоритм с 30 особями и числом итераций 400–600, так как при данных параметрах достигается компромисс между временем идентификации и значением ошибки вывода.

Самой эффективной модификацией является использование функции предпочтения. Эксперименты показали, что использовать функцию предпочтения необходимо на 300–400 итерациях, после чего следует запускать алгоритмы, основанные на производных, либо алгоритм настройки консеквентов правил.

При аппроксимации первой и второй тестовых функций порядок ошибки определялся восьмым знаком. На рис. 1 и 2 (см. 3-ю сторону обложки) представлены функции принадлежности, полученные при аппроксимации второй тестовой функции.

Были проведены сравнения алгоритма роящихся частиц с другими алгоритмами идентификации нечетких систем. Параметры нечетких систем, аппроксимирующих заданные функции, представлены в таблице.

При аппроксимации третьей тестовой функции для алгоритма *Mitaim* и *Kosko* [14] ошибка аппроксимации составила 1,426, для алгоритма *Lisin* и *Gennert* [14] — 0,247, для алгоритма роящихся частиц — 0,008.

При аппроксимации четвертой тестовой функции для алгоритма *Rojas*, *Pomares*, *Ortega*, *Prieto* [15] ошибка аппроксимации составила 0,026, для алгоритма роящихся частиц — 0,002.

Таким образом, можно сделать вывод, что алгоритм роящихся частиц точнее сравниваемых алгоритмов.

При идентификации параметров функций принадлежности алгоритмом роящихся частиц использовалась функция предпочтения, коэффициент инерции изменялся случайным образом независимо от предыдущей итерации $a = 0,3$; $c_1 = 1,4$; $c_2 = 1,4$; число итераций — 200; число особей — 30. Настройку параметров консеквентов проводили на следующих параметрах алгоритма: $c_1 = 1,4$; $c_2 = 1,4$; $w = 0,7$; число итераций — 200; число особей — 3.

Параметры нечетких систем

№ функции	Число входных переменных	Число точек в таблице наблюдений	Число термов, представляющих входную переменную
1	2	81	5
2	2	121	5
3	1	100	12
4	2	400	5

Идентификация параметров нечеткой системы велась итерационно: вначале настраивались консеквенты, затем антецеденты.

Заключение

Адаптивный алгоритм роящихся частиц основан на динамической корректировке параметров алгоритма, он сохраняет хорошее равновесие между ранней сходимостью и разнообразием. Результаты аппроксимации тестовых функций показали, что алгоритм имеет хорошую способность поиска и высокую скорость сходимости при решении задачи аппроксимации нелинейных функций.

Список литературы

1. **Ходашинский И. А.** Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. № 4. С. 15–23.
2. **Innovations in Swarm Intelligence** / С. P. Lim, L. C Jain, S. Dehuri (Eds.) // Studies in Computational Intelligence. Vol. 248. Berlin; Speinger-Verlag, 2009. 254 p.
3. **Ходашинский И. А.** Идентификация параметров нечетких моделей типа сингтон на основе алгоритма роящихся частиц // Информационные технологии. 2009. № 6. С. 8–11.
4. **Juang C.-F., Hsiao C.-M., Hsu C.-H.** Hierarchical Cluster-Based Multispecies Particle-Swarm Optimization for Fuzzy-System Optimization // IEEE Transactions on Fuzzy Systems. 2010. Vol. 18, N 1. P. 14–26.
5. **Ratnaweera A., Halgamuge S. C., Watson H. C.** Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients // IEEE Transactions on Evolutionary Computation. 2004. Vol. 8, N 3. P. 240–255.
6. **Chen H.-C., Chen O. T.-C.** Particle Swarm Optimization incorporating a Preferential Velocity-Updating Mechanism and Its Applications in IIR Filter Design // IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan. 2006. P. 1190–1195.
7. **Zhan Z.-H., Zhang J., Y. Li, Chung H. S.-H.** Adaptive Particle Swarm Optimization // IEEE Transactions on Systems, Man, and Cybernetics — PART B: Cybernetics. 2009. Vol. 39, N 6. P. 1362–1381.
8. **Montes de Oca M. A., Stützle T., Birattari M., Dorigo M.** Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm // IEEE Transactions on Evolutionary Computation. 2009. Vol. 13, N. 5. P. 1120–1132.
9. **Arumugam M. S., Rao M. V. C.** On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems // Applied Soft Computing. 2008. Vol. 8. P. 324–336.
10. **El-Abd M., Kamel M. S.** A cooperative particle swarm optimizer with migration of heterogeneous probabilistic models // Swarm Intelligence. 2010 Vol. 4. P. 57–89.
11. **Iqbal M., Montes de Oca M. A.** An estimation of distribution particle swarm optimization algorithm / M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stutzle (Eds.), LNCS: Vol. 4150 // Proc. of the Fifth International workshop on ant colony optimization and swarm intelligence. Berlin: Springer, 2006. P. 72–83.
12. **Yan Y., Osadciw L. A.** Density estimation using a new dimension adaptive particle swarm optimization algorithm // Swarm Intelligence. Vol. 3. P. 275–301.
13. **Banks A., Vincent J., Anyakoha C.** A review of particle swarm optimization. Part I: background and development // Natural Computing. 2007. Vol. 6. P. 467–484.
14. **Lisin D., Gennert M. A.** Optimal Function Approximation Using Fuzzy Rules // Proc. 18th Int. Conf. North American Fuzzy Information Processing Society. 1999. P. 184–188.
15. **Teng Y.-W., Wang W.-J., Chiu C.-H.** Function approximation via particular input space partition and region-based exponential membership functions // Fuzzy Sets and Systems. 2004. Vol. 142. P. 267–291.

В. И. Масликов, доц.,
 ГОУ ВПО Дальневосточный государственный
 гуманитарный университет, г. Хабаровск,
 e-mail: vlad@email.kht.ru

Универсумная система категорических суждений силлогистики

Разработан полный базис суждений силлогистики типов А, Е, I, О, а, е, i, о. Определена связь всех типов суждений с булевой алгеброй, диаграммами Эйлера—Венна и древовидными структурами. Полученная система логических отношений между суждениями применима для разработок в областях баз знаний, нейронных и интеллектуальных систем.

Ключевые слова: универсум, логические отношения, булева алгебра, силлогизм, силлогистика, искусственный интеллект, нейросети

Введение

Ряд методологических проблем и некоторых парадоксов не позволяет в полной мере использовать потенциал силлогистики как одного из важных разделов логики, на базе которого могут развиваться современные информационные технологии. Одной из причин такого положения является отсутствие максимально универсальной, единой системы описаний логических отношений, которая, удовлетворяя условиям различных языковых систем, была бы совместима и в достаточной мере соответствовала потребностям смежных направлений научных исследований.

Несмотря на богатый логический инструментарий большинство специалистов опирается всего на пять—семь основных логических понятий, чаще всего используя отношения равнообъемности, импликация, перекрещивания, дополнительности, противоречия и соподчинения. Между тем отмечается, что "логика классическая с ее материальной импликацией не может быть признана удачным описанием условной связи, а значит, и логического следования" [1, с. 265]. Отсутствие четкой определенности описания логических отношений, в свою очередь, неизбежно приводит к некоторым неопределенностям в теории силлогистики.

Потребность в универсальной системе логических функций (подобной периодической системе химических элементов), интегрирующих булеву алгебру, теорию множеств, силлогистику, графы, древовидные структуры, описания алгоритмов и другие привычные компоненты научного инстру-

ментария, несомненна, ведь универсальность, кроме всего прочего, — один из надежных критериев истинности.

Для создания такой системы может быть использована триединая философская система "Материя—Информация—Мера" [2, с. 317], полная функция управления [2, с. 16] и универсумно-фреймовый метод описания объектов окружающего мира [3, с. 10], приложенный к классическим фундаментальным отношениям между понятиями, применяемыми в логике.

Универсумная методология решения интеллектуальных задач

В основу современного мироописания положена философская система предельно обобщенных философских категорий "Материя — Энергия — Пространство — Время". Эта система не лишена недостатков [2]. Альтернативная философская система триединства "Материя — Информация — Мера" позволяет определить **Универсум** (Universum, U) — единую, универсальную единицу описания любого элемента Вселенной как единства Материи (М), Информации (И) и Меры, в которой категория Мера определяет некоторое численное соотношение между Материей и Информацией.

Функционирование универсумов отражается в процессах, которые можно назвать информационно-материальными, материально-информационными (ИМ, МИ) или, обобщенно, универсумными

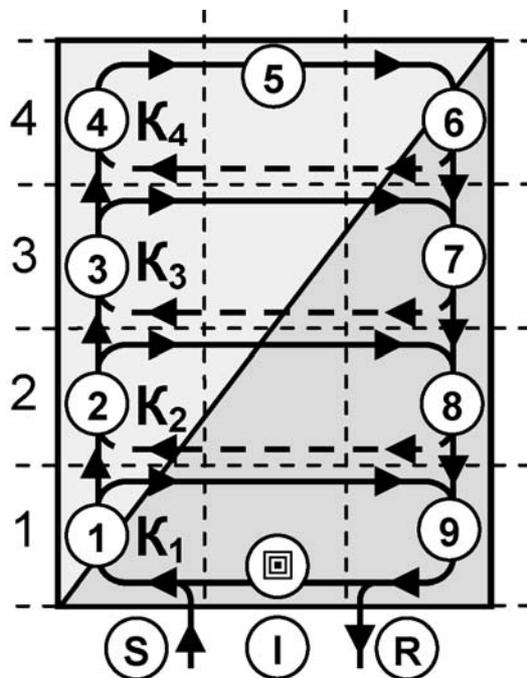


Рис. 1. Графическое представление универсума класса 4U3 (S — входной, I — внутренний, R — выходной U-поток)

U-потоками. Графически универсум можно определить как единую совокупность Материи и Информации, представленную как прямоугольник, разграниченный Мерой — диагональной линией, условно отделяющей пространства М (нижняя часть прямоугольника под диагональю, рис. 1) и И (верхняя часть над диагональю, рис. 1).

Протекающие в универсуме U-потоки могут перемещаться в любых направлениях, но в пределе разбиваются всего на два класса:

- претерпевающие только количественные изменения при распространении в пределах универсума, т. е. такие потоки, при движении которых их ИМ-соотношение не меняется (обозначены горизонтальными стрелками);
- претерпевающие только качественные изменения при распространении в пределах универсума, т. е. такие потоки, при движении которых их ИМ-соотношение меняется (обозначены вертикальными стрелками).

Перемещение U-потока вверх по универсуму изменяет МИ-соотношение от М, бесконечно близкого к 100 %, до уровня И, бесконечно близкого к 100 %, перемещение вниз — наоборот. Перемещение горизонтального U-потока не меняет его МИ-соотношение.

Универсумная логика подразумевает последовательное изменение процентного соотношения между материей и информацией при движении U-потоков в вертикальном направлении и фиксации этого соотношения при распространении в горизонтальной плоскости.

Класс универсума — число горизонтальных и вертикальных фреймов (дискретов) разбиения универсума. Так, универсум с тремя горизонтальными разбиениями по числу полученных фреймов записывается как 4U, универсум с четырьмя вертикальными разбиениями — пятью каскадами — как U5.

Матричный универсум допускает оба вида дискретизации: и по горизонтали, и по вертикали.

В приведенном на рис. 1 обобщенном описании матричного универсума класса 4U3 отображены четыре горизонтальных контура внутреннего обращения U-потоков: контур K_1 содержит максимальную часть материальной составляющей, в контурах K_2 и K_3 доля информационной составляющей последовательно возрастает, контур K_4 содержит максимальную меру информационной составляющей.

Как любой объект мироздания, универсум является открытой системой. Каскад S учитывает входной U-поток, каскад R — выходной, промежу-

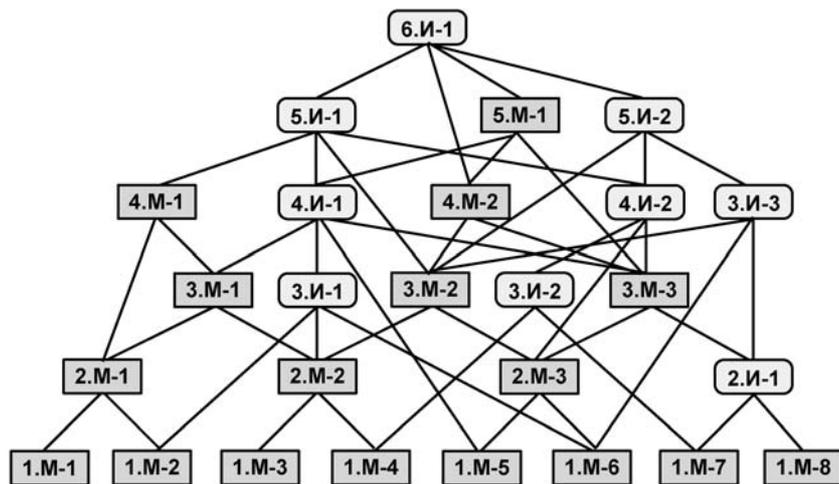


Рис. 2. Шестиуровневая база знаний, организованная в виде универсумного "дерева"

точный набор фреймов I является каскадом внутренней (интеллектуальной) обработки U-потоков.

Универсумная методология основана на предельно простом принципе — стратификации явлений, объектов, процессов и знаний по мере изменения в них соотношения "Материя—Информация".

В универсумном описании любая система обработки знаний может быть представлена в виде ориентированного графа, вершины которого состоят из преобразователей материального (прямоугольники "М" на рис. 2) и информационного (прямоугольники "И" со скругленными углами на рис. 2) U-потока, стратифицированных (снизу вверх) по мере увеличения информационной составляющей. Первый слой вершин "1.М-Х" отвечает преимущественно за материальную обработку U-потока, второй слой наряду с материальными преобразователями "2.М-Х" содержит уже и информационный преобразователь "2.И-1". От слоя к слою доля информационных вершин по сравнению с материальными возрастает.

Граф, ребрами объединяющий "М" и "И" вершины, представляет собой сложную систему типа "дерево", поддающуюся формальному описанию средствами классической логики. Например, вершины типа "М" и "И" могут представлять систему двух логических переменных и отражать субъектно-предикатные отношения умозаключений (суждений, высказываний) силлогистики, а ребра графа — описывать связи между терминами.

Полная система логических отношений двух переменных

Универсумный подход позволяет получить полную систему логических отношений (ПСЛО) для любого числа переменных, для чего в качестве универсума из окружающего мира выделяются определенные объемы логических множеств А, В, С и

т. д., что позволяет их рассматривать как взаимосвязанную, цельную (исчерпываемую) систему, между элементами (множествами, фреймами) которой возникают определенные отношения, поддающиеся формальному описанию.

Фрейм (Frame, F) — структурный элемент итерационной декомпозиции универсума.

Фундаментальные отношения между понятиями — совместимость ($A \cap B \neq \emptyset$), исчерпаемость ($A \cup B = U$) и включенность одного понятия в другое ($A \supset B$ и $B \supset A$) — позволяют установить четыре нефундаментальных отношения между объемами и содержимым одной переменной (ПСЛО-1) и 16 для двух переменных (ПСЛО-2), заданных в определенном универсуме.

Для одной переменной A возможны только два типа фундаментальных отношений: совместимость и исчерпаемость. Фреймовое описание множеств под записью двух переменных, например AB, подразумевает не только обычное логическое произведение $A \cdot B$, а используется и как имя фрейма, содержащего соответствующие множества A и B, отношения между которыми на данном уровне рассмотрения могут точно не определяться. Такое использование принципов системного подхода позволяет сосредоточиться на межфреймовых отношениях в универсуме, что, в свою очередь, не мешает исследователю в случае необходимости рассмотреть любой фрейм как отдельный универсум [3, 8] или определить внутрифреймовую систему отношений в зависимости от нужд решаемой задачи.

Идея универсумно-фреймового метода (UF-метода, UFM) состоит в систематизации процесса описания межфреймовых взаимосвязей, интересующих исследователя групп универсумных фреймов. Все возможные фреймы можно выделить из универсума, например комбинационным способом, обозначив в их наименованиях имена тех множеств, из которых он состоит. Затем проводится исследование отношений между фреймами.

UF-описанию двух переменных можно сопоставить множества с субъектно-предикатным вариантом описания отношений, например:

- AB — объекты A, имеющие свойство B;
- A'B — объекты, не входящие в A, но имеющие свойство B;
- AB' — объекты A, не имеющие свойства B;
- A'B' — объекты, не входящие в A и не имеющие свойства B.

Из этого описания следует, что из четырех фреймов для двух переменных A и B универсум составляют только три множества — AB, A'B, AB', множество же A'B' в общем случае определяет объем и ту границу U, за которой находится все неисчерпаемое множество "объектов, не входящих в A и не имеющих свойства B".

Также очевидно, что для понятий A и B два фундаментальных отношения совместимости и исчерпаемости должны дополниться родовидовыми (иерархическими, подчиненными, субординационными) отношениями включенности одного понятия в другое, а именно: отношением включения A в B ($A \supset B$) и отношением включения B в A ($B \supset A$). Четыре типа фундаментальных отношений — совместимость, исчерпаемость и встречные родовидовые — устанавливают 16 производных U-отношений.

Понятие "граница универсума" для полной системы логических отношений двух переменных (ПСЛО-2) определяется фундаментальным свойством "исчерпаемость" и в диаграммах (табл. 1) обозначается прямоугольником, выделенным штриховой и/или сплошной линией. Штриховой линией на кругах Эйлера универсумов отмечены границы родовидовых отношений. В приложении к решаемой задаче определения полной системы отношений силлогистики обратим внимание на универсумные функции U-9—U-14 (табл. 1), поскольку именно эти функции ПСЛО-2 определяют и классифицируют различные типы логических высказываний.

Полная система логических отношений силлогистики

Полученная на основе ПСЛО-2 система отношений силлогистики (табл. 2) содержит следующие основные универсумные функции:

- **Функция U-9 "Эквивалентность"** (Равнозначность, $A \equiv B$) определяется фундаментальным отношением совместимости множеств A и B (фрейм AB). Занимая в U одно и то же пространство, тем не менее, множества A и B иерархически не подчинены друг другу. Функция U-9 обеспечивает установление эквивалентности независимых, не имплицитных простых и сложных фреймов (имен, терминов, суждений и т. д.).
- **Функция U-10 "Отрицание B"** (Инверсия B, $\neg B$, "B не подчиняется A") определяет связь отрицания родовидовых отношений между A и B, где A — род, а B — вид. Видовой объем B размещен во фреймах AB и A'B, родовой объем A — во фреймах AB и AB'. Функция U-10 определяет подчинение вида (антецедента, основания) A роду (консеквенту, следствию) B, а именно: описывает ситуацию, когда положение дел, имеющееся в A, не соответствует положению дел, имеющемуся в B.
- **Функция U-11 "Импликация от B к A"** ($B \rightarrow A$, $B \supset A$, "B подчиняется A") определяет наличие родовидовых отношений между A и B, где A — род, а B — вид. Род A размещен в условно едином объеме фреймов AB и AB', видовой объем

Таблица 1

ПОЛНАЯ СИСТЕМА ЛОГИЧЕСКИХ ОТНОШЕНИЙ ДВУХ ПЕРЕМЕННЫХ																																											
№, назв. функции, фундам. отнош, табл. истинности	Представление в классическом виде множеств	№, назв. функции, фундам. отнош, табл. истинности	Представление в классическом виде множеств																																								
1	2	1	2																																								
0. Константа 0 <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	0	А - род, В - вид	1	0	0	Совместимость	1	1	0		8. Элемент Пирса <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	0	А - род, В - вид	1	0	0	Совместимость	1	1	0	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	0																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	0																																								
1. Конъюнкция <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	0	А - род, В - вид	1	0	0	Совместимость	1	1	1		9. Эквивалентность <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	0	А - род, В - вид	1	0	0	Совместимость	1	1	1	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	1																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	1																																								
2. Запрет по А <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	0	А - род, В - вид	1	0	1	Совместимость	1	1	0		10. Отрицание В <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	0	А - род, В - вид	1	0	1	Совместимость	1	1	0	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	0																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	0																																								
3. Переменная А <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	0	А - род, В - вид	1	0	1	Совместимость	1	1	1		11. Имплик. от В к А <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	0	А - род, В - вид	1	0	1	Совместимость	1	1	1	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	1																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	0																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	1																																								
4. Запрет по В <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	1	А - род, В - вид	1	0	0	Совместимость	1	1	0		12. Отрицание А <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	1	А - род, В - вид	1	0	0	Совместимость	1	1	0	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	0																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	0																																								
5. Переменная В <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	1	А - род, В - вид	1	0	0	Совместимость	1	1	1		13. Имплик. от А к В <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	1	А - род, В - вид	1	0	0	Совместимость	1	1	1	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	1																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	0																																								
Совместимость	1	1	1																																								
6. Сумма по мод. 2 <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	1	А - род, В - вид	1	0	1	Совместимость	1	1	0		14. Штрих Шеффера <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>0</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	1	А - род, В - вид	1	0	1	Совместимость	1	1	0	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	0																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	0																																								
7. Дизъюнкция <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	0	В - род, А - вид	0	1	1	А - род, В - вид	1	0	1	Совместимость	1	1	1		15. Генератор 1 <table border="1"> <tr><td>Фунд.отнош.</td><td>A</td><td>B</td><td>U</td></tr> <tr><td>Исчерпаемость</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>В - род, А - вид</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>А - род, В - вид</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>Совместимость</td><td>1</td><td>1</td><td>1</td></tr> </table>	Фунд.отнош.	A	B	U	Исчерпаемость	0	0	1	В - род, А - вид	0	1	1	А - род, В - вид	1	0	1	Совместимость	1	1	1	
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	0																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	1																																								
Фунд.отнош.	A	B	U																																								
Исчерпаемость	0	0	1																																								
В - род, А - вид	0	1	1																																								
А - род, В - вид	1	0	1																																								
Совместимость	1	1	1																																								

Таблица 2

UF-ОПИСАНИЕ КЛАССИЧЕСКИХ СУЖДЕНИЙ В ПСЛО-2	
Представление в виде диаграмм Эйлера-Венна	Структурная схема субъектно-предикатного (S-P) описания классических типов утверждений в привязке к полной системе логических функций двух переменных (ПСЛО-2)
1	2
<p>U-9 a</p>	<p>(P) А - Металл (S) В - Электропроводен а: МЕТАЛЛ – Электропроводен (P) А - Электропроводен (S) В - Металл</p>
<p>U-10 E</p>	<p>(S) А-Род Металл (P) В-Вид Плавкий, В'-НеПлавкий E: МЕТАЛЛ (А) – Не-НеПлавок (В') А: МЕТАЛЛЫ (А) – Плавки (В) а: МЕТАЛЛ – ВЕЩЕСТВО</p>
<p>U-11 i, I</p>	<p>(S) А-Род Металл (P) В-Вид Мягкий, А'-НеМягкий I: Нек. МЕТАЛЛЫ (А) – Мягкие (В) O: Нек. МЕТАЛЛЫ (А) – НеМягкие (В') а: МЕДЬ – МЕТАЛЛ</p>
<p>U-12 o, O</p>	<p>(S) В-Род Металл (P) А-Вид Мягкий, А'-НеМягкий O: Нек. МЕТАЛЛЫ (В) – НеМягкие (А') I: Нек. МЕТАЛЛЫ (В) – Мягкие (А) а: МЕДЬ – МЕТАЛЛ</p>
<p>U-13 A</p>	<p>(P) В-Род Плавкий (S) А-Вид Металл, А'- НеМеталл А: МЕТАЛЛ (А) – Плавкий (В) E: МЕТАЛЛ (А) – Не-НеПлавкий (В') а: МЕТАЛЛ – ВЕЩЕСТВО</p>
<p>U-14 e</p>	<p>(S) А-Род Металл (В') (P) В-Вид НеЭлектропроводен (А') e: МЕТАЛЛ – Не-НеЭлектропроводен (S) А-Вид Металл (В') (P) В-Род НеЭлектропроводен (А')</p>

В — во фреймах АВ и А'В. Функция U-11 определяет подчинение части вида В (фрейм АВ) роду А (совокупности фреймов АВ и АВ'), т. е. описывает ситуацию, когда положение дел, имеющееся в В, частично соответствует положению дел, имеющемуся в А.

- **Функция U-12 "Отрицание А"** (Инверсия А, ¬А, "А не подчиняется В") определяет связь отрицания родовидовых отношений между В и А, где В — род, а А — вид. Вид А размещен в ус-

ловно едином объеме соседних фреймов АВ и АВ', родовой объем В размещен во фреймах АВ и А'В. Функция U-12 определяет подчинение части вида А (фрейм АВ) роду В (фреймы АВ и АВ'), т. е. описывает ситуацию А'В, когда положение дел, имеющееся в А, частично не соответствует положению дел, имеющемуся в В.

- **Функция U-13 "Импликация от А к В"** (Материальная импликация, А → В, А ⊃ В, "А подчиняется В") определяет наличие родовидовых

отношений между А и В, где В — род, а А — вид. Род В размещен в условно едином объеме соседних фреймов АВ и А'В, видовой объем А размещен во фреймах АВ и АВ'. U-13 определяет ситуацию, когда положение дел, имеющееся для вида А, соответствует положению дел, имеющееся в роде В.

- **Функция U-14 "Штрих Шеффера"** (Отрицание дизъюнкции, "Либо А, либо В") определяется несовместимостью множеств А и В (фреймов АВ', А'В). Составляя единое целое, тем не менее, А и В иерархически подчинены друг другу, причем А подчинено В, и В подчинено А инверсным образом, т. е. в том случае, когда родом является множество А, В автоматически принимает значение вида, и наоборот, когда родом становится В, значение вида принимает А. Можно сказать, что функция U-14 предназначена для контрадикторного разделения дополняющих U множеств А и В.

В полную систему силлогистики также входят функции U-8 и U-15 (см. табл. 1), где U-8 определяет границы, т. е. исчерпаемость U, а U-15 обеспечивает динамическую реорганизацию структуры U.

Табл. 2 устанавливает соответствие между составляющими фреймов А, А', В, В' и субъектно-предикативными (S и P) частями высказываний, их смысловое наполнение представлено также в виде древовидной структуры, описывающей классические умозаключения силлогистики на конкретных примерах, подтверждающих однозначное соответствие между U-описанием и а-, Е-, I-, O-, А- и е-утверждениями силлогистики. На схемах в колонке 2 табл. 2 двойной линией отмечена связь типа "а", простой линией — связь типа "А", простой линией со встречно направленными стрелками — связь типа "Е", штриховой линией — связь типа "I", штриховой линией со встречно направленными стрелками — связь типа "O", двойной линией со встречно направленными стрелками — связь типа "е".

Универсумно-логический квадрат силлогистики

Сведение всех утверждений в общей U-схеме позволяет получить единое, универсумное описание всех отношений а-, Е-, I-, O-, А-, е-высказываний, точно соответствующее общеизвестному логическому квадрату формальной логики (рис. 3).

На основании этого соответствия вполне можно утверждать, что полученная из булевых функций посредством UF-метода система описания силлогистических высказываний в виде диаграмм

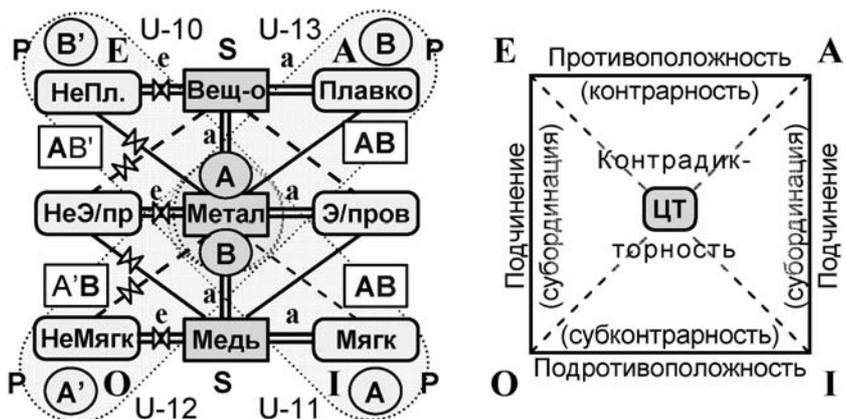


Рис. 3. Универсумное и классическое описание логического квадрата (ЦТ — Центральный Термин)

Эйлера—Венна и структур типа "дерево", по крайней мере, не противоречит некоторым ранее полученным логическим схемам. Кроме того, UF-описание позволяет внести некоторые уточнения в систему категорических высказываний.

Так, например, из U-схемы становится очевидным, что все суждения Е-, I-, O- и А-типов относятся к родовидовым (подчиненным, субординационным) отношениям. Частные же высказывания типа а и е, описывающие нижние границы, пределы дискретизации понятий, должны быть дополнены частными высказываниями типа і и о. В самом деле, если в моделях типа "дерево", построенных по принципу "от общего к частному", высказывания а и е определяют крайние ветви структур, то почему этими краями не могут стать высказывания типа і и о? Ведь многие множества окружающего мира имеют мульти- и дихотомическую природу и, следовательно, определенная часть ветвей должна и может завершаться процессом именно такого деления.

Единичные частные і- и о-высказывания

Схема образования единичного частного утвердительного высказывания і представлена на рис. 4. От родовидового утверждения типа I, связывающего два субъективных термина (например, "Металл" — "Медь", см. рис. 3) и один предикативный термин ("Мягкий", см. рис. 3), частные термины і и о отличаются связыванием трех субъективных терминов — одного родового ("Металлы") и, как минимум, двух видовых ("Медь" и "Платина", рис. 4). Очевидно, что множество "Металлы" может быть продолжено рядом элементов, принадлежащих этому же множеству: "Железо", "Золото", "Свинец" и т. д. При этом часть множества "Металлы", определяющая отношения типа і, получит предикатное свойство "Мягкий" ("Медь", "Золото", "Свинец"), а другая часть, определяющая тип отношения о, обязательно получит свойство "Твердый" ("Платина", "Железо").

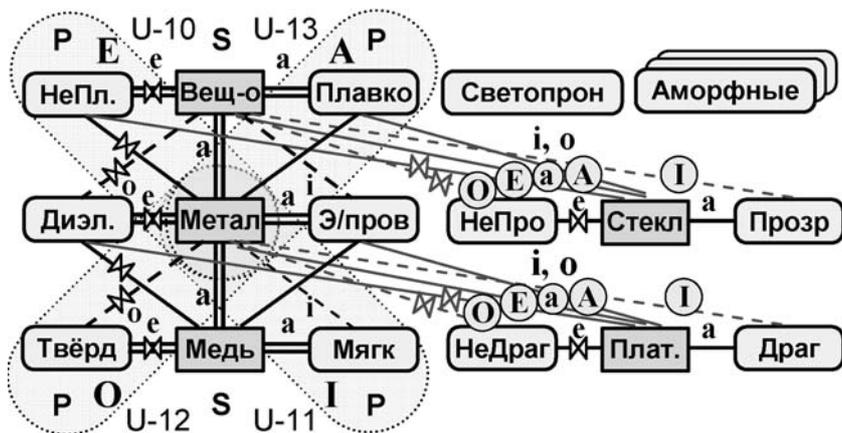


Рис. 4. Отношения подчинения (субординации, родовидовые) в универсумном (UF) описании терминов и суждений

В остальном единичные отношения типа *i* и *o* принципиально не отличаются от родовидовых *I* и *O*, и в колонке 1 табл. 2 они правомерно отнесены к одним и тем же универсумным функциям *U-11* и *U-12*.

Следует подчеркнуть три особенности единичного деления:

- для получения из отношений типа *a* и *e* отношения типа *i* и *o* объем делимого понятия должен состоять более чем из одного элемента с несовпадающими предикатами;
- при наличии только одного элемента в подчиненном множестве с *i*-связью она автоматически преобразуется в связь типа *a*; при наличии только одного элемента с *o*-связью связь приобретает тип *e*;
- в статической универсумной схеме отношений (см. рис. 2) родовидовые связи не могут содержать пересекающихся множеств, поскольку такие множества относятся к функции *U-15*, определяющей динамический процесс реорганизации структур.

Полная система категорических суждений силлогистики

Полная система категорических суждений силлогистики содержит восемь описанных выше элементов: родовидовые общие суждения *E*, *A*; родовидовые частные суждения *I*, *O*; единичные общие суждения *a*, *e*; единичные частные суждения *i*, *o*.

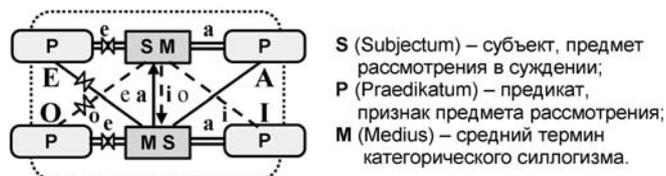


Рис. 5. Полная схема всех типов SP-суждений простых категорических силлогизмов

Взаимосвязь этих суждений между собой приведена на рис. 5, в котором описание субъектно-предикатных отношений дополнено включением в схему позиции *M* — среднего термина категорических силлогизмов. Наличие на этой схеме встречно направленных стрелок *a* и *i* между субъектами *SM* и *MS* отражает факт смены родовидовых "полюсов" на границе между универсумами *U-11* и *U-12* (см. также вершины *A* и *B* на рис. 3). Вопрос описания силлогизмов, в том числе и содержащих родовидовую связь *e*- и *o*-типов, в данном исследовании не ставился. Точечными линиями на рис. 5 отмечены четыре до-

полнительных варианта структур силлогизмов, анализ которых также не входит в тематику данного исследования.

Заключение

Полученная с использованием UF-метода ПСЛО-2 позволила:

- определить и описать связь между булевыми функциями, множествами, графами и силлогистическими суждениями;
- точно описать классические типы суждений *a*, *E*, *I*, *O*, *A*, *e*, дополнив этот список *i*- и *o*-суждениями;
- определить области использования всех восьми типов суждений, связав их со структурами типа "дерево" на конкретных примерах;
- уточнить некоторые требования, предъявляемые к силлогистическим структурам.

В результате проведенного исследования получена универсумная схема категорических суждений силлогистики, расширяющая возможности научного инструментария исследователя и поддающаяся эффективной компьютерной алгоритмизации логических вычислений.

Кроме того, полученная система позволяет точно определить многие понятия формальной логики, систематизировать алгоритмику осуществления дедуктивных и индуктивных методов исследований, снять ряд противоречий классической логики и разработать новые методы решения научных и практических задач.

СПИСОК ЛИТЕРАТУРЫ

1. Ивин А. А., Никифоров А. Л. Словарь по логике. М.: Гуманит. изд. центр ВЛАДОС, 1997. 384с.
2. Достаточно общая теория управления. Постановочные материалы учебного курса факультета прикладной математики — процессов управления Санкт-Петербургского государственного университета (1997—2003 гг.). Санкт-Петербург: 2003. 419 с.
3. Масликов В. И. Универсум: эволюция мыслящей материи. Хабаровск: Изд-во Приамурского географического общества (ПГО). 2008. 192 с.

С. С. Садыков, д-р техн. наук, проф.,
e-mail: sadykovss@yandex.ru,

С. В. Савичева, аспирант,
e-mail: savicheva.svetlana2010@yandex.ru,
Муромский институт (филиал)

Владимирского государственного университета

Идентификация реальных плоских объектов на основе единственного признака точек их внешних контуров

Предложен алгоритм идентификации реальных плоских объектов, использующий лишь значения кривизны точек внешних контуров их бинарных изображений. Работа алгоритма иллюстрируется примерами.

Ключевые слова: идентификация реальных объектов, значение кривизны, бинарное изображение, функция кривизны, признак

Введение

В настоящее время системы технического зрения (СТЗ) находят все более широкое применение в различных областях промышленного производства.

Важной производственной задачей, требующей применения СТЗ, является упорядочение неориентированных деталей без участия человека. Решение этой задачи необходимо для автоматизации таких операций, как кассетирование, сортировка деталей, комплектация компонентов для сборки и т. п. В функции СТЗ здесь входит определение типа детали, ее положения и ориентации [2]. Определение типа детали в поле зрения СТЗ является задачей распознавания и идентификации образов, требующей выбора и использования набора эффективных признаков. Подавляющее большинство СТЗ используют для классификации, идентификации и распознавания объектов значительное число признаков, что увеличивает временные затраты [3]. Это затрудняет выполнение процесса идентификации в реальном времени. Поэтому задача разработки алгоритмов идентификации объектов на основе минимального числа признаков является актуальной.

Ниже предлагается алгоритм идентификации с использованием лишь одного признака точек контура реальных объектов.

Алгоритм идентификации

Алгоритм идентификации состоит из следующих этапов.

1. Предварительная обработка полутоновых изображений плоских объектов (сглаживание шумов, сегментация-бинаризация).

2. Приведение бинарных изображений объектов к исходному (стандартному) положению. Для этого:

а) вычисляется центр тяжести бинарного изображения объекта;

б) осуществляется совмещение центра тяжести бинарного изображения объекта с центром поля зрения СТЗ;

в) определяется длина линии (линия длины объекта), соединяющей максимально удаленные друг от друга две точки бинарного изображения объекта;

г) вычисляется угол наклона линии длины объекта к горизонтали;

д) бинарное изображение объекта поворачивается вокруг центра тяжести на вычисленный угол наклона так, чтобы линия длины объекта совпала с горизонтальной линией.

3. Выделение безразрывного контура бинарного изображения объекта.

4. Вычисление значения кривизны α в каждой точке контура и формирование α -функции объекта:

а) вычисление значения кривизны α в каждой точке полученного контура путем последовательного его обхода;

б) формирование α -функции объекта; α -функция объекта представляет собой последовательность (совокупность) значений кривизны всех точек контура объекта, расположение значений кривизны в данной последовательности (α -функции) определяется правилом обхода точек контура.

5. Идентификация объектов путем сравнения их α -функций.

Этап 1 реализуется с использованием алгоритма, рассмотренного в работе [1].

Этапы 2, 3 реализуются аналогичными алгоритмами, приведенными в работах [2, 3].

Вычисление значения кривизны α в каждой точке контура (этап 4, а). Пусть отрезок непрерывной линии ω (рис. 1) представлен в памяти машины дискретными точками K, L, M, N, O .

Вычисление кривизны состоит в следующем.

1. Последовательно рассматривается каждая точка контура.

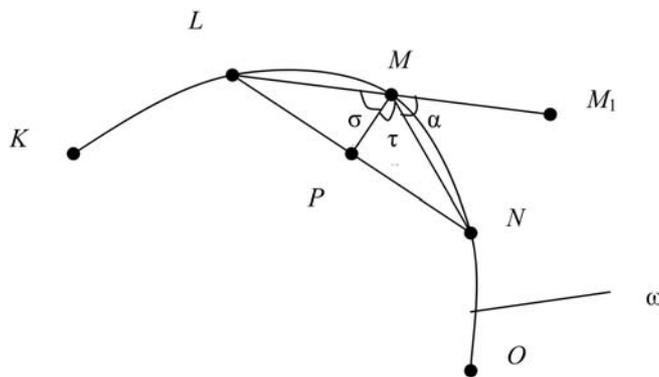


Рис. 1

Таблица 1

№ п/п точки контура	Объект 1	Объект 2	...	Объект N
1	α_{11}	α_{12}	...	α_{1N}
2	α_{21}	α_{22}	...	α_{2N}
...
M	α_{M1}	α_{M2}	...	α_{MN}

Пусть M — текущая точка, где значение кривизны α кривой ω следует определить с использованием ее соседних точек справа и слева, L и N соответственно. Соединив точки L , M и N , получим треугольник LMN .

2. Вычисления длин сторон $LM = d$, $MN = e$, $LN = f$ треугольника LMN выполняются на основе следующих формул [4]:

$$LM = d = \sqrt{(x_L - x_M)^2 + (y_L - y_M)^2};$$

$$MN = e = \sqrt{(x_M - x_N)^2 + (y_M - y_N)^2};$$

$$LN = f = \sqrt{(x_L - x_N)^2 + (y_L - y_N)^2},$$

где (x_L, y_L) , (x_M, y_M) , (x_N, y_N) — координаты точек L , M и N данного треугольника LMN .

3. Длина перпендикуляра (высота) $PM = g$ треугольника LMN , опущенного из вершины M на сторону LN определяется по формуле [4]

$$PM = g = 2 \frac{\sqrt{p(p-d)(p-e)(p-f)}}{f};$$

$$p = \frac{d+e+f}{2}.$$

4. Значения углов α , σ , τ вычисляются по формулам

$$\alpha = 180^\circ - (\sigma + \tau);$$

$$\sigma = \arccos \frac{g}{d};$$

$$\tau = \arccos \frac{g}{e},$$

где α — угол между линиями NM и MM_1 , принятый за угол кривизны контура в точке M (значение этого угла дискретной кривой пропорционально значению кривизны в данной точке непрерывной кривой).

Аналогичным образом происходит вычисление значения кривизны α в точке N рассматриваемой дискретной кривой по предложенному алгоритму путем построения треугольника MNO и т. д.

В процессе обхода контура вычисляются значения кривизны всех его точек и формирование α -функций объекта.

Формирование α -функций объекта (этап 4, б).

Контуров объектов представляют собой замкнутые дискретные кривые и во всех их точках могут быть определены значения кривизны. Обычно точки контура нумеруют и значения их кривизны заносят в таблицу (табл. 1).

Последовательность значений кривизны точек контура бинарного изображения объекта, расположенная в каждом столбце табл. 1, также называется α -функцией объекта.

Таблица 2

	Эталон 1	Эталон 2	...	Эталон N
Неизвестный объект 1	Z_{11}	Z_{12}	...	Z_{1N}
Неизвестный объект 2	Z_{21}	Z_{22}	...	Z_{2N}
...
Неизвестный объект P	Z_{P1}	Z_{P2}	...	Z_{PN}

Идентификация объектов (этап 5). В процессе обучения в память СТЗ заносятся α -функции эталонных объектов. При идентификации осуществляется сравнение α -функции экзаменационного (неизвестного) объекта со всеми α -функциями эталонов на основе следующей формулы:

$$Z_{kl} = \sum_{i=1}^M |\alpha_{ik} - \alpha_{il}|,$$

где Z_{kl} — сумма абсолютных величин отклонений значений кривизны точек эталонного и неизвестного объектов (аналог суммарного среднего квадратического отклонения), назовем ее суммарным отклонением; M — число точек контура объекта, т. е. число значений кривизны объекта; i — номера точек эталонного и экзаменационного объектов, разность значений кривизны которых определяется в текущем шаге вычисления ($i = 1 \dots M$); k — номер эталонного объекта ($k = 1 \dots N$); l — номер экзаменационного объекта ($l = 1 \dots P$).

Рассчитанные по каждой паре kl значения Z_{kl} заносятся в табл. 2. Значения элементов каждого столбца табл. 2 являются результатом сравнения неизвестных объектов с одним из эталонов.

Идентификация неизвестного объекта l при заданном множестве эталонов и экзаменационных объектов осуществляется на основе минимального значения суммарного отклонения Z_{kl} по табл. 2, т. е. определяется $\min_k Z_{kl} = \min_k \{ \min_l Z_{kl} \}$, где $\min_k Z_{kl}$ означает выбор минимального значения Z_{kl} среди всех Z_{kl} при $k = 1 \dots N$ и постоянном l ; $\min_l Z_{kl}$ означает выбор минимального значения Z_{kl} среди всех Z_{kl} при $l = 1 \dots P$ и постоянном k .

N п/п точек контура	Номера объектов														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	44,99	44,99	44,99	44,99	44,99	134,9	44,99	44,99	44,99	44,99	89,99	44,99	44,99	44,99	134
2	44,99	0	44,99	0	0	89,99	44,99	44,99	44,99	0	0	0	0	0	0
3	89,99	44,99	0	0	0	0	44,99	44,99	44,99	0	0	0	0	0	0
4	44,99	44,99	44,99	0	44,99	44,99	44,99	0	89,99	0	0	0	0	0	0
...
1685	—	—	—	—	—	—	—	—	—	—	—	—	—	—	89,99
1686	—	—	—	—	—	—	—	—	—	—	—	—	—	—	49,99

Таблица 4

№ эталона	Объект 4	Объект 7
Эталон 1	10114,524	12148,714
Эталон 2	7923,299	14173,181
Эталон 3	12716,599	11980,682
Эталон 4	0,000225	11239,270
Эталон 5	10062,229	13982,599
Эталон 6	11595,449	14294,039
Эталон 7	9031,074	0,000220
Эталон 8	10335,970	12834,914
Эталон 9	10123,769	14140,004
Эталон 10	9552,221	14466,495
Эталон 11	9788,487	13845,962
Эталон 12	8402,083	12147,758
Эталон 13	6093,083	12148,714
Эталон 14	10123,764	12149,192
Эталон 15	12147,758	14195,919

Уменьшение временных затрат на обработку и анализ изображений объектов

Известно, что полутоновые изображения реальных объектов содержат различные шумы разной интенсивности. Обработка и анализ этих изображений в СТЗ обычно осуществляются быстрыми, с целью сокращения затрат времени на идентификацию, детерминированными алгоритмами. Поэтому не удается максимально устранить шумы, обеспечить оптимальность значения порога сегментации-бинаризации. В связи с этим безразрывные одноточечные внешние контуры бинарных изображений объектов (формируемые на основе четырех связанных точек [2]) получают достаточно изрезанными. В целях сглаживания таких контуров и уменьшения временных затрат на вычисление значений кривизны, а также на процесс идентификации, можно использовать процедуру прореживания точек контура. Она заключается в вычислении значений кривизны не всех, а каждой второй, третьей, четвертой и так далее точек контура (коэффициенты прореживания при этом 2, 3, 4 и т. д.).

Такая процедура (с помощью данных табл. 1 выполняется очень просто) позволяет уменьшить влияние изрезанности контуров, а также сократить длины α -функций (число входящих в α -функцию

Экспериментальное исследование

Возможности разработанного алгоритма идентификации реальных объектов по их полутоновым изображениям с использованием единственного информационного признака (кривизны точки контура бинарного изображения) объекта исследовались на обучающей (эталон) и экзаменационной (неизвестные объекты) выборке, состоящей из 15 реальных изображений объектов (рис. 2). Максимальный размер объектов 512×512 и 1024×1024 элементов.

На рис. 3 показаны результаты выполнения этапов 2 и 3 алгоритма по обработке изображений двух объектов.

Часть числовых данных (результатов) по расчету α -функций объектов рис. 2 приведена в табл. 3.

В качестве экзаменационного объекта берется любой из исходных объектов (рис. 2) и проводится идентификация. Результаты расчетов значений функции суммарного отклонения Z_{kl} для двух объектов (№ 4 и № 7) приведены в табл. 4.

Из табл. 4 видно, что $Z_{4,4}$ и $Z_{7,7}$ имеют на объектах № 4 и № 7, соответственно, минимальное значение.



Рис. 2

Объект № 10



Объект № 13



Рис. 3

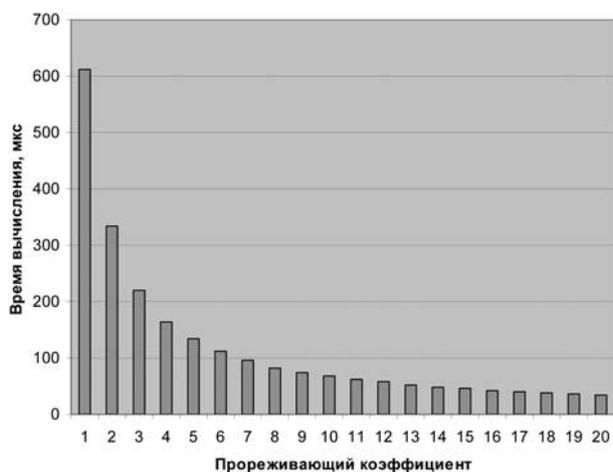


Рис. 4

значений кривизны точек) объектов, следовательно, время идентификации.

В результате исследования временных затрат на выполнение алгоритма идентификации при использовании укороченных α -функций (с меньшим числом значений кривизны точек) было установлено, что с уменьшением длины α -функции время идентификации сокращается.

Результаты исследования по объекту № 12 при различных коэффициентах прореживания представлены на рис. 4. Аналогичные данные дают эксперименты на других объектах.

Следует отметить, что при значении прореживающего коэффициента больше 5 возникают ошибки при идентификации некоторых объектов, а когда значение коэффициента прореживания больше 10, появляются ошибки при идентификации уже многих объектов.

Заключение

1. На основе данного алгоритма создано программное приложение, которое выполняет следующие функции: сглаживание шумов полутоновых изображений реальных объектов; сегментация полутоновых изображений в целях получения бинарных; определение центра тяжести, длины, угла наклона объекта (местоположение, ориентация); приведение бинарного изображения объекта к исходному положению; выделение однотоочечного безразрывного контура бинарного изображения объекта; вычисление значения кривизны точек контура и формирование на ее основании α -функции объекта; определение типа объекта путем сравнения α -функций объектов; оценка временных затрат на идентификацию объектов при различных значениях прореживающего коэффициента.

2. Показана возможность создания алгоритма идентификации плоских реальных объектов на основе единственного признака — кривизны точки бинарного контура.

3. Основными достоинствами данного алгоритма являются: простота реализации; инвариантность к перемещению и повороту объекта на дискретной плоскости; возможность изменения времени идентификации объектов путем использования различных значений прореживающего коэффициента; возможность идентификации объектов в реальном масштабе времени.

4. К недостаткам можно отнести неинвариантность к изменению масштаба на дискретной плоскости.

5. Совершенствование алгоритма проводится в следующих направлениях: возможность идентификации алгоритмом наложенных двух, трех объектов как однотипных, так и разнотипных; обеспечение инвариантности алгоритма к изменению масштаба (в определенных пределах); оценка вероятности ошибки идентификации при больших выборках; выбор и обоснование оптимального коэффициента прореживания, позволяющего: а) минимизировать временные затраты при сохранении требуемого качества идентификации объектов; б) уменьшить (в лучшем случае исключить) влияние изрезанности контура на надежность идентификации объектов.

Список литературы

1. Прэтт У. Цифровая обработка изображений: пер с англ. — М.: Мир, 1982. 788 с.
2. Садыков С. С., Стулов Н. Н. Методы и алгоритмы выделения признаков объектов в системах технического зрения. М.: Горячая линия — Телеком, 2005. 204 с.
3. Садыков С. С., Стародубов Д. Н. Алгоритмы определения длины и ширины дискретных площадных объектов // Автоматизация и современные технологии. 2007, № 10.
4. Выгодский М. Я. Справочник по элементарной математике. М.: Недра, 1975. 416 с.

УДК 004.93

В. А. Бобков, д-р. техн. наук, зав. лаб.,
Ю. И. Роншин, вед. инж.-программист
Институт автоматизи-
ки и процессов управления ДВО РАН,
e-mail: bobkov@iacp.dvo.ru

Реализация алгоритма построения карт глубин на графических процессорах¹

Предложен алгоритм построения карт глубин по множественной последовательности изображений и его реализация на графических процессорах (GPU) на языке CUDA. Алгоритм основан на применении space-sweep стратегии и адаптивной методики сопоставления окон. Получены оценки сравнительной производительности двух модификаций алгоритма на CPU и GPU.

Ключевые слова: 3D-реконструкция, карта глубин, сопоставление изображений, CUDA

Введение

Проблема построения 3D-моделей объектов по последовательности изображений (видов) по-прежнему остается актуальной сегодня в компьютерном зрении. Существуют различные подходы к ее решению. Некоторый обзор соответствующих методов/алгоритмов с их классификацией представлен, например, в работе [1]. Одним из апробированных является подход, основанный на вычислении карт глубин [2, 3]. Карты глубин используются для получения триангуляционной поверхности. Методы расчета карт глубин основываются на алгоритмах локального сопоставления — оценка сходства точек строится на сравнении окон вокруг точки на опорном изображении и точек на других изображениях. Основной трудностью при сопоставлении окон на изображениях разных видов является наличие в сцене полузакрытых областей (*semiocluded regions*) (пиксели видимы на некоторых, но не на всех изображениях). Как следствие, ситуации с невидимыми областями могут порождать ошибочные сопоставления. Для преодоления указанной трудности в ряде работ были предложены различные механизмы (см. например, [4]), суть которых сводится к использованию сдвигаемых окон,

отбора предположительно более достоверных кадров и адаптивного размера окна сравнения.

Однако применение таких механизмов приводит к существенному увеличению объема вычислений. Известно, что производительность существующих алгоритмов 3D-реконструкции на CPU, в том числе основанных на построении карт глубин, пока является недостаточной для приложений, связанных с режимом реального времени, например, таких приложений, как навигация роботов с помощью компьютерного зрения. Поэтому появилась потребность в ускорении обработки за счет применения параллельных вычислений. В последнее время наметилась тенденция к использованию для этой цели эффективной SIMD-архитектуры современных графических процессоров (*Graphics Processing Unit*). Ранее для программирования приложений использовали язык CG и шейдерные языки с существенным учетом аппаратной специфики GPU. Позже стали применять предложенную NVIDIA более эффективную технологию CUDA, а также аналоги OpenCL, OpenMP. Применительно к задаче 3D-реконструкции по изображениям были осуществлены реализации на GPU в ряде работ для разных алгоритмов и с разной степенью эффективности [5—9]. Во всех них отмечается необходимость дальнейшего повышения эффективности параллельной обработки в алгоритмах на основе GPU.

В данной статье представлен оригинальный алгоритм построения карт глубин и его реализация на GPU средствами CUDA. Были выполнены реализации двух вариантов алгоритма как на CPU, так и на CUDA, с оценкой их сравнительной производительности и достигаемого ускорения на GPU. Предложенные варианты алгоритма обеспечивают возможность выбора требуемого соотношения качество/скорость для конкретного приложения.

Последующий текст организован следующим образом. В разделе 1 приведены основные решения предлагаемого алгоритма, в разделе 2 описана параллельная реализация алгоритма на CUDA для GPU, в разделе 3 представлены результаты вычислительных экспериментов и выводы.

1. Алгоритм построения карт глубин

Весь процесс построения триангуляционной модели сцены по множественной последовательности видов выполняется согласно применяемому подходу в три этапа:

- построение карты глубин для отдельного вида по соседним видам;

¹ Работа выполнена при поддержке РФФИ (проект № 11-07-00088-а).

- построение триангуляционной модели для отдельного вида с использованием карты глубин;
- построение единой триангуляционной модели сцены посредством объединения моделей всех видов последовательности.

Предлагаемый в данной статье алгоритм относится к первому этапу. Карты глубин (расстояния до видимых точек в узлах сетки) строятся для отдельных видов, которые рассматриваются в качестве опорных. При построении карты глубин опорного вида используются несколько соседних видов, что, предположительно, повышает достоверность результата. Эта задача решается с помощью предлагаемого авторами алгоритма, который сконструирован с учетом известного опыта, включая подход *space-sweep* [10], и некоторых авторских алгоритмических решений. Алгоритм может применяться в двух режимах — для определения глубины одного пикселя и для построения плотной карты глубин в узлах заданной сетки. При реализации второго режима используется пространственная когерентность для сокращения объема вычислений. Предполагается что смещения между соседними кадрами незначительные.

Основные решения алгоритма определения глубин

Применение подхода *space-sweep*. Поиск глубин видимых точек объектов осуществляется методом *space-sweep*. Для каждого пикселя изображения опорного кадра строится луч из центра проекций в сцену (рис. 1). Выполняется перебор глубин на луче с заданным шагом. В каждой из точек на луче (глубины) задается ориентация плоскости (нормаль), которая определяет нормаль поверхности объекта в данной 3D-точке. Выполняется проецирование охватывающего окна на опорном виде на другие виды. Проецирование определяется задваемым преобразованием плоской томографии.

Особенностью предлагаемой алгоритмической схемы является реализация проверки множества

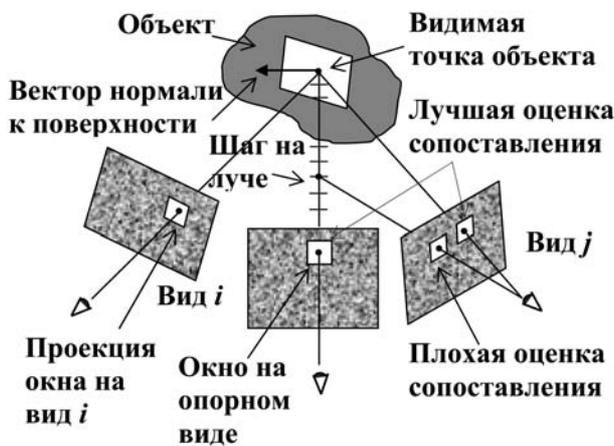


Рис. 1. Поиск глубины для пикселя опорного изображения. Лучшая оценка сопоставления соответствует правильной глубине

гипотез-нормалей к поверхности объекта, что позволяет выбрать наиболее оптимальную (близкую к истинному наклону поверхности). Сопоставление точек на разных видах осуществляется с помощью вычисляемой оценки сходства (текстурной близости) по окну. Оценкой служит сумма квадратов разностей интенсивностей цветов:

$$D = \Sigma(rp_{1i} - rp_{2i})^2 + (gp_{1i} - gp_{2i})^2 + (bp_{1i} - bp_{2i})^2,$$

где $rp_{1i}, gp_{1i}, bp_{1i}$ — компоненты цвета i -й точки 1-го окна; а $rp_{2i}, gp_{2i}, bp_{2i}$ — компоненты цвета соответствующей точки 2-го окна. То есть наилучшей оценке соответствует минимальное значение оценки, а наихудшей — ее максимальное значение. Наилучшая по всему пространству изменяемых параметров (глубина на луче, ориентация поверхности) интегральная (по всем видам) оценка определяет искомую глубину для рассматриваемого пикселя.

Поиск глубины осуществляется в два этапа: грубый с большим шагом по лучу и последующий точный поиск с маленьким шагом в окрестности найденного грубого решения.

При построении плотной карты глубин применяется вычислительная схема, учитывающая пространственную когерентность. Она основывается на комбинированном использовании двух режимов вычисления глубины. В режиме 1 выполняется полный проход по лучу, т. е. по всей глубине сцены, в режиме 2 (оптимизированном) выполняется короткий проход, ограниченный окрестностью глубины, найденной для соседней точки (пикселя) изображения.

Схема вычислений строится следующим образом. Для начального пикселя строки изображения выполняется обработка в первом режиме, для последующих пикселей — в оптимизированном режиме. Если же полученная в режиме 2 оценка не удовлетворяет заданному порогу (настроечный параметр), то выполняется переключение в режим 1 с повторным поиском уже на всем диапазоне глубины. По логике переключение должно происходить тогда, когда точка на другом виде оказывается в зоне невидимости или когда истинная глубина оказывается вне короткого прохода, т. е. эта схема учитывает различие в обработке ситуаций непрерывной поверхности и разрыва по глубине.

Томография. Знание наклона поверхности объекта в рассматриваемой точке позволяет построить правильное преобразование окна на других видах и, соответственно, обеспечить корректное сравнение окон. Поэтому для поиска и учета правильного наклона поверхности для каждого луча осуществляется полный перебор нормалей по полусфере (с заданной дискретностью). Для каждого значения нормали строится преобразование плоской томографии *Hom*. Томография *Hom* — это преобразование, отображающее точки на заданном (1-м) виде в точки на другом (2-м) виде. Оно пред-

ставляется как матрица размерностью 3×3 , преобразующая 2P-проективные точки: $p_2 = p_1 \cdot Hom$.

Матрица *Hom* вычисляется как произведение двух проективных матриц:

$$Hom = P_{plane} \cdot P_{12},$$

где P_{plane} — матрица размерностью 4×3 , преобразующая 2P-проективные точки в 3P-проективные точки; P_{12} — матрица размерностью 3×4 , преобразующая 3P-проективные точки в 2P-проективные точки — проекционная матрица для 3D-точек, заданных в системе координат 1-го вида, на 2-й вид:

$$P_{plane} = \begin{pmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \cdot f \\ n.x & n.y & n.z \cdot f \end{pmatrix},$$

где d — расстояние от начала системы координат 1-го вида до заданной плоскости; f — фокус для 1-го вида; n — вектор нормали плоскости.

Методика сравнения окон. Как уже было отмечено выше, основной трудностью при сопоставлении окон на изображениях разных видов является наличие в сцене *semiocluded regions*, которые порождаются разрывностью глубин. Применение стандартного центрального окна вокруг точки для сравнения двух видов может в этом случае не дать правильного результата (рис. 2). Для преодоления указанной трудности в применяемой методике оконного сравнения реализуются следующие механизмы:

а) *сдвигаемые окна на обоих сравниваемых видах.* Последовательно выполняется несколько вариантов смещений окна (рис. 3) с вычислением соответствующих сравнительных оценок. Предпо-

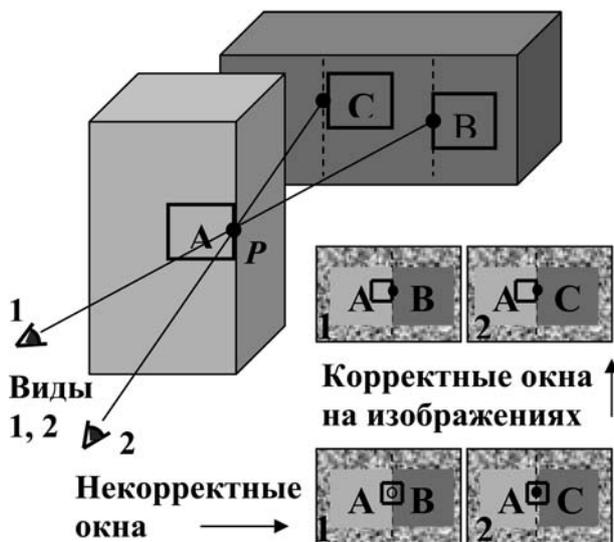


Рис. 2. Пример полузакрытых областей в 3D-сцене. Область *C* невидима на виде 1 и видима на виде 2. Варианты корректного и некорректного сравнения окон на видах 1, 2 для точки *P*

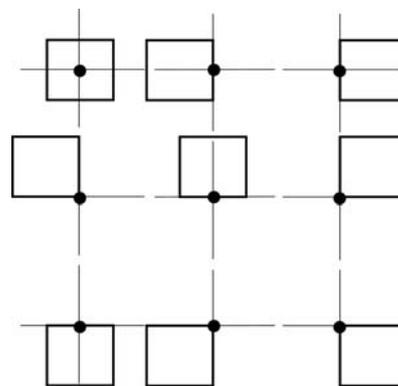


Рис. 3. Схема сдвига прямоугольных окон

жительно хотя бы для одного из вариантов окно полностью окажется в видимой области и обеспечит корректное сравнение окон на обоих видах. Именно для этого варианта оценка должна быть наилучшей;

б) *темпоральный отбор кадров.* Суть этого механизма заключается в том, что при формировании интегральной (по всем видам) сравнительной оценки применяется не усреднение локальных оценок по всем парам (опорный + другой вид), а отбор наилучших по локальным оценкам кадров. В основе такого отбора лежит предположение, что наилучшим кадром соответствует ситуация видимости точки. Как правило, в случае "разрывности" поверхности точка видима только на предшествующей или только на последующей последовательности кадров (по отношению к кадру, где изменяется ее статус видимости). Наиболее простой критерий отбора наилучших кадров — выбор 50 % наилучших оценок. Из них удаляются оценки больше заданного порога P . Если оценок не осталось, возвращается максимальное значение оценки. В противном случае возвращается средняя оценка по оставшимся. Возможны другие варианты отбора наилучших пар с более точным учетом хороших и плохих оценок;

в) *адаптивное изменение размера окна.* Этот механизм позволяет адаптироваться к качеству текстуры поверхности. Для слаботекстурированной области требуется больший размер окна для получения надежной оценки сходства. Другое преимущество использования изменяемого окна — повышение производительности алгоритма. Реализуется двухпроходная схема тестирования глубин по лучу: вначале проход с грубым шагом и маленьким окном, затем проход с маленьким шагом и большим окном в окрестности найденного грубого решения. Такая схема, как показали эксперименты, дает существенную экономию вычислений сравнительных оценок.

"Линейные" окна. В алгоритме предложен и реализован метод "линейных" окон вместо традиционно используемых прямоугольных окон, что позволило сократить объем вычислений без сущест-

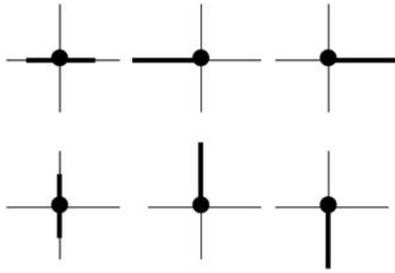


Рис. 4. Схема сдвига линейных окон

венного снижения качества результата. В качестве "линейного" окна используется последовательность точек вдоль отрезка. В предлагаемой схеме применяются два линейных окна — отрезок по вертикали и отрезок по горизонтали изображения (такой выбор гарантирует корректное сравнение окон, по крайней мере, для линейной границы видимости в окрестности точки). Применительно к линейному окну также используются все перечисленные выше механизмы, в частности, при сдвиге — каждый из отрезков может занимать одно из трех положений по отношению к анализируемой точке (рис. 4). Число точек, по которым подсчитывается оценка, в случае "линейного" окна существенно меньше, чем при квадратном окне. Число подокон также меньше — 6 вместо 9. Еще одно вычислительное преимущество "линейного" окна — число возможных наклонов поверхности (нормалей), от которых зависит положение окна-отрезка на втором виде, также существенно меньше, чем в случае квадратного окна. На рис. 5 видно, что множество нормалей, которые необходимо учитывать при полном переборе, определяется уже полуокружностью, а не полусферой. Геометрия вычисления нормалей на полуокружности применительно к горизонтальному линейному окну показана на рис. 5. Вначале берется стандартный набор нормалей на полуокружности, лежащей в плоскости XZ системы координат i -го вида.



Рис. 5. Вычисление набора нормалей для горизонтального линейного окна

Затем выполняется преобразование поворота на угол $\alpha = \arctg(y/f)$, где y — координата y точек линии окна, а f — фокус 1-го вида.

Для получения сравнительных оценок были реализованы две версии алгоритма — с прямоугольным и линейным окном как на CPU, так и на GPU.

2. Реализация алгоритма построения карты глубин на GPU по технологии CUDA

Многопоточный мультимикропроцессорный чип nVidia, специализированный для вычислений с плавающей точкой одинарной точности с помощью технологии CUDA, позволяет реализовывать алгоритмы на языке программирования C. Известно, что CUDA обладает рядом специфических для графических ускорителей на базе nVidia ограничений, что требует не только адаптации переносимых программ, но и серьезной переработки существующих алгоритмов для достижения наилучшего результата по скорости вычислений. В нашей задаче за один запуск CUDA необходимо обработать $N_p \cdot N_n \cdot 9$ нитей (процессов) для алгоритма с прямоугольным окном или $N_p \cdot N_n$ нитей для алгоритма с линейным окном (N_p — число точек на луче, N_n — число нормалей, 9 — число подокон в первой модификации алгоритма 1). Трехмерные массивы нитей разбиваются на блоки. Исходя из требования, чтобы размер блока был кратен 32 (размер "warp") были выбраны следующие блоки: $4 \cdot 8 \cdot 9$ для модификации 1 и $4 \cdot 8$ для модификации 2.

Выходом работы на CUDA является трехмерный массив оценок. Определение минимума этих оценок проводится на CPU. Минимуму соответствуют оптимальная 3D-точка и оптимальная нормаль для этой точки. Ниже описан алгоритм с линейным окном для GPU (т. е. последовательность обработки данных в рамках одного процесса-нити).

1. Из идентификатора нити определяются индексы нормали, точки и направления.
2. Выбирается нормаль из набора нормалей, расположенных равномерно по полуокружности, определенной в зависимости от пикселя и направления.
3. Вычисляется точка в зависимости от индекса точки.
4. Выполняется цикл по парам видов (между заданным видом и другими).
 - 4.1. Для пары вычисляется томография Hom в зависимости от нормали и точки.
 - 4.2. Выполняется цикл по подокнам расширенного окна (отрезком) сравнения.
 - 4.2.1. Вычисляется оценка для подокна в зависимости от Hom .
 - 4.2.2. Определяется минимум этих оценок — оценка для пары.
 - 4.3. Оценка для пары заносится в массив оценок для пар.

Сравнение алгоритмов на CPU

Параметры эффективности	Алгоритм 1	Алгоритм 2
Время обработки одной точки, мс	1847,4	77,1
Сопоставленные пиксели, %	85,8	86,6
Ошибки в среднем, м	0,0196	0,0227

Таблица 2

Сравнение алгоритмов на GPU

Параметры эффективности	Алгоритм 1	Алгоритм 2
Время обработки одной точки, мс	109,9	1,45
Сопоставленные пиксели, %	85,6	80,8
Ошибки в среднем, м	0,0194	0,0271
Ускорение (по отношению к CPU)	17	53

5. Из массива оценок для пар вычисляется оценка для нормали, точки и направления.

6. Эта оценка заносится в выходной трехмерный массив оценок.

3. Вычислительные эксперименты

Вычислительные эксперименты проводились на PC с CPU Intel® Core™2 Duo с видеокартой GeForce GTS 250 на модельной сцене, поскольку на модельной сцене возможна количественная оценка геометрической точности реконструкции. Использовали пять изображений (видов) с разрешением 2048×1536 пикселей. Проводили два типа сравнений. Во-первых, сравнивались между собой модификация алгоритма с прямоугольным окном (алгоритм 1) и модификация с линейным окном (алгоритм 2) как на CPU, так и на GPU, по скорости и качеству. Под качеством понималось число определенных точек и точность в среднем по всем точкам. Карта глубин вычислялась алгоритмом 1 и алгоритмом 2 для опорного вида (рис. 6).

Во-вторых, оценивалось ускорение, достигаемое за счет использования GPU. Использовали следующие значения настроечных параметров: грубый шаг по лучу 0,5 м (число шагов 220), точный шаг 0,01 м (число шагов 100); размер прямоугольного окна 9×9 пикселей, размер линейного окна —

15 пикселей; число нормалей (на полусфере) в наборе для алгоритма 1—190, для алгоритма 2 (на полуокружности) — 36. Все расстояния приведены в условных единицах, поскольку сцена модельная. Результаты представлены в табл. 1, 2. Из таблиц видно, что алгоритм с линейным окном превосходит алгоритм с прямоугольным окном по скорости в 24 раза на CPU и в 76 раз на GPU при практически равноценном качестве — 80—85 % определяемых точек со средней точностью вычисленных глубин 2—3 см. По отношению к глубине сцены (200 м) полученная точность составляет меньше 0,1 %. Такой же результат по точности был получен и на сцене с криволинейной поверхностью. Эти результаты подтверждают правомерность применения линейного окна. Для алгоритма 1 получено ускорение за счет GPU в 17 раз, для алгоритма 2 в 53 раза.

Также был проведен эксперимент для проверки возможности ускорения алгоритма за счет использования двух размеров окна — маленького окна для поиска грубого решения и большого окна для поиска точного решения в окрестности найденного грубого решения (по аналогии с описанной выше схемой двухэтапного поиска решения с большим и маленьким шагом по лучу). Были получены графики зависимости оценок соответствия от глубины (по лучу) для двух размеров окна. Сравнение графиков показало их хорошую корреляцию, что позволило использовать этот факт для реализации двухэтапной процедуры поиска глубины. Эксперименты показали, что применение такой процедуры дает дополнительный прирост скорости обработки на 30 %.

Заключение

Представлены две модификации предложенного алгоритма определения карты глубин — с использованием традиционного прямоугольного и линейного окна сопоставления — и их реализации на CPU и GPU со сравнительными оценками

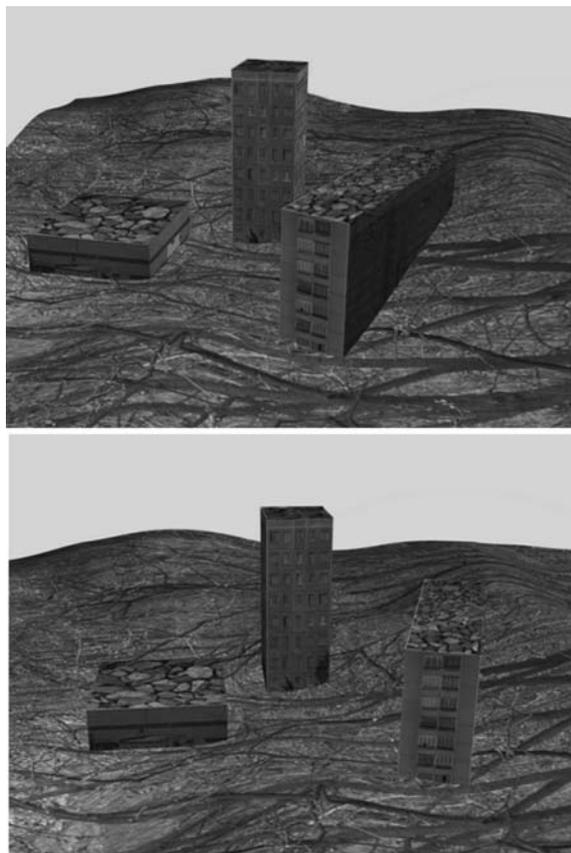


Рис. 6. Два вида модельной сцены

производительности. Алгоритм основывается на применении:

- модификации стратегии space-sweep с грубым и точным поиском глубин 3D-точек;
- адаптивного метода сдвигаемых окон и отбора лучших соседних изображений;
- гомографии с плотным набором гипотез-нормалей и оптимизации вычислений за счет учета пространственной когерентности.

Эксперименты показали существенное превосходство в скорости модификации алгоритма с линейным окном. Реализация алгоритмов на GPU, в свою очередь, дала значительное ускорение их работы. Таким образом, ускорение предложенного алгоритма вычисления карт глубин достигается как за счет алгоритмической модификации, так и за счет многопроцессорной обработки на GPU средствами CUDA. В дальнейшем предполагаются улучшение характеристик алгоритма за счет привлечения информации о линейных границах на изображениях и устранения избыточных вычислений, тестирование на реальных сценах, а также реализация алгоритмов фильтрации и построения единой триангуляционной модели по множеству локальных моделей для отдельных опорных видов. Также планируется применение алгоритма в приложениях, связанных прежде всего с реконструкцией сцен городской обстановки и навигацией автономных подводных аппаратов по видеоинформации.

1. Seitz S., Curless B., Diebel J., Scharstein P., & Szeliski R. A comparison and evaluation of multi-view stereo reconstruction algorithms // Proceedings of the IEEE international conference on computer vision and pattern recognition. 2006. P. 519–526.
2. Szeliski R. A multi-view approach to motion and stereo // CVPR. 1999. Vol. 1. P. 157–163.
3. Mordohai P., Frahm J.-M., Akbarzadeh A., Clipp B., Engels C., Gallup D., Merrell P., Salmi C., Sinha S., Talton B., Wang L., Yang Q., Stewenius H., Towles H., Welch G., Yang R., Pollefeys M., and Nistler D. Real-time video-based reconstruction of urban environments // In 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-ARCH), Zurich, Switzerland, July 2007. 3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures, (ETH Zurich, Switzerland), July 2007.
4. Kang S., Szeliski R. and Chai J. Handling occlusion in dense multi-view stereo // Int. Conf. on Computer Vision and Pattern Recognition. 2001. P. 103–110.
5. Yang R., Pollefeys M. Multi-resolution Real-Time Stereo on Commodity Graphics Hardware // CVPR. 2003. P. 211–218.
6. Mairal J., Keriven R. and Chariot A. Fast and efficient dense variational stereo on GPU // Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06). 2006. P. 797–104.
7. Grauer-Gray S., Kambhamettu C., and Palaniappan K. GPU Implementation of Belief Propagation Using CUDA for Cloud Tracking and Reconstruction // 5th IAPR Workshop on Pattern Recognition in Remote Sensing. 2008.
8. Grauer-Gray S., Kambhamettu C. Hierarchical Belief Propagation To Reduce Search Space Using CUDA for Stereo and Motion Estimation // IEEE Workshop on Applications of Computer Vision (WACV). 2009.
9. Gibson J., and Marques O. Stereo Depth with a Unified Architecture GPU // Computer Vision and Pattern Recognition Workshops (CVPRW) 2008. P. 1–6.
10. Collins R. T. A space-sweep approach to true multi-image matching // Int. Conf. on Computer Vision and Pattern Recognition. 1996. P. 358–363.

УДК 004.93

В. К. Гулаков, канд. техн. наук, проф.,
А. О. Трубаков, ст. преп.,
Брянский государственный
технический университет,
e-mail: trubakovao@gmail.com

Введение

Эффективный алгоритм преобразования полноцветного изображения к палитре для систем поиска по содержанию

*Проанализированы алгоритмы получения палитры в разных цветовых пространствах, таких как RGB, HSV, $L * a * b$, $L * C * h$. Выделены критерии, выдвигаемые к палитре в системах поиска изображений по содержанию. Предложен новый способ генерации палитры и алгоритм преобразования полноцветного изображения в цвета палитры с помощью карты трансформации.*

Ключевые слова: обработка изображений, цветовая палитра, системы поиска изображений по содержанию, CBIR

Система поиска изображений по содержанию (*Content-Based Image Retrieval, CBIR*) — это информационно-поисковая система, выполняющая индексирование и поиск изображений на основе его внутренних характеристик. Системы подобного класса очень востребованы в настоящее время ввиду бурного развития мультимедийного контента в сети Internet. Очень часто в распоряжении поисковой системы находится только само изображение без какой-либо дополнительной информации. При этом в задачи системы входит обработка изображения и организация поиска на основе таких характеристик, как цвет, текстура, форма и контуры однородных объектов. Из всех перечисленных критериев для большого класса изображений основным является цвет [1]. Основываясь только на цветовом подобию, можно делать достаточно точные обработки.

Одной из самых распространенных практик оценки цветового подобия является сравнение гистограмм изображений, которые получаются после

перевода полноцветного изображения в индексированное, с применением некоторой палитры. При этом выбор палитры может значительно повлиять на процесс сравнения и результат работы системы. В данной статье предлагается алгоритм перевода изображения к цветам палитры на основе адаптивной карты преобразования, полученной из цветового пространства *HSV*. Такой алгоритм достаточно прост в реализации, является вычислительно эффективным и при этом показывает достаточно хорошие результаты по качеству перевода, что подтверждается проведенными экспериментами.

В первом разделе статьи дается общее понятие палитры, ее формальное представление. Также выдвигаются основные требования к палитре, которую можно использовать в системах поиска изображений по содержанию. Во втором разделе приведено описание цветовых пространств, выделены их особенности и стандартные способы получения палитр по ним. В третьем разделе описан предлагаемый алгоритм получения адаптивной палитры. Результаты тестирования палитры приведены в четвертом разделе статьи.

1. Цветовая палитра и требования к ней

Цветовая палитра — фиксированный набор цветов и оттенков, имеющий цифровую реализацию в том или ином виде. Все палитры компьютерной графики можно разделить на равномерные и нелинейные. Равномерная палитра получается путем квантования осей некоторого цветового пространства на равные отрезки. Для обозначения таких палитр далее будем использовать запись вида $ColSpace[L_1, L_2, L_3]$, где $ColSpace$ — цветовое пространство, L_1, L_2, L_3 — число уровней квантования для каждой из осей пространства. Например, 64-цветная равномерная палитра пространства *RGB* будет обозначаться $RGB [4,4,4]$. При этом формально подобные палитры определяют следующим образом:

$$ColSpace[L_1, L_2, L_3] = \{col_{i,j,k}\},$$

$$i = \overline{1, L_1}, j = \overline{1, L_2}, k = \overline{1, L_3}.$$

Равномерные палитры легки в понимании и не требуют сложных математических расчетов. Если изображение уже переведено в соответствующее цветовое пространство, то для преобразования к палитре остается только провести квантование цвета каждого пикселя. Вычислительная эффективность таких преобразований намного лучше других способов перевода. Однако равномерные палитры не всегда дают приемлемые результаты оценки визуального подобия с точки зрения человека. Это связано с тем, что описанные преоб-

разования не учитывают особенностей человеческого восприятия.

Нелинейные палитры нельзя получить с помощью простых операций квантования или подобных им. Чаще всего для получения подобных палитр вводятся сложные зависимости одних компонентов от других, "сжимаются" или "расширяются" отдельные части цветового пространства. Большинство подобных схем лучше учитывают психологию восприятия цвета, однако они сложно реализуемы и требуют больших вычислительных затрат. Этим можно объяснить их малую распространенность в системах поиска изображений по содержанию.

В данной статье предлагается средний подход — адаптивная неравномерная палитра, получаемая на основе простых операций. Под адаптивностью в данном контексте понимается не подбор палитры под конкретное изображение, а ее адаптация к особенностям восприятия человека. Также стоит заметить, что данный алгоритм построен на основе цветового пространства *HSV* и меры подобия *CIE2000*, но при этом он не требует перевода изображения из *RGB* пространства, что существенно повышает его вычислительную эффективность.

Для анализа существующих и разработанной методик введем критерии, которым должна соответствовать цветовая палитра для систем поиска изображений по содержанию.

1. При переводе изображения к цветам заданной палитры должны использоваться простые операции, т. е. алгоритм преобразования должен быть вычислительно эффективен. Это связано с тем, что информационно-поисковые системы оперируют большими объемами информации и для них крайне нежелательно наличие дополнительной нагрузки.

2. Исходными данными для преобразования являются изображения в пространстве *RGB*. Это связано с тем, что практически все современные цифровые изображения хранятся в этом формате. Если палитра работает в другом пространстве, то дополнительно стоит учитывать нагрузку на преобразование между пространствами.

3. Палитра должна быть малоцветной. Большинство систем поиска изображений по содержанию не может оперировать большими палитрами или полноцветными изображениями. Их эффективность нелинейно зависит от размера палитры.

4. Палитра должна учитывать психологию восприятия цвета человеком и быть ориентирована на человеческое зрение.

2. Цветовые пространства

Все широко используемые в компьютерной графике палитры основаны на принципе деления некоторого цветового пространства на части. При

этом в зависимости от того, какое цветовое пространство выбрано за основу, получаются совершенно разные варианты преобразования. Рассмотрим самые популярные цветовые пространства, которые использовались при тестировании новой палитры [2].

Линейное цветовое пространство RGB. Данное цветовое пространство является самым популярным в компьютерной графике. Именно оно используется практически во всех графических форматах. Название оно получило от трех базисных цветов: *Red* (красный, длина волны 645,16 нм); *Green* (зеленый, длина волны 526,32 нм); *Blue* (синий, длина волны 444,44 нм). Наглядно цветовое пространство *RGB* представляется как трехмерная система координат, каждая из осей которой отвечает за отдельный компонент (рис. 1). Для получения определенного цвета необходимо на каждой из осей отложить нужное количество данного компонента и на пересечении отсчетов будет находиться нужный цвет. Обычно область определения по каждой оси соответствует отрезку [0; 255], что легко реализуется в компьютерных системах.

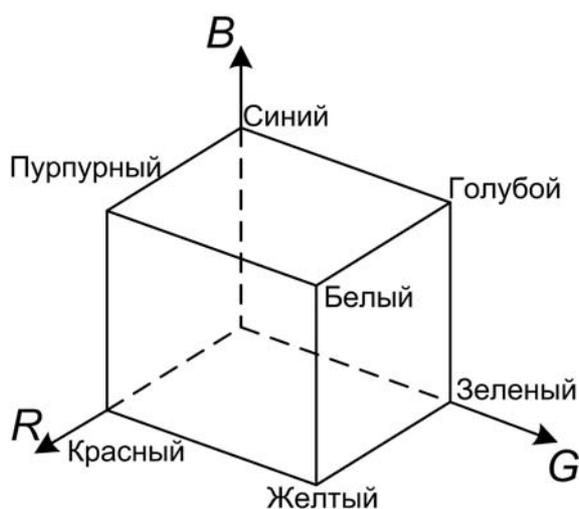


Рис. 1. Цветовое пространство RGB

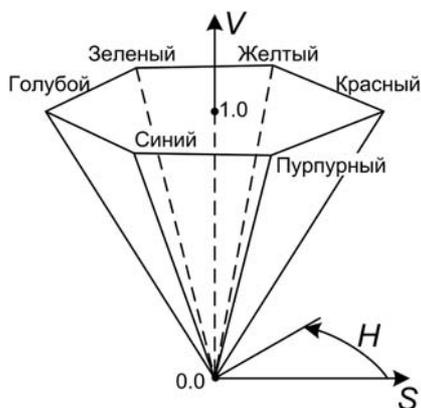


Рис. 2. Цветовое пространство HSV

Таким образом, цвет точки в пространстве *RGB* требует для хранения в памяти 24 бита (или 3 байта).

Цветовое пространство HSV. Данное пространство так же, как и рассмотренное выше, является трехкомпонентным. Однако оно более приближено к психологическому восприятию человека. Название пространства происходит от трех выбранных компонентов: *Hue* (цветовой тон); *Saturation* (насыщенность); *Brightness* (яркость). Получение определенного цвета начинается с выбора необходимого цветового тона (рис. 2). Для этого существует цветовой круг, на котором отображены все необходимые тона. Цветовой тон определяется значением угла на этом круге и принимает значение в диапазоне [0; 360].

Насыщенность (процент добавления белого) — это параметр, определяющий чистоту создаваемого оттенка. С помощью этого параметра можно получать как яркие и насыщенные тона, так и блеклые, "выцветшие". Параметр насыщенности меняется в диапазоне [0; 1], при этом 1 соответствует насыщенному цвету, 0 — выцветшему до белого.

Яркость (процент добавления черного) — компонент, субъективно определяющий освещенность или затененность цвета. Этот параметр также определяется в диапазоне [0; 1] (1 — полная яркость, 0 — минимальная яркость, черный цвет).

Процесс получения цвета в этом пространстве очень похож на интуитивное восприятие человеком. Именно поэтому данная модель хорошо подходит для систем поиска изображений по содержанию.

Цветовое пространство XYZ. В данном пространстве используются следующие три компонента: один — для определения интенсивности освещения, два других — для определения цвета (хроматичность в диапазоне красный/зеленый и желтый/синий).

Данная система построена на основе зрительных возможностей "стандартного наблюдателя", т. е. гипотетического зрителя, возможности которого были тщательно изучены и зафиксированы в ходе длительных исследований человеческого зрения, проведенных комитетом *CIE*. В настоящее время чаще применяется не само пространство *XYZ*, а полученные на основе него пространства *L*a*b* или *L*C*h*.

Цветовое пространство L*a*b. *L*a*b* также является разработкой международной комиссии по освещенности *CIE*. В нем был исправлен ряд недочетов пространства *XYZ*, связанных с восприятием человеком диапазонов синего и желтого цветов.

Любой цвет пространства *L*a*b* определяется двумя хроматическими компонентами (рис. 3) — *a*, отвечающим за изменение от зеленого до красного (*green/magenta*); *b*, отвечающим за изменение от синего до желтого (*blue/yellow*). Компонент *L* от-



Рис. 3. Цветовое пространство L^*a^*b

вечает за освещенность (*luminosity*). Пространство L^*a^*b получено с помощью нелинейного преобразования пространства XYZ . Недостатком XYZ является его неоднородность и погрешность наблюдения цветов в различных участках цветового диапазона. Эти недостатки были учтены с помощью эмпирических коэффициентов.

3. Преобразование в цвета адаптивной палитры

Равномерная палитра. Самым простым подходом к созданию палитры является получение набора цветов с помощью квантования осей цветового пространства. Например, если разбить каждую ось пространства RGB на четыре части, получим решетку, которая делит это пространство на 64 кубика (рис. 4). Каждый кубик можно положить в основу одного цвета палитры. Для этого выбирается средний цвет по всему кубику:

$$col_{i,j,k} = \left\langle SzL_1i - \frac{SzL_1}{2}, SzL_2j - \frac{SzL_2}{2}, SzL_3k - \frac{SzL_3}{2} \right\rangle,$$

где $col_{i,j,k}$ — цвет, соответствующий позиции $[i, j, k]$ в цветовом пространстве; SzL_1, SzL_2, SzL_3 — шаг квантования по соответствующей оси пространства.

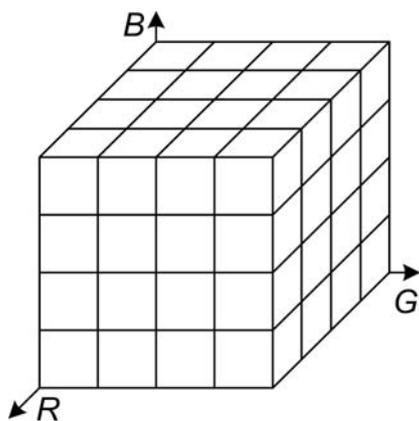


Рис. 4. Равномерное деление RGB пространства

Таким образом получаем палитру из 64 цветов $RGB[4, 4, 4]$. Аналогично можно получить 64-цветную палитру в любом цветовом пространстве.

Одним из недостатков описанного выше метода получения палитры является невозможность формирования палитры произвольного размера. Например, нельзя с помощью подобного способа получить палитру из 63 цветов. Также стоит учитывать, что большинство цветовых пространств при равномерном делении образуют палитру, которая плохо подходит для восприятия человеком.

Адаптивная палитра произвольного размера.

Для получения палитры произвольного размера с адаптацией под человеческое восприятие предлагается следующий подход. В качестве исходных данных для расчета выбирается равномерная палитра RGB пространства увеличенного размера. В частности, как показали эксперименты, для получения палитры до 150 цветов хорошо подходит палитра $RGB[8, 8, 8]$, содержащая 512 цветов.

После получения палитры из 512 цветов проводится ее разбиение на кластеры. Число кластеров, на которое необходимо разбить исходные данные, соответствует числу цветов в палитре, которое необходимо получить. Например, если необходимо получить палитру из 60 цветов, то выполняют алгоритм кластеризации на 60 множеств.

В качестве алгоритма кластеризации был выбран алгоритм k -средних (k -means), который старается минимизировать дисперсию на точках каждого кластера:

$$D = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2,$$

где k — число кластеров; S_i — полученные кластеры ($i = \overline{1, k}$); μ_i — центр масс векторов $x_j \in S_i$. После выполнения алгоритма кластеризации i -й цвет палитры определяется как средний цвет i -го кластера μ_i .

Для выполнения алгоритма кластеризации необходимо ввести функцию расстояния между любыми двумя элементами множества $RGB[8,8,8]$. В качестве такой функции предлагается использовать меру цветового подобия, предложенную международной комиссией по освещенности $CIE2000$ [3]. Конечно, для расчета этой меры необходимо временно перевести цвета палитры в пространство L^*C^*h , однако такой перевод нужен только на этапе кластеризации, и он не повлияет ни на принцип получения палитры, ни на скорость работы в дальнейшем.

Используя описанный выше алгоритм можно получить палитру любого размера. При этом палитра будет адаптирована к человеческому восприятию, так как функция цветового подобия $CIE2000$ учи-

тывает все известные на сегодняшний день особенности человеческого зрения.

Особенно стоит остановиться на выборе начальных приближений для алгоритма кластеризации. Известно, что алгоритм *k-means* чувствителен к такому выбору, и разный выбор начальных приближений может значительно повлиять на результат работы [4]. В процессе экспериментов было рассмотрено несколько подходов к получению начальных приближений. Наиболее удачным из них оказался выбор средних цветов при равномерном делении пространств *RGB* или *HSV*. Причем, как показали эксперименты, приведенные ниже, выбор средних цветов из пространства *HSV* дает немного лучшие результаты.

Еще одной особенностью предлагаемого метода является реакция на получение пустых множеств при кластеризации. В процессе работы алгоритма возможны случаи, при которых один из кластеров окажется практически пустым или даже полностью пустым. Это происходит вследствие того, что в начальном приближении палитры могут находиться цвета, которые с точки зрения человеческого восприятия очень близки. Подобные ситуации нужно отслеживать в процессе генерации палитры и кластеры с маленьким наполнением удалять. При этом на место удаленных цветов палитры необходимо добавить цвета, являющиеся средним арифметическим самых удаленных цветов в палитре текущего шага:

$$Col_{new} = \frac{Col_1 + Col_2}{2},$$

где Col_{new} — новый цвет, добавляемый вместо удаленного; Col_1 и Col_2 — цвета, выбранные по принципу максимальной дистанции на основе функции цветового подобия *CIE2000*:

$$\begin{aligned} \text{DistCIE2000}(Col_1, Col_2) &= \\ &= \max_i \{ \min_j \{ \text{DistCIE2000}(Col_i, Col_j) \} \}. \end{aligned}$$

Преобразование в цвета адаптивной палитры.

Палитра, полученная с помощью описанного выше метода, является неравномерной. Поэтому необходимо не только разработать метод генерации подобной палитры, но и выработать алгоритм перевода изображения из полноцветного пространства *RGB* к цветам палитры.

Для быстрого перехода от 24-битной картинки в пространстве *RGB* к предложенной палитре предлагается ввести понятие *карты трансформации*. Под картой трансформации понимается массив чисел, который однозначно сопоставляет цвета палитры *RGB*[8,8,8] и цвета полученной адаптивной палитры. Так как палитра *RGB*[8,8,8] состоит всего из 512 элементов, то карта трансформации будет содержать 512 чисел. При этом для ее хра-

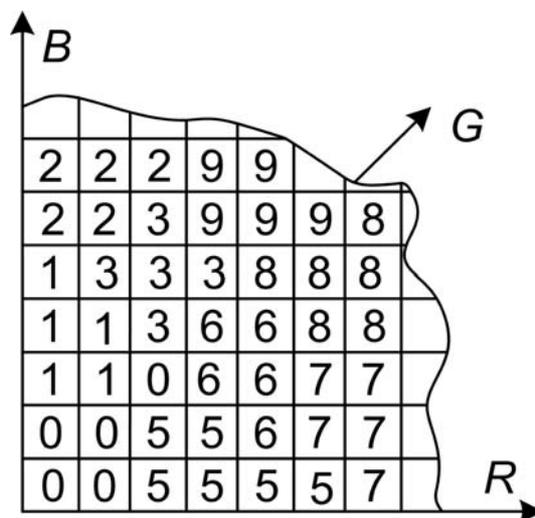


Рис. 5. Карта трансформации

нения в оперативной памяти понадобится всего 512 байт, что сравнительно немного по современным меркам.

Карту трансформации удобно представлять в виде трехмерного массива размером $8 \times 8 \times 8$ (рис. 5). Каждая ячейка этого массива соответствует одному из цветов палитры *RGB*[8,8,8]. При этом число, записанное в этой ячейке, соответствует номеру цвета из адаптивной палитры.

Для использования карты трансформации необходимо выполнить простейшие преобразования:

$$r = Col.R \gg 5;$$

$$g = Col.G \gg 5;$$

$$b = Col.B \gg 5;$$

$$Col_{pal} = \text{Map}[r][g][b],$$

где Col — цвет, который нужно привести к палитре; $Col.R$, $Col.G$, $Col.B$ — красный, зеленый и синий компоненты цвета; Col_{pal} — операция побитового сдвига; Map — карта трансформации; Col_{pal} — цвет палитры, соответствующий цвету Col .

Таким образом, разработанная палитра не только является адаптированной к восприятию человека, но и эффективной с точки зрения вычислительных ресурсов.

4. Экспериментальные результаты

Для проведения тестирования было выбрано четыре равномерные палитры *RGB*[4,4,4], *HSV*[4,4,4], $L*a*b$ [4,4,4], $L*C*h$ [4,4,4] и две адаптивные палитры, полученные по описанному выше принципу. Первая из них была получена с помощью начального приближения из *RGB* пространства (*AdaRGB*), вторая — из *HSV* пространства (*AdaHSV*). Как было указано выше, применяемый для кластеризации алгоритм *k-means* чувствителен

Таблица 1

Состав тестовой коллекции

Цветовой оттенок	Тематика	Соотношение изображений в коллекции, %
Зеленый	Природа, пейзажи леса и полей	39
Синий, голубой	Небо, море	49
Красный, фиолетовый, розовый	Крупный план цветов, фотографии экзотических рыб	42
Желтый, оранжевый	Крупный план цветов, фотографии экзотических рыб	38
Черный, серый	Фотографии гор, скал, подводных пещер	30

к начальному приближению. Поэтому полученные адаптивные палитры имеют значительные отличия друг от друга.

Все выбранные палитры имеют одинаковый размер — 64 цвета. Это связано с тем, что используемые в тестировании равномерные палитры не поддерживают произвольный размер.

Тестовые коллекции. Тестирование проводили с помощью специально подготовленного набора изображений. Тематика тестовой коллекции достаточно разнообразна. В нее были включены природные пейзажи, изображения животных, крупный план цветов и экзотических рыб, городские фотографии. Подобный выбор обусловлен тем, что выбранные изображения охватывают разные цветовые оттенки, включающие практически всю палитру (особенно фотографии цветов и подводного мира). Это позволяет всесторонне исследовать выбранные палитры, оценить их сильные и слабые стороны.

Число изображений разных оттенков и их тематика представлена в табл. 1. В последней колонке указано процентное соотношение изображений, в которых данный оттенок занимает значительное место (не менее 50 % фона изображения или является преобладающим в центральном объекте). Вследствие того, что в коллекции очень много морских и природных пейзажей, оттенки синего выделяются из всей палитры. Однако все остальные цвета были сбалансированы приблизительно поровну.

Результаты проведения тестов. Для тестирования использовали специально разработанное приложение. Пользователю необходимо было ответить на 30 вопросов, в каждом из которых предоставлялось оригинальное изображение и шесть вариантов, преобразованных к выбранным

палитрам. Пользователю необходимо было выбрать то изображение, которое в результате преобразования потеряли как можно меньше в своем качестве и осталось больше всего похоже на оригинал.

В тестировании приняло участие 150 человек, среди которых преподаватели и студенты Брянского государственного технического университета. Обработанная статистика ответов представлена на рис. 6.

Как видно из полученных результатов лучше всего показала себя адаптивная палитра, полученная с помощью начального приближения из цветового пространства *HSV*. В более 40 % случаев пользователи выбирали изображения данной палитры как наиболее удачные при преобразовании в малое число цветов. На все остальные протестированные пять палитр приходится в сумме чуть меньше 59 % ответов.

Интерпретация результатов. Разработанная методика получения палитры с помощью адаптивной карты трансформации показала свою состоятельность в проведенных экспериментах. При этом в качестве начального приближения для подобной палитры лучше выбирать палитру из пространства *HSV*, так как она имеет большую ориентацию на психологию восприятия цвета человеком.

На втором месте неожиданно оказалась равномерная палитра *RGB[4,4,4]*. По своим показателям она обошла и адаптивную палитру *RGB*, и равномерную палитру *HSV*. Такие результаты, скорее всего, связаны с тем, что полученные с использованием этой палитры изображения являются более контрастными, это и повлияло на выбор пользователей.

Неожиданным может показаться чрезвычайно низкий показатель равномерной палитры *HSV[4,4,4]*. Цветовое пространство *HSV* достаточно хорошо описывает модель восприятия цвета человеком. Эта модель легка для понимания и должна была показать хорошие результаты. Однако, как показала практика, пользователи не оценили результат преобразования по равномерной палитре *HSV[4,4,4]*. Это можно объяснить тем,

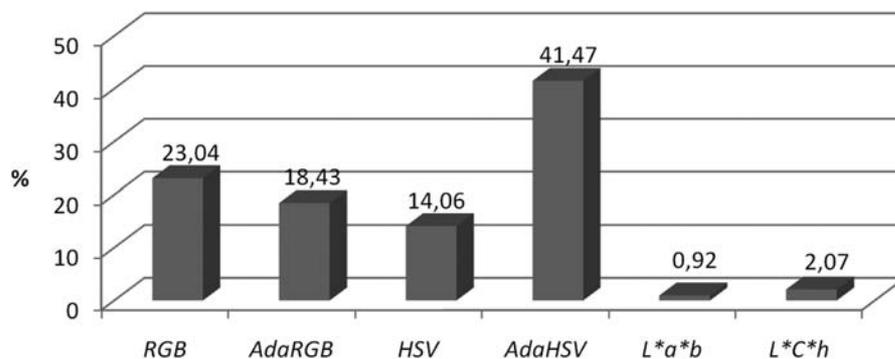


Рис. 6. Результат тестирования палитр

что получение палитры путем равномерного квантования осей в пространстве HSV не учитывает такие особенности системы зрительного восприятия человека, как меньшая чувствительность к изменению оттенка цвета при недостаточной или чрезмерной яркости, при недостаточной насыщенности цвета. Более адекватные результаты можно было бы получить с помощью усложнения схемы квантования и введения зависимости от значений компонентов. Но подобная схема становится достаточно сложной как для реализации, так и для анализа и интерпретации. К тому же ввод дополнительных зависимостей приведет к еще большей вычислительной сложности метода.

Аналогичные выводы можно сделать по поводу палитр L^*a^*b и L^*C^*h . У этих палитр результаты еще более скромные. Для них также плохо подходит способ равномерного квантования. При этом преобразование к палитре для этих пространств является самым неэффективным с точки зрения вычислительной нагрузки.

Оценка скорости преобразования. Помимо качества преобразования очень важно учитывать скорость обработки. Ранее были выдвинуты требования, которым должна удовлетворять палитра для систем поиска изображений по содержанию. Так как информационно-поисковые системы опери-

руют большими объемами данных, в качестве одного из важных требований к палитре выдвигалось применение простых операций и большая скорость обработки.

Для проведения тестирования было написано консольное приложение, выполняющее перевод изображений к цветам палитры в автоматическом режиме. Чтобы уменьшить субъективную составляющую, приложение загружало изображение в память и измеряло только скорость обработки без учета операций работы с диском. Тестирование проводили многократно на разных компьютерах и изображениях. Данные были усреднены. Результаты тестирования представлены в табл. 2 и на рис. 7. За 100 % взята скорость преобразования к палитре $RGB[4,4,4]$, так как это самый быстрый вариант преобразования.

Как и следовало ожидать, преобразование в адаптивную палитру не намного по скорости отличается от преобразования в равномерную палитру пространства RGB . Такое положение объясняется тем, что для преобразования в адаптивную палитру требуется всего три операции побитового сдвига и прямое обращение к памяти.

Палитры HSV , L^*a^*b , L^*C^*h показали плохую производительность. Это связано с вычислительной сложностью алгоритмов перевода в разные цветовые пространства.

Таблица 2

Скорость работы алгоритмов преобразования

Палитра	Скорость обработки
$RGB [4, 4, 4]$	100,00
$AdaRGB$	118,96
$HSV [4, 4, 4]$	657,32
$AdaHSV$	119,09
$L^*a^*b [4, 4, 4]$	3739,27
$L^*C^*h [4, 4, 4]$	4647,05

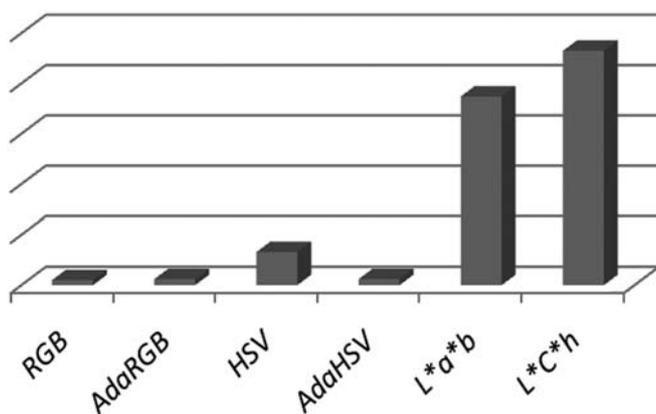


Рис. 7. Диаграмма скорости преобразования

Заключение

Согласно проведенным тестам разработанная адаптивная палитра показывает лучшее качество преобразования по сравнению с другими. При этом она практически не уступает по скорости работы преобразованию в равномерную палитру RGB . Поэтому данный подход решает поставленные в начале статьи задачи по созданию палитры для системы поиска изображений по содержанию.

Список литературы

1. Васильева Н., Марков И. Синтез цветовых и текстурных признаков при поиске изображений по содержанию // Труды Российского семинара по оценке методов информационного поиска РОМИП 2007–2008. С. 135–144.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
3. Sharma G., Wu W., Dalal E. N. The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations // Color Research and Application. 2005. February. Vol. 30, N 1.
4. Kanungo T., Mount D. M., Netanyahu N. S., Piatko C. D., Silverman R., Wu A. Y. An efficient k-means clustering algorithm: Analysis and implementation // IEEE Trans. Pattern Analysis and Machine Intelligence. 2002. P. 881–892.

УДК 004.7

С. Л. Беляков, д-р техн. наук, проф.,
М. Л. Белякова, канд. техн. наук, доц.,
Д. С. Самойлов, аспирант,
Таганрогский технологический институт
Южного федерального университета,
e-mail: beliacov@yandex.ru

Геоинформационный сервис ситуационного центра

Анализируется вариант построения геоинформационного сервиса, ориентированного на предоставление картографических материалов с учетом опыта их использования при принятии решений. Рассматриваются особенности постановки и реализации оптимизационных поисковых задач, реализуемых при построении новых решений.

Ключевые слова: ситуационный центр, геоинформационные системы, сервисы, сетевые сообщества

Ситуационные центры создаются как инструмент коллективного принятия решений в условиях недостатка, неполноты и неопределенности исходной информации [1]. Системы данного типа строятся на различных уровнях государственного и производственного управления. Средства сбора и отображения данных в ситуационных центрах призваны стимулировать творческий процесс решения экспертами сложных управленческих задач. Современная концепция сетевого сервиса дает возможность использовать в процессе принятия решений информационные ресурсы интрасетей и сети Internet. Одним из таких сервисов является геоинформационный — получение ответов на запросы пространственно-временного содержания.

В настоящее время существует достаточно много разномасштабных геоинформационных систем (ГИС), способных выполнять пространственно-временные запросы. Однако, как показывает анализ, возможность получить ответ на запрос не является определяющей при постановке и решении задач в ситуационных центрах. Ответы на запросы — это не решения. Решение строится коллективом, поэтому важное значение приобретает последовательный процесс устного обмена знаниями, гипотезами, обсуждение эвристических стратегий и тактик экспертами. Соответственно, запросы к ГИС не являются одиночными, а связаны некой логикой поиска

решения задачи, стоящей перед ситуационным центром. Поисковый характер процедуры построения решения, свойственный неформализуемым и неопределенным постановкам проблем, требует введения в состав ГИС дополнительных компонентов.

Целью настоящей работы является анализ особенностей организации геоинформационного сервиса для поддержки поискового режима решения прикладных задач.

Известно, что поиск и принятие решений экспертами строится на основе интуиции, которая базируется на опыте [2]. Образное представление геоинформационным сервисом пространственно-временных данных является стимулом и мотивирует мыслительный процесс экспертов. Поэтому важно, чтобы логика работы сервиса соответствовала практической интуитивной логике, которая используется экспертами. Анализ показал, что процесс изучения ситуаций в заданной области пространства и времени базируется на следующих стратегиях отбора информации, сводящихся к нахождению данных:

- 1) о ситуациях, похожих на заданную ситуацию в той же местности;
- 2) о ситуациях, похожих на заданную ситуацию в другой местности;
- 3) о похожих между собой ситуациях в той же местности;
- 4) о похожих ситуациях в похожей местности.

Первые три варианта направлены на поиск похожей ситуации и использование принятых в ней решений как базы для конструирования нового решения путем обобщения и адаптации к заданной местности. Последняя стратегия используется для синтеза решений на основе аналогии.

Результатом применения геоинформационного сервиса является множество ресурсов, предоставленных эксперту в сеансе работы

$$E = \{e_1, e_2, \dots, e_N\}.$$

Под ресурсом $e_i \in E$ понимается картографическое изображение, включающее ссылки на тексты, гипертексты, таблицы, схемы, инфографику, видео- и аудиозаписи, анимацию, тематические каталоги, базы данных. Изучение содержания ресурсов экспертом включается в процесс поиска решения. Следует подчеркнуть, что рассматриваемый в данной работе процесс поиска и результат поиска связаны с интуитивным осмыслением ин-

формации экспертом, но не с нахождением ресурса, содержащего ответ на поставленный вопрос.

В процессе поиска информации посредством геоинформационного сервиса аналитик реализует некоторую оптимизационную стратегию поиска. Таковых можно сформулировать две. Первая стратегия:

$$\begin{cases} I(E) \rightarrow \max, \\ \left\{ \begin{array}{l} t \leq t^*, \\ R \leq R^*, \end{array} \right. \end{cases} \quad (1)$$

где I — функция оценки информативности результата поиска; t — время поиска; t^* — заданное ограничение; R — необходимые сетевые и вычислительные ресурсы; R^* — заданное ограничение. В такой постановке решается задача пользователем, желающим найти "несколько наиболее известных (полезных) примеров решения проблемы".

Вторая стратегия:

$$\begin{cases} t \rightarrow \min, \\ \left\{ \begin{array}{l} I \geq I^*, \\ R \leq R^*, \end{array} \right. \end{cases} \quad (2)$$

соответствует стремлению "быстро найти что-нибудь полезное, имеющее отношение к проблеме".

Следует подчеркнуть, что под временем поиска t понимается интервал времени, необходимый аналитику для решения прикладной задачи. С практической точки зрения именно этот показатель определяет качество геоинформационного сервиса. Время, затраченное геоинформационной системой на построение ответа на любой одиночный запрос, характеризует в большей степени работу подсистемы управления данными и не является показателем качества найденных ресурсов. В то же время поиск в широком понимании предполагает циклическое повторение операций запроса и анализа ответа, т. е. временная модель процесса использования геоинформационного сервиса при анализе имеет вид

$$\begin{aligned} & \langle T_A, E_A \rangle, \\ T_A &= \{(\Delta t_i^{\text{ГИС}}, \Delta t_i^A)\}, \\ E_A &= \{e_i^A\}, \end{aligned} \quad (3)$$

где T_A — множество пар интервалов, соответствующих построению ответа на запрос геоинформационным сервисом ($\Delta t_i^{\text{ГИС}}$) и анализу полученного промежуточного результата пользователем (Δt_i^A). Множество E_A включает в себя частичные результаты каждого шага поиска. Исходя из этого,

$$t = \sum_i^{|T_A|} (\Delta t_i^{\text{ГИС}} + \Delta t_i^A).$$

Можно видеть, что стремление сократить время поиска приводит к необходимости сокращать зна-

чения $|T_A|$, $\Delta t_i^{\text{ГИС}}$, Δt_i^A . Сделать это исключительно за счет совершенствования механизмов поиска в базе данных и построения изображений геоинформационной системой невозможно.

Таким образом, геоинформационный сервис должен обеспечить поддержку модели поиска решения (1)—(3). Основой решения оптимизационных задач является оценка информативности. Известны трудности оценки информативности картографических материалов [3], которые обусловлены различиями в областях применения карт, планов и схем, а также субъективизмом пользователей в оценке уровня информативности. Чтобы решить проблему, предлагается следующий подход.

1. Информационную базу сервиса хранить как ресурс специализированного сетевого сообщества, члены которого решают задачи выбранной прикладной области и готовы делиться опытом их решения. Для выполнения этого требования реализация сервера ГИС должна быть дополнена структурами данных, обеспечивающих наполнение картографических материалов субъективным опытом и оценками информативности уже использованных материалов. Тем самым создается основа для накопления коллективного опыта субъектами.

2. Связывать субъективные оценки информативности конкретных фрагментов карты с концептами онтологии, принятой экспертами-пользователями сервиса. Развитие социальных сетей как способ обмена знаниями специалистов дает путь к согласованному пониманию полезности картографических материалов. Внутри сетевых сообществ действуют понятные всем термины и концепции, что значительно облегчает повторное использование геоинформационных ресурсов для решения задач определенного класса. Таким образом возникает основа повторного использования полученных отдельными субъектами результатов.

3. Использовать для решения задач (1) и (2) средства индексирования и поиска в базе данных геоинформационного сервиса по значению информативности. Ключами поиска и индексирования являются субъективные оценки информативности, выраженные через концепты онтологии сетевого сообщества. База данных, таким образом, должна включать оценки информативности отдельных ресурсов, т. е. множество $\{I_1(e_1), I_2(e_2), \dots, I_N(e_N)\}$. Эти оценки указываются членами сетевого сообщества, которые пользовались отдельными ресурсами. Здесь $I_i(e_i)$ — функция информативности отдельного ресурса. Каждый ресурс оценивается индивидуально соответственно своему происхождению, актуальности, виду представления, полноте и непротиворечивости.

4. Предоставлять пользователю ограниченный набор формулировок запросов. Каждая формулировка соответствует одной из привычных для ана-

литика стратегий отбора полезной информации. Параметром каждого запроса является картографическое отображение текущей анализируемой ситуации. Обслуживание запроса строится на основе оценки информативности всей совокупности найденных ресурсов $I(I_1(e_1), I_2(e_2), \dots, I_N(e_N))$. Здесь $I(x_1, x_2, \dots, x_N)$ — интегральная функция информативности набора ресурсов. Данная оценка отражает свойство человека-аналитика обобщать сведения из разных источников.

Полученный в сеансе поиска материал может быть применен повторно, если его смогут идентифицировать другие пользователи, установив в соответствии контексту собственного поиска решения. Точность идентификации, соответствие опыта текущей ситуации, отсутствие несущественных сведений позволяют эффективно решить поставленную прикладную задачу и оценить полученные картографические материалы как информативные. Обобщенно сценарий использования геоинформационного сервиса для поиска решения, таким образом, включает в себя следующие действия.

1. Нахождение известных картографических материалов, описывающих ситуацию, близкую по смыслу к анализируемой. Для этого вначале отбираются концепты онтологии, с помощью которых можно охарактеризовать ситуацию и принятые решения. Затем запускается процедура, соответствующая стратегии (1) или (2), результатом которой является набор ссылок на информативные ресурсы.

2. Субъективная оценка информативности найденных результатов. Если оценка недостаточна, потребуется переформулировать постановку задачи и вернуться к предыдущему этапу;

3. Изучение полученных информационных ресурсов, конструирование возможных решений прикладной задачи.

Описанная последовательность циклически повторяется, если анализ информации изменил взгляд пользователя на проблему и смысловую направленность дальнейшего поиска. Следует подчеркнуть, что доля геоинформационного исследования в рассматриваемой процедуре относительно невелика. Большую значимость приобретает поисковый процесс информативных ресурсов.

На рис. 1 и 2 (см. вторую сторону обложки) показан пример использования геоинформационного сервиса. Рис. 1 иллюстрирует исходную точку поиска решения путем привлечения картографических данных: на карте задается область рассматриваемой ситуации, для которой должны быть найдены аналогичные прецеденты. На рис. 2 показан пример ответа, отображающего похожие ситуации в другой местности. Описание похожих ситуаций сопровождается описанием решений на специально выделенном слое электронной карты. В качестве примера схемы решения показана схема на рис. 3.



Рис. 3. Пример схемы на слое принятия решений

Информативность картографического ресурса предлагается оценивать с помощью нечетких дескрипторов. Нечетким дескриптором $D(e)$ ресурса e будем называть вектор

$$(d_1, d_2, \dots, d_L), d_i = \mu_i/c_i, i = \overline{1, L},$$

в котором $c_i \in C$ — концепт онтологии предметной области поиска $O = \langle C, R, F \rangle$, $\mu_i \in [0, 1]$ — степень принадлежности ресурса e смысловому содержанию концепта. При $\mu_i = 0$ найденные данные не имеют смыслового отношения к концепту, при $\mu_i = 1$ смысл данных полностью соответствует сути концепта.

В дескриптор включаются только элементы со значениями $\mu_i > 0$, полагая, что отсутствие координаты $c_b \in C$ означает, что в выражении μ_b/c_b $\mu_b = 0$.

Дескриптор можно рассматривать как точку в $|C|$ -мерном пространстве понятий предметной области. Значение μ_i отражает степень уверенности эксперта в том, что ресурс e будет полезен при решении задач, смысл которых отображается концептом c_i . Содержательно под $c_i \in C$ может пониматься объект, факт, явление, отношение, класс объектов или отношений, процесс, метод или область знаний.

Проанализируем эффективность предлагаемого механизма поиска с использованием оценок информативности по критерию времени получения результата.

При сравнении будем анализировать отношение полезности информационных ресурсов $\Theta \subseteq E \times E$, которое отражает их информативность. Наличие такого отношения лежит в основе любой задачи информационного поиска. При работе пользователя-аналитика с геоинформационным сервисом имеются две реализации данного отношения:

- отношение Θ_A , подразумеваемое аналитиком;
- отношение $\Theta_{ГИС}$, поддерживаемое геоинформационным сервисом.

Если обе стороны используют одно и то же отношение, т. е.

$$\Theta_A = \Theta_{\text{ГИС}},$$

то в соответствии с временной моделью поиска (3),

$$t_{\min} = \sum_i \frac{|T_A|}{\Delta t_i^{\text{ГИС}} + \Delta t_i^A},$$

причем $\Delta t_i^A \rightarrow \min$ и $|T_A| \rightarrow \min$ независимо от личности аналитика и решаемой задачи. Но на практике

$$\Theta_A \neq \Theta_{\text{ГИС}},$$

что заставляет аналитика затрачивать дополнительное время на поиск (или, возможно, упорядочение) предоставленных геоинформационным сервисом ресурсов. Как минимум, растет Δt_i^A , очень вероятен рост $|T_A|$. Поэтому в реальности $\Theta_A \approx \Theta_{\text{ГИС}} \Rightarrow t > t_{\min}$.

Случай близких отношений:

$$\Theta_A \approx \Theta_{\text{ГИС}}$$

соответствует использованию аналитиком геоинформационного сервиса, использующего опыт. В этой ситуации $t \approx t_{\min}$.

Близость двух отношений понимается следующим образом:

$$\Theta_A \approx \Theta_{\text{ГИС}} : \Theta_A / (\Theta_A \cap \Theta_{\text{ГИС}}) \neq \emptyset \ \& \ \Theta_{\text{ГИС}} / (\Theta_A \cap \Theta_{\text{ГИС}}) \neq \emptyset \ \& \ \frac{|\Theta_A \cap \Theta_{\text{ГИС}}|}{|\Theta_A|} > \varepsilon \ \& \ \frac{|\Theta_A \cap \Theta_{\text{ГИС}}|}{|\Theta_{\text{ГИС}}|} > \varepsilon, \\ 0 < \varepsilon < 1.$$

Здесь ε является степенью близости отношений. Значение $\varepsilon = 0,5$ рассматривается как пороговое, при $\varepsilon \leq 0,5$ считается, что $\Theta_A \neq \Theta_{\text{ГИС}}$.

Предлагаемый в данной работе вариант организации поисковой системы приводит к получению $\Theta_A \approx \Theta_{\text{ГИС}}$, поскольку:

- онтология создается и поддерживается в рамках известного круга интересов сетевого сообщества;
- каждый элемент априорно оценен соответственно смыслу и его дескриптор находится в согласованном пространстве признаков;
- как следствие, существует общая определенность в оценке значимости информационных ресурсов и это отражено в значениях дескрипторов. Поэтому справедливо считать, что $\varepsilon > 0,5$.

Таким образом, в данной работе описана и проанализирована модель геоинформационного сервиса, включающая в себя наравне с традиционными инструментами пространственного анализа средства повторного использования картографических материалов для принятия решений в различных ситуациях. Эти средства предполагают использование опыта в рамках сетевого сообщества экспертов. Основой процесса передачи опыта является использование нечетких дескрипторов информационных ресурсов в рамках используемой сетевым сообществом онтологии.

СПИСОК ЛИТЕРАТУРЫ

1. Райков А. Н. Развитие России и единое информационное пространство // Вестник РФФИ. 1999. № 3. С. 29–34.
2. Майерс Д. Интуиция. Возможности и опасности. СПб.: Питер, 2010.
3. Зеркаль О. В. Понятия "информация" и "геоинформация", развитие взглядов и современные воззрения // Геоинформатика. 2008. № 4. С. 21–33.

ПРЕДСТАВЛЯЕМ КНИГУ



С. М. Авдошин, Е. Ю. Песоцкая

Информатизация бизнеса. Управление рисками. М.: ДМК Пресс, 2011

Проблема управления рисками при информатизации бизнеса является одной из наиболее актуальных и значимых в ИТ-индустрии. В предлагаемом учебно-практическом пособии затронуты как теоретические, так и практические вопросы управления рисками, раскрывается специфика механизма управления рисками при реализации проектов в области информационных технологий.

В основу учебного пособия положен многолетний опыт преподавания авторами дисциплины "Управление рисками" на отделении программной инженерии Высшей школы экономики.

Книга предназначена для студентов магистратуры, обучающихся по направлениям 080500.68 "Бизнес-информатика" и 231000.68 "Программная инженерия", а также для ИТ-специалистов, разработчиков и заказчиков программных продуктов, менеджеров ИТ-проектов.

В. И. Струченков, д-р техн. наук, проф.,
А. Н. Козлов, аспирант, **А. С. Егунов**, аспирант,
 Московский институт радиотехники,
 электроники и автоматики
 (технический университет),
 e-mail: str1942@mail.ru

Динамическое программирование в проектировании трасс линейных сооружений¹

Рассматривается задача аппроксимации плоской кривой, заданной дискретной последовательностью точек, которая не является графиком однозначной функции. Элементами аппроксимирующей кривой, число которых неизвестно, служат дуги окружностей и отрезки прямых. Аппроксимирующая кривая должна удовлетворять системе ограничений и минимизировать значение принятого критерия оптимальности. Задача возникает при разработке САПР трасс линейных сооружений. Излагается алгоритм решения задачи по методу динамического программирования.

Ключевые слова: аппроксимация, ограничения, динамическое программирование

Постановка задачи

При разработке систем автоматизированного проектирования трасс линейных сооружений (железных и автомобильных дорог, трубопроводов различного назначения, траншей для водо- и газоснабжения и др.) возникают задачи аппроксимации плоских кривых, заданных дискретной последовательностью точек, проектными кривыми, состоящими из элементов заданного вида, параметры которых должны удовлетворять системе ограничений. Ограничения обусловлены необходимостью соблюдения строительных норм и правил, условиями землепользования и другими факторами. При проектировании трасс различных сооружений в качестве элементов аппроксимирующей гладкой кривой могут использоваться отрезки прямых, парабол второй степени, окружностей и клотоид.

Число элементов неизвестно и должно быть установлено в процессе решения задачи при выполнении всех ограничений и минимизации заданного показателя качества аппроксимации. В качестве такого показателя можно рассматривать максимальное по абсолютной величине отклонение от ис-

¹ Данная статья является продолжением статьи "Кусочно-параболическая аппроксимация плоских кривых при наличии ограничений специального вида". См. журнал "Информационные технологии", № 7, 2011. С. 44–48.

ходной кривой, сумма квадратов отклонений в заданных точках или интеграл от квадрата разности исходной и аппроксимирующей функций. Задачи кусочно-линейной и кусочно-параболической аппроксимации, возникающие при проектировании продольного профиля железных и автомобильных дорог соответственно, решены с применением метода динамического программирования [1–3].

При проектировании плана трассы возникают дополнительные сложности, обусловленные неоднозначностью аппроксимируемых функций.

В данной статье рассматривается именно такая задача, в которой элементами аппроксимирующей кривой являются дуги окружностей, сопрягаемые встык или отрезками прямых.

При этом должны быть выполнены ограничения четырех типов:

1) отклонения проектной кривой от исходной в заданных точках не должны превосходить заданных значений. В частности, возможно задание фиксированных точек;

2) кривизна аппроксимирующей кривой во всех точках по абсолютной величине не больше заданной;

3) длины элементов не должны быть меньше заданных значений;

4) на границах элементы должны иметь общую касательную.

Считаются заданными начальная и конечная точка искомой кривой, а также начальное и конечное направления.

Применение динамического программирования

Исходную кривую будем рассматривать как ломаную линию. В точках перелома построим нормали (по направлению к центру окружности, соединяющей три смежные точки). При необходимости нормали могут быть построены и в промежуточных точках (рис. 1).

Ограничения первого типа задают область поиска, т. е. крайние точки на каждой нормали.

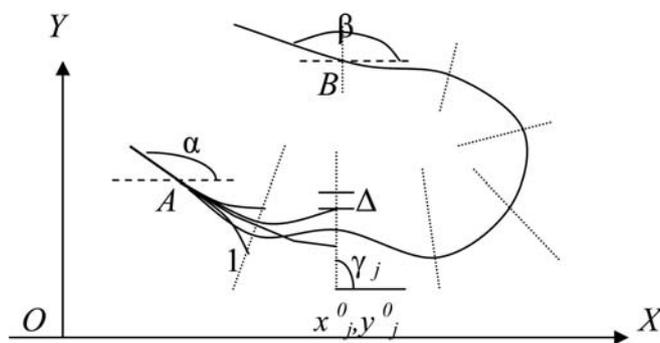


Рис. 1. Первый шаг алгоритма

Для каждой нормали задается угол γ_j с осью X и разбивается сетка с шагом Δ относительно исходной ломаной. Декартовы координаты узлов сетки на каждой нормали вычисляются по формулам

$$x_j^i = x_j^0 + n_j^i \Delta \cos \gamma_j; \quad y_j^i = y_j^0 + n_j^i \Delta \sin \gamma_j,$$

где n_j^i — номер i -й точки на j -й нормали (нумерация начинается с нуля).

Очередной этап динамического программирования — переход к следующей нормали, но не к следующему элементу, число которых неизвестно. В процессе построения аппроксимирующей кривой в одну точку могут приходиться несколько кривых, состоящих из различного числа элементов. Основное понятие в динамическом программировании — "состояние системы" — в данной задаче соответствует точке на нормали и углу между касательной к окружности в этой точке и осью X . Другими словами, задача рассматривается как двухпараметрическая. Начальное состояние — точка A и угол α , конечное состояние — точка B и угол β (рис. 1). Рассмотрим сначала задачу, в которой допускается сопряжение дуг окружностей без прямых вставок.

Заданную минимальную длину элемента обозначим L_{\min} . Максимальную длину элемента L_{\max} примем равной $2L_{\min}$, так как элементы длиной больше $2L_{\min}$ можно рассматривать как два различных элемента.

Первый шаг алгоритма динамического программирования

Для каждого положения точки $D(x_d, y_d)$ первой нормали по ее декартовым координатам, координатам точки A и углу α строится дуга окружности (рис. 2). При этом сначала определяются координаты центра окружности x_c, y_c , а затем ее радиус R :

$$x_c = ((y_d - y_a)/2 + (x_d^2 - x_a^2)/(2(y_d - y_a)) - x_a \operatorname{ctg} \alpha) / ((x_d - x_a)/(y_d - y_a) - \operatorname{ctg} \alpha); \quad (1)$$

$$y_c = y_a + x_a \operatorname{ctg} \alpha + \operatorname{ctg} \alpha (x_a \operatorname{ctg} \alpha - ((y_d + y_a)/2 + (x_d^2 - x_a^2)/(2(y_d - y_a)))) / ((x_d - x_a)/(y_d - y_a) - \operatorname{ctg} \alpha); \quad (2)$$

$$R = AC.$$

При $\alpha = 0$ или $\alpha = 180^\circ$ и $y_d \neq y_a$ вместо формул (1), (2) имеем

$$x_c = x_a; \quad y_c = (y_a + y_d)/2 + (x_d - x_a)^2 / (y_d - y_a) / 2.$$

Если $\operatorname{tg} \alpha = 0$ и $y_d = y_a$, то $x_c = (x_a + x_d)/2$; $y_c = y_a$.

При $y_d = y_a$ и $\operatorname{tg} \alpha \neq 0$ имеем $x_c = (x_a + x_d)/2$; $y_c = y_a - \operatorname{ctg} \alpha (x_a - x_d)/2$.

Случай $(x_d - x_a)/(y_d - y_a) = \operatorname{ctg} \alpha$ означает, что конечная точка расположена на касательной и вместо дуги окружности имеем отрезок прямой.

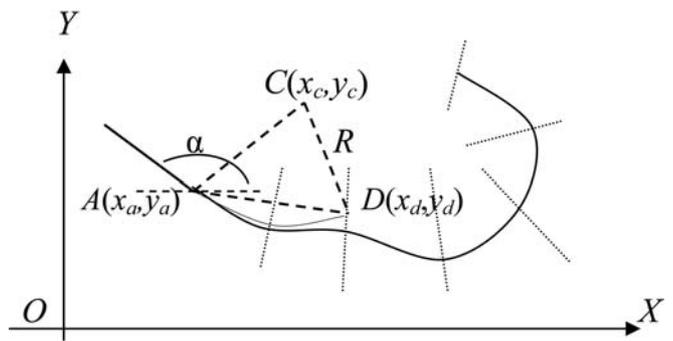


Рис. 2. Построение первого элемента

Если $y_d = y_a$ и $\operatorname{tg} \alpha = 0$ этот отрезок прямой параллелен оси X .

Если радиус меньше допустимого, т. е. не выполнено ограничение типа 2, переходим к следующей точке, в противном случае определяем длину дуги $L = 2R \arcsin(AD/(2R))$. Если $L < L_{\min}$, т. е. не выполнено ограничение типа 3, переходим к следующей точке, в противном случае проверяем выполнение ограничений типа 1. Для этого приходится искать точку пересечения дуги окружности поочередно с каждой нормалью в пределах данной дуги.

Если ни одна из точек первой нормали не может быть соединена с точкой A дугой окружности, удовлетворяющей всем ограничениям, переходим ко второй нормали, сохраняя точку A в качестве начальной. Для допустимых дуг вычисляется и запоминается величина принятого критерия оптимальности, например, сумма квадратов отклонений по всем нормальям в пределах данной дуги. Для каждой точки в конце дуги вычисляется и запоминается угол между касательной и осью X .

Процесс продолжается для нормалей, отстоящих от точки A не более чем на L_{\max} , так, чтобы длины элементов получались не более L_{\max} .

Построены все варианты первого элемента аппроксимирующей кривой. Для каждого из них вычислен угол между касательной и осью X в конечной точке элемента и значение критерия, которые занесены в память. Другими словами, вычислен "веер" исходящих из точки A элементов и их характеристики (см. рис. 1). Каждая точка может быть концом только одного элемента, все элементы имеют общее начало. Запоминание организовано так, что с каждой точкой каждой нормали, которая рассматривается как конец элемента, связано число входящих элементов (на первом шаге алгоритма это 0 или 1), для каждого из входящих элементов запоминаются номер нормали начала элемента (для первого элемента это ноль), номер точки на ней (для первого элемента это ноль), угол между касательной и осью X в начале элемента (для первого элемента это α), угол между касательной и осью X в конце элемента и критерий оптимальности.

Общий шаг алгоритма динамического программирования

В качестве начала элемента рассматривается очередная нормаль. На втором шаге это первая нормаль. Если ни в одном из принадлежащих ей узлов сетки нет входящих элементов, переходим к следующей нормали. В противном случае точка рассматривается как начало элемента. Если с этой точкой связано более одного элемента, то для каждого из них строятся и проверяются по ограничениям все варианты следующего элемента с использованием формул (1), (2) при замене координат точки A на координаты рассматриваемой точки и угла α на угол между касательной входящего элемента и осью X .

Во избежание полного перебора необходима отбраковка бесперспективных вариантов положения элементов и их продолжений. В каждую точку приходит уже не один, как на первом шаге, а "веер" входящих элементов. Если два или более аппроксимирующих кривых имеют общую конечную точку и в ней общую касательную, то в соответствии с принципом оптимальности Р. Беллмана [1] из них следует оставить лучший вариант, а худший вариант и все его продолжения далее не рассматривать. Однако ситуация, при которой имеется общая касательная, маловероятна. Подобно алгоритму кусочно-параболической аппроксимации [3] можно рассчитать максимальный угол ε между касательными двух окружностей, проходящих в заданную точку, при котором на следующем элементе аппроксимирующей кривой при его длине L_{\max} отклонение при прочих равных условиях не превышает Δ . Другими словами все кривые, сходящиеся в одной точке, считаем сравнимыми, если $|\varphi_i - \varphi_j| < \varepsilon$, где φ_i и φ_j — углы между касательными и осью X в конце сходящихся кривых, при этом кривые могут отличаться числом элементов. Величина ε , как и Δ , задается в числе исходных данных. Фактически это дискрет по второму параметру, характеризующему "состояние системы".

Построение вариантов последнего элемента

Если при построении очередного элемента аппроксимирующей кривой без нарушения ограничения по максимальной длине достигнута последняя нормаль, то делается попытка в качестве его конечной точки использовать точку B . Если для соответствующего варианта выполнены все ограничения и $|\varphi - \beta| < \varepsilon_1$, где φ — угол между касательной в конце элемента и осью X , β — заданный угол в конце трассы (см. рис. 1), ε_1 — заданное число, определяемое требуемой точностью расчета, то этот вариант считается допустимым и для него вычисляется критерий оптимальности. При наличии нескольких таких вариантов выбирается и запоминается лучший из них по критерию оптимальности.

Восстановление оптимальной трассы

Если для очередной нормали нет вариантов построения очередного элемента аппроксимирующей кривой или все нормали рассмотрены, анализируется вариант, который запомнили ранее. При его отсутствии задача не имеет решения. Это может быть при противоречивости системы ограничений или при задании слишком малого значения ε_1 . Если вариант положения аппроксимирующей кривой с окончанием в точке B есть, то осуществляется обратный разворот. Для каждой кривой, проходящей в точку B , были записаны номер нормали, точки на ней и угол между касательной и осью X в начале элемента. По этим данным восстанавливается последний элемент. Далее его начало рассматривается как конец предпоследнего элемента, для которого известно начало, и т. д. до тех пор, пока началом элемента не будет нулевая нормаль, т. е. точка A . В итоге определяется и число элементов и вся трасса.

Заметим, что при построении дуг окружностей прямолинейные элементы могут появляться как частный, но маловероятный случай.

Задача существенно усложняется, если прямолинейные элементы обязательны, т. е. если не допускается сопряжение двух дуг окружностей встык, а требуется сопрягать их прямолинейным элементом, длина которого неизвестна, но не может быть меньше заданной. Для решения этой задачи не удастся применить изложенный алгоритм, рассматривая отрезок прямой как отдельный элемент, так как конец этого элемента может не совпадать с имеющимися точками на нормалях, а его привязка к какой-либо точке на нормали нарушает ограничение типа 4, т. е. условие общей касательной на стыках элементов. Поэтому приходится в качестве элемента рассматривать связку "дуга окружности + отрезок прямой" с переменными параметрами — длиной прямой и углом между касательной и концом. При построении очередной связки рассматриваются последовательно все точки на очередной нормали (нормали и точки на них задаются, как и ранее). Для каждой нормали с учетом ее ориентации задается диапазон углов $\varphi_{j\min} \leq \varphi_j \leq \varphi_{j\max}$ между входящим "веером" и осью X . Границы диапазона $\varphi_{j\min} = \gamma_j - \pi/2 - \psi$ и $\varphi_{j\max} = \gamma_j - \pi/2 + \psi$, где ψ — заданный угол (рис. 3).

На этом рисунке C и D — точки на нормалях. Прямая BD , как вероятное положение касательной, получена поворотом нормали на $\pi/2$, ее угол с осью X $\varphi_j = \gamma_j - \pi/2$.

В каждой точке, рассматриваемой как конец связки (на рис. 3 это точка D), для каждой точки и угла в начале связки (точка C и угол φ_i между прямой CB и осью X) будут последовательно с заданным дискретом ε перебираться значения углов в полученном диапазоне. Тем самым "состояние

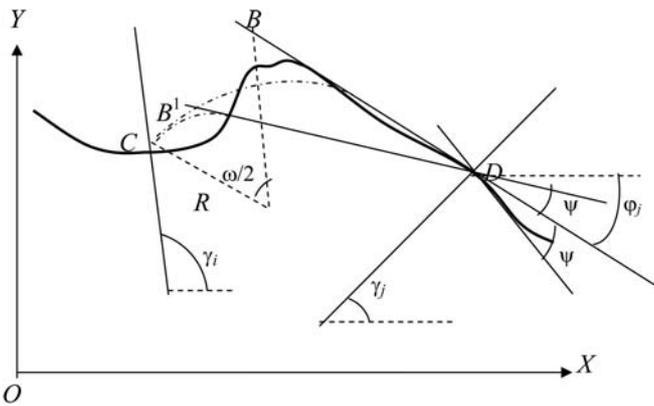


Рис. 3. Варианты построения связки дуга + отрезок прямой

системы" то же, что и ранее (точка + угол), но углы не вычисляются, а назначаются с заданной дискретностью. Фактически изначально разбивается сетка не только по нормали с шагом Δ , но и по углам с шагом ϵ .

По заданному начальному (x_c, y_c, φ_i) и конечному состоянию (x_d, y_d, φ_j) однозначно определяются все параметры связки: сначала вершина угла (координаты x_b, y_b точки B на рис. 3) и сам угол поворота ω , затем длина BC и радиус окружности R , затем длина дуги окружности $L_{кр}$ и длина отрезка прямой $L_{п}$:

$$x_b = (y_d - y_c + x_c \operatorname{tg} \varphi_i - x_d \operatorname{tg} \varphi_j) / (\operatorname{tg} \varphi_i - \operatorname{tg} \varphi_j);$$

$$y_b = y_c + (x_b - x_c) \operatorname{tg} \varphi_i;$$

$$\omega = \varphi_j - \varphi_i;$$

$$R = BC \operatorname{ctg}(\omega/2);$$

$$L_{кр} = R\omega; L_{п} = BD - BC.$$

Приведенные формулы должны быть преобразованы, если φ_i или φ_j равны $\pm\pi/2$.

Далее проверяется выполнение всех ограничений, допустимые соединения оцениваются по критерию оптимальности и для каждой точки и угла в конце связки запоминаются точка и угол наилучшего начала. Задача остается двухпараметрической, но число вариантов и сложность оценки вариантов возрастают по сравнению с аппроксимацией только круговыми элементами.

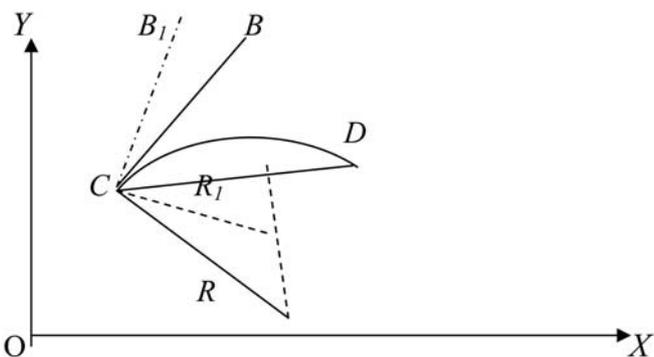


Рис. 4. Прекращение перебора при нарушении ограничения

Может показаться, что изложенные алгоритмы требуют полного перебора всех возможных комбинаций углов при заданных точках начала и конца элемента при сопряжении встык или при построении связки. Однако это не так. Поскольку перебор упорядочен от меньших значений к большим, то при нарушении ограничений может быть выявлена бессмысленность дальнейшего увеличения варьируемого параметра, так как при этом невязка в ограничении увеличивается. В этом случае требуется перейти к следующему состоянию. Частный случай такой ситуации показан на рис. 4.

На рис. 4. точки C и D рассматриваются как концы очередной дуги окружности (при сопряжении окружностей встык). Прямая BC соответствует углу между касательной и осью X в точке C , при котором радиус окружности R принимает минимально допустимое значение. При увеличении этого угла при фиксированном положении точки D радиус $R_1 < R$.

Область практического применения

Предложенные алгоритмы можно использовать при решении различных задач проектирования трасс линейных сооружений, а именно:

- при автоматизированном проектировании продольного профиля автомобильных дорог — как альтернатива кусочно-параболической аппроксимации. Эта задача проще рассмотренной, так как исходная функция однозначна и вместо нормалей можно использовать прямые параллельные оси ординат;
- при проектировании плана трассы автомобильных дорог низких категорий, в частности лесовозных, которые не предназначены для высокоскоростного движения, вследствие чего на них не устраиваются виражи и соответственно не используются клотоиды;
- при проектировании плана трасс трубопроводов большого диаметра.

Данная задача и ее обобщения на случай поиска аппроксимирующей кривой, состоящей из других элементов, может иметь и другие приложения, далекие от проектирования трасс линейных сооружений. Наибольшую сложность представляет случай использования клотоид для обеспечения непрерывности не только касательной, но и кривизны. Этот случай требует отдельного рассмотрения.

Список литературы

1. Белман Р. Динамическое программирование. М.: ИЛ, 1960.
2. Струченков В. И., Козлов А. Н., Егунов А. С. Кусочно-линейная аппроксимация плоских кривых при наличии ограничений // Информационные технологии. 2010. № 12. С. 32–34.
3. Струченков В. И., Козлов А. Н., Егунов А. С. Кусочно-параболическая аппроксимация плоских кривых при наличии ограничений специального вида // Информационные технологии. 2011. № 7. С. 44–48.

Д. Н. Кобзаренко,
канд. техн. наук, ст. науч. сотр.,
Институт проблем геотермии
Дагестанского НЦ РАН,
e-mail: kobzarenko_dm@mail.ru

Особенности применения некоторых алгоритмов вычислительной геометрии в геоинформационных технологиях

Рассматриваются особенности двух алгоритмов: определение наличия пересечения двух отрезков и определение принадлежности точки произвольному полигону. Решение, казалось бы, на первый взгляд простых задач имеет некоторые особенности применения при разработке программного обеспечения геоинформационных систем, которые необходимо учитывать. Такие нюансы не описаны в литературе, однако их игнорирование в определенных случаях приводит к ошибкам и некорректным результатам.

Ключевые слова: геоинформационные технологии, компьютерная графика

Определение наличия пересечения отрезков

Входными данными являются координаты отрезков AB и CD , пересечение которых мы проверяем. Считаем, что отрезки AB и CD строго пересекаются, если точка пересечения P не совпадает с A, B, C и D , т. е. ни с одной из вершин (рис. 1, а).

Направленные отрезки AB и CD задаются следующими параметрическими уравнениями [1]:

$$P = A + r(B - A), \quad P \in AB; \tag{1a}$$

$$Q = C + s(D - C), \quad Q \in CD. \tag{1б}$$

Если отрезки AB и CD пересекаются, то

$$A + r(B - A) = C + s(D - C). \tag{2}$$

Перепишем это векторное соотношение в координатном виде:

$$a_x + r(b_x - a_x) = c_x + s(d_x - c_x), \tag{3a}$$

$$a_y + r(b_y - a_y) = c_y + s(d_y - c_y). \tag{3б}$$

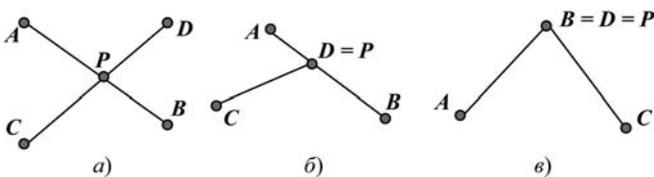


Рис. 1. Виды пересечений отрезков:
а — строгое пересечение отрезков; б, в — нестрогое пересечение отрезков (касание)

Эта система линейных алгебраических уравнений при

$$(b_x - a_x)(d_y - c_y) \neq (b_y - a_y)(d_x - c_x) \tag{4}$$

имеет единственное решение:

$$r = \frac{(a_y - c_y)(d_x - c_x) - (a_x - c_x)(d_y - c_y)}{(b_x - a_x)(d_y - c_y) - (b_y - a_y)(d_x - c_x)}, \tag{5a}$$

$$s = \frac{(a_y - c_y)(b_x - a_x) - (a_x - c_x)(b_y - a_y)}{(b_x - a_x)(d_y - c_y) - (b_y - a_y)(d_x - c_x)}. \tag{5б}$$

Если получившиеся значения r и s принадлежат интервалу $[0, 1]$, то отрезки AB и CD пересекаются и точка пересечения может быть найдена из параметрических уравнений. В случае, когда оба или одно из полученных значений не принадлежат интервалу $[0, 1]$, отрезки AB и CD не пересекаются, но пересекаются соответствующие прямые. Равенство $(b_x - a_x)(d_y - c_y) = (b_y - a_y)(d_x - c_x)$ означает, что отрезки AB и CD параллельны.

Возможны следующие модификации алгоритма проверки пересечения (всего таких модификаций может быть 16):

- 1) $0 \leq r \leq 1$ и $0 \leq s \leq 1$ — учитывается любое пересечение, включая касания;
- 2) $0 < r \leq 1$ и $0 \leq s \leq 1$ — точка пересечения не может быть равна A ;
- 3) $0 < r < 1$ и $0 \leq s \leq 1$ — точка пересечения не может быть равна A, B ;
- 4) $0 < r < 1$ и $0 < s \leq 1$ — точка пересечения не может быть равна A, B, C ;
- 5) $0 < r < 1$ и $0 < s < 1$ — точка пересечения не может быть равна A, B, C, D .

Проблема в использовании текущего алгоритма в реальных задачах, таких как, например [2], заключается в том, что при малых длинах отрезков возникает ошибка в точности определения r и s по формулам (5). Ошибка возникает при выполнении операции деления, и даже введение при расчетах 80-битовых переменных вместо 64-битовых не спасает положение. К тому же увеличение разрядности влечет за собой замедление работы алгоритма проверки пересечения.

Одним из вариантов решения описанной проблемы является введение погрешности при выполнении сравнений. Например, вместо $r < 1$ сравниваем $r < 1 - \Delta$, где Δ — допустимая погрешность (очень маленькое число). Но тогда возникает вопрос, какой взять эту допустимую погрешность, чтобы не перегнуть палку в сторону некорректного определения r ?

На самом деле имеется простое эффективное решение данной проблемы. Оно заключается в том, чтобы избавиться от операции деления, вызывающей ошибку. Избавление от операции деления влечет за собой замену операций сравнения для r и s другими операциями сравнения, как показано

Замена операций сравнения для коэффициентов z и s

Классический вариант определения пересечения отрезков	Модифицированный вариант определения пересечения отрезков под программную реализацию
$d = (b_x - a_x)(d_y - c_y) - (b_y - a_y)(d_x - c_x)$ $r = \frac{(a_y - c_y)(d_x - c_x) - (a_x - c_x)(d_y - c_y)}{d}$ $s = \frac{(a_y - c_y)(b_x - a_x) - (a_x - c_x)(b_y - a_y)}{d}$	$d = (b_x - a_x)(d_y - c_y) - (b_y - a_y)(d_x - c_x)$ $r = (a_y - c_y)(d_x - c_x) - (a_x - c_x)(d_y - c_y)$ $s = (a_y - c_y)(b_x - a_x) - (a_x - c_x)(b_y - a_y)$
Для $r(s) \in [0, 1] (r \geq 0) \cup (r \leq 1)$	$(abs(r) \leq abs(d)) \cup (sign(r) \cdot sign(d) > -1)$
Для $r(s) \in (0, 1) (r > 0) \cup (r < 1)$	$(abs(r) < abs(d)) \cup (sign(r) \cdot sign(d) = 1)$
Для $r(s) \in (0, 1] (r > 0) \cup (r \leq 1)$	$(abs(r) \leq abs(d)) \cup (sign(r) \cdot sign(d) = 1)$
Для $r(s) \in [0, 1) (r \geq 0) \cup (r < 1)$	$(abs(r) < abs(d)) \cup (sign(r) \cdot sign(d) > -1)$

в таблице. Здесь abs — операция модуля числа; \cup — логическое "И"; $sign$ — операция извлечения знака (для положительного числа $sign = 1$, для отрицательного числа $sign = -1$, для нуля $sign = 0$).

Из таблицы следует, что после модификации алгоритма определения пересечения отрезков мы избавляемся от двух трудоемких операций деления, которые могут привести к ошибке, и добавляем три операции модуля и три операции $sign$ для r, s, d . При этом число условий и логических "И" остается неизменным. Получается, что вместо двух многотактовых операций деления имеем шесть одноктактовых операций, а это при многократном использовании процедуры в цикле дает еще и экономию по времени выполнения.

В зависимости от необходимости определения чистых пересечений или касаний алгоритм имеет 16 модификаций. Можно написать одну процедуру, покрывающую все возможные случаи. Но в реальных задачах в ней нет смысла, поскольку чаще всего, как например в работе [2], используются 2—3 модификации и гораздо эффективнее иметь по процедуре для каждой необходимой модификации алгоритма.

Определение принадлежности точки произвольному полигону

Классический алгоритм для решения этой задачи заключается в том, что из искомой точки $P(x, y)$ выпускается луч и определяется число пересечений луча с границей многоугольника. Если отбросить случай, когда луч проходит через вершину

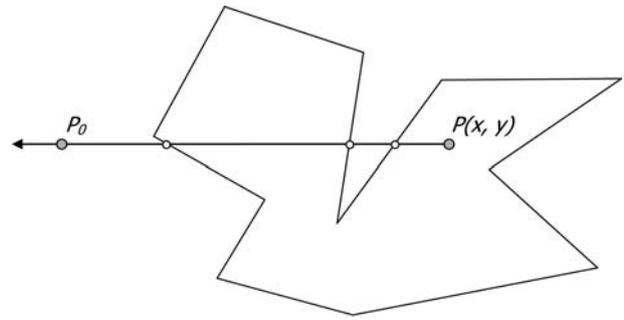


Рис. 2. Определение принадлежности точки полигону с помощью выпускания луча

многоугольника, то решение задачи тривиально — точка лежит внутри, если общее число пересечений нечетно, и точка лежит снаружи, если это число четно [1] (рис. 2).

Ясно, что для любого многоугольника всегда можно построить луч, не проходящий ни через одну из вершин. Однако построение такого луча связано с трудностями и, кроме того, проверку пересечения границы многоугольника с произвольным лучом провести сложнее, чем с фиксированным, например — горизонтальным. Исходя из этого можно построить следующий алгоритм: выпускаем из точки P горизонтальный луч в направлении оси Ox и все ребра многоугольника, кроме горизонтальных, проверяем на пересечение с этим лучом. В случае, когда луч проходит через вершину, т. е. формально пересекает сразу два ребра, сходящихся в этой вершине, засчитаем это пересечение только для тех ребер, для которых эта вершина является верхней (или нижней) [1].

Описанное выше широко известное теоретическое решение задачи с точки зрения программно-алгоритмической реализации состоит в следующем. Разрабатывается локальная процедура проверки наличия пересечения двух отрезков (описана выше). В этой процедуре наличие пересечения засчитывается, если коэффициенты r и s , найденные по формулам (5), принадлежат промежуткам: $0 < r < 1$ (или $0 \leq r < 1$), $0 < s \leq 1$ (или $0 \leq s < 1$). Входные данные полигона должны быть представлены ориентированными ребрами либо по возрастанию Y , либо по убыванию. Основная часть алгоритма состоит в том, чтобы в цикле для всех ребер проверить наличие их пересечения с отрезком P_0P (рис. 2) и посчитать эти пересечения. Такая программная реализация приводит к результатам, показанным на рис. 3, $a, б, в$.

На рис. 3, $a, б, в$ серым цветом показана область заданного полигона, а черные полосы — это

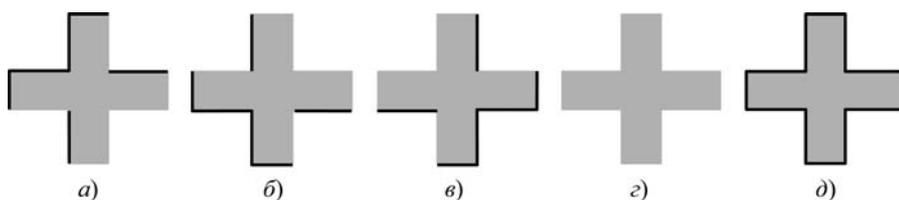


Рис. 3. Закраска фигуры с помощью алгоритма определения принадлежности точки полигону: a — при $0 < r < 1$ и $0 < s \leq 1$; $б$ — при $0 < r < 1$ и $0 \leq s < 1$; $в$ — при $0 \leq r < 1$ и $0 \leq s < 1$; $г$ — корректный результат с включением контура полигона; $д$ — корректный результат без включения контура полигона

пропущенные участки в результате не совсем корректной работы алгоритма. Исходя из рис. 3, а, б, в получается, что если ребра полигона вертикальные или горизонтальные, то они могут либо включаться, либо не включаться при попадании на них проверяемой точки. Для реальных задач геоинформатики такое неприемлемо, здесь могут стоять только две четкие задачи: контур полигона полностью включается в рассмотрение задачи (рис. 3, з) или контур полигона полностью исключается из рассмотрения (рис. 3, д). При этом не стоит забывать, что такого рода алгоритмы в реальных задачах

работают в длинных циклах и желательно позаботиться об оптимальном времени их выполнения.

Самое простое решение, приводящее алгоритм к корректным результатам (рис. 3, з, д), заключается в добавлении процедуры принадлежности точки отрезку. Тогда, используя эту процедуру, в самом начале алгоритма выполняется проверка: попадает ли искомая точка на какое-либо ребро полигона? В случае такого попадания выдается либо результат = "истина" (для варианта рис. 3, з), либо результат = "ложь" (для варианта рис. 3, д) и алгоритм досрочно заканчивает работу.

Нами предлагается альтернативный вариант решения данной проблемы, который более эффективен как с точки зрения размера программного кода, так и времени выполнения алгоритма (в среднем на 15 % быстрее, как показало тестирование), что немаловажно. Решение заключается в некоторой модификации алгоритма проверки наличия пересечения двух отрезков (рис. 4) и модификации основного алгоритма проверки принадлежности точки полигону (рис. 5), учитывающих некорректности частных случаев на горизонтальных и вертикальных ребрах полигона (см. рис. 3, а, б, в).

Модификация алгоритма проверки наличия пересечения двух отрезков (см. рис. 4) состоит во введении дополнительного возвращаемого параметра *Flag* логического типа. Когда *Flag* = "истина" — это означает, что искомая точка попадает на одно из ребер полигона. Блок 5 (см. рис. 4) обрабатывает исключительную ситуацию попадания на горизонтальное ребро, а блок 8 — попадания на остальные ребра.

Модификация алгоритма проверки принадлежности точки полигону представлена на рис. 5. Предполагается, что направленные отрезки полигона *Lines* подготавливаются заранее. Во входных данных X_0 — левая граница луча P_0P (см. рис. 2), т. е. координата *x* для точки P_0 .

Выводы

1. Решена проблема неточного определения наличия пересечения двух отрезков в программной реализации стандартного алгоритма решения задачи. Модификация стандартного алгоритма исключает операцию деления, вследствие которой возникает данная проблема.

2. Рассмотрены некорректности стандартного алгоритма проверки принадлежности точки полигону и предложена модификация данного алгоритма. Предложенный алгоритм является более эффективным как с точки зрения объема программного кода, так и с точки зрения увеличения скорости работы (около 15 %).

СПИСОК ЛИТЕРАТУРЫ

1. Шикин Е. В., Боресков А. В. Компьютерная графика. Полигональные модели. М.: Диалог-МИФИ, 2001. 461 с.
2. Кобзаренко Д. Н., Аскеров С. Я. Внутреннее отсечение триангуляционного объекта, проецируемого на плоскость XY // Геоинформатика. 2007. № 4. С. 26—35.

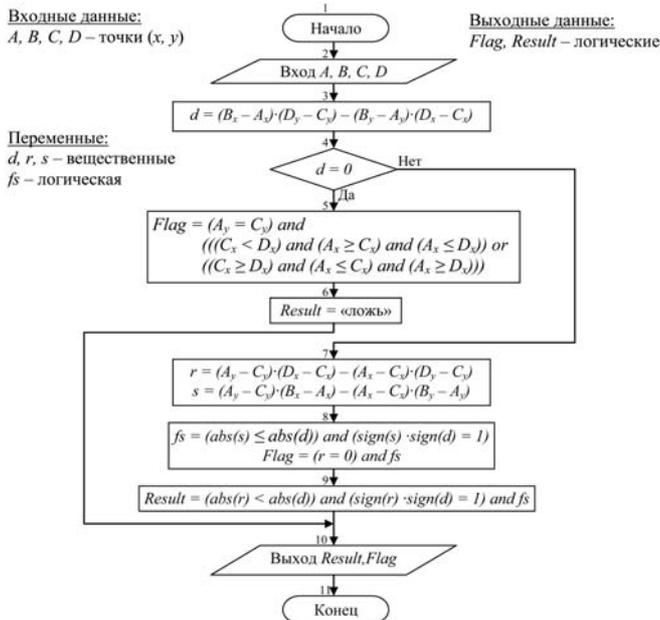


Рис. 4. Модифицированный алгоритм проверки наличия пересечения двух отрезков

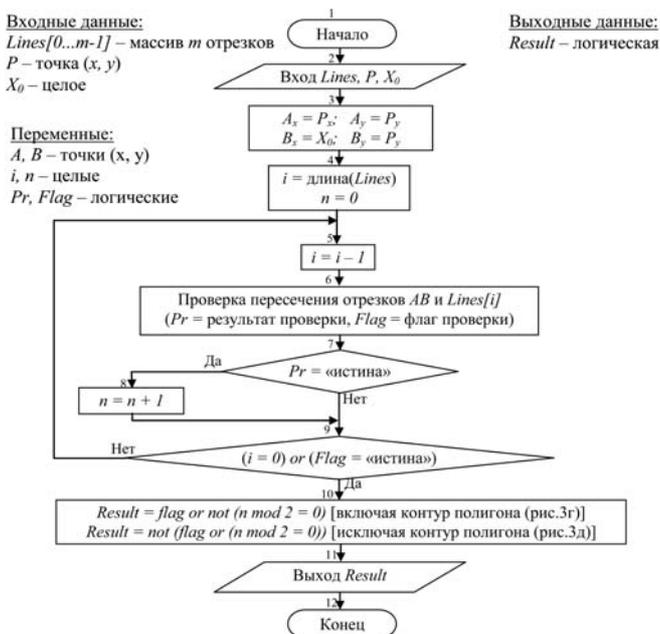


Рис. 5. Модифицированный алгоритм проверки принадлежности точки полигону

УДК 621.396:004.021

Я. Е. Львович, д-р техн. наук, проф.,
И. Я. Львович, д-р техн. наук, проф.,
А. П. Преображенский, канд. физ.-мат. наук, доц.,
С. О. Головинов, аспирант,
 e-mail: GolovinovSO@yandex.ru
 АНОО ВПО Воронежский институт
 высоких технологий

Исследование метода трассировки лучей при проектировании беспроводных систем связи

Приведено описание алгоритма расчета числа лучей распространения электромагнитных волн в условиях городской застройки. По результатам исследования предложенного алгоритма показано уменьшение числа рассматриваемых лучей в точке приема с использованием метода трассировки лучей.

Ключевые слова: метод трассировки лучей, городская застройка, коэффициенты корреляции

В настоящее время для проектирования беспроводных систем связи разработано большое число разного рода методов и алгоритмов, которые применяют для расчета зоны покрытия базовой станции (БС).

В данной работе приведено краткое описание модели [1], а также предложен модуль расчета числа лучей при заданной погрешности расчетов, что позволяет сократить время выполнения вычислений.

Для моделирования передачи данных по беспроводным системам связи примем за $H(f, r)$ функцию распространения сигнала в канале, где f — частота, r — длина пути. Тогда полной функцией среды распространения будет [1]

$$H_t = H_{RR} + H_{RDR}, \quad (1)$$

где H_{RR} и H_{RDR} — полные функции распространения RR и RDR групп лучей соответственно; RR — группа лучей, отражающихся от главной и второстепенных улиц; RDR — группа лучей, которые дифрагируют и отражаются от главной и второстепенных улиц. Далее определим выражения для расчета функций распространения каждой группы.

Функция распространения группы, состоящей из лучей RR между передающей и приемной антеннами, может быть определена [1] по формуле

$$H_{V,H} = \left(\frac{\lambda}{4\pi}\right) \sum_{i \equiv (m, S, n, u, g)} \left[f_B(\theta_i, \phi_i) f_M(\Theta_i, \Phi_i) \times \right. \\ \left. \times (\mathfrak{R}_{V,H}^i)^g (R_{H,V}^{in})^n (R_{H,V}^{im})^m e^{-jkr_i} \right], \quad (2)$$

где λ — длина волны; i -й луч представлен рядом пяти целых чисел (m, S, n, u, g) ; m и n — число отражений от стен вдоль главной и второстепенной улиц, соответственно; $g = 0, 1$ для отражения от земли; k — волновое число; $\mathfrak{R}_{V,H}^i$, $R_{H,V}^{im}$ и $R_{H,V}^{in}$ — известные коэффициенты отражения Френеля от земли и стен на главной и второстепенной улицах, соответственно, с передачей сигнала при вертикальной и горизонтальной поляризации, соответственно; $f_B(\theta, \phi)$ и $f_M(\Theta, \Phi)$ — диаграммы направленности с учетом поляризации для базовой станции (БС) и мобильной станции (МС), соответственно. Углы (θ, Θ) и (ϕ, Φ) — углы в вертикальной и горизонтальной плоскостях для БС и МС, соответственно [1].

У углов зданий есть важная особенность в отклонении сигналов на второстепенные улицы. Дифракция в каждом углу уличного пересечения способствует полному полученному сигналу. Функция распространения лучей, которая соответствует дифракции на четырех углах здания [1],

$$H_{RDR} = H_{C1} + H_{C2} + H_{C3} + H_{C4}, \quad (3)$$

где H_{Cl} — функция распространения лучей, испытывающих дифракцию на угле l . Электрическое поле для луча, полученного в процессе дифракции на углу здания, рассчитывается на основе геометрической теории дифракции (ГТД). Функция распространения из лучей, испытывающих явления отражения — дифракции — отражения перед приемной антенной, может быть вычислена [1] по формуле

$$H_{V,H} = \left(\frac{\lambda}{4\pi}\right) \sum_{i \equiv (m, S, n, u, g, C)} \left[f_B(\theta_i, \phi_i) f_M(\Theta_i, \Phi_i) \times \right. \\ \left. \times (\mathfrak{R}_{V,H}^i)^g (R_{H,V}^{im})^m \overline{D_{V,H}^i} (R_{H,V}^{in})^n \times \right. \\ \left. \times \frac{1}{\sqrt{D_1 D_2 (D_1 + D_2)}} e^{-jk(D_1 - D_2)} \right], \quad (4)$$

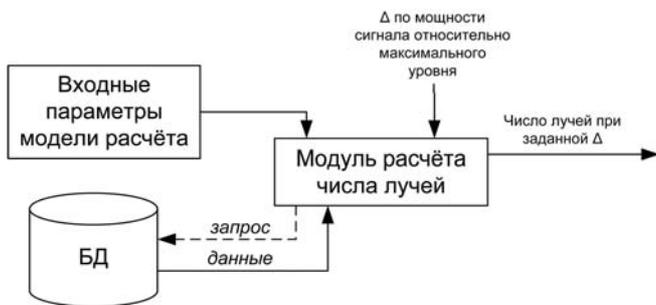


Рис. 1. Структурная схема модуля расчета числа лучей

где $D_{V,H}^i$ — коэффициент дифракции на вертикальном краю угла здания; $D_{1,2}$ — расстояния от БС и МС до места дифракции, соответственно.

Уровень сигнала в данной точке при расчете с использованием данного подхода определяется суммой лучей. При этом число лучей и вносимый уровень сигнала каждым лучом может меняться в зависимости от положения МС, ширины улиц и т. п. В связи с этим, если в приведенной модели учитывать лучи с наибольшим вносимым уровнем сигнала, то можно сократить число итераций при расчетах. Модуль, приведенный на рис. 1, реализует указанный подход, при этом в качестве выходного результата получаем число лучей, при заданной погрешности Δ относительно максимального уровня сигнала в данной точке. В качестве входных параметров модели расчета примем:

- положение БС на главной улице;
- ширину главной и второстепенной улиц;
- координаты углов перекрестка главной и второстепенной улиц;
- координаты МС;
- расстояние от БС до МС.

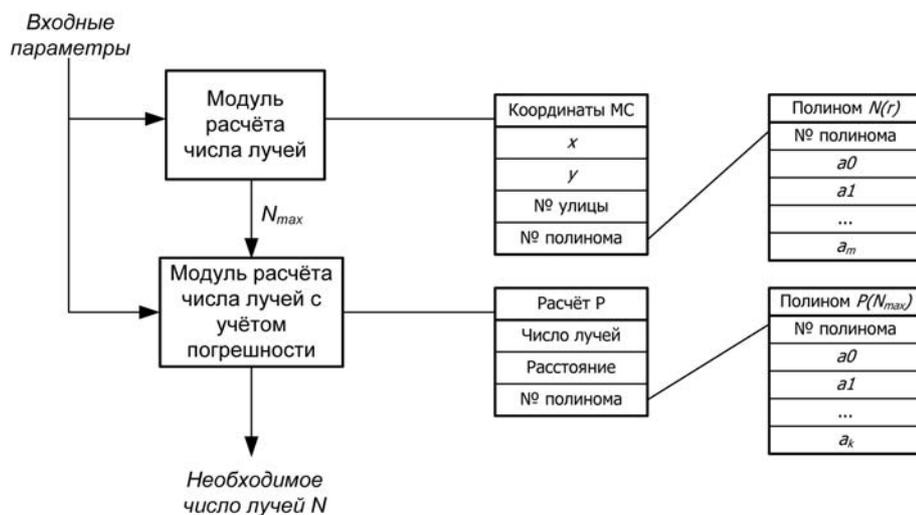


Рис. 2. Функциональная схема модуля расчета числа лучей

БД содержит результаты исследования модели в различных входных параметрах модели. Для уменьшения размерности БД распределение мощности по лучам приводится в виде полинома порядка N .

В качестве дополнительного входного параметра для модуля расчета числа лучей принимается Δ по мощности сигнала относительно максимального уровня.

Функционирование модуля (рис. 2) расчета числа лучей осуществляется в три этапа [2, 3]:

- с использованием коэффициентов корреляции рассчитывается полином зависимости числа лучей от расстояния от БС до МС;
- с использованием коэффициентов корреляции рассчитывается полином зависимости суммарной мощности от мощности, вносимой каждым лучом;
- на основе найденного полинома рассчитывается число лучей с учетом заданной погрешности.

Рассмотрим алгоритм функционирования предложенного модуля.

1. Ввод входных параметров.

2. С использованием коэффициента корреляции рассчитываем номер полинома в БД, зависимости числа лучей от расстояния от БС до МС.

3. По выбранному полиному рассчитываем полное число лучей.

4. С использованием коэффициента корреляции рассчитываем номер полинома в БД зависимости мощности сигнала в точке приема от числа лучей.

5. По выбранному полиному рассчитываем число лучей с учетом введенной погрешности Δ .

6. Вывод числа лучей.

На первом шаге задаются входные параметры, которые впоследствии используются для расчета необходимого числа лучей. Входные параметры можно представить в виде вектора $\alpha_{ВП}$. Элементами данного вектора являются функциональные зависимости

положения БС на главной улице, ширина главной и второстепенной улиц, координаты углов перекрестка главной и второстепенной улиц, координаты МС, расстояния от БС до МС.

Обозначим через $\alpha_{N_{max}}$, $\alpha_{N_{\Delta}}$ векторы-функции, содержащие соответствующие зависимости для элементов, имеющих в БД. Все перечисленные выше векторы имеют одинаковую размерность K .

Процесс нахождения оптимального числа лучей N сводится к выполнению следую-

шего ряда процедур отыскания максимумов соответствующих коэффициентов корреляции [2—7]:

$$k_{N_{\max j}} = \frac{\int_{S_1 S_2 \dots S_L} \alpha_{N_{\max}}(\eta_1, \eta_2 \dots \eta_L) \alpha_{\text{ВП}}(\eta_1, \eta_2 \dots \eta_L) d\eta_1 d\eta_2 \dots d\eta_L}{\sqrt{\int_{S_1 S_2 \dots S_L} \alpha_{N_{\max}}(\eta_1, \eta_2 \dots \eta_L)^2 d\eta_1 d\eta_2 \dots d\eta_L \int_{S_1 S_2 \dots S_L} \alpha_{\text{ВП}}(\eta_1, \eta_2 \dots \eta_L)^2 d\eta_1 d\eta_2 \dots d\eta_L}}, \quad (5)$$

$$k_{N_{\Delta j}} = \frac{\int_{S_1 S_2 \dots S_L} \alpha_{N_{\Delta}}(\eta_1, \eta_2 \dots \eta_L) \alpha_{\text{ВП}}(\eta_1, \eta_2 \dots \eta_L) d\eta_1 d\eta_2 \dots d\eta_L}{\sqrt{\int_{S_1 S_2 \dots S_L} \alpha_{N_{\Delta}}(\eta_1, \eta_2 \dots \eta_L)^2 d\eta_1 d\eta_2 \dots d\eta_L \int_{S_1 S_2 \dots S_L} \alpha_{\text{ВП}}(\eta_1, \eta_2 \dots \eta_L)^2 d\eta_1 d\eta_2 \dots d\eta_L}}, \quad (6)$$

где S_1, S_2, \dots, S_L — области изменения параметров $(\eta_1, \eta_2, \dots, \eta_L)$.

Для каждого из векторов коэффициентов корреляции с помощью стандартной процедуры находятся максимальные элементы, номера которых соответствуют оптимальным номерам полиномов в БД для заданных входных параметров.

На рис. 3 приведен график зависимости уровня сигнала от номера луча в точке приема.

Из рис. 3 видно, что для расчета уровня сигнала в точке приема необходимо учесть около 90 лучей, причем закон распределения вносимой мощности каждого луча — нелинейный. В связи с этим для удобства хранения в БД кривая, приведенная на рис. 3, будет представлена в виде функции. Для аппроксимации приведенной кривой будем использовать метод наименьших квадратов (МНК).

Особенностью, описанной выше модели, является то, что при отдалении МС от БС, число лучей в точке приема увеличивается и, следовательно, увеличивается порядок полинома.

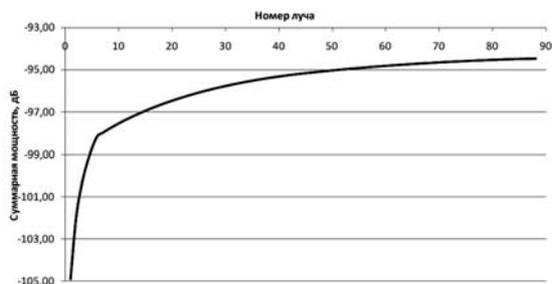


Рис. 3. Суммарный уровень сигнала

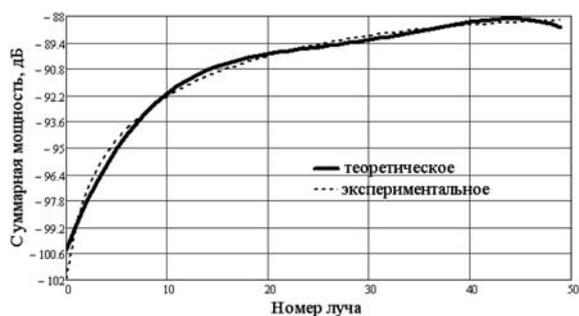


Рис. 4. Аппроксимация зависимости суммарной мощности от номера луча

В качестве критерия оценки порядка полинома выберем разность между экспериментальными и теоретическими значениями, равную 5 %.

На рис. 4 представлены две кривые:

- экспериментальная (на основе значений, полученных при расчете);
- теоретическая (на основе аппроксимации с использованием МНК).

При заданной погрешности (5 %) порядок полинома равен 4, следовательно, хранение в БД только коэффициентов полинома вместо экспериментальных значений, позволяет сократить объем информации.

В соответствии с алгоритмом на шаге 5 по выбранному полиному рассчитывается число лучей с учетом введенной погрешности Δ , относительно максимального уровня. Так, например, для графика, изображенного на рис. 4, с учетом погрешности $\Delta = 2$ дБ, число лучей, учитываемых в точке приема, сокращается с 50 до 15.

В связи с этим предложенный подход с использованием коэффициентов корреляции и аппроксимации МНК позволяет сократить число итераций при расчетах уровня сигнала с использованием метода трассировки лучей. При $\Delta = 2$ дБ число итераций сокращается в 2—3 раза (в зависимости от отдаленности МС).

Список литературы

1. El-Sallabi H. M., Vainikainen P. Radio wave propagation in perpendicular streets of urban street grid for microcellular communications // Progress In Electromagnetics Research. 2003. PIER 40. P. 229—254.
2. Андерсон Т. Введение в многомерный статистический анализ: пер. с англ. М.: Государственное издательство физико-математической литературы, 1963. 499 с.
3. Афифи А., Эйзен С. Статистический анализ: Подход с использованием ЭВМ: пер. с англ. М.: Мир, 1982. 488 с.
4. Вуколов Э. А. Основы статистического анализа. Практикум по статистическим методам и исследованию пакетов STATISTICA и EXEL.: учеб. пособие. — 2-е изд. М.: ФОРУМ, 2008. 464 с.
5. Халафия А. А. STATISTICA 6. Статистический анализ данных. 3-е изд.: учебник. М.: Бинوم-Пресс, 2007. 512 с.
6. Чертыкин Е. М. Статистические методы прогнозирования. Изд. 2-е, пераб. и доп. М.: Статистика, 1977. 200 с.
7. Эфрон Б. Нетрадиционные методы многомерного статистического анализа: сб. статей: пер. с англ. М.: Финансы и статистика, 1988. 263 с.

А. П. Мошевикин, канд. физ.-мат. наук, доц.,
e-mail: alexmou@lab127.karelia.ru,

А. С. Галов, аспирант, программист 1 кат.,
e-mail: suhona@karelia.ru,

А. С. Волков, программист 1 кат.,
e-mail: volk@frgroup.ru

Петрозаводский государственный университет
(ПетрГУ)

Локация в беспроводных сетях датчиков стандарта nanoLOC (IEEE 802.15.4a)

Дано описание технологии nanoLOC™ (IEEE 802.15.4a), применяемой для построения локальных систем позиционирования мобильных объектов. В ней также описаны принципы расчета локации, приведены результаты экспериментов и даны рекомендации по оценке абсолютной точности системы расчета местоположения.

Ключевые слова: беспроводные сети датчиков, локальные системы позиционирования объектов, расчет местоположения, nanoLOC™, IEEE 802.15.4a

Введение

Все системы определения местоположения объектов можно разделить на два класса: глобального позиционирования и локального позиционирования. Наиболее популярными для большинства приложений являются системы глобального позиционирования (GPS, Galileo, ГЛОНАСС), обеспечивающие расчет локации объекта в масштабах всего мира с возможной точностью в несколько метров. Существуют также технологии улучшения точности локации, например, Assisted GPS (A-GPS), WAAS, EGNOS и другие. Основная проблема перечисленных систем — невозможность определения локации внутри помещения без дополнительного дорогостоящего оборудования.

Набирают популярность системы определения локации в сотовой связи (GSM), не требующие использования специального оборудования (необходим лишь сотовый телефон). Такие системы уже можно отнести к классу локальных.

Необходимо заметить, что и глобальные, и локальные системы не всегда демонстрируют удовлетворительную точность расчетов вследствие проблем с определением расстояний в условиях отсутствия прямой видимости между мобильным объектом (МО) и базовыми станциями (БС), которые являются типичными для города.

Для позиционирования внутри помещений существуют системы определения местоположения, использующие лазерную и ультразвуковую аппаратуру. За счет этого достигается высокая точ-

ность (сантиметры), однако ввиду высокой стоимости и сложности инсталляции такие системы не получили достаточного распространения [1].

В последнее время системы определения локации разрабатывают на основе беспроводных технологий передачи данных. В качестве примера таких распространенных стандартов можно привести ZigBee, Bluetooth, WiFi, RFID и др. Эти технологии чаще всего базируются на одном из трех подходов: анализа силы сигнала от БС до объекта (*receive signal strength*, RSS), времени распространения сигнала (*time of flight*, TOF) и угла приема сигнала (*angle of arrival*, AOA).

Согласно выполненному обзору публикаций по данной тематике [1–4] не существует локальной системы позиционирования, которая одновременно обладала бы следующими характеристиками:

- высокая точность локации;
- большая пропускная способность сети (возможность одновременного определения локаций большого числа объектов);
- низкое энергопотребление МО и БС (для автономного использования радиомодулей);
- невысокая стоимость оборудования;
- простота инсталляции системы.

Одним из решений, обеспечивающих высокую точность определения расстояния при минимальном потреблении энергии и высоких скоростях передачи данных, является использование радиотехнологии nanoLOC™ (IEEE 802.15.4a), представленной компанией *Nanotron Technologies GmbH*. На основе этой технологии уже ведутся разработки систем по определению локации МО [5], однако пока они носят исследовательский характер.

Способы расчета локации объектов

В условиях идеального эксперимента (при отсутствии ошибок в измерении расстояний между мобильным объектом и базовыми станциями) для расчета локации объекта достаточно знать положения трех базовых станций (x_1, y_1) , (x_2, y_2) , (x_3, y_3) и трех соответствующих расстояний R_1 , R_2 , R_3 от базовых станции до объекта. Локация объекта (x, y) определяется (рис. 1) из системы уравнений:

$$(x - x_i)^2 + (y - y_i)^2 = R_i^2, \quad i = 1 \dots n.$$

В реальных условиях неизбежно появление ошибок, связанных как с измерениями расстояний, так и с нанесением местоположения БС на карту (привязке их к местности). Последнее становится особенно критичным в случае обеспечения расчетов повышенной точности (около 1–2 м). В англоязычной литературе для систем позиционирования встречаются две оценки точности: *accuracy* (абсолютная точность) и *precision* (кучность).

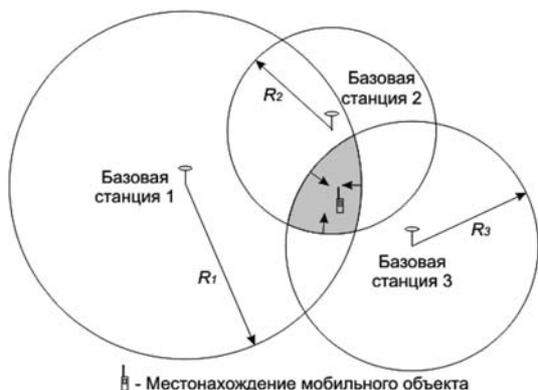


Рис. 1. Определение локаций объекта

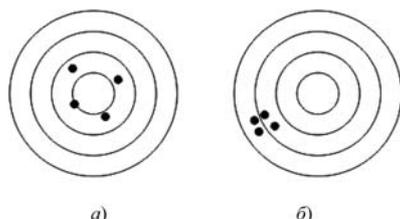


Рис. 2. Кучность и точность

Проиллюстрировать различие между оценками можно на примере двух мишеней. На рис. 2, а обозначена ситуация с плохой кучностью, но хорошей абсолютной точностью, на рис. 2, б — наоборот.

Точность расчета локаций объектов

Точность расчета местоположения МО напрямую связана с процедурой измерения расстояний между двумя объектами. Точность **измерения расстояний** радиометодом зависит от:

- способа измерения расстояний (TOF, RSS, AoA и т. д.) (ΔR_{method});
- аппаратных характеристик стабильности измерений во времени: температуры, электропитания, внешнего электромагнитного окружения (ΔR_{time});
- значений и характера разброса значений параметров среди используемых модулей (ΔR_{var});
- особенностей местности (завышение расстояний из-за переотражений, изменение спектральных характеристик сигнала из-за переотражений) (ΔR_{NLOS});
- возможности учета завышения расстояний вследствие габаритных размеров антенны и длины антенно-фидерных трактов ($\Delta R_{\text{antenna}}$);
- фактического расстояния между объектами (ΔR_{dist});
- мощности радиосигналов (ΔR_{power}).

Суммарную ошибку измерений можно представить следующим образом:

$$\Delta R = \Delta R_{\text{method}} + \Delta R_{\text{time}} + \Delta R_{\text{var}} + \Delta R_{\text{NLOS}} + \Delta R_{\text{antenna}} + \Delta R_{\text{dist}} + \Delta R_{\text{power}} \quad (1)$$

Вклад одних величин носит систематический характер, других — случайный.

Точность результатов **расчета локаций покоящегося объекта** на основе проведенных замеров расстояний зависит также от:

- точности определения реального положения БС на местности;
- учета высоты расположения антенн над поверхностью земли;
- внешних факторов (помех);
- алгоритма расчета;
- времени накопления (в случае использования усреднения оценок по времени).

Кроме этого точность результатов **расчета локаций движущегося объекта** зависит от:

- алгоритма учета предыдущих положений объекта;
- алгоритма расчета возможного следующего местоположения объекта;
- ограничений, налагаемых на вид движения объекта;
- реальных ограничений по перемещению на местности (учет стен).

Систематические ошибки (ошибки, связанные с аппаратной частью радиомодулей и антеннами, расположением БС на местности и т. д.) можно устранить. Основную сложность составляет учет случайных ошибок и ошибок, связанных с природой и условиями распространения электромагнитной волны. Как правило, наиболее весомый вклад в ошибку расчетов вносят измерения, выполненные в условиях отсутствия "прямой видимости" между БС и МО (NLOS, Non-Line-Of-Sight).

Возвращаясь к рис. 1, следует отметить, что существует множество различных алгоритмов определения локаций внутри области пересечения окружностей. В работах [2, 3] рассмотрены преимущества и недостатки основных алгоритмов, основанных на измерении RSS: Simple Point Matching (SPM), Area Based Probability (ABP), Bayesian Network (BN), Point-Based, Bayesian Point, Averaged Bayesian, RADAR (R1), Averaged RADAR (R2), Gridded RADAR (GR), Highest Probability, Averaged Highest Probability, Gridded Highest Probability.

В работе [6] перечислены основные алгоритмы для TOF методов и описаны их ограничения. В случае определения положения покоящегося объекта применяют алгоритмы Non-linear Least Squares NLS, Weighted Non-linear Least Squares, Maximum Likelihood ML, Gaussian ML GML.

В случае движущегося объекта: The Least Mean Square (LMS), Recursive Least Squares (RLS), Extended Kalman Filter (EKF), Particle Filter (PF).

При использовании метода TOF локация объекта определяется по измеренным расстояниям до базовых станций, в связи с чем нет большой необходимости проводить калибровку системы. Кроме того, применение данного метода позволяет использовать меньшее число БС для определения локаций с равной точностью (по сравнению

с системами, основанными на измерении RSS). Однако при реализации метода существует необходимость точной синхронизации источника и приемника сигнала по времени. До недавнего времени не существовало беспроводных технологий локальных сетей, позволяющих провести измерение по методу TOF с точностью до 1 м. Технология nanoLOC™ это позволяет [4].

Общие сведения о радиотехнологии nanoLOC™

Стандарт nanoLOC™ (IEEE 802.15.4a) предложен компанией *Nanotron Technologies GmbH* (<http://www.nanotron.com/>) и применяется для построения беспроводных сетей датчиков с возможностью автоматического измерения расстояний.

Максимальная заявленная скорость передачи данных составляет 2 Мбит/с, несущая частота — 2,4 ГГц (радиочастотный диапазон ISM, разрешенная мощность передатчика до 100 мВт), ширина спектра сигнала — 80 МГц.

Для адресации узлов сети используются 6-байтовые числа (MAC-адреса), которые могут назначаться как централизованно, так и разработчиком приложений этой технологии; возможна также широковещательная рассылка кадров в сети.

Для передачи данных используются электромагнитные сигналы с плавно изменяющейся частотой (CSS-модуляция — *Chirp Spread Spectrum*).

Принцип измерения расстояний в технологии nanoLOC™

Как уже было сказано выше, в технологии nanoLOC™ предусмотрен специальный режим обмена кадрами в эфире, в результате которого радиомодули рассчитывают расстояние друг до друга на основании регистрации времени распространения сигнала. Измерения проводят с помощью разновидности метода TOF под названием Symmetric Double Sided Two Way Ranging (SDS-TWR, симметричное двустороннее двухступенчатое измерение расстояний). Данный метод является усовершенствованием метода *Round Trip Time* (RTT).

Для измерения расстояний методом RTT (рис. 3) [4] между узлом А и узлом В узел А посылает узлу В пакет DATA, содержащий запрос на измерение, и

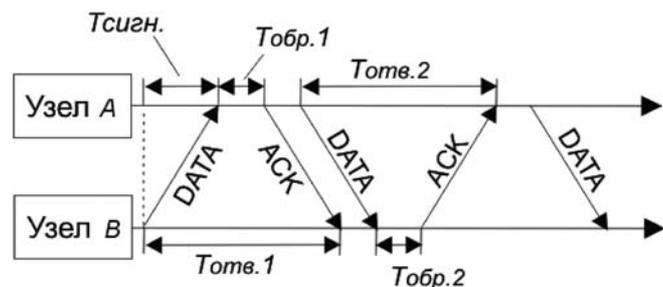


Рис. 3. Схема метода RTT

фиксирует время отправки. Узел В, получив пакет от узла А, отправляет узлу А подтверждение — ACK-пакет. Узел А, получив ACK-пакет, фиксирует время его получения. Метод RTT использует аппаратную генерацию ACK-пакета, где время обработки пакетов считается одинаковым для обоих объектов. Регистрация времени между отправкой пакета, содержащего запрос на измерение, и приемом ACK-пакета происходит также аппаратно. Это позволяет вычислить время распространения сигнала t_p по формуле:

$$t_p = \frac{(T_{RTT} - T_{отв})}{2},$$

где T_{RTT} — время, измеренное узлом А с момента отправки пакета узлу В до получения ACK-пакета от узла В; $T_{отв}$ — время, измеренное узлом В с момента получения пакета от узла А до отправки ACK-пакета.

Считая скорость распространения сигнала в среде известной и постоянной величиной, можно вычислить расстояние между объектами.

В методе SDS-TWR измерение расстояний проводится обеими сторонами, а затем берется среднее значение. Это минимизирует эффект, связанный с рассинхронизацией приемопередатчиков вследствие непредвиденного сдвига центральной частоты у разных МО (например, вследствие температурной разницы, ΔR_{time} и ΔR_{var}).

Результаты экспериментов по точности измерения расстояний между стационарными объектами в условиях прямой видимости

Резюмируя все эксперименты [7, 8], проведенные исследовательским коллективом за последние два-три года, можно привести следующие значения погрешностей при измерении расстояний для технологии nanoLOC™ (в терминах формулы (1)): $\Delta R_{method} \leq 1$ м; $\Delta R_{time} < 0,5$ м; $\Delta R_{var} < 0,5$ м; ΔR_{NLOS} — от 0 до нескольких десятков метров, $\Delta R_{antenna}$ — в зависимости от длины кабеля и габаритных размеров антенны; $\Delta R_{dist} \sim 1$ м; $\Delta R_{power} \sim 1$ м.

Поэтому при расхождении измеренного и реального расстояний на значение менее 2 м можно утверждать, что измерение проведено достаточно точно и его можно корректировать, вычитая известную систематическую погрешность (например, ΔR_{power} или $\Delta R_{antenna}$).

Распознавание ситуации NLOS

Проблема непрямолинейного распространения сигнала (NLOS, Non-Line-Of-Sight) является основной проблемой, приводящей к ошибкам в системах позиционирования объекта, построенных на основе беспроводных сетей датчиков.

Данный эффект приводит к возникновению завышения в измерениях и неправильному расчету локации (ошибка NLOS складывается из дополнительного пути, который прошел сигнал по отно-

шению к прямолинейному пути между передатчиком и приемником).

Основными факторами, приводящими к завышению измеряемых значений, являются препятствия, связанные с геометрией помещения, и экранирование радиоизлучения телом человека.

В большинстве случаев распределение ошибки, связанной с NLOS, зависит от положения объекта. В случае, когда мобильный объект неподвижен или незначительно перемещается около одного положения, ошибка, связанная с NLOS, является постоянной (не изменяется со временем).

Существуют как аппаратные, так и алгоритмические методы распознавания и устранения ошибок, связанных с нелинейным распространением сигнала.

Алгоритмические методы распознавания ошибок сводятся, в основном, к использованию различных модификаций фильтра Калмана [9—11]. Физической основой эффективности данного метода является то, что объект движется по непрерывной траектории, подчиняющейся законам кинематики, а, следовательно, имеющей необходимые условия применения фильтра Калмана. Использование данного метода возможно только в случае частого опроса датчика (координата объекта не должна изменяться слишком сильно за интервал времени между двумя последовательными изменениями), иначе невозможно использовать кинематические уравнения для вычисления передаточной матрицы [12].

На практике могут возникать ситуации, когда возможно проводить лишь редкий опрос датчика. Этот случай, когда нет априорной информации о распределении завышений, связанных с NLOS, и нет информации о движении объекта, является одним из наиболее трудных для распознавания и устранения ошибки. При использовании технологии nanoLOC™ описанные выше методы распознавания NLOS не применимы (в чипе nanoLOC™ не предусмотрена возможность аппаратного распознавания NLOS).

Точность расчета локаций при однократном измерении расстояний

Для оценки локации при однократных измерениях используется один из двух подходов [3]: *point-based* (определение конкретной точки, в которой находится объект, в обозначенных выше терминах — *accuracy*) и *area-based* (определение области нахождения объекта, *precision*).

В случае использовании технологии nanoLOC™ область локации может быть представлена областью пересечения окружностей с центрами в соответствующих базовых станциях, т. е., с одной стороны, локацию удобно описывать с помощью *area-based* критерия, с другой стороны, применяя дополнительные алгоритмы расчета, внутри области можно определить точку вероятного нахождения объекта, т. е. использовать *point-based* величину.

Определить эффективность данных подходов для оценки точности локации можно из эксперимента.

Для случая покоящегося объекта точность результатов расчета локаций может определяться как:

а) зона стопроцентного¹ нахождения объекта (внутри окружности или фигуры, построенной пересечением нескольких сложных кривых);

б) зона вероятного нахождения объекта (ограниченная каким-то уровнем вероятности нахождения объекта внутри нее);

в) статистический разброс расчетов локации относительно центра масс или кучность (наиболее вероятного местонахождения объекта);

г) расстояние между истинным местонахождением объекта на местности и рассчитанным (абсолютная погрешность).

Первые два критерия относятся к категории *area-based*, вторые два — к *point-based*.

Для случая движущегося объекта точность результатов расчета локаций обычно описывают по совпадению траектории движения объекта с рассчитанной траекторией [13].

Проверка точности расчета локаций

В сетях датчиков и локальных системах позиционирования имеет смысл говорить о двумерной (а не 3D) геометрии местности. В таких условиях легко проверить точность расчетов местоположения объекта, если знать с заранее заданной и достаточной точностью места расположения базовых станций и истинное местоположение статического узла, для которого проводятся накопление результатов (рис. 4).

Разница между средним значением, измеренным за некоторый период, и истинным положением радиоузла будет соответствовать значению абсолютной погрешности. Этот способ позволяет оценить точность всей системы позиционирования, включающей в себя аппаратное обеспечение и методы измерения расстояний, сервер и алгоритмы расчета и коррекции местоположения, сетевую инфраструктуру.

Следует отметить, что накопление данных с последующим усреднением в случае неподвижного объекта сравнимо по результату с применением априорной и апостериорной фильтрации данных для подвижного объекта.

Основная проблема в оценке точности состоит в том, что определить абсолютную погрешность системы локации можно только экспериментально

¹ Термин "стопроцентный доверительный интервал", вообще говоря, не корректен. Любое измерение от базовой станции до объекта — неточно. Хотя значения измеренных расстояний в технологии nanoLOC в идеальном случае всегда завышены, существует ненулевая вероятность, что истинное расстояние окажется выше измеренного вследствие ошибок измерений. Кроме того, координаты самих базовых станций известны с некоторой погрешностью и эта погрешность может быть настолько существенной, что может давать истинное расстояние как в минус, так и в плюс, по отношению к измеренному.

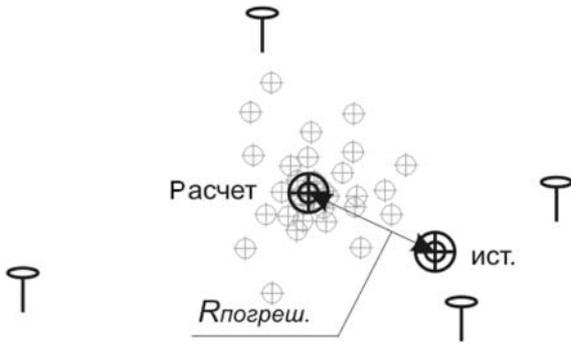


Рис. 4. Проверка точности расчета локаций

и только для конкретных точек проведения замеров. Она может быть вычислена автоматизированной системой только в случае, если есть в наличии данные об истинном местоположении объекта.

Кроме эксперимента оценку абсолютной точности позиционирования может дать совмещение радиотехнологии с другой измерительной системой, например системой анализа изображений с видеокамер.

Методика экспериментальной оценки абсолютной точности системы локации

Проведенные авторами эксперименты показали, что абсолютная точность расчетов местоположения МО сильно зависит от окружающей среды и места проведения эксперимента.

Для оценки абсолютной точности локации предлагается выполнить следующие действия и соблюсти следующие условия.

1. Базовые точки расставить на местности, их положение нанести на карту.

2. Число экспериментальных точек, в которых будут проводить замеры, должно превышать 20.

Места экспериментов определять следующим образом:

а) 5 разных точек в условиях открытого пространства и прямой видимости до 4—5 базовых станций, с минимальным расстоянием до ближайших базовых станций не менее 30 м;

б) 5 разных точек в условиях открытого пространства и прямой видимости до 4—5 базовых станций, с минимальным расстоянием до ближайшей базовой станции не более 10 м;

в) 5 разных точек в условиях открытого пространства и прямой видимости 2—3 базовых станций (с расстояниями до них в диапазоне 70...200 м);

г) 5 разных точек внутри здания сложной геометрии.

3. Для каждой точки проводить накопление результатов 60 измерений для каждого мобильного узла (5 мин, если измерения проводятся один раз в 5 с).

4. На основании собранных данных путем усреднения полученных оценок определить точность локации всей системы.

Резюмируя сказанное выше, можно утверждать, что, используя уникальные аппаратные характеристики технологии nanoLOC™, можно разработать систему локального позиционирования мобильных объектов с плотностью расположения базовых станций около 5—7 шт./га для открытого пространства и средней абсолютной точностью позиционирования 2—3 м.

В условиях сильно пересеченной местности, в закрытых помещениях, плотность размещения базовых станций должна быть увеличена в несколько раз для сохранения той же точности локаций.

Авторы благодарят Гордеева Д. Е. за финансовое обеспечение проекта и Сураева А. И. за высказанные критические замечания при подготовке статьи.

Данное исследование проведено в рамках работы над проектами компании ООО "РТЛ-Сервис", а также при поддержке ИТ-парка Петрозаводского государственного университета и научно-образовательного центра по фундаментальным проблемам приложений физики низкотемпературной плазмы (проект RUX0-000013-PZ-06, Министерство образования и науки РФ, Правительство Республики Карелия, CRDF).

Список литературы

1. **Rainer M.** Overview of Current Indoor Positioning Systems // Geodesy and Cartography. 2009. N 35 (1). P. 18—22.
2. **Chandrasekaran G., Ergin M.** et al. Empirical Evaluation of the Limits on Localization Using Signal Strength // Proc. of IEEE SECON. June 2009.
3. **Elnahrawy E., Li X.** et al. The Limits of Localization Using Signal Strength: A Comparative Study // Proc. of IEEE SECON. Oct. 2004.
4. **Real Time Location Systems (RTLS).** Nanotron Technologies GmbH, Berlin, Germany. White paper NA-06-0248-0391-1.02, Apr. 2007.
5. **Muller R.** Localization of Sensor Nodes in a Wireless Sensor Network Using the nanoLOC TRX Transceiver // Proc. of VTC 2009. Apr. 2009.
6. **Patwari N., Ash J.** et al. Locating the nodes: cooperative localization in wireless sensor networks // Signal Processing Magazine, IEEE. 2005. Vol. 22, N 4. P. 54—69.
7. **Дмитриев С., Екимов К.** и др. Изучение возможности применения технологии nanoLOC // Беспроводные технологии. 2008. № 3 (12). С. 52—56.
8. **Гоголев А., Екимов Д.** и др. Точность определения расстояний с помощью технологии nanoLOC // Беспроводные технологии. 2008. № 3 (12). С. 48—51.
9. **Röhrig C., Müller M.** Indoor Location Tracking in Non-line-of-Sight Environments Using a IEEE 802.15.4a Wireless Network // Proc. of the IEEE/RSJ (IROS 2009), USA, Oct. 2009, pp. 552—557.
10. **Bosch M., Najar M.** Unscented Kalman Filter for location in Non-Line-Of-Sight // Proc. of European Signal Processing Conference (EUSIPCO). Italy, Sept. 2006.
11. **Le B., Ahmed K.** et al. Mobile location estimator with NLOS mitigation using Kalman filtering // Proc. of IEEE Wireless Communications and Networking Conf. Mar. 2003. Vol. 3. P. 1969—1973.
12. **Julier S., Uhlmann J.** Unscented Filtering and Nonlinear Estimation // Proc. of the IEEE. Mar. 2004. Vol. 92, N 3. P. 401—422.
13. **Röhrig C., Spieker S.** Tracking of Transport Vehicles for Warehouse Management using a Wireless Sensor Network // Proc. of IEEE/RSJ (IROS 2008), France. Sep. 2008. P. 3260—3265.

УДК 004.424.72:519.863

Н. А. Гильманова, аспирант,
Уфимский государственный авиационный
технический университет,
e-mail: hopediesthelast@yahoo.com

Мультиметодный алгоритм трехмерной упаковки с регулируемым провисанием объектов в контейнере*

Представлен мультиметодный алгоритм решения задачи трехмерной упаковки контейнера с учетом дополнительных условий из практики хранения и перевозки грузов. С помощью численного эксперимента устанавливается зависимость качества получаемого решения от сочетаний различных параметров алгоритма. Показано, каким образом путем комбинирования эвристик можно получать рациональные решения за короткое время.

Ключевые слова: трехмерная упаковка, мультиметодная технология, комбинирование эвристик

Введение

Задача, рассматриваемая в данной статье, относится к широкому классу задач раскроя-упаковки и заключается в поиске способа размещения множества объектов-параллелепипедов внутри контейнера-параллелепипеда. Она включает в себя условия, характерные для абстрактного размещения параллелепипедов и реального размещения ящиков, коробок, паллет на складах или в транспортных средствах.

1. Расположение объектов в контейнере должно быть ортогональным и объекты не должны перекрываться между собой и с гранями контейнера.

2. Набор допустимых ориентаций устанавливается для каждого объекта индивидуально, поскольку на практике размещаемые объекты могут иметь различные маркировки и приспособления для погрузки-разгрузки на одной из сторон (чем обуславливается запрет на их поворот); кроме того, может быть запрещено переворачивание некоторых объектов.

3. В абстрактных задачах пакуемые объекты могут соприкасаться с другими на ничтожную долю

* Работа выполнена при поддержке РФФИ (грант № 10-07-91330-ННИО_а).

площади дна, фактически "висеть в воздухе". В реальных условиях это невозможно: грузы должны располагаться устойчиво. В упрощенном виде это требование сводится к тому, что дно каждого объекта должно иметь под собой твердую опору достаточной площади. Для этого вводится учет **коэффициента непровисания** — минимальной доли площади дна объекта, которая не должна провисать.

При поиске схемы размещения объектов преследуются различные цели: поиск набора объектов с максимальной ценностью, который можно загрузить в один контейнер заданных размеров [1]; проверка возможности загрузки заданного набора объектов в один контейнер [2]; минимизация длины занятой части контейнера; поиск минимального числа контейнеров, в которые можно загрузить заданный набор объектов [3] и т. д. Описанный в настоящей статье подход к построению трехмерной схемы размещения объектов не имеет строгой привязки к какой-либо из этих целей, т. е. он может быть использован для решения разных задач трехмерной упаковки. В основе подхода лежит **мультиметодная технология (ММТ)**; ее идея в виде комбинирования эвристик была предложена в 1999 г. И. П. Норенковым для решения различных задач проектирования [4]. Детализация подхода для задачи двумерной упаковки в виде мультиметодного алгоритма приведена в статье [5]. В работе [6] описано расширение алгоритма для задачи трехмерной упаковки, но в данной версии алгоритма запрещались любые провисания, что приводило к снижению плотности упаковки, а исследование эффективности алгоритма не проводилось.

В настоящей работе подробно описан мультиметодный алгоритм трехмерного размещения объектов (3D-MMA) с дополнительными ограничениями; экспериментально исследована его эффективность на различных классах задач при различных целевых функциях.

1. Постановка задачи

1.1. Описание задачи. Имеется m объектов-параллелепипедов, каждый из которых характеризуется длиной l_i , шириной w_i и высотой h_i , $i = \overline{1, m}$. Для каждого объекта задан вектор $r_i = (r_{i1}, \dots, r_{i6})$, где $r_{ij} \in \{0, 1\}$ означает запрет или разрешение на расположение объекта в каждой из шести ориентаций, отвечающих требованию ортогональности. Задан коэффициент непровисания a . Пусть даль-

няя нижняя левая вершина контейнера с длиной L , шириной W и высотой H совпадает с началом координат, а исходящие из нее ребра лежат соответственно на осях Ox , Oy и Oz . Поиск допустимого решения заключается в определении возможности $p_i \in \{0, 1\}$ размещения в контейнере каждого объекта $i = \overline{1, m}$, а также в отыскании координат (x_i, y_i, z_i) дальних нижних левых вершин упакованных объектов и их проекций $(d_i^x; d_i^y; d_i^z)$ на оси координат (проекции характеризуют ориентацию объекта, рис. 1).

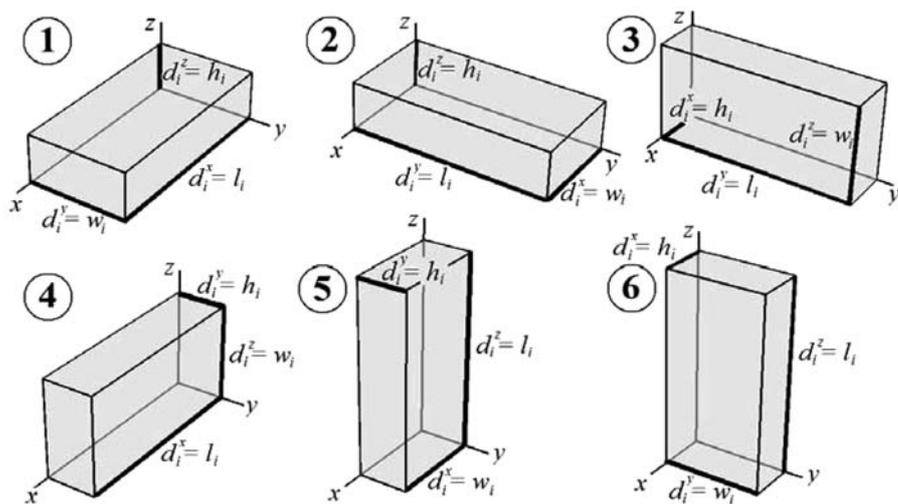


Рис. 1. Возможные ориентации объекта i с длиной l_i , шириной w_i и высотой h_i

Рассматриваются две возможные цели решения задачи:

а) максимизация коэффициента упаковки, рассчитываемого как отношение суммарного объема упакованных объектов к объему контейнера;

б) минимизация длины занятой части контейнера (вдоль оси Ox) — для сравнения двух схем размещения с одинаковыми коэффициентами упаковки.

1.2. Математическая модель. Требуется найти размещение объектов в контейнере, доставляющее оптимум следующим целевым функциям:

а) максимизация коэффициента упаковки

$$PC = \frac{\sum_{i=\overline{1, m}} p_i d_i^x d_i^y d_i^z}{LWH};$$

б) минимизация длины занятой части контейнера

$$\Lambda = \max_{i=\overline{1, m}} p_i (x_i + d_i^x)$$

при следующих условиях:

1) допустимая ориентация упакованных объектов $i = \overline{1, m} : p_i = 1$

$$\left[\begin{array}{l} r_{i1} = 1 \wedge d_i^x = l_i \wedge d_i^y = w_i \wedge d_i^z = h_i \\ r_{i2} = 1 \wedge d_i^x = w_i \wedge d_i^y = l_i \wedge d_i^z = h_i \\ r_{i3} = 1 \wedge d_i^x = h_i \wedge d_i^y = l_i \wedge d_i^z = w_i \\ r_{i4} = 1 \wedge d_i^x = l_i \wedge d_i^y = h_i \wedge d_i^z = w_i \\ r_{i5} = 1 \wedge d_i^x = w_i \wedge d_i^y = h_i \wedge d_i^z = l_i \\ r_{i6} = 1 \wedge d_i^x = h_i \wedge d_i^y = w_i \wedge d_i^z = l_i \end{array} \right.$$

2) *неперекрывание* упакованных объектов $i, j = \overline{1, m} : i \neq j; p_i = p_j = 1$ между собой

$$\begin{aligned} x_i \geq x_j + d_j^x \vee y_i \geq y_j + d_j^y \vee z_i \geq z_j + d_j^z \vee x_j \geq \\ \geq x_i + d_i^x \vee y_j \geq y_i + d_i^y \vee z_j \geq z_i + d_i^z; \end{aligned}$$

3) *неперекрывание* упакованных объектов $i = \overline{1, m} : p_i = 1$ с *гранями* контейнера

$$\begin{aligned} x_i \geq 0 \wedge y_i \geq 0 \wedge z_i \geq 0 \wedge x_i + d_i^x \leq \\ \leq L \wedge y_i + d_i^y \leq W \wedge z_i + d_i^z \leq H; \end{aligned}$$

4) *провисание* упакованных объектов $i = \overline{1, m} : p_i = 1$ на *долю* дна не более a

$$\left[\begin{array}{l} z_i = 0, \\ \forall j = \overline{1, m} : p_j = 1 \wedge z_j + d_j^z = \\ = z_i \wedge x_i < x_j + d_j^x \wedge x_j < x_i + d_i^x \wedge \\ y_i < y_j + d_j^y \wedge y_j < y_i + d_i^y \wedge z_i < z_j + d_j^z \wedge z_j < z_i + d_i^z; \\ \sum_j (\min\{x_i + d_i^x, x_j + d_j^x\} - \max\{x_i, x_j\}) \times \\ \times (\min\{y_i + d_i^y, y_j + d_j^y\} - \max\{y_i, y_j\}) \geq a d_i^x d_i^y. \end{array} \right.$$

2. Описание алгоритма

2.1. Общая схема работы алгоритма. Идея решения задач дискретной оптимизации путем комбинирования нескольких простых эвристик возникла достаточно давно. На западе такой подход получил название **гипер-эвристик** (см. обзор [7]) и предполагает наличие "эвристики верхнего уровня", которая управляет применением эвристик нижнего уровня. В России подход с комбиниро-

ванием эвристик описан в работе И. П. Норенкова [4]. Применительно к задачам раскроя-упаковки подход реализован в виде мультиметодной технологии [5], предполагающей применение простых эвристик с заданными вероятностями на каждом этапе построения упаковки. При этом каждая простая эвристика содержит оптимизационный механизм, требуя установки объекта, наиболее соответствующего некоторому критерию.

Мультиметодный алгоритм в своей основе имеет идею разбиения задачи размещения набора объектов на множество подзадач установки отдельных объектов; каждая подзадача решается с помощью некоторой элементарной эвристики. При решении каждой подзадачи последовательно выполняются следующие шаги: 1) выбор точки для установки очередного предмета; 2) выбор предмета и его размещение (если такой предмет найден); 3) поиск новых точек установки. Шаги выполняются до тех пор, пока не будут упакованы все объекты либо пока не будет установлена невозможность их дальнейшего размещения.

В следующих подразделах подробно описан каждый шаг алгоритма. Однако до перехода к шагу 1 целесообразно выполнить некоторые действия по обеспечению в дальнейшем более быстрой работы основных процедур алгоритма (**препроцессинг**).

Препроцессинг. Построение упорядоченных списков ориентированных объектов. Рассмотрим два неодинаковых параллелепипеда. Предположим, что они имеют фиксированную ориентацию. Существует шесть типов "жадных" правил, согласно которым можно сделать однозначный выбор одного из этих двух параллелепипедов, каковы бы ни были их измерения (Ребро₁, Ребро₂, Ребро₃) и образуемые ими грани (Грань₁₋₂, Грань₁₋₃, Грань₂₋₃),

а также объем. Каждое правило заключается в последовательном сравнении объектов по трем критериям. Общий вид этих типов правил:

1. $\max \text{Ребро}_1 \rightarrow \max \text{Ребро}_2 \rightarrow \max \text{Ребро}_3$;
2. $\max \text{Ребро}_1 \rightarrow \max \text{Грань}_{2-3} \rightarrow \max \text{Ребро}_2$;
3. $\max \text{Грань}_{1-2} \rightarrow \max \text{Ребро}_1 \rightarrow \max \text{Ребро}_3$;
4. $\max \text{Грань}_{1-2} \rightarrow \max \text{Ребро}_3 \rightarrow \max \text{Ребро}_1$;
5. $\max \text{Объем} \rightarrow \max \text{Ребро}_1 \rightarrow \max \text{Ребро}_2$;
6. $\max \text{Объем} \rightarrow \max \text{Грань}_{1-2} \rightarrow \max \text{Ребро}_1$.

Например, если используется правило типа 5, то нужно выбрать предмет большего объема; если объемы одинаковы, то необходимо выбрать предмет, у которого больше одно из ребер (вдоль одной из осей координат); если и соответствующие ребра равны, то необходимо сравнить ребра вдоль другой оси координат. Если же и эти ребра одинаковы, то третьи ребра окажутся одинаковыми автоматически, и объекты идентичны. Таким образом, правило любого другого вида либо будет приводить к неоднозначному выбору объекта, либо будет содержать избыточность. Перечисленные шесть типов правил далее используются в качестве элементарных эвристик выбора объекта для установки в свободную область. С учетом шести возможных ориентаций общее число эвристик выбора объекта составляет 36. Ориентация эвристики, при которой $\text{Ребро}_1 = d^x$, $\text{Ребро}_2 = d^y$, $\text{Ребро}_3 = d^z$, считается исходной; соответствующие эвристики назовем базовыми. Прочие наборы эвристик назовем поворотными. Все эвристики перечислены в табл. 1. Каждая эвристика позволяет *однозначно* выбрать объект и его ориентацию. Непосредственно **препроцессинг** заключается в том, что до на-

Таблица 1

Эвристики выбора объекта

Тип эвристики	Ориентация эвристик (1 — "базовые" эвристики; 2–6 — "поворотные" эвристики)		
	1	2	3
1	$\max d^x \rightarrow \max d^y \rightarrow \max d^z$	$\max d^y \rightarrow \max d^x \rightarrow \max d^z$	$\max d^z \rightarrow \max d^x \rightarrow \max d^y$
2	$\max d^x \rightarrow \max d^y d^z \rightarrow \max d^y$	$\max d^y \rightarrow \max d^x d^z \rightarrow \max d^x$	$\max d^z \rightarrow \max d^x d^y \rightarrow \max d^x$
3	$\max d^x d^y \rightarrow \max d^x \rightarrow \max d^z$	$\max d^x d^y \rightarrow \max d^y \rightarrow \max d^z$	$\max d^x d^z \rightarrow \max d^z \rightarrow \max d^y$
4	$\max d^x d^y \rightarrow \max d^z \rightarrow \max d^x$	$\max d^x d^y \rightarrow \max d^z \rightarrow \max d^y$	$\max d^x d^z \rightarrow \max d^y \rightarrow \max d^z$
5	$\max d^x d^y d^z \rightarrow \max d^x \rightarrow \max d^y$	$\max d^x d^y d^z \rightarrow \max d^y \rightarrow \max d^x$	$\max d^x d^y d^z \rightarrow \max d^z \rightarrow \max d^x$
6	$\max d^x d^y d^z \rightarrow \max d^x d^y \rightarrow \max d^x$	$\max d^x d^y d^z \rightarrow \max d^x d^y \rightarrow \max d^y$	$\max d^x d^y d^z \rightarrow \max d^x d^z \rightarrow \max d^x$
Тип эвристики	Ориентация эвристик (1 — "базовые" эвристики; 2–6 — "поворотные" эвристики)		
	4	5	6
1	$\max d^x \rightarrow \max d^z \rightarrow \max d^y$	$\max d^y \rightarrow \max d^z \rightarrow \max d^x$	$\max d^z \rightarrow \max d^y \rightarrow \max d^x$
2	$\max d^x \rightarrow \max d^y d^z \rightarrow \max d^z$	$\max d^y \rightarrow \max d^x d^z \rightarrow \max d^z$	$\max d^z \rightarrow \max d^x d^y \rightarrow \max d^y$
3	$\max d^x d^z \rightarrow \max d^x \rightarrow \max d^y$	$\max d^y d^z \rightarrow \max d^y \rightarrow \max d^x$	$\max d^y d^z \rightarrow \max d^z \rightarrow \max d^x$
4	$\max d^x d^z \rightarrow \max d^y \rightarrow \max d^x$	$\max d^y d^z \rightarrow \max d^x \rightarrow \max d^y$	$\max d^y d^z \rightarrow \max d^x \rightarrow \max d^z$
5	$\max d^x d^y d^z \rightarrow \max d^x \rightarrow \max d^z$	$\max d^x d^y d^z \rightarrow \max d^y \rightarrow \max d^z$	$\max d^x d^y d^z \rightarrow \max d^z \rightarrow \max d^y$
6	$\max d^x d^y d^z \rightarrow \max d^x d^z \rightarrow \max d^x$	$\max d^x d^y d^z \rightarrow \max d^y d^z \rightarrow \max d^y$	$\max d^x d^y d^z \rightarrow \max d^y d^z \rightarrow \max d^z$

Таблица 2

Данные для определения ориентации объекта

		Ориентация эвристики					
		1	2	3	4	5	6
Ориентация объекта на базовой эвристике	1	1	2	5	4	3	6
	2	2	1	4	5	6	3
	3	3	4	1	6	5	2
	4	4	3	6	1	2	5
	5	5	6	3	2	1	4
	6	6	5	2	3	4	1

чала работы основного алгоритма список всех объектов во всех шести ориентациях упорядочивается в соответствии с описанными эвристиками. Это позволяет сэкономить время на поиск подходящего объекта. Не нужно упорядочивать список ориентированных объектов согласно всем 36 эвристикам — достаточно сделать это для шести "базовых". Для прочих эвристик список номеров объектов будет тем же, меняться будут лишь ориентации объектов.

Определить номер ориентации объекта для каждой поворотной эвристики, зная его ориентацию согласно базовой эвристике, можно с помощью табл. 2. Числа в данной таблице образуют "латинский квадрат": во всех строках и столбцах цифры не повторяются; при этом по диагоналям расположены одинаковые цифры.

Например, пусть список из двух объектов упорядочен в соответствии с базовой эвристикой 1.2 ($\max d^x \rightarrow \max d^y d^z \rightarrow \max d^y$) и имеет следующий вид (используется обозначение "объект^{ориентация}"):

$$1^3 - 1^4 - 2^1 - 2^5 - 1^5 - 2^3 - 2^4 - \\ - 1^6 - 1^2 - 1^1 - 2^6 - 2^2.$$

Тогда упорядочение согласно "поворотной" эвристике 3.2 будет следующим:

$$1^1 - 1^6 - 2^5 - 2^3 - 1^3 - 2^1 - 2^6 - \\ - 1^2 - 1^4 - 1^5 - 2^2 - 2^4.$$

При препроцессинге необходимо составить списки объектов во всех шести ориентациях, упорядоченные согласно шести "базовым" эвристикам (итого 6 списков, в каждом из которых $6m$ пар "объект^{ориентация}").

Алгоритм 3D-ММА состоит из выполнения следующих шагов.

Шаг 1. Выбор точки установки. При конструировании упаковки на каждом шаге определяется множество точек-кандидатов для размещения в них очередного предмета дальним левым нижним углом. У разных авторов эти точки называются нормальными позициями [2], допустимыми точками установки [8], угловыми точками и т. д. Как правило, такие точки образуются на пересечении трех взаимно перпендикулярных плоскостей, являющихся стенками контейнера либо гранями

уже установленных объектов. Будем считать, что определение точек установки не привязывается к структуре упаковки; такими точками являются некоторые позиции, в которые можно установить очередной предмет. В начале работы алгоритма есть только одна точка установки: $(0, 0, 0)$. При выполнении шага 3 множество точек установки обновляется. В рамках шага 1 точка для размещения очередного объекта выбирается из множества точек установки согласно одной из элементарных эвристик, приводящих к однозначному выбору точки, ближайшей к какой-либо из граней контейнера, к дальнему ребру или к началу координат.

Мы использовали следующий набор эвристик:

1. $\min x \rightarrow \min z \rightarrow \min y$;
2. $\min y \rightarrow \min x \rightarrow \min z$;
3. $\min z \rightarrow \min x \rightarrow \min y$;
4. $\min(x^2 + z^2) \rightarrow \min x \rightarrow \min y$;
5. $\min x \rightarrow \min y \rightarrow \min z$;
6. $\min(x^2 + y^2 + z^2) \rightarrow \min x \rightarrow \min z$.

Шаг 2. Выбор объекта для размещения. После выбора точки необходимо выбрать предмет для установки в эту точку с соблюдением перечисленных в разд. 1 ограничений. В целях максимизации плотности упаковки принимается правило: *предмет должен быть в некотором смысле наибольшим из вмещающихся в свободную область и не слишком провисающих*. Для выбора каждого очередного объекта используется одна из эвристик, перечисленных в табл. 1. При наличии построенных при препроцессинге упорядоченных списков ориентированных объектов и выбранной на шаге 1 точки установки порядок действий по выбору объекта для размещения следующий:

- 1) выбрать эвистику (табл. 1), установить ее тип и ориентацию;
- 2) просматривая список ориентированных объектов, соответствующий данному типу эвристики, в порядке убывания соответствия пар "объект^{ориентация}" данному типу эвристики, выбрать первый неупакованный объект, который находится в допустимой ориентации согласно табл. 2 и условию 1, при установке в выбранную точку не перекрывается с другими объектами и стенками контейнера (условия 2, 3) и не слишком сильно провисает, т. е. имеет твердую опору под установленной долей площади дна (условие 4).

В результате просмотра списка объектов может оказаться, что в текущую точку установки и соответствующую ей свободную область невозможно установить ни один объект ни в одной из ориентаций. Вне зависимости от того, найден ли предмет для установки, следующей процедурой будет поиск новых точек установки.

Шаг 3. Поиск новых точек установки. В случае установки нового предмета точки определяются путем просмотра плоскостей верхней, правой и

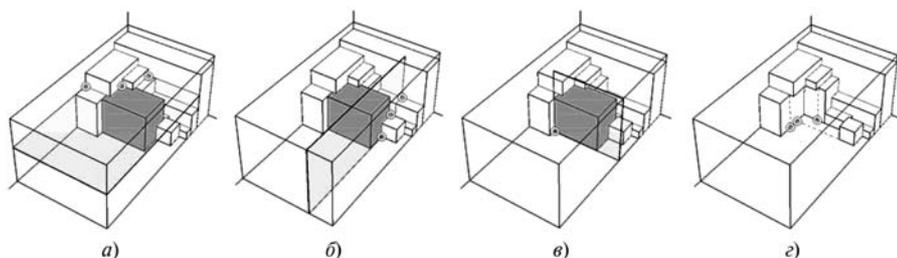


Рис. 2. Поиск новых точек установки:

a — в плоскости верхней грани нового объекта; *b* — в плоскости правой грани нового объекта; *c* — в плоскости передней грани нового объекта; *d* — объект для установки не найден

передней граней этого предмета, как показано на рис. 2, *a–c*. Во множество точек установки добавляются точки, находящиеся на стыке трех твердых поверхностей (граней ящиков или стенок контейнера) и находящиеся "перед обрывом" (слева или сзади от которых — пустое пространство). Если объект для установки найден не был, новые точки определяются иным способом, который проиллюстрирован на рис. 2, *d*. Для этого просматриваются ортогональные плоскости, проходящие через текущую точку установки. При наличии нависающих над свободной областью объектов, а также объектов, ограничивающих свободную область справа или спереди, новые точки установки располагаются в плоскостях их передних и правых граней.

Из множества точек установки необходимо исключить повторяющиеся точки и точки, перекрытые вновь установленным объектом. Важно, чтобы алгоритм поиска новых точек установки учитывал как можно больше возможных ситуаций и не допускал появления в схеме размещения пустот вследствие отсутствия в этих областях точек установки.

2.2. Внешний поиск рациональных решений.

Описанный алгоритм трехмерного размещения объектов вследствие случайного выбора элементарных эвристик может давать совершенно различные результаты. ММТ разумно использовать в рамках метаэвристики, которая за короткое время сможет перебрать множество схем размещения и выбрать наилучшую. Простейшая метаэвристика — одноточечный эволюционный алгоритм (1 + 1)-EA [9], сравнивающий полученное на каждой итерации решение с текущим рекордом. Применительно к ММТ данный алгоритм обозначается (1 + 1)-ММЕА [6]. В разд. 2.1 перечислено 6 эвристик выбора точки установки и 36 эвристик выбора объекта, работа которых осуществляется на основе ММТ. Какая из эвристик сработает на очередном этапе построения упаковки, определяется случайно. Однако не все эвристики одинаково эффективны. Поэтому одним из элементов ММТ является механизм дискриминации эвристик, т. е. выставления им различных вероятностей применения. Установить эти вероятности можно экспертно, но более правильным способом является численный эксперимент. Далее с помощью предварительного

численного эксперимента определяется эффективность отдельных эвристик, а затем проводится дискриминация и исследуется эффективность окончательного алгоритма.

3. Численные эксперименты

Для выявления аспектов работы алгоритма трехмерной упаковки (1 + 1)-ММЕА было проведено его обширное тестирование на задачах разного типа. Преследовались следующие цели: сравнить эффективность различных комбинаций отдельных эвристик выбора точки и выбора объекта; выявить влияние разрешения/запрета поворотов и переворачиваний объектов на качество полученного решения; установить зависимость эффективности работы алгоритма от заданного коэффициента непровисания; определить, как соотносятся эффективности различных конфигураций алгоритма при решении задач загрузки контейнера (при максимизации коэффициента упаковки) и задач упаковки в полубесконечный контейнер (при минимизации длины занятой части контейнера). Численные эксперименты проводили на примерах, сгенерированных по методике Neminki [10] и имеющих типичные свойства производственных задач загрузки. Для каждого примера объекты генерируют до тех пор, пока их суммарный объем не превысит 90 % объема контейнера с длиной 590, шириной 230 и высотой 230. Сгенерировано по 50 примеров для трех классов задач.

1. Сильно разнородные задачи (*strongly heterogeneous instances*): генерируются отдельные объекты с длиной, шириной и высотой, равномерно распределенными на интервале [25, 115]. Полученные примеры содержат от 70 до 104 объектов (в среднем 85,36 объекта).

2. Слабо разнородные задачи (*weakly heterogeneous instances*): сначала генерируется 20 различных типов объектов; для каждого типа объекта длину, высоту и ширину выбирают с равной вероятностью из интервала [25, 115]. Затем последовательно генерируют объекты путем случайного выбора из 20 типов с равной вероятностью. Полученные таким образом примеры содержат от 56 до 117 объектов (в среднем 84,4 объекта).

3. Однородные задачи (*homogeneous instances*): имеется лишь один тип объектов с длиной, шириной и высотой, случайно выбранными из интервала [25, 115]. В этом случае разброс очень большой: сгенерированные примеры содержат от 22 до 408 объектов (в среднем 135,28 объекта).

Эксперимент № 1. Сравнение результатов работы алгоритма при различных комбинациях отдельных эвристик. Исследовали результаты рабо-

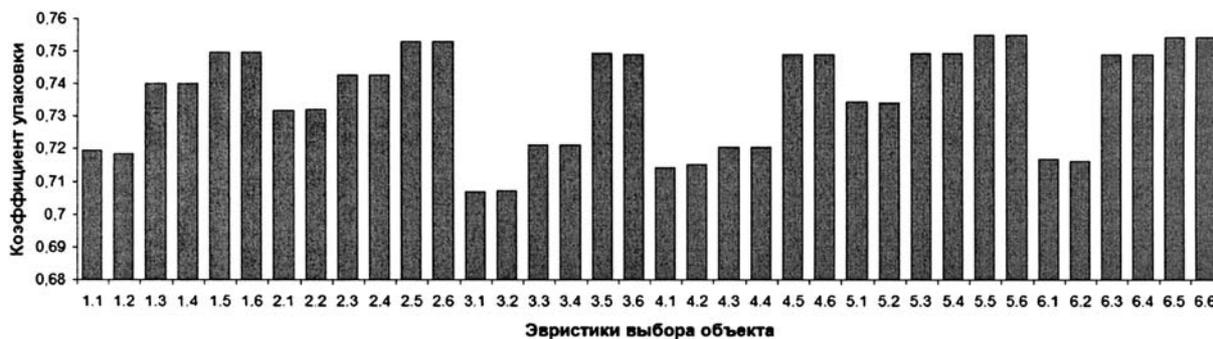


Рис. 4. Средние значения коэффициента упаковки при использовании различных эвристик выбора объекта для установки

ты алгоритма для различных сочетаний допустимых ориентаций объектов, коэффициента непровисания, эвристик выбора точки установки и эвристик выбора объекта. Все объекты устанавливали в контейнер с длиной 590, шириной 230 и высотой 230 согласно одному и тому же правилу, и результат решения каждой задачи для каждой такой комбинации был детерминированным. Применяли три варианта наборов допустимых ориентаций объектов: запрет любых поворотов и переворачиваний (1 допустимая ориентация), разрешение поворота объектов в плоскости основания (2 допустимые ориентации) и разрешение любых поворотов и переворачиваний (6 допустимых ориентаций). Перебирались все значения коэффициента непровисания от 0 до 1 с шагом 0,05. Таким образом, 3 набора допустимых ориентаций, 21 значение коэффициента непровисания, 6 эвристик выбора точки и 36 эвристик выбора объекта образуют 13608 комбинаций параметров. Именно столько раз решали каждый пример.

Зависимость коэффициента упаковки, усредненного по всем задачам и всем комбинациям прочих параметров, от числа допустимых ориентаций объектов представлена на рис. 3, а (см. 3-ю сторону обложки). В среднем решения, полученные при разрешении поворотов, на 2,9 % лучше, чем решения с фиксированной ориентацией объектов, а решения при разрешении любых вращений лучше на 6,8 %. На рис. 3, б проиллюстрирована зависимость среднего коэффициента упаковки от шести используемых эвристик выбора точки установки. Эвристики 2 ($\min y \rightarrow \min x \rightarrow \min z$) и 3 ($\min z \rightarrow \min x \rightarrow \min y$) проигрывают остальным, в то время как эвристика 6 ($\min(x^2 + y^2 + z^2) \rightarrow \min x \rightarrow \min z$) лидирует, что является достаточно неожиданным результатом. При этом сильно разнородные задачи более чувствительны к изменению перечисленных параметров алгоритма, чем слабо разнородные и однородные

(качество решения однородных задач практически одинаково при любых эвристиках выбора точки установки).

На рис. 4 представлены средние по задачам всех классов значения коэффициента упаковки, полученные при использовании различных эвристик выбора объекта (см. табл. 1). Качество решения зависит от используемой эвристики достаточно сильно; наилучшим образом себя проявили эвристики 5.5, 5.6, 6.5 и 6.6, использующие в качестве главного критерия выбора объекта его объем.

Зависимость качества решения от установленного коэффициента непровисания определялась для двух целевых функций: максимизации коэффициента упаковки контейнера с длиной 590, шириной 230 и высотой 230 и минимизации длины занятой части контейнера с такими же шириной и высотой и неограниченной длиной. Тенденции, представленные на рис. 5, являются удивительно четкими и гладкими: чем более разнородны объекты в примере, тем быстрее при возрастании коэффициента непровисания падает качество упаковки. Интерес представляет то, что коэффициент упаковки принимает максимальные значения, а длина занятой части полосы — минимальные значения, при коэффициенте непровисания в промежутке от 0,2 до 0,4.

Общие выводы по эксперименту № 1 заключаются в том, что эффективность работы отдельных

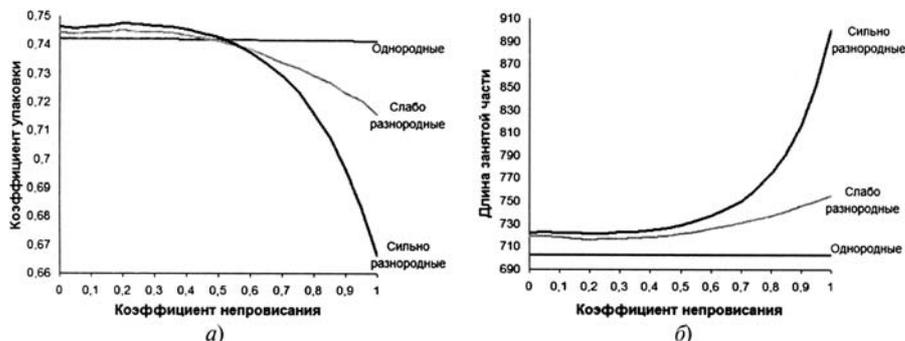


Рис. 5. Средняя эффективность работы алгоритма при различных значениях коэффициента непровисания: а — при максимизации коэффициента упаковки; б — при минимизации длины занятой части контейнера

Вероятности применения эвристик

	№ эвристики	Эвристика	Вероятность
Эвристики выбора точки установки	1	$\min x \rightarrow \min z \rightarrow \min y$	0,444
	5	$\min x \rightarrow \min y \rightarrow \min z$	0,389
	6	$\min(x^2 + y^2 + z^2) \rightarrow \rightarrow \min x \rightarrow \min z$	0,167
Эвристики выбора объекта	5.3	$\max d^y d^z \rightarrow \max d^y \rightarrow \max d^x$	0,361
	5.4	$\max d^y d^z \rightarrow \max d^z \rightarrow \max d^x$	0,361
	6.3	$\max d^y d^z \rightarrow \max d^x \rightarrow \max d^y$	0,139
	6.4	$\max d^y d^z \rightarrow \max d^x \rightarrow \max d^z$	0,139

эвристик выбора точки установки и объекта в значительной степени зависят от структуры решаемой задачи: от целевой функции, допустимости различных ориентаций и коэффициента непровисания. Можно выявить некоторые статистические закономерности, однако результат решения конкретных примеров различными комбинациями эвристик непредсказуем. Это подтверждает целесообразность применения эволюционного алгоритма.

Эксперимент № 2. Комбинирование эвристик.

В результате проведения эксперимента № 1 были выявлены такие параметры и эвристики, при которых алгоритм упаковки работает наиболее эффективно. Далее было проведено тестирование мультиметодного алгоритма при разрешении всех возможных ориентаций и случайном выборе коэффициента непровисания из промежутка $[0; 0,5]$. Эвристике выбора точки установки и объекта были присвоены вероятности в соответствии с частотой получения ими лучших решений при эксперименте № 1 (табл. 3). Определяли лучшее решение, полученное за 5000 итераций алгоритма, что занимало от 5 с для однородных примеров до 3 мин для сильно разнородных примеров. Результаты эксперимента представлены в табл. 4.

Решения, полученные с помощью комбинирования эвристик, во всех случаях существенно превосходят решения, полученные с помощью отдельных эвристик. Для многих однородных примеров были найдены решения с упаковкой всех доступных объектов. Наивысшая средняя эффективность работы алгоритма была достигнута на сильно разнородных задачах.

Заключение

В работе подробно описан мультиметодный алгоритм трехмерной упаковки, усовершенствованный по сравнению с его предыдущей версией. Введен учет коэффициента провисания, позволяющего обеспечить устойчивость схемы размещения объектов, не жертвуя при этом плотностью упаковки. Мультиметодную технологию применяют не только при выборе объектов для установки, но и при выборе точек, в которые помещают данные объекты. Обосновано применение набора из 36 "жадных" эвристик, которые всегда приводят к однозначному выбору объекта и его ориентации.

С помощью численного эксперимента исследована эффективность описанных эвристик, а также показано, каким образом изменяется качество получаемых решений при разрешении и запрете поворотов и переворачиваний объектов и при различных значениях коэффициента непровисания. Полученные данные использованы для подбора параметров алгоритма и установки вероятностей применения отдельных эвристик, что позволило быстро получать решения высокого качества.

Список литературы

1. **Pisinger D.** Heuristics for the container loading problem // European Journal of Operational Research. 1997. N 141. P. 143–153.
2. **Gendreau M., Iori M., Laporte G., Martello S.** A tabu search approach for a routing and container loading problem. // Transportation Science. 2006. N 40. P. 342–350.
3. **Martello S., Pisinger D., Vigo D.** The three-dimensional bin-packing problem // Operations Research. 2000. Vol. 48. P. 256–267.
4. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2–7.
5. **Валиахметова Ю. И., Филиппова А. С.** Мультиметодный генетический алгоритм для решения задач ортогональной упаковки. // Информационные технологии. 2007. № 12. С. 50–57.
6. **Валиахметова Ю. И., Мухачева Э. А., Филиппова А. С., Гильманова Н. А., Карипов У. А.** Мультиметодная технология ортогональной упаковки и ее применение в задачах транспортной логистики // Информационные технологии. 2009. № 12. Приложение. 32 с.
7. **Burke E. K., Hart E., Kendall G., Newall J., Ross P., Schulenburg S.** Hyper-heuristics: An emerging direction in modern search technology, Handbook of Metaheuristics (F. Glover and G. Kochenberger, eds.), Kluwer, 2003, pp. 457–474.
8. **Scheithauer G.** Algorithms for the container loading problem // Operations Research Proceedings, 1991. Berlin, Heidelberg; Springer-Verlag, 1992. P. 445–452.

Таблица 4

Значения	Коэффициент упаковки				Число неупакованных объектов			
	Однородные задачи	Слабо разнородные задачи	Сильно разнородные задачи	Среднее по всем задачам	Однородные задачи	Слабо разнородные задачи	Сильно разнородные задачи	Среднее по всем задачам
Лучшее	0,91566	0,90229	0,89947	0,90581	0	1	1	0,67
Худшее	0,61327	0,86257	0,87747	0,78444	44	8	6	19,33
Среднее	0,85295	0,88154	0,88791	0,87413	8,26	3,92	3,22	5,13

9. **Борисовский П. А., Еремеев А. В.** О сравнении некоторых эволюционных алгоритмов // Автоматика и телемеханика. 2004. № 3. С. 3–9.

10. **Hemminki J.** Container loading with variable strategies in each layer // ESI-X, EURO Summer Institute, Jouy-En-Josas, France, July 2–15. 1994.

А. А. Ковалевич, аспирант,
 А. И. Якимов, канд. техн. наук, доц.,
 Д. М. Албкеират, аспирант,
 ГУВПО "Белорусско-Российский университет",
 г. Могилев, Беларусь,
 e-mail: ykm@tut.by

Исследование стохастических алгоритмов оптимизации для применения в имитационном моделировании систем

Проведен сравнительный анализ стохастических алгоритмов оптимизации: роя частиц, имитации отжига и дифференциальной эволюции. Для проверки алгоритмов использовались функции Branin RCOS, модифицированная версия RCOS, Easom и Goldstein-Price. Показано, что алгоритм дифференциальной эволюции является лучшим алгоритмом, так как стабильно находит оптимум функции за минимальное время.

Ключевые слова: стохастический алгоритм оптимизации, рой частиц, имитация отжига, дифференциальная эволюция, имитационное моделирование

Введение

Основной идеей имитационного моделирования часто является построение модели для последующей оптимизации. Поэтому этап оптимизации имитационных моделей является ключевым для всего имитационного моделирования [1]. От эффективности оптимизации модели напрямую зависит эффективность моделирования для конечного пользователя модели. В настоящее время существует множество алгоритмов поиска оптимального решения, поэтому выбор действительно лучшего из них является довольно трудной задачей. Для выбора эффективного алгоритма используются критерии оптимальности найденного решения (то есть, какую часть от заранее известного оптимального решения составляет решение, найденное с помощью выбранного алгоритма) и времени, затраченного на поиск этого решения.

Анализ пакетов оптимизации, используемых при имитационном моделировании, показывает, что в имитационном эксперименте используются алгоритмы случайного поиска. Наиболее известные оптимизационные пакеты, применяемые в имитационном моделировании, следующие: *AutoStat* (AutoSimulations, Inc.; www.autosim.com) — использует эволюционные стратегии, генетические алгоритмы, применяется в системе имитации

AutoMod; *Evolutionary Optimizer* (ImagineThat, Inc; www.imaginethatinc.com) — основан на эволюционных стратегиях, генетических алгоритмах, применяется в *Extend*; *OptQuest* (OptTek Systems, Inc.; www.opttek.com) — использует случайный поиск, поиск с запретами, нейронные сети, встроен в системы имитации *AnyLogic*, *Arena*, *Crystal Ball*, *CSIM19*, *Enterprise Dynamics*, *Micro Saint*, *ProModel*, *Quest*, *SimFlex*, *SIM-PROCESS*, *SIMUL8*, *TERAS*; пакет *RISKOptimizer* (Palisade Corp.; www.palisade.com), использующий генетические алгоритмы, применяется в *@RISK*; *WITNESS Optimizer* (banner Group, Inc.; www.lanner.com/corporate) — использует алгоритм имитации отжига, поиск с запретами, применяется в системе имитационного моделирования *WITNESS* [2, 3].

Преимущества эволюционных подходов к решению оптимизационных задач: повышенное быстроедействие; высокая надежность и помехоустойчивость; высокая робастность, т. е. малая чувствительность к нерегулярностям поведения целевой функции; сравнительно простая внутренняя реализация; малая чувствительность к росту размерности задачи оптимизации; возможность естественного ввода в процесс поиска операции обучения и самообучения; легкое построение новых алгоритмов, реализующих различные эвристические процедуры адаптации в рамках известных схем случайного поиска.

Недостатки их использования: отсутствие гарантий качества получаемых решений; невозможность проверки на всех мыслимых классах задач; необходимость предварительной настройки параметров алгоритма для его эффективной работы.

В данной статье проводится сравнение трех алгоритмов стохастической оптимизации: алгоритма имитации отжига (ИО), роя частиц (РЧ) и дифференциальной эволюции (ДЭ).

1. Алгоритм имитации отжига

Алгоритм имитации отжига — общий алгоритмический метод решения задачи глобальной оптимизации, особенно дискретной и комбинаторной оптимизации, является одним из примеров метода Монте-Карло и относится к классу пороговых алгоритмов. Он создан в 1983 году [4]. Алгоритм основан на имитации процесса кристаллизации вещества при переходе из жидкого состояния в твердое. Процесс протекает при постоянно понижающейся температуре. При этом предполагается, что атомы уже выстроились в кристаллическую решетку, но допустимы переходы отдельных атомов из одной ячейки в другую. Переход происходит с некоторой вероятностью, которая

уменьшается с понижением температуры. Устойчивая кристаллическая решетка соответствует минимуму энергии атомов, поэтому атом переходит в состояние с меньшим уровнем энергии либо остается на месте.

Пусть $f_{\min}: R^D \rightarrow R^+$ — целевая функция, которую требуется минимизировать. Схема работы алгоритма ИО, используемого для проведения исследований, выглядит следующим образом.

Шаг 1. Положить $i = 0$, $i \in I = \{0, 1, 2, \dots\}$. Выбрать начальное решение $\mathbf{x}_i = x_1, x_2, \dots, x_D$ из пространства решений X и начальную температуру T_i . При этом скорость и закон убывания температуры задаются, например, по формуле

$$T_i = 1 \cdot 10^{19} \cdot 0,95^i | i \in I. \quad (1)$$

Шаг 2. Оценить начальное решение по известной целевой функции $f(\mathbf{x}_i)$.

Шаг 3. Случайным образом изменить решение \mathbf{x}_i и получить \mathbf{x}^* .

Шаг 4. Оценить полученное решение $f(\mathbf{x}^*)$.

Шаг 5. Проверить возможную замену текущего решения \mathbf{x}_i измененным решением \mathbf{x}^* . Точка \mathbf{x}^* становится точкой \mathbf{x}_{i+1} с вероятностью $P(\mathbf{x}_{i+1} := \mathbf{x}^* | \mathbf{x}_i)$, которая вычисляется в соответствии с распределением Гиббса:

$$P(\mathbf{x}_{i+1} := \mathbf{x}^* | \mathbf{x}_i) = \begin{cases} 1 & |f(\mathbf{x}^*) - f(\mathbf{x}_i) < 0; \\ \exp\left(-\frac{f(\mathbf{x}^*) - f(\mathbf{x}_i)}{T_i}\right) & |f(\mathbf{x}^*) - f(\mathbf{x}_i) \geq 0, \end{cases}$$

где T_i — элементы произвольной убывающей, сходящейся к нулю положительной последовательности, являющейся аналогом падающей температуры в кристалле (см. (1) на шаге 1).

Шаг 6. Уменьшить температуру, т. е. задать $i := i + 1$. При этом температура уменьшится в соответствии с формулой (1).

Шаг 7. Шаги 3—6 повторять, пока не выполнен критерий останова $T_i < 1 \cdot 10^{-322}$, принятый в настоящих исследованиях.

Алгоритм ИО достаточно широко распространен благодаря своей простоте, гибкости и эффективности, поскольку для данного алгоритма удается аналитически исследовать его свойства и доказать асимптотическую сходимость [4].

2. Алгоритм роя частиц

Алгоритм роя частиц — метод численной оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции. Алгоритм РЧ был доказан Кеннеди, Эберхартом и Ши и изначально предназначался для имитации социального поведения. Алгоритм был упрощен и было замечено, что он пригоден

для выполнения оптимизации. Алгоритм РЧ оптимизирует функцию, поддерживая популяцию возможных решений, называемых частицами, и перемещая эти частицы в пространстве решений согласно простой формуле. Перемещения подчиняются принципу наилучшего найденного в этом пространстве положения, которое постоянно изменяется при нахождении частицами более выгодных положений [5].

Пусть $f_{\min}: R^D \rightarrow R^+$ — целевая функция, выбранные параметры этой функции $\mathbf{x} = (x_1, \dots, x_D)$, которые необходимо оптимизировать, и определена область для поиска оптимальных значений: $\mathbf{b}_{\text{sup}}, \mathbf{b}_{\text{inf}}$ — верхняя и нижняя границы пространства решений соответственно; $|S|$ — число частиц в рое S (в данной работе $|S| = 60$), каждой из которых сопоставлена координата $\mathbf{x}_i \in R^D$ в пространстве решений и скорость $\mathbf{v}_i \in R^D$. Пусть также \mathbf{p}_i — лучшее из известных положений i -й частицы (ЛПЧ); \mathbf{g} — наилучшее известное положение роя (НПР) в целом. Схема работы алгоритма РЧ выглядит следующим образом.

Шаг 1. Сгенерировать начальное положение каждой частицы с помощью случайного вектора

$$\mathbf{x}_i := \text{rand}(\mathbf{b}_{\text{sup}}, \mathbf{b}_{\text{inf}}) | i \in S = \{1, 2, \dots, |S|\},$$

имеющего многомерное равномерное распределение. Задаются скорости их движения

$$\mathbf{v}_i := \text{rand}(-(\mathbf{b}_{\text{sup}} - \mathbf{b}_{\text{inf}}), (\mathbf{b}_{\text{sup}} - \mathbf{b}_{\text{inf}})).$$

Шаг 2. Инициализация \mathbf{p}_i и \mathbf{g} . Для каждой частицы рассчитывается функция пригодности $f(\mathbf{x}_i)$. Присвоить лучшему известному положению частицы его начальное положение $\mathbf{p}_i := \mathbf{x}_i$. Лучшая частица с точки зрения целевой функции f объявляется значением НПР:

$$\mathbf{g} := \mathbf{p}_i | f(\mathbf{p}_i) \rightarrow \min.$$

Шаг 3. Сгенерировать случайные векторы

$$\mathbf{r}_p := \text{rand}(-1, +1) \text{ и } \mathbf{r}_g := \text{rand}(-1, +1).$$

Введение \mathbf{r}_p и \mathbf{r}_g в оптимизацию предназначено для моделирования незначительного непредсказуемого реального поведения роя.

Шаг 4. Корректировка скорости частицы. Скорость частицы меняется в соответствии с взаимным расположением позиций ЛПЧ и НПР. Они стремятся в направлении этих позиций в соответствии с формулой

$$\mathbf{v}_i := \omega \mathbf{v}_i + \varphi_p \mathbf{r}_p \times (\mathbf{p}_i - \mathbf{x}_i) + \varphi_g \mathbf{r}_g \times (\mathbf{g} - \mathbf{x}_i),$$

где операция \times означает покомпонентное умножение.

Шаг 5. Обновить положение частицы переносом \mathbf{x}_i на вектор скорости

$$\mathbf{x}_i := \mathbf{x}_i + \mathbf{v}_i.$$

Частица может вылетать за пределы разрешенной области, но полученные ею значения не учитываются до тех пор, пока она не вернется обратно.

Шаг 6. Если $f(\mathbf{x}_i) < f(\mathbf{p}_i)$, то обновить ЛПЧ $\mathbf{p}_i := \mathbf{x}_i$. Если $f(\mathbf{p}_i) < f(\mathbf{g})$, то обновить НПП $\mathbf{g} := \mathbf{p}_i$. Теперь \mathbf{g} содержит лучшее из найденных решений.

Шаг 7. Шаги 3–6 выполняются для каждой частицы, пока не выполнен критерий останова (например, достижение заданного числа итераций или необходимого значения целевой функции). В исследованиях критерием останова является число итераций, равное $1 \cdot 10^4$.

Алгоритм имеет три параметра: ω , φ_p и φ_g . При проведении экспериментов использовали следующие значения: $\omega = 0,93$, $\varphi_p = 0,7$ и $\varphi_g = 1,2$.

Алгоритм РЧ можно эффективно распределить на несколько параллельных процессов, за счет чего повысится его быстродействие. По сравнению с генетическим алгоритмом, операторы которого могут быть реализованы различным образом, имеется лишь один оператор — вычисление скорости, что делает его более простым в использовании.

3. Алгоритм дифференциальной эволюции

Дифференциальная эволюция — алгоритм многомерной математической оптимизации, относящийся к классу стохастических алгоритмов оптимизации (т. е. работает с использованием случайных чисел) и использующий некоторые идеи генетических алгоритмов.

Алгоритм ДЭ предназначен для нахождения глобального экстремума недифференцируемых, нелинейных, мультимодальных (имеющих, возможно, большое число локальных экстремумов) функций от многих переменных. Алгоритм был предложен Р. Сторном и К. Прайсом, впервые опубликован ими в 1995 г. и разработан в дальнейшем в их более поздних работах. На первом международном конкурсе по эволюционным вычислениям, который проводился в мае 1996 г., алгоритм ДЭ занял третье место, при этом он был лучшим среди генетических алгоритмов и проиграл только двум алгоритмам, которые не являются в общем случае универсальными, однако быстрее решили предложенный набор задач.

Подобно другим эволюционным алгоритмам ДЭ рассматривает случайную популяцию решений. Пусть начальная популяция решений состоит из NP индивидов и пространство поиска является D -мерным. Популяция n -й итерации может быть представлена следующим образом:

$$\mathbf{X}(n) = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{NP}),$$

где \mathbf{x} — возможное решение в D -мерном пространстве поиска. Пусть существует некоторый критерий качества $f(\mathbf{x})$ такой, чтобы найти

$$\mathbf{x}^* : f(\mathbf{x}^*) = \min_{\mathbf{x}} f(\mathbf{x}).$$

Пусть $f: R^D \rightarrow R^+$ — целевая функция, которую требуется минимизировать. Алгоритм ДЭ включает три эволюционных процесса: мутацию, скрещивание, выбор.

Шаг 1. Оператор мутации случайным образом выбирает три различных индивида из текущей популяции и создает нового измененного индивида

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \times (\mathbf{x}_{r2} - \mathbf{x}_{r3}) | F \in [0, 2],$$

где F — управляющий параметр, обычно выбираемый в интервале $[0, 2]$, с наилучшими значениями в диапазоне $[0,5, 0,9]$. Оператор мутации является стохастическим отображением

$$T_m : S^{NP} \rightarrow S,$$

где $S = R^D$; S^{NP} — пространство популяции (в исследованиях $NP = 60$).

Шаг 2. Оператор скрещивания создает новое решение копированием компонентов мутационного вектора \mathbf{v}_i и выбранного вектора \mathbf{x}_i :

$$u_{ji} = \begin{cases} v_{ji} | r_b \leq CR \wedge j = r_p, i = 1, \dots, NP; \\ x_{ji} | r_b > CR \vee j \neq r_p, i = 1, \dots, D, \end{cases}$$

где $r_b = \text{rand}[0, 1]$ — случайное число в интервале $[0, 1]$; $r_p = \text{rand}[0, D]$ — случайное целое число в интервале $[1, D]$; $CR \in [0, 1]$ — управляющий параметр скрещивания.

Оператор скрещивания является стохастическим отображением

$$T_r : S^2 \rightarrow S.$$

Шаг 3. Оператор выбора является детерминированным процессом в алгоритме ДЭ и выбирает индивида с лучшим значением целевой функции для следующего поколения:

$$\mathbf{x}_i(n+1) = \begin{cases} \mathbf{u}_i | f(\mathbf{u}_i) \leq f(\mathbf{x}_i); \\ \mathbf{x}_i | f(\mathbf{u}_i) > f(\mathbf{x}_i). \end{cases}$$

Процесс выбора описывается детерминированным оператором

$$T_s : S^2 \rightarrow S.$$

Оператор выбора гарантирует, что лучшее значение целевой функции не может быть пропущено, что приводит к быстрой сходимости. Алгоритм ДЭ может быть описан следующим образом:

$$\begin{aligned} \mathbf{X}(n+1) &= \{\mathbf{x}_i(n+1) | \mathbf{x}_i(n+1) = \\ &= T_s \circ T_r \circ T_m(\mathbf{X}(n)), i = 1, \dots, NP\}. \end{aligned}$$

После этого шаги 1—3 повторяются, пока не будет сформировано нужное число поколений, которое в исследованиях принято равным $1 \cdot 10^4$.

Алгоритм прост в реализации и использовании (содержит малое число управляющих параметров, требующих подбора: коэффициент мутации F и вероятность скрещивания CR), легко распараллеливается [6, 7].

4. Функции, используемые для проверки алгоритмов

Для проверки алгоритмов использовали функции *Branin RCOS*, модифицированная версия *RCOS*, *Easom* и *Goldstein-Price*. Этот выбор обусловлен тем фактом, что функции специально предложены и используются для тестирования алгоритмов оптимизации, основанных на случайном поиске.

Для функции *Branin RCOS* характерно наличие множества локальных оптимумов. Определение функции *Branin RCOS*:

$$\begin{aligned} & \text{Branin}(x_1, x_2) = \\ & = (x_2 - b(x_1)^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e, \end{aligned} \quad (2)$$

где $b = 5,1/4\pi^2$; $c = 5/\pi$; $d = 6$; $e = 10$; $f = 1/8\pi$; $-5 \leq x_1 \leq 10$; $0 \leq x_2 \leq 15$.

Глобальный минимум этой функции:

$$\begin{aligned} & \text{Branin}(x_1, x_2) = 0,397887 | (x_1, x_2) = \\ & = (9,42478; 2,475), (-\pi; 12,275), (\pi; 2,275). \end{aligned} \quad (3)$$

График функции (2) представлен на рис. 1.

Функция *Easom* характеризуется наличием пологих участков и глобального минимума, площадь которого достаточно мала по сравнению с площадью пространства поиска. Определяется функция *Easom* следующим образом:

$$\begin{aligned} & \text{Easom}(x_1, x_2) = \\ & = -\cos(x_1)\cos(x_2)\exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2)), \end{aligned} \quad (4)$$

где $-100 \leq x_i \leq 100$, $i = 1, 2$.

Глобальный минимум этой функции:

$$\text{Easom}(x_1, x_2) = -1,0 | (x_1, x_2) = (\pi, \pi). \quad (5)$$

График функции (4) представлен на рис. 2.

Для модифицированной функции *RCOS* характерно наличие шести локальных экстремумов и одного глобального. Определение функции:

$$\begin{aligned} & M_Branin(x_1, x_2) = 1/[a(x_2 + bx_1^2 + cx_1 - d)^2 + \\ & + e(1 - f)\cos(x_1)\cos(x_2) + \log(x_1^2 + x_2^2 + 1) + e], \end{aligned} \quad (6)$$

где $a = 1$, $b = 5,1/4\pi$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/8\pi$.

Область поиска: $-5 \leq x_1 \leq 10$, $0 \leq x_2 \leq 15$.

Глобальный максимум этой функции:

$$\begin{aligned} & M_Branin(x_1, x_2) = 0,689087 | (x_1, x_2) = \\ & = (3,06699; 2,99805 \cdot 10^{-16}). \end{aligned} \quad (7)$$

График функции (6) представлен на рис. 3.

Для функции *Goldstein-Price* характерно наличие нескольких оптимумов, разделенных широким плато, один из которых — глобальный мак-

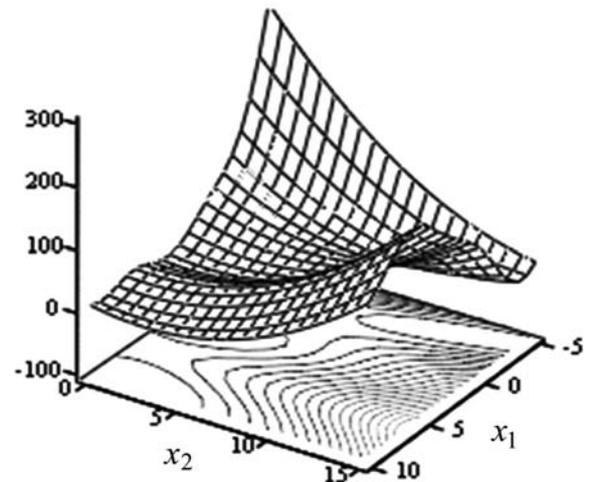


Рис. 1. График функции *Branin*(x_1, x_2)

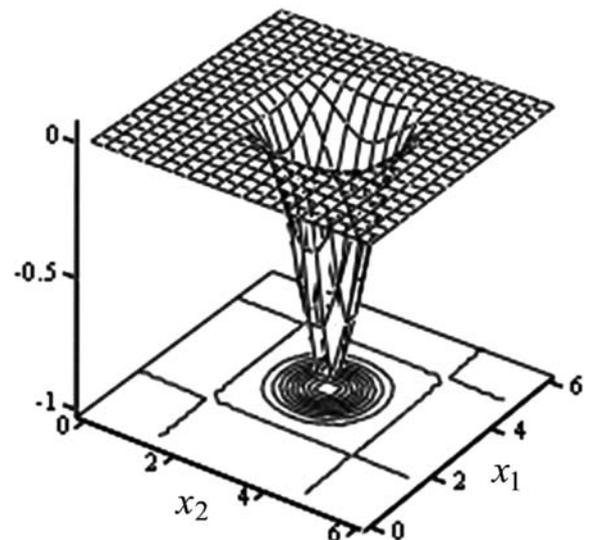


Рис. 2. График функции *Easom*(x_1, x_2)

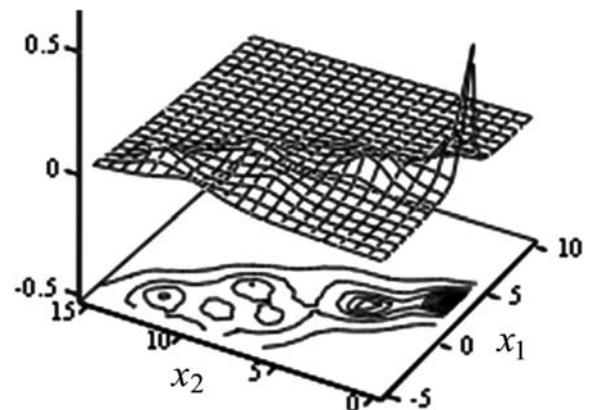


Рис. 3. График функции *M_Branin*(x_1, x_2)

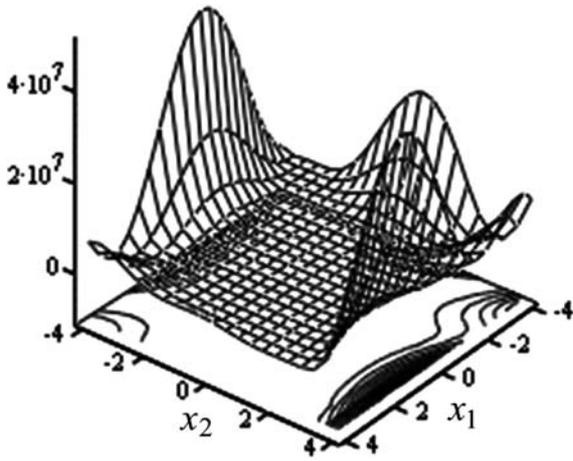


Рис. 4. График функции $Golden(x_1, x_2)$

симум. Определяется функция $Goldstein-Price$ выражением

$$Gold(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)], \quad (8)$$

где $-2 \leq x_i \leq 2, i = 1, 2$.

Глобальный минимум этой функции

$$Gold(x_1, x_2) = 3,0 \mid (x_1, x_2) = (0; -1). \quad (9)$$

График функции (8) представлен на рис. 4.

5. Результаты исследований и их обсуждение

Для проведения исследований использовали компьютер с процессором *AMD Athlon 64 3500+* и 1 ГБ ОЗУ под управлением операционной системы *Windows XP SP3*. Реализации алгоритмов были написаны на языке C++.

При анализе чувствительности параметров F и CR алгоритма ДЭ было отмечено, что алгоритм не чувствителен к этим параметрам для всех функций кроме модифицированной функции $RCOS$. Результаты исследования чувствительности параметров F и CR для модифицированной функции $RCOS$ приведены в табл. 1.

Для проведения исследований использовали следующие значения параметров алгоритма ДЭ: $F = 0,9$; $CR = 0,5$.

При исследовании функции $Branin RCOS$ глобальный минимум был найден только с помощью алгоритма ДЭ (табл. 2). Так как с помощью ДЭ на поиск лучшего решения было потрачено не больше времени, то следует признать, что для функции $Branin RCOS$ оптимальным алгоритмом поиска глобального минимума является алгоритм ДЭ. Далее следует алгоритм роя частиц, который нашел в среднем решение за то же время, но на 0,04 %

хуже. Последним оказался алгоритм имитации отжига, с помощью которого решения находились медленнее на 3,8 % и на 10,5 % хуже оптимального.

При исследовании функции $Easom$ (табл. 3) глобальный минимум был найден только с помощью алгоритма ДЭ. В среднем алгоритм роя частиц находил решение на 0,21 %, а имитации отжига — на 13,04 % хуже. Учитывая, что с помощью ДЭ на поиск лучшего решения было потрачено еще и меньшее время (на 22,6 % и на 13,1 % быстрее, чем с помощью алгоритмов роя частиц и имитации отжига соответственно), следует считать, что для функции $Easom$ оптимальным алгоритмом поиска глобального минимума является алгоритм ДЭ. Далее следует алгоритм РЧ и худшее решение получено с помощью алгоритма ИО.

Таблица 1
Результаты исследований чувствительности параметров ДЭ для функции $M_Branin(x_1, x_2)$

Коэффициент мутации F	Вероятность скрещивания CR	Решение
0,1	0,1	0,167853
0,1	0,5	0,628472
0,1	0,9	0,167826
0,9	0,1	0,167853
0,9	0,9	0,689087
2	0,1	0,689087
2	0,5	0,689087
2	0,9	0,689087

Таблица 2
Результаты исследований для функции $Branin(x_1, x_2)$

Наименование алгоритма	Время работы, с	Решение	Оптимальность решения, %
Дифференциальная эволюция	0,556	0,397887	100
Рой частиц	0,556	0,398056	99,96
Имитация отжига	0,577	0,439660	89,50

Таблица 3
Результаты исследований для функции $Easom(x_1, x_2)$

Наименование алгоритма	Время работы, с	Решение	Оптимальность решения, %
Дифференциальная эволюция	0,718	-1,0	100
Рой частиц	0,880	-0,997890	99,79
Имитация отжига	0,812	-0,869631	86,96

Таблица 4
Результаты исследований для функции $M_Branin(x_1, x_2)$

Наименование алгоритма	Время работы, с	Решение	Оптимальность решения, %
Дифференциальная эволюция	0,609	0,68909	100
Рой частиц	0,703	0,68845	99,91
Имитация отжига	0,765	0,68831	99,89

Таблица 5

Результаты исследований для функции *Golden* (x_1, x_2)

Наименование алгоритма	Время работы, с	Решение	Оптимальность решения, %
Дифференциальная эволюция	0,562	3,0	100
Рой частиц	0,666	3,00295	99,90
Имитация отжига	0,723	3,00667	99,78

При исследовании модифицированной функции *RCOS* (табл. 4) глобальный максимум был найден только с помощью алгоритма ДЭ. Учитывая, что с помощью ДЭ на поиск лучшего решения было потрачено еще и меньшее время, следует считать, что для модифицированной функции *RCOS* оптимальным алгоритмом поиска глобального максимума является алгоритм ДЭ. Далее следует алгоритм роя частиц. Стоит отметить, что для данной функции алгоритм имитации отжига нашел решение, которое хуже оптимального на 0,11 %, алгоритм роя частиц нашел решение на 0,09 % хуже оптимального.

При исследовании функции *Goldstein-Price* (табл. 5) алгоритм РЧ в среднем находил решение на 0,1 % хуже и на 18,5 % медленнее, чем алгоритм ДЭ. Алгоритм ИО отставал от алгоритма ДЭ на 0,22 % по оптимальности решения и на 28,6 % по скорости. Поэтому оптимальным алгоритмом следует считать алгоритм ДЭ [8].

Таким образом, для четырех функций (*Branin RCOS*, модифицированной *RCOS*, *Goldstein-Price* и *Easom*) наиболее быстрым и точным алгоритмом из трех исследуемых алгоритмов является алгоритм дифференциальной эволюции. За ним следует алгоритм роя частиц. Замыкает тройку алгоритм имитации отжига. Это можно объяснить тем фактом, что алгоритм ИО лучше всего подходит для оптимизации дискретных функций, а для функций, которые не являются дискретными, оптимальное решение, найденное с его помощью, как правило, не является глобальным.

Также стоит заметить, что при проведении экспериментов над всеми функциями алгоритм ДЭ всегда выдавал одно и то же оптимальное решение для каждой исследуемой функции, в то время как алгоритмы РЧ и алгоритм ИО каждый раз выдавали разное оптимальное решение. Таким обра-

зом, для выбранных функций алгоритм дифференциальной эволюции характеризуется стабильностью поиска решений, в отличие от двух других алгоритмов.

Заключение

Алгоритм дифференциальной эволюции является лучшим алгоритмом, так как стабильно находит оптимум функции и за минимальное время. Учитывая еще и исследования [9], где алгоритм дифференциальной эволюции сравнивался с генетическим алгоритмом, метагенетическим алгоритмом, методом "ветвей и границ" и также оказался лучшим, то следует признать, что алгоритм дифференциальной эволюции является лидирующим алгоритмом для оптимизации непрерывной функции от многих переменных. Еще одной положительной особенностью алгоритма является то, что его можно использовать и в режиме распределенных вычислений для сокращения времени проведения этапа оптимизации.

Работа выполнена при финансовой поддержке гранта Ф09М-171 БРФФИ.

Список литературы

1. Якимов А. И. Технология имитационного моделирования систем управления промышленных предприятий: монография. Могилев: Белорус. Рос. ун-т, 2010. — 304 с.
2. Law A. M., McComsas M. G. Simulation-Based Optimization // Proc. of 2002 Winter Simulation Conference. San Diego: IEEE Press, 2002. P. 41—44.
3. Fu M. C., Glover F., April J. Simulation Optimization: A Review, New Developments, and Applications // Proc. of 2005 Winter Simulation Conference. Orlando: IEEE Press, 2005. P. 83—95.
4. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing // Science. — 1983. — № 4598. — P. 671—680.
5. Weise T. Global Optimization Algorithms. Theory and Application 2nd Ed. 2008. 652 p. URL: <http://www.it-weise.de/projects/book.pdf>. — Дата доступа 20.06.2010.
6. Price K. V., Storn R. M., Lampinen J. A. Differential evolution. A practical approach to global optimization. Leipzig: Springer-Verlag Berlin Heidelberg, 2005. 539 p.
7. Zhang X. Y., Duan H. B., Yu Y. X. Receding horizon control for multi-UAVs close formation control based on differential evolution // Science China. Information Sciences. — 2010. — Vol. 53. — № 2. — P. 223—235.
8. Ковалевич А. А. Сравнительный анализ алгоритмов многопараметрической оптимизации в имитационном моделировании систем // Математическое и имитационное моделирование систем. МОДС 2010: тез. докл. пятой науч.-практич. конф. с междунар. участием, 21—25 июня 2010 г. Киев: ИПММС НАН Украины, 2010. С. 218—220.
9. Пупков К. А., Феоктистов В. А. Алгоритм дифференциальной эволюции для задач технического проектирования // Информационные технологии. 2004. № 8. С. 25—31.

В. Н. Пашенко, канд. техн. наук, доц.,
e-mail: pashenkovn@inbox.ru,
В. А. Мартынюк, канд. техн. наук, доц.,
e-mail: mart@rk6.bmstu.ru,
А. С. Монилов, студент,
e-mail: abidzhail@gmail.com,
В. В. Пашенко, студент,
e-mail: pashenkovv@inbox.ru
МГТУ им. Н. Э. Баумана

Исследование влияния типа конечных элементов на продолжительность и точность расчета температурных напряжений в среде ANSYSWorkbench

Предлагается методика решения задачи инженерного анализа термических деформаций сложных размерно-зависимых конструкций с точными привязками, входящих в состав малых космических аппаратов. Проведено исследование влияния типа конечных элементов на результаты решения. Построена конечно-элементная модель с использованием одномерных, двухмерных и трехмерных конечных элементов. Приведены результаты численного эксперимента, демонстрирующие преимущество данного подхода.

Ключевые слова: одномерные конечные элементы, линейное тело, точечная сварка, температурные напряжения, расчет термических деформаций, выбор конечных элементов, технические системы с точными размерными привязками, расчет антенных систем, вычисления в среде ANSYSWorkbench

Введение

Корпусные и рамные конструкции, антенные системы, входящие в состав малых космических аппаратов, определяют широкий класс сложных размерно-зависимых инженерных конструкций с точными привязками, работающих в условиях перепада температур от $-150\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$. Подобные эксплуатационные тепловые воздействия вызывают температурные деформации, способные исказить геометрию конструкции, существенно изменить взаимное угловое положение приборов и, как следствие, существенно снизить точность их измерений.

В инженерных расчетах сложных конструкций используют как аналитические, так и численные методы [4, 5]. Первые не могут быть применены для решения указанного выше класса задач, поскольку

базируются на математических методах решения краевых задач, обычно сложных и трудоемких, и ограничены простыми геометрическими формами тел и схем нагружений [2–5]. Численные методы, к которым относятся, в частности, метод конечных разностей, метод граничных интегральных уравнений, метод граничных элементов, метод конечных элементов и другие методы, не ограничены ни формой тел, ни способом приложения нагрузки [4].

При эскизной проработке термосистем начинают с оценочных вычислений вручную или с применением табличных программ. На следующих этапах используется программное обеспечение инженерного анализа (CAE-системы) для моделирования температурных полей и напряженно-деформированного состояния конструкции.

Основной проблемой при разработке (назначении) моделей является компромисс между достаточно полным и адекватным описанием рассматриваемого явления и доступностью (трудоемкостью) расчета на основе принятых моделей. С одной стороны, возможно построение подробной конечно-элементной модели исследуемого объекта, с другой стороны, время и ресурсы, необходимые для расчета, зачастую не позволяют выполнить поставленную задачу [1–3].

Постановка задачи

В настоящей статье рассматривается задача разработки методики анализа термических деформаций размерно-зависимых конструкций с точными привязками, одним из представителей которых являются антенно-фидерные системы бортовой приемопередающей аппаратуры магистрального тракта (АФС МТ) малых космических аппаратов. АФС МТ предназначена для работы в составе бортового радиоретрансляционного комплекса изделия МЦА (модуль целевой аппаратуры). Геометрическая модель одной из возможных конструкций представлена на рис. 1.

АФС МТ должна сохранять свою работоспособность в условиях максимальных перепадов температур. При этом температуры стабилизации, поддерживаемые на кронштейне и двигателях, при воздействии и отсутствии воздействия солнечного излучения могут быть различными. В рассматриваемом примере выделено четыре расчетных случая, соответствующие максимальным возможным перепадам температур (табл. 1).

АФС МТ испытывает тепловое воздействие:

- от нагрева поверхностей зеркал и облучателей Солнцем (остальные элементы конструкции изолированы от действия солнечного излучения);
- через контактные поверхности кронштейна;

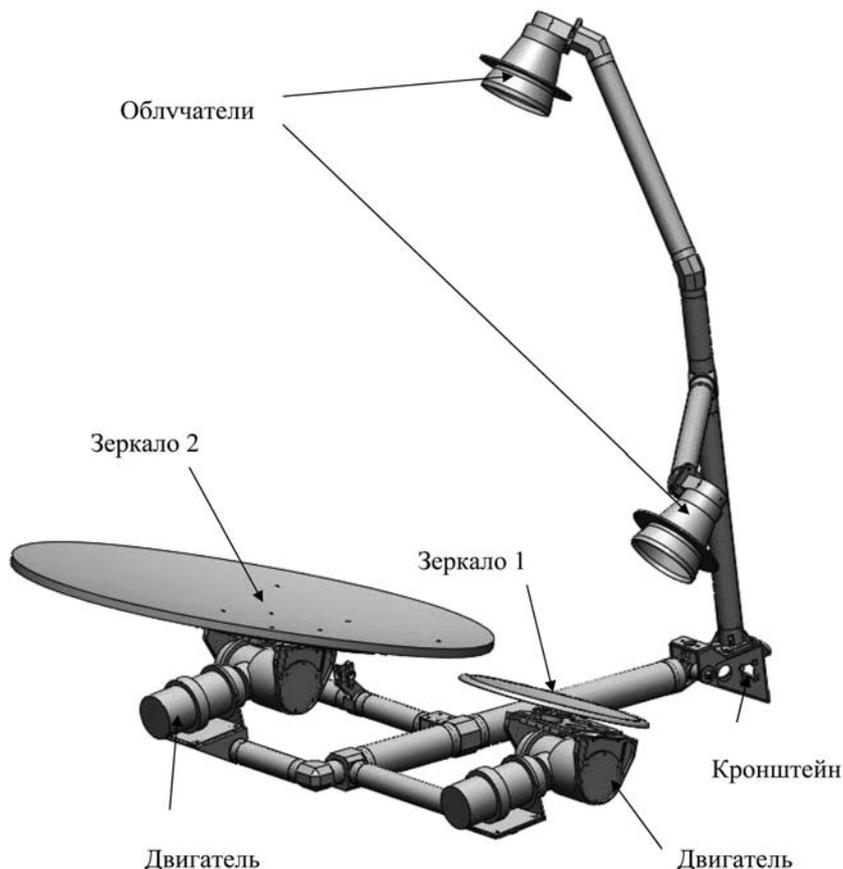


Рис. 1. Геометрическая модель АФСМТ

- посредством передачи температуры от двигателей стабилизации;
- через передачу теплоты от одних элементов конструкции к другим.

Теплота не передается с зеркал на трубы, что является следствием термической "отвязки" тарелки от труб системой приводов.

Вследствие нагрева (охлаждения) элементов конструкции происходит деформация конструкции. Деформация конструкции приводит к дефокусировке и децентровке.

Под дефокусировкой Δ_{ϕ} понимается смещение TZ зеркала относительно своего облучателя по оси Z расчетной координатной системы:

$$\Delta_{\phi} = TZ.$$

Таблица 1

Расчетные случаи возможных перепадов температур

Расчетный случай	Температура зеркал и облучателей, °С	Температура кронштейна, °С	Температура двигателей, °С
1	+125	+40	+50
2	+125	-20	-45
3	-150	-20	-45
4	-150	+40	+50

Под децентровкой $\Delta_{ц}$ принимается смещение зеркала относительно своего облучателя в плоскости, перпендикулярной оси Z , т. е. суммарное перемещение по оси X и Y :

$$\Delta_{ц} = \sqrt{TX^2 + TY^2},$$

где TX и TY — смещения зеркала относительно своего облучателя по оси X и Y соответственно.

При расчете температурных напряжений сначала необходимо рассчитать температуру конструкции, а на основе результатов расчета определить деформацию.

Реализация методики анализа температурных деформаций

Для расчета температурных напряжений в среде инженерного анализа ANSYSWorkbench существует специальный шаблон Thermal-Stress (расчет температурных напряжений). Он включает в себя два типа анализа: Steady-StateThermal (тепловой анализ в установившемся режиме) и StaticStructural (статический конструкционный анализ).

Математическая модель изделия представляет собой конечно-элементное описание конструкции при заданных граничных условиях и нагрузках, построенное в рамках определенных допущений.

ANSYSWorkbench поддерживает функцию автоматической генерации конечно-элементной модели. Для АФСМТ конечно-элементная модель (рис. 2, см. четвертую сторону обложки) содержит 528 234 узла и 175 187 элементов. Провести расчет на компьютерах с 32-битной версией Windows невозможно из-за особенностей ее архитектуры: операционная система не может выделить для ANSYSWorkbench требуемый объем оперативной памяти. Эта проблема частично решается использованием 64-битной версии операционной системы.

В методе конечных элементов точность и время проведения анализа зависит от правильного подбора типа конечного элемента и его размера. При использовании конечно-элементной модели, автоматически построенной в ANSYSWorkbench, погрешность расчета может достигать 20 %. Выбор типа и размера конечных элементов для каждого конкретного случая индивидуален и основан на мировом опыте применения метода конечных элементов [1].

Как показали исследования, увеличение размера конечных элементов в трубах незначительно

повлияло на точность расчета, но увеличило скорость проведения анализа за счет уменьшения числа узлов и элементов в конечно-элементной модели, а следовательно, уменьшило размер требуемой оперативной памяти для проведения расчета, что в рассматриваемом случае позволило провести расчет и в 32-битной операционной системе Windows. При оптимизации конструкции или при большом числе возможных расчетных случаев требуется существенное увеличение скорости проведения расчета, даже если это приведет к незначительной потере точности.

В рассматриваемой модели при нагревании (охлаждении) трубы меняются все их размеры, но при вычислении дефокусировки и децентровки имеет значение только изменение ее длины, и использование трехмерных конечных элементов приводит к "лишним" вычислениям. В таких случаях используются одномерные конечные элементы, например, при расчете балок, стержней, штанг и любых протяженных тонкостенных конструкций [6]. В ANSYSWorkbench можно использовать инструментарий linebody (линейное тело) и crosssection (поперечное сечение) [5]. Для передачи нагрузок между линейными телами к ним необходимо применить точечный контакт. Для построения конечно-элементной модели кронштейна применили трехмерные конечные элементы, зеркал — двухмерные, а для остальных элементов использовали одномерные конечные элементы. В полученной упрощенной модели (рис. 3, см. четвертую сторону обложки), число узлов и конечных элементов

снизилось на порядок, что уменьшило время расчета в несколько раз.

Среднее время проведения анализа для расчетного случая разных конечно-элементных моделей, а также число узлов и элементов приведены в табл. 2. Оценить точность вычислений можно на основе полученных значений дефокусировок и децентровок для четырех расчетных случаев (рис. 4, см. четвертую сторону обложки). По горизонтали откладывается номер расчетного случая, по оси Y — значения децентровки (дефокусировки) в миллиметрах для модели, конечно-элементная модель которой сгенерирована автоматически (обозначена синим цветом), и для модели с использованием одномерных конечных элементов (обозначена красным цветом).

Заключение

В статье были рассмотрены основные проблемы, возникающие при проведении инженерного анализа сложных конструкций с точными размерными привязками, а также проблемы, возникающие в штатных орбитальных условиях всевозможных конструкций малых космических аппаратов.

Предлагаемый подход получения конечно-элементной модели исследуемого объекта позволяет существенно уменьшить время, необходимое для оптимизации конструкции при большом числе расчетных случаев, без существенного снижения точности расчета.

Список литературы

1. Каплун А. Б., Морозов Е. М., Олферьева М. А. ANSYS в руках инженера: Практическое руководство. М.: Едиториал УРСС, 2003. 272 с.
2. Норенков И. П., Кузьмик П. К. Информационная поддержка наукоемких изделий. CALS-технологии. М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. 320 с.
3. Основы САПР (CAD/CAM/CAE). СПб.: Питер, 2004. 560 с.
4. Сабоннадьер Ж.-К., Клон Ж.-Л. Метод конечных элементов и САПР: Пер. с франц. М.: Мир, 1989. 190 с.
5. Шаповалов Л. А. Моделирование в задачах механики элементов конструкций (Б-ка расчетчика / Ред. кол.: В. А. Светлицкий (пред.) и др.). М.: Машиностроение, 1990. 288 с.
6. Чернов. Балки в ANSYSWorkbench // ANSYS Solution. Русская редакция, 2006. № 2. С. 38—39.

Таблица 2

Сравнительная таблица конечно-элементных моделей

Параметры модели	Автоматически сгенерированная модель	Модель, полученная посредством изменения размеров конечных элементов	Модель, полученная посредством изменения типа конечных элементов
Число узлов	528 234	320 549	24 622
Число элементов	175 187	139 619	16 623
Время генерации	≥20 ч	≈1 ч	≈2 мин

Х. Г. Асадов, д-р техн. наук, нач. отдела,
Е. Н. Алиева, диссертант,
 НИИ аэрокосмической информатики
 Национального аэрокосмического агентства,
 г. Баку
 e-mail: asadzade@rambler.ru

О возможности применения метода многоуровневой оптимизации для оптимальной информационной оценки разнообразия видов в экологических системах

Предложен способ многоуровневой оптимизации, где в верхней ступени оптимизации решается задача условной вариационной оптимизации, а в нижней — задача линейного программирования, что позволяет существенно уменьшить число оптимальных решений, полученных в первой ступени. Предложенный способ применен для оценки разнообразия видов в экологических системах.

Ключевые слова: экологические системы, оптимизация, вариационное исчисление, линейное программирование, экологическое разнообразие

Идея многоуровневой оптимизации в общем виде впервые была предложена социологом Этзионе в 1968 г. [1] в его теории адаптивного общества в качестве метода для оптимизации социальной структуры общества. Со временем эта идея стала применяться для оптимизации как в экономических, так и технических системах. Как отмечено в работе [2], каждая сложная система, к которой можно отнести и экономические системы, состоит из большого числа взаимосвязанных объектов, каждый из которых можно интерпретировать как подсистему этой сложной системы.

Узловой и чрезвычайно сложной проблемой при моделировании таких систем является проблема согласования решений, найденных для каждой подсистемы в отдельности (локальных оптимумов), с глобальным оптимумом всей системы в целом. Очевидно, что изучение проблемы согласования локальных оптимумов с глобальным тесно связано с проблемой оптимальной декомпозиции (децентрализации) целевых установок до соответствующих целевых установок каждой подсистемы в отдельности. Таким образом, обеспечивается согласование целей на уровне непротиворечивости интересов и реализуется принцип: то, что необходимо системе (например, обществу в целом),

должно быть выгодно каждому элементу системы в отдельности.

Следует отметить, что идеи многоуровневой оптимизации широко применяются в различных сферах техники.

Как отмечено в работе [3], важным принципом построения АСУ промышленными предприятиями является рассмотрение их как "больших" систем с многоуровневой иерархической структурой. Интегральная оптимизация функционирования предприятия — задача, на первый взгляд кажущаяся сформулированной удачно и достаточно просто, в действительности еще очень далека от возможности практической реализации ее вследствие огромной размерности модели и необходимости учета в них возмущающих воздействий, характеризующихся высокой степенью неопределенности. Согласно [1], такую трудность можно преодолеть путем формирования иерархической модели, принципиальное преимущество которой заключается в возможности решения задачи управления на базе локальных решений, осуществимых на основе подмоделей, соответствующих уровням иерархии модели.

Книга [4] является одной из первых в мировой литературе работ, посвященных систематическому изложению и математической формализации идей и методов управления большими системами, построенными по иерархическому принципу. В работе Л. Ласдона [5] развита теория многоуровневой оптимизации, позволяющая заменить "большие" задачи оптимизации последовательностью более мелких задач. Дальнейшее свое развитие и первое применение к промышленному объекту метод многоуровневой оптимизации Ласдона [5] получил в работе [6]. В [7] идея многоуровневой оптимизации развита применительно к управлению сложными комплексами с непрерывным характером производства. Система управления промышленным предприятием представляется как четырехуровневая иерархическая система, между уровнями которой разделены функции управления: оптимальное планирование — оперативное управление — оптимизация режимов установок — регулирование процессов. Обсуждаются вопросы декомпозиции общей задачи управления объектом на основе характеристик возмущений; построение моделей и алгоритмов отдельных уровней системы; задачи их взаимосвязки.

Идея многоуровневой оптимизации в технике применяется не только при построении АСУ, но и находит широкое применение в высокотехнологичных отраслях производства. Так, в работе [8] излагаются задача и методический аппарат эффективной модернизации космической системы дис-

танционного зондирования Земли (ДЗЗ), при наличии ограничений. Задача эффективной модернизации космической системы — это комплексная согласованная оптимизация параметров модификации космических аппаратов (КА) и модернизация космической системы. Оптимизация параметров модификации КА и модернизации космической системы позволяет:

- расширить число решаемых задач, область применения космической системы, сроки ее эффективного использования;
- при наличии новых технических и технологических возможностей повысить эффективность космической системы.

Поиск решения при комплексной согласованной оптимизации параметра модификации КА и модернизации космической системы проводится итерационным методом. При этом алгоритм решения задачи исключает последовательное решение трех основных задач:

- оптимизации параметра космической системы при модернизации и наличии ограничений;
- оптимизации параметров модификации КА с учетом динамики функциональных связей;
- направленной адаптации проектных зависимостей и управления согласованием решений.

Методический аппарат представлен в виде схемы многоуровневой оптимизации проектных параметров, моделей и алгоритмов.

Известен многоуровневый декомпозиционный метод, где исходно сформулированная задача решается по иерархической схеме, и в этом случае оптимизационная задача верхнего уровня считается решенной, если решены все оптимизационные подзадачи нижнего уровня [9].

В настоящей статье предлагается несколько иной подход к общей идее многоуровневой оптимизации, который заключается в следующем.

На первом уровне оптимизации применяется выбранный первый метод, который дает определенное множество решений $A = \{X_j\}$. На втором уровне применяется второй выбранный метод, который дает подмножество решений $B = \{X_j\}$, содержащее несколько меньшее число решений, где $A \supset B$. На третьем уровне оптимизации применяется третий выбранный метод, который дает подмножество решений $C = \{x_k\}$, содержащее еще меньшее число решений, т. е. $A \supset B \supset C$. И, наконец, на завершающем уровне удается применить последний выбранный метод, который дает подмножество решений $D = \{X_j\}$, содержащее всего один элемент, где $A \supset B \supset C \supset D$.

Далее в настоящей статье мы рассмотрим способ реализации изложенного выше подхода к идее многоуровневой оптимизации с применением методов вариационной оптимизации и линейного программирования в задаче двухуровневой оптимизации.

Общий алгоритм предлагаемого способа предусматривает оптимизацию на двух уровнях.

1. На первом уровне оптимизации выполняются следующие операции.

1.1. Формируется интегральный функционал цели в виде

$$\int_0^{T_{\max}} F[S(T)]dT, \quad (1)$$

где T_{\max} — верхний временной предел интегрирования; $S(T)$ — искомая функция, которая должна придать функционалу (1) его максимальное значение; F — известная функция от $S(T)$.

1.2. Формируется ограничительное условие в виде

$$\int_0^{T_{\max}} L[S(T)]dT = C, \quad (2)$$

где $C = \text{const}$; L — оператор дискретной модуляции значений функций $S(T)$.

Действие оператора L сводится к мультипликативному преобразованию $S(T)$, т. е.

$$L[S(T)] = a(T)S(T).$$

1.3. Составляется уравнение вариационной оптимизации

$$\Phi[S(T)] = \int_0^{T_{\max}} F[S(T)]dT + \lambda \int_0^{T_{\max}} L[S(T)]dT, \quad (3)$$

где λ — множитель Лагранжа.

1.4. Допускается существование множества операторов L_i

$$\{L_i\}, i = \overline{1, n},$$

и, соответственно, множества $\{C_i\}$ и множества $\{a_i(T)\}$.

1.5. Для каждого L_i находится функция $a_i(T)S(T)$, приводящая функционал (3) к его максимальному значению. Таким образом, получаем множество оптимальных решений первого уровня $\{a_i(T)S_i(T)\}$.

2. На втором уровне оптимизации выполняются следующие операции.

2.1. Для нахождения множества оптимальных решений нижнего уровня промежуток $0 - T_{\max}$ разбивается на m участков и для фиксированных точек $T_j, j = \overline{1, m}$, а также для разных операторов $L_i, i = \overline{1, n}$, с учетом условия (2), приняв

$$\int_0^{T_{\max}} L_i[S(T)]dT = C_i = \sum_{j=1}^m a_{ij}(T_j)S(T_j)\Delta T,$$

составляется система уравнений для $i = \overline{1, n}$:

$$\left. \begin{aligned} a_{11}(T_1)S(T_1) + a_{12}(T_2)S(T_2) + \dots + \\ + a_{1m}(T_m)S(T_m) = c_1; \\ a_{21}(T_1)S(T_1) + a_{22}(T_2)S(T_2) + \dots + \\ + a_{2m}(T_m)S(T_m) = c_2; \\ \dots \\ a_{n1}(T_1)S(T_1) + a_{n2}(T_2)S(T_2) + \dots + \\ + a_{nm}(T_m)S(T_m) = c_n. \end{aligned} \right\} \quad (4)$$

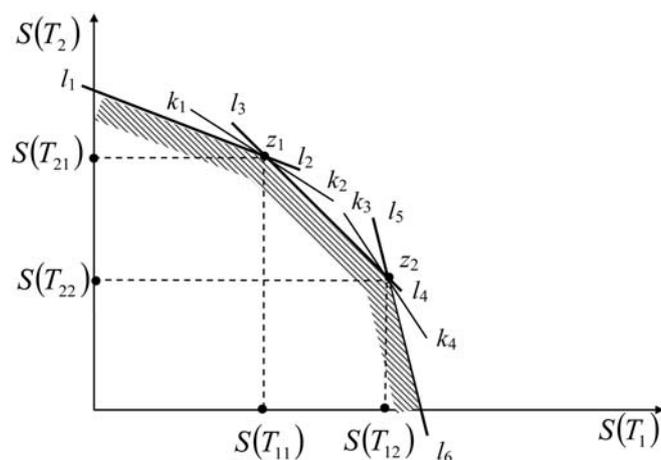
где $c_i = \frac{C_i}{\Delta T}$.

2.2. Вводится новый целевой функционал в виде линейной суммы

$$\Phi_1 = \sum_{j=1}^m b_j S(T_j), \quad (5)$$

где b_j — весовые коэффициенты.

2.3. Знаки равенства в уравнениях системы (4) заменяются на знаки "равно или меньше" и с учетом функционала (5) составляется и решается задача линейного программирования. Решение этой задачи позволяет значительно сузить множество оптимальных решений, найденных на первом уровне оптимизации. На втором уровне оптимизации в результате решения задачи линейного программирования выделяются только те прежние оптимальные решения, которые соответствуют узловым точкам, отображающим решение оптимизационной задачи линейного программирования. Например, при $m = 2$, $n = 3$ на первом уровне оптимизации можно получить три оптимальных решения при решении дискретного аналога задачи оптимизации (3). Однако на втором уровне оптимизации происходит сокращение одного из оптимальных



Графическая интерпретация оптимизационной задачи линейного программирования

решений, что иллюстрируется рисунком. Дадим необходимое пояснение по рисунку, где приняты следующие обозначения: отрезки линий l_1l_2 , l_3l_4 и l_5l_6 соответствуют следующим ограничениям:

$$a_{11}(T_1)S(T_1) + a_{12}(T_2)S(T_2) < c_1; \quad (6)$$

$$a_{21}(T_1)S(T_1) + a_{22}(T_2)S(T_2) < c_2; \quad (7)$$

$$a_{31}(T_1)S(T_1) + a_{32}(T_2)S(T_2) < c_3. \quad (8)$$

Отрезки линий k_1k_2 и k_3k_4 обозначают основные опорных плоскостей, соответствующих двум разным функционалам цели, используемым в оптимизации второго уровня, которые могут быть обозначены как

$$\Phi_1^{(1)} = b_{11}S(T_1) + b_{12}S(T_2); \quad (9)$$

$$\Phi_1^{(2)} = b_{21}S(T_1) + b_{22}S(T_2). \quad (10)$$

Как видно из рисунка, решение задачи линейного программирования в зависимости от выбора функционалов оптимизации (9) или (10) позволяет найти различные оптимальные узловые точки z_1 и z_2 и соответствующие оптимальные пары $[S(T_{21}), S(T_{11})]$ и $[S(T_{22}), S(T_{12})]$. При этом в каждом из этих решений не участвует одно уравнение систем (6)—(8), т. е. происходит сокращение множества решений, полученных на первом уровне оптимизации.

Рассмотрим возможность применения изложенного выше способа многоуровневой оптимизации для решения задачи оптимальной информационной оценки разнообразия видов в экологических системах.

Хорошо известно [10], что для оценки структурированности экологических систем используется формула К. Шеннона, определяющая информированность (степень упорядоченности) системы

$$H = - \sum_{i=1}^k P_i \log P_i, \quad (11)$$

где P_i — вероятность события.

Как указывается в [10], при вычислении разнообразия экосистем величину, выражающую количество информации на один элемент (особь, единицу биомассы и пр.), обозначают как \bar{H} :

$$\bar{H} = - \sum_{i=1}^m \frac{n_i}{N} \log_2 \frac{n_i}{N}, \quad (12)$$

где N — общее число элементов в экосистеме; n_i — число элементов данной группы; m — число групп.

Информация всей экосистемы или ее части в единице пространства (объем, площадь), равная

произведению \bar{H} на число элементов, определяется по формуле

$$H = - \sum_{i=1}^m n_i \log_2 \frac{n_i}{N}. \quad (13)$$

Далее для упрощения математических выкладок рассматривается упрощенный случай равенства числа элементов во всех группах. В этом случае формула (13) приобретает следующий вид:

$$H = N \log_2 S. \quad (14)$$

Рассматривая временную динамику развития экосистемы, можно предположить, что число групп, а также общее число элементов в экосистеме изменяются во времени. В этом случае выражение (14) может быть записано в следующем виде:

$$H = N(T) \log_2 S(T). \quad (15)$$

Введем на рассмотрение новый показатель, который может быть назван интегральной оценкой динамики разнообразия экологической системы:

$$H_n = \int_0^{T_{\max}} N(T) \log_2 S(T) dt. \quad (16)$$

Допустим, что известны или прогнозируются различные сценарии временного саморегулирования экологической системы. При этом критерием саморегулирования в смысле обеспечения стабильности экосистемы является равенство

$$\int_0^{T_{\max}} a(T) S(T) = C, \quad (17)$$

где $a(T)$ — мультипликативный коэффициент саморегулирования; $C = \text{const}$.

Так как сценариев динамики развития экосистемы несколько, то имеем

$$\int_0^{T_{\max}} a_i(T) S(T) = C_i; \quad i = \overline{1, n}. \quad (18)$$

С учетом формул (16) и (17) составим функционал условной оптимизации, подобный функционалу (3):

$$H_{uy} = \int_0^{T_{\max}} N(T) \log_2 S(T) dT + \lambda \int_0^{T_{\max}} a(T) S(T) dT. \quad (19)$$

Для решения вариационной задачи оптимизации применим метод Эйлера. Если представить функционал (19) в виде

$$H_{uy} = \int_0^{T_{\max}} W[N(T), S(T), \lambda, a(T)] dT, \quad (20)$$

то согласно методу Эйлера должно быть удовлетворено условие

$$\frac{dW}{dS(T)} = 0. \quad (21)$$

С учетом формул (19), (20) и (21) получаем

$$\frac{N(T)}{(\ln 2) S(T)} + \lambda a(T) = 0. \quad (22)$$

Из выражения (22) получаем

$$S(T) a(T) = - \frac{N(T)}{\lambda \ln 2}. \quad (23)$$

С учетом выражений (17) и (23) имеем

$$- \int_0^{T_{\max}} \frac{N(T)}{\lambda \ln 2} = C. \quad (24)$$

Из выражения (24) имеем

$$\lambda = - \int_0^{T_{\max}} \frac{N(T)}{C \ln 2}. \quad (25)$$

На основании выражений (22) и (25) находим

$$S(T) a(T) = \frac{CN(T)}{T_{\max} \int_0^{T_{\max}} N(T) dT}. \quad (26)$$

Очевидно, что для трех различных сценариев изменения $a(T)S(T)$ во времени можно получить различные решения, аналогичные формуле (26).

С учетом известных значений функций $a(T)$ и $N(T)$ из формулы (26) нетрудно составить следующие уравнения для трех разных сценариев изменения $S(T)$:

$$\left. \begin{aligned} S(T_1) a_{11}(T_1) + S(T_2) a_{12}(T_2) &= C_1; \\ S(T_1) a_{21}(T_1) + S(T_2) a_{22}(T_2) &= C_2; \\ S(T_1) a_{31}(T_1) + S(T_2) a_{32}(T_2) &= C_3, \end{aligned} \right\} \quad (27)$$

где

$$C_1 = C^{(1)} \left(\frac{N(T_1)}{\int_0^{T_{\max}} N(T) dT} + \frac{N(T_2)}{\int_0^{T_{\max}} N(T) dT} \right);$$

$$C_2 = C^{(2)} \left(\frac{N(T_1)}{\int_0^{T_{\max}} N(T) dT} + \frac{N(T_2)}{\int_0^{T_{\max}} N(T) dT} \right);$$

$$C_3 = C^{(3)} \left(\frac{N(T_1)}{\int_0^{T_{\max}} N(T) dT} + \frac{N(T_2)}{\int_0^{T_{\max}} N(T) dT} \right),$$

$C^{(1)}$, $C^{(2)}$, $C^{(3)}$ — значения постоянной C для трех сценариев.

Дальнейший ход рассуждений повторяет последовательность мыслей, изложенных в первой части статьи. Уравнения системы (27) превращаются в неравенства, аналогичные неравенствам (6), (7), (8). Далее определяется функционал цели аналогично (9) или (10).

Аналогично ранее изложенному на этапе оптимизации нижнего уровня выясняется вопрос о выборе оптимальной узловой точки z_1 или z_2 (см. рисунок). Выбор любой из этих узловых точек обеспечивает уменьшение числа решений, полученных на первом уровне, до двух, что дает существенную определенность для окончательного решения задачи оптимизации.

Отметим, что формирование неравенств задачи линейного программирования в предложенном методе многоуровневой оптимизации может быть осуществлено двумя методами:

- на основании ограничительного условия (2), как это было показано в первой части статьи при изложении теоретических основ предлагаемого метода;
- на основании решения оптимизационной задачи верхнего уровня, как это было сделано в рассматриваемом примере оптимизации био-разнообразия в экологических системах.

* * *

В заключение сформулируем основные выводы и положения проведенного исследования.

- ♦ Предложен новый многоуровневый метод оптимизации, где на верхнем уровне оптимизации решается задача условной вариационной оптимизации, а на нижнем уровне решается задача

линейного программирования, что позволяет существенно уменьшить число оптимальных решений, полученных на первом уровне.

- ♦ На основе предложенного метода многоуровневой оптимизации сформулирована и решена задача оптимальной оценки разнообразия видов в экологических системах.
- ♦ Показано, что неравенства, используемые при решении оптимизационной задачи нижнего уровня методом линейного программирования, могут быть сформулированы как на основе ограниченного условия верхнего уровня оптимизации, так и решения задачи оптимизации верхнего уровня.

Список литературы

1. **Goertzel B.** Structure of Intelligence. URL: http://www.goertzel.org/books/intel/chaptel_two.html
2. **Шимко П. Д.** Оптимальное управление экономическими системами: учеб. пособие. СПб.: Издательский дом "Бизнес-пресса", 2004. 240 с.
3. **Абдуллаев А. А., Алиев Р. А., Уланов Г. М.** Принципы построения автоматизированных систем управления промышленными предприятиями с непрерывным характером производства // Под ред. акад. Б. Н. Петрова. М.: Энергия, 1975.
4. **Месарович М., Махо Д., Такахара И.** Теория иерархических многоуровневых систем. М.: Мир, 1973. 344 с.
5. **Lasdon L. S.** Optimization theory for large systems London, 1970. 523 p.
6. **Brosilon C., Nunez E.** Multi-level optimization applying to catalytic cracting plant // Canadian Journal of Chemical Engineering. 1968. N 3. P. 205—211.
7. **Абдуллаев А. А., Уланов Г. М., Алиев Р. А.** Вопросы теории, построения и внедрения инвариантных систем автоматического управления в химической, нефтехимической и нефтеперерабатывающей промышленности // Теория инвариантности и теория чувствительности автоматических систем. Ч. III. Киев: КВИАВУ ВВС, 1971. С. 717—743.
8. **Матвеев Ю. А., Ламзин В. В.** Метод многоуровневой оптимизации параметров модификации космических аппаратов и модернизации космической системы ДЗЗ // Научные чтения памяти К. Э. Циолковского, Калуга, 2007. URL: <http://readings.gmik.ru/lecture/2007-METHOD-MNOGOUROVNEV...>
9. **Choudhary R., Papalambros P. Y., Malkawi A. M.** Simulation — based design by bierchical optimization // Proceedings of Ninth International IBPSA Conference. Montreal, Canada. August 15—18. 2005. P. 175—182.
10. **Дулупов В. И., Лескова О. А., Майоров И. С.** Системная экология. Сайт цифровых учебно-методических матер. ВГУ-ЭС//abc.vvsu.ru/page0032. URL: http://abc.vvsu.ru/Books/sistemnaya_ekologiya_up/default.asp.

УДК 53.082.9:612.172.4 + 612.081.2

А. А. Кузнецов, канд. физ.-мат. наук., доц.,
Владимирский госуниверситет,
e-mail: artemi-k@mail.ru

Системная связь между процессами ритма сердца и динамики систолического потенциала

По данным обработки 58 двадцатиминутных регистраций ЭКГ условно здоровых молодых людей разного пола проведен поиск системной взаимосвязи между процессами ритма сердца и динамики систолического потенциала. Применение интегрального метода анализа и двухпараметрической гистограммы позволило выделить нелинейную связь в относительно узком интервале распределения значений R-R интервалов и трактовать данные ЭКГ с позиций предложенных механизмов взаимосвязи. Приведены характеристики этих механизмов. Предложена модель системной взаимосвязи. Обнаружено, что структура кардиосигнала формируется в рамках кардиоинтервала и задается ритмом, а управление и регуляция ритма осуществляется посредством изменения систолического потенциала.

Ключевые слова: электрокардиограмма, ритм сердца, систолический потенциал, системная связь, двухпараметрическая гистограмма

Постановка задачи

Результат системной циклической работы сердца реализуется на электрокардиограмме (ЭКГ) в форме последовательности участков с качественно одинаковой топологической структурой. Каждый такой участок (кардиоинтервал) представляет полную последовательную картину изменения электрического потенциала в течение цикла и характеризуется граничными значениями макропараметров: амплитудным значением зубца R и R - R интервалом [1–3]. Каждый пик зубца R соответствует, с одной стороны, максимальному значению электрического потенциала, а с другой стороны, сигналу на сокращение миокарда желудочков сердца. Для здоровых молодых людей эти события совпадают по моменту времени. Кроме этого в прямоугольной системе координат ЭКГ указанные параметры ортогональны и потому не могут напрямую влиять друг на друга. Это позво-

ляет применить технологию обработки ЭКГ, сохранения данных и анализа динамической параметрической связи построением амплитудно-фазовой характеристики (АФХ) ЭКГ для поиска системной связи, определяющей опосредованные причинно-следственные отношения между динамиками этих параметров в рамках механизмов регуляции и управления функциями системы сердца [4]. Ставится задача определения параметра первичной или исходной динамики в этих отношениях по форме и свойствам АФХ.

Цель работы — поиск системной взаимосвязи между процессами ритма сердца и динамики систолического потенциала.

Экспериментальная часть

Двадцатиминутные регистрации ЭКГ условно здоровых обследуемых (УЗО) проводились монитором Холтера лицензированного комплекса амбулаторной регистрации электрокардиосигнала "AnnAFlash3000" [5] с применением двухполюсных отведений (по Небу), соответствующих стандартному отведению II с максимальной амплитудой зубцов R [3–5].

Регистрации ЭКГ проводили в состоянии относительного покоя в положении сидя для двух групп УЗО студентов обоих полов в возрасте 18–24 года. Разовые регистрации ЭКГ первой смешанной группы (17 человек, 17 регистраций) проводили по 2–3 через неделю в течение февраля — марта 2008 года в дневное (14⁰⁰–16⁰⁰) и вечернее (18⁰⁰–20⁰⁰) время. Комбинированные регистрации ЭКГ второй группы (8 девушек, 41 регистрация) проводились по 3–4 через неделю в течение сентября — ноября 2009 года в дневное (12⁰⁰–14⁰⁰) время.

В программе обработки, записи и хранения информации "Анализ ЭКГ" определялись координаты пиков зубцов R и в текстовом формате записывались и сохранялись цифровые ряды последовательностей R - R интервалов, амплитудных значений зубцов R , расчетных значений площадей S_R и средних значений электрического потенциала φ_S^* кардиоциклов [6, 7]. По длине записи ЭКГ уровень изоэлектрической линии не постоянен, в частности вследствие проявлений дыхательного ритма, поэтому за условный "ноль" принимали минимальное значение разности потенциалов. Применение общей для всей длины записи горизонтальной линии "нулевого уровня" при оп-

ределении амплитудных значений зубцов R позволяло учитывать захваченные ритмом сердца естественные волновые процессы внутренних и внешних влияний [3].

Результаты регистрации, обработки и анализа ЭКГ

Процесс ритма сердца реализуется последовательностью R - R интервалов размером n на ЭКГ, определенной в цифровом или графическом формате в виде условного процесса $RR(n)$. Динамика систолического электрического потенциала реализуется последовательностью амплитудных значений зубца R и определена в цифровом или графическом формате в виде условного процесса $R(n)$. Удельная электрическая энергия электропроводящей системы сердца (ЭПСС), реализуемая в морфологию кривой кардиоцикла, оценивается значением величины S_R . По технологии расчета эта величина представляет площадь, ограниченную кардиоинтервалом на кривой ЭКГ и "нулевой линией".

Интегральная величина S_R , заключая в себе информацию об обоих макропараметрах кардиоцикла в форме "свертки", является наиболее информативной. С ее помощью в круг анализа вводятся относительные параметры: среднее значение элек-

трического потенциала кардиоцикла $\phi_S^* = S_R/RR$, среднее значение электрического потенциала ЭКГ $\langle \phi_S^* \rangle = S_R(n)/RR(n)$, а также среднеинтегральное значение $RR_S^* = SR(n)/R(n)$ [6, 7].

Резкие амплитудные изменения на диаграмме $RR(n)$ (рис. 1, а) сопровождаются импульсными откликами на диаграмме $R(n)$ (рис. 1, в) и усилены в 4 раза на диаграммах $S_R(n)$ и $S_R/RR(n)$ (рис. 1, б, г). Это указывает на чувствительность к изменениям не только амплитуды зубца R , но и иных морфологических составляющих кардиоциклов на ЭКГ.

Диаграмма $RR(n)$ имеет среднее значение $\langle RR \rangle = 631$ мс, среднее квадратическое отклонение $\sigma_{RR} = 49$ мс и одновершинную гистограмму с коэффициентом асимметрии $A_{RR} = 0,578$. Диаграмма $R(n)$ имеет среднее квадратическое отклонение $\sigma_R = 0,2$ мВ и гистограмму с тремя вершинами с координатами R : 2,47, 2,67 и 2,87 мВ. Модальные участки распределения отчетливо выделяются даже на графике $R(n)$ в форме "каймы" на нижней и верхней границах диаграммы и среднего участка наполнения (рис. 1, в).

Превалирование вклада R - R интервалов или амплитудных значений зубца R при формировании цифрового ряда $S_R(n)$ удобно исследовать на диаграммах $S_R(RR)$ и $S_R(R)$ (рис. 2, а, б). Во всех случаях очевидна практически линейная функциональная зависимость $S_R(RR)$ по всей длине диаграмм $S_R(n)$ и $RR(n)$, и, напротив, сложная нелинейная связь на фоне сильного шума между параметрами S_R и R .

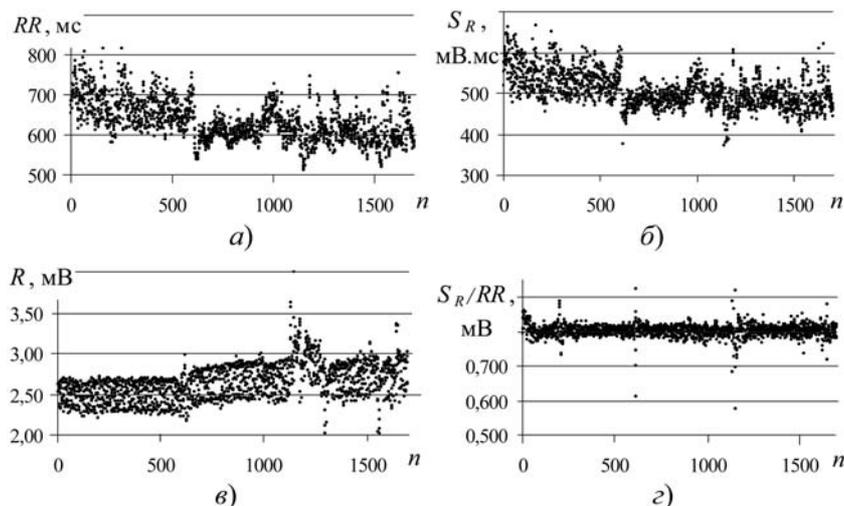


Рис. 1. Временные параметрические диаграммы ЭКГ УЗО (06.10.09. 12³⁰): RR -грамма (ритмограмма) (а); S_R -грамма (б); R -грамма (в) и S_R/RR -грамма (г). Размер $n = 1700$ отсчетов

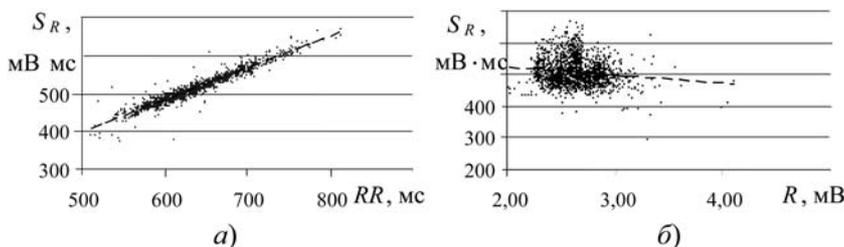


Рис. 2. Диаграммы $S_R(RR)$ (а) и $S_R(R)$ (б), рассчитанные по данным рис. 1, а, б, в. Приведены линии тренда: с уравнением $S_R = 0,84RR - 26$ при достоверности аппроксимации $r^2 = 0,92$ и уравнением $S_R = -25,7R + 574$ при $r^2 = 0,02$

Двухпараметрическая гистограмма $RR(R, m)$

Диаграмма $R(RR)$, представляющая АФХ или вольт-секундную характеристику ЭКГ и построенная по двум временным параметрическим диаграммам $R(n)$ и $RR(n)$ (см. рис. 1, а, в), представлена на рис. 3, а. Множество точечных значений разбито на три подмножества также как и на диаграммах $R(n)$ (см. рис. 1, в) и $S_R(R)$ (см. рис. 2, б). Такая форма точечного распределения представляет интерес для фрактального анализа [8—10].

При применении статистического метода анализа диаграммы, приведенной на рис. 3, а, и для удобства визуального восприятия проведем осреднение согласованных цифровых рядов $RR(n)$ и $R(n)$ по методу средневзвешенных значений и по заданному постоянному интервалу $\sigma_R/4 = 0,05$ мВ

и поменяем местами координаты на графике рис. 3, а. Полученный экспериментальный полигон двухпараметрической гистограммы (ДПГ) в форме массовой диаграммы $RR(R, m)$ на фазовой плоскости параметров (рис. 3, б) позволяет избавиться от проявлений непрерывной высокочастотной генерации импульсов и определяет распределение и по исследуемым параметрам, и по представительству m их значений по интервалам усреднения. На диаграмме $RR(R, m)$ значения m выделены размерами точек: 50 — 2 пикселя; 50—100 — 3 пикселя; 100—150 — 4 пикселя; 200—300 — 6 пикселей, ... (рис. 3, б). Создание массовой диаграммы $RR(R, m)$ методически ничем не отличается от построения полигона гистограммы по одному параметру. Выбранный интервал осреднения (карман) согласованных параметрических рядов равен 0,05 мВ, а их количество не превышает удвоенного числа k столбцов гистограммы, определенное по формуле Старджеса [11].

Применение ДПГ позволяет выделить статистическую нелинейную связь в диапазоне значений R - R интервалов шириной не более $\pm\sigma_{RR}$ и трактовать данные рис. 1 с позиций разных ритмических режимов, определяемых разными интервалами значений систолического потенциала. На общем, монотонно падающем уровне средневзвешенных значений R - R интервалов последовательно выделяются по оси $0R$ три ритмических режима (рис. 3, б). Кривая ДПГ имеет форму фигуры из трех сопряженных параболических кривых ($\cap\cap\cap$), что позволяет также уточнить местоположения максимумов на гистограмме диаграммы $R(n)$: 2,3; 2,65 и 2,87 мВ.

Обсуждение экспериментальных результатов

Наличие волновых и импульсных структур на диаграмме $RR(n)$ предопределяет характер изменений как макроструктуры диаграммы $R(n)$, так и формы ее полигона распределения. Если на графике $RR(n)$ выделяется волна влияния с периодом 1000 отсчетов (12 мин) со слабым (10...15 %) амплитудным размахом 80...100 мс, то тренд графика $R(n)$ синхронно повторяет тренд графика $RR(n)$ с тем же периодом. На периодах не более 200 отсчетов и амплитудным размахом не более 100 мс динамика средних уровней параметров находится в прямой зависимости.

Если амплитуда волны на диаграмме $RR(n)$ превышает 100 мс, то связь становится обратной. Например, при падении среднего уровня на диаграмме $RR(n)$ на 20 %, средний уровень на диаграмме $R(n)$ растет на 15 %, а на диаграмме $S_R(n)$ падает на 10 %. На диаграмме $S_R/RR(n)$ возникают импульсы шириной 10—50 отсчетов и амплитудным размахом 1,5—2,5 мВ (см. рис. 1, з). Они приходятся на наиболее резкие амплитудные из-

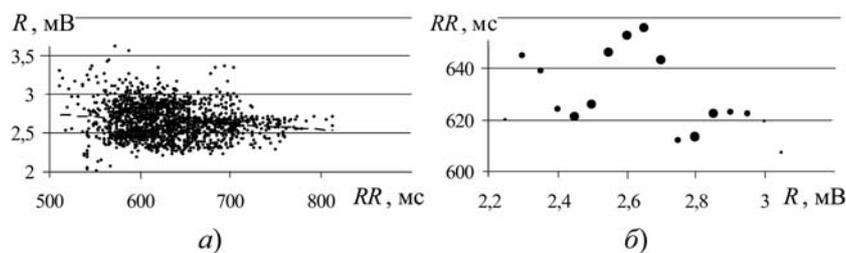


Рис. 3. Диаграмма распределения $R(RR)$ по рис. 1, а, в с линией тренда $R = -RR \cdot 10^{-4} + 3,1$ при $r^2 = 0,0223$ (а); двухпараметрическая гистограмма распределения $RR(R, m)$, где m — вариационный ряд отсчетов значений (б)

менения ритма. Процесс $R(n)$ "гасит" эти изменения, но с переменным запаздыванием 1—5 с, поэтому динамика среднего уровня потенциала кардиоцикла очень неустойчива на протяжении всей записи. Механизм обратной связи тормозит падение среднего уровня на диаграмме $S_R(n)$.

Медленный рост среднего уровня на диаграмме $RR(n)$ сопровождается медленным подъемом среднего уровня диаграммы $R(n)$. При этом фазовые импульсации с амплитудой 50 мс и периодом 80—110 отсчетов вызывают импульсные фрагментарные изменения на диаграмме $R(n)$. Кратковременный неглубокий спад на диаграмме $RR(n)$ сопровождается малым спадом и на диаграмме $R(n)$. Таким образом, малая скорость изменения среднего уровня на диаграмме $RR(n)$ приводит к прямой связи между средневзвешенными значениями $\langle RR_i \rangle$ и $\langle R_i \rangle$ по интервалу 0,05 мВ.

Быстрое падение среднего уровня на диаграмме $RR(n)$ не приводит к резкому падению соответствующих значений S_R . Появляется своеобразный защитный механизм, проявляющийся в росте среднего уровня на диаграмме $R(n)$. Результат работы этой защиты можно оценить качественно — по форме диаграммы $S_R/RR(n)$ или количественно — по достоверности аппроксимации линейной зависимости $S_R(RR)$, определяемой по значению коэффициента Пирсона r^2 . Для 57 проведенных регистраций ЭКГ значение r^2 не было ниже 0,85. Замечено, что с ростом коэффициента корреляции между рядами $RR(n)$ и $R(n)$ линейная связь между рядами $S_R(n)$ и $RR(n)$ слабеет. Это подтверждает известные выводы о возможности использования критерия S_R для оценки общего состояния организма человека [6, 7].

При резком падении среднего уровня на диаграмме $R(n)$ нижняя граница этой диаграммы "рвется", средняя область наполнения "сыплется", а верхняя граница лишь деформируется. Так происходит при прямой связи. При резком росте среднего уровня на диаграмме $RR(n)$ синхронно "рвутся" и верхняя, и нижняя граница диаграммы $R(n)$, а область наполнения "высыпается вверх" (см. рис. 1, в). Так происходит при обратной связи. При любом резком изменении среднего уровня $RR(n)$ нижняя граница диаграммы $R(n)$ исчезает, а форма фигуры на ДПГ меняет качество.

По данным 58 регистраций встречались следующие простые формы фигур ДПГ: линейная (J) — 13 регистраций (рис. 4, z); нижняя половина S -образной кривой (S) — 6 (рис. 4, a); верхняя половина S -образной кривой (Γ) — 3; S -образная кривая (S) — 5 (рис. 4, b); параболическая кривая (\cap) — 16; вертикальный острый угол (Λ) — 7 (рис. 4, $в$). В восьми случаях встречались сложные фигуры составных форм: ($\cap\cap$, $\cap\cap\cap$, $\Lambda\Lambda$, $\Lambda\cap$, $J\cap$, $J\cap/J$).

Наибольший интерес представляет переменная связь параметров, при которой кривая графика ДПГ (\cap , Λ) делится почти поровну около среднего значения — "ломается" по границе превалирования прямой или обратной связи (рис. 4, $в$). Превалирование прямой или обратной связи определяется представительностью значений для форм \cap и Λ на той или иной ветке кривой ДПГ. Для графиков ДПГ с формами фигур $\cap\cap$, $\cap\cap\cap$, $\Lambda\Lambda$ характерна знакопеременная связь между макропараметрами в форме волны ("волновая связь") (см. рис. 3, b), образуемая уплощенным одномодальным распределением диаграммы $RR(n)$ и трехвершинным распределением диаграммы $R(n)$.

С одной стороны, при одновершинном распределении по диаграмме $RR(n)$ формы фигур ДПГ полностью определены формой кривой полигона распределения по диаграмме $R(n)$. Если на полигоне распределения диаграммы $R(n)$ представлена лишь одна мода, то эти фигуры ДПГ обретают простые формы. С другой стороны, все простые фигуры являются фрагментами сложных фигур, форма которых определена числом мод и характером распределения по диаграмме $R(n)$. Таким образом, форму искомой взаимосвязи предопределяет процесс $R(n)$.

Вне зависимости от пола УЗО, времени года и суток данные средних значений параметров поддерживают линейную связь (рис. 5, $a, б$). Средние значения параметров на групповых диаграммах захватывают сектор на фазовой плоскости. Распределение групповых средних значений параметров внутри фазовых секторов позволяет проводить классификацию обследуемых по признаку отношения $\langle R \rangle / \langle \varphi_s^* \rangle$. Это отношение, определенное на диаграмме рис. 5, a как тангенс угла наклона трендовой линии, показывает, насколько средний уровень систолического потенциала превосходит средний уровень потенциала кардиоцикла, оценивая относительный вклад других участков кардиоциклов.

Для всех $N = 58$ регистраций формирование ряда $S_R(n)$ происходит за счет соответствующей ритмической составляющей с организацией линейной функциональной связи (см. рис. 2, a), однако между средними значениями $\langle S_R \rangle$ и $\langle RR \rangle$ отсутствует даже статистическая связь.

Системная связь ритма сердца и динамики потенциала систолы

По принципу частотного кодирования ослабление интенсивности воспринимаемого организмом человека внешнего стимула любой природы приводит к падению частоты разрядов генераторного потенциала [12]. Разреженный сигнал меньшей частоты заставляет расти средний уровень диаграммы $RR(n)$, что приводит к возможности большего кровенаполнения желудочков сердца. Для выброса увеличенного ударного объема крови при систоле желудочков требуется большая энергия, поэтому амплитуда систолического электрического сигнала должна расти.

При реализации обратной связи, определяемой по форме соответствующих простых и любых сложных фигур

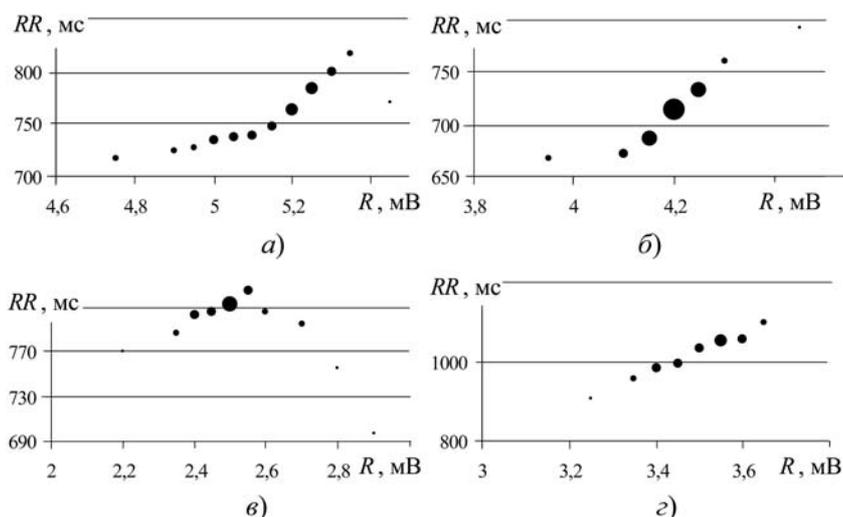


Рис. 4. Формы диаграмм $RR(R, m)$ с принятыми обозначениями: J — 1-го УЗО 06.10.09. 13⁰⁰ (a); S — 2-го УЗО 29.09.09. 13³⁰ ($б$); Λ — 3-го УЗО 06.10.09 12³⁰ ($в$); $/$ — 4-го УЗО 06.10.09. 12³⁰

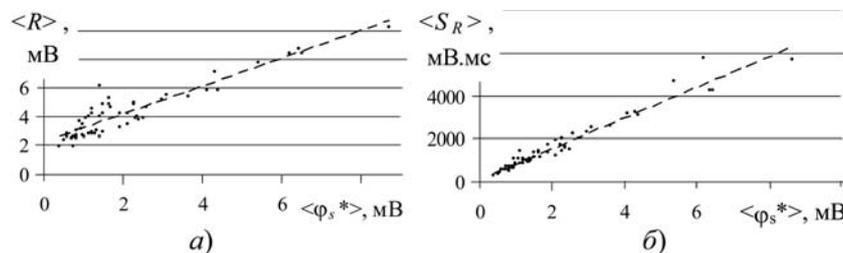


Рис. 5. Точечные групповые диаграммы зависимостей: $\langle R \rangle \{ \langle \varphi_s^* \rangle \}$ с уравнением линии тренда $\langle R \rangle = 0,98 \langle \varphi_s^* \rangle + 2$ [мВ] при достоверности $r^2 = 0,86$ (a) и $\langle S_R \rangle \{ \langle \varphi_s^* \rangle \}$ с уравнением линии тренда $\langle S_R \rangle = 714 \langle \varphi_s^* \rangle + 111$ [мВ · мс] при достоверности $r^2 = 0,95$ ($б$)

на ДПГ, результаты серийных измерений второй группы обследуемых показывают сопровождающееся резкое падение среднего уровня на диаграммах $RR(n)$. Это приводит к сжатию интервала общей паузы кардиоинтервала до критического значения. Исчезает своеобразный "люфт упругой деформации" записи ЭКГ. В этих случаях резко падает значение σ_{RR} , ритм становится более "жестким".

Основываясь на данных ДПГ лишь 58 регистраций ЭКГ УЗО, можно предположить, что возникновение обратной связи вызвано одной общей причиной — резким изменением среднего уровня на диаграмме $RR(n)$. Для механизма обратной связи характерны относительная задержка (до 5 с) вторичного процесса $R(n)$ и зависимость формы его распределения от скорости изменения среднего уровня на диаграмме $RR(n)$. В рамках этого предположения механизм обратной связи может иметь индукционную природу: не позволяет развивать большие скорости изменений площадей и среднего уровня потенциала и является защитным для сохранения морфологии комплексов при сильном уменьшении интервала общей паузы.

Индукционный механизм защиты включается в области верхней границы (третий режим на диаграмме $R(n)$). Поэтому в большинстве случаев на уровне наиболее вероятного значения ($d(\langle RR_0 \rangle)/d(\langle R_0 \rangle) = 0$) на ДПГ возникает конкуренция прямого и обратного механизмов связи: механизма регуляции амплитуды электрического сигнала для выброса задаваемого ритмом объема крови и механизма защиты морфологии кардиоцикла как системной единицы. Последовательность стадий действия такого комбинированного механизма в процессе перебора вариантов приводит к результирующему отсутствию корреляционной связи в интервале регистрации и впечатлению хаоса на диаграмме $R(RR)$.

Таким образом, топологическая структура кривой кардиосигнала формируется в рамках кардиоинтервала и задается ритмом сердца, а регуляция ритма выполняется посредством изменения электрического потенциала при систоле с задержкой от 1 до 5 с, что, вероятно, можно объяснить дыхательным ритмом. Первичными являются ритмические процессы внутренних и внешних влияний, а динамика систолического потенциала ($R(n)$) относится к реализации непрерывного запаздывающего действия механизмов регуляции и управления.

Выводы

1. Технология обработки ЭКГ построением по двум ее макропараметрам амплитудно-фазовой характеристики (АФХ) привело к двум возможным путям анализа динамической параметрической связи в согласованной последовательности амплитудных значений зубца R и временных $R-R$ интер-

валов: фрактальному анализу точечной АФХ на фазовой плоскости и статистическому анализу распределения в форме экспериментального полигона двухпараметрической гистограммы (ДПГ).

2. При использовании статистического метода анализа диаграмм $RR(R)$ экспериментальный полигон ДПГ в форме массовой диаграммы $RR(R, m)$ на фазовой плоскости параметров выделяет статистическую взаимосвязь двух согласованных параметрических рядов $RR(n)$ и $R(n)$ с классификацией по форме фигур ДПГ и распределению параметра m .

3. Волновые и импульсные структуры на диаграмме $RR(n)$ предопределяют характер изменений как макроструктуры диаграммы $R(n)$, так и формы ее полигона распределения. Если изменения средних уровней на диаграммах $RR(n)$ и $R(n)$ происходят медленно, в рамках значений стандартных отклонений и интервала общей паузы, то работает механизм прямой взаимосвязи, определенный регуляцией амплитуды электрического сигнала при систоле, задаваемый ритмом объема крови. В противном случае (быстрые изменения) включается индукционный механизм обратной связи, определенный энергетической моделью ЭКГ.

4. На ДПГ в области наиболее вероятного значения параметра выделяется эффект конкуренции двух механизмов взаимосвязи: механизма регуляции амплитуды электрического сигнала при переменном ударном объеме крови и механизма защиты морфологии кривой кардиоцикла. Формы фигур и распределений на ДПГ могут служить операторами "жесткой" и "мягкой" связи механизмов регуляции при генерации ЭКГ сигнала.

5. В общем случае формой системной связи, определяющей опосредованные причинно-следственные отношения между динамикой макропараметров в рамках механизмов регуляции и управления функциями системы сердца здорового человека, является форма "волновой связи".

6. Топологическая структура кривой кардиосигнала формируется в рамках кардиоинтервала и задается ритмом сердца, а регуляция ритма осуществляется посредством изменения электрического потенциала при систоле с задержкой от 1 до 5 с.

Список литературы

1. Шмидт Р., Тевс Г. Физиология человека: В 3 т. Т. 2. М.: Мир, 1996. — 313 с.
2. Heart rate variability. Standards of measurement, physiological interpretation, and clinical use. Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology // European Heart J. 1996. V. 17. P. 354—381.
3. Баевский Р. М., Берсенева А. П. Введение в донозологическую диагностику. М.: Слово, 2008. 176 с
4. Кузнецов А. А., Плеханов А. А., Степанова Д. В. $R-RR$ характеристика электрокардиограммы человека // Физика и радиоэлектроника в медицине и экологии. ФРЭМЭ'2010. — Владимир-Суздаль: Изд. ВлГУ. 2010. С. 119—123.

5. Прилуцкий Д. А., Кузнецов А. А., Плеханов А. А. Накопитель ЭКГ "AppA Flash2000" // Методы и средства измерений физических величин. Н. Новгород: Изд. НГТУ. 2006. С. 31.
6. Кузнецов А. А. Методы анализа и обработки электрокардиографических сигналов: Новые подходы к выделению информации. Владимир: Изд. ВлГУ. 2008. 140 с.
7. Кавасма Р., Кузнецов А., Сушкова Л. Автоматизированный анализ и обработка электрокардиографических сигналов. Методы и система. М.: Сайнс-пресс, 2006. 144 с.
8. Федер Е. Фракталы: пер. с англ. М.: Мир. 1991. 254 с.

9. Меклер А. А. Применение аппарата нелинейного анализа динамических систем для обработки сигналов ЭЭГ // ВНМТ. 2007. Т. XIV, № 1. С. 73—79.
10. Кузнецов А. А., Плеханов А. А. О технике оценки фрактальной размерности фазовых портретов динамических систем // Физика и радиоэлектроника в медицине и экологии. ФРЭ-МЭ'2008. В 2-х кн. Кн. 1. — Владимир-Суздаль: Изд. ВлГУ. 2008. С. 152—157.
11. Новицкий П. В., Зограф И. А. Оценка погрешностей результатов измерений. Л.: Энергоатомиздат. 1991. 304 с.
12. Рубин А. Б. Биофизика: В 2 т. Т. 1: Теоретическая биофизика: учебник для вузов. М.: Университет, 1999. 448 с.

УДК 004.942

В. А. Галкин*,

д-р физ.-мат. наук, проф., зав. каф.,

Ю. А. Белый**, д-р мед. наук, проф.,

зам. директора по научной работе,

А. А. Кучеров*, аспирант,

e-mail: adomis@yandex.ru; kaluga40@mail.ru

* Обнинский институт атомной энергетики (ИАТЭ) Национального ядерного исследовательского университета (НИЯУ) Московского инженерно-физического института (МИФИ)

** Калужский филиал ФГУ "МНТК

"Микрохирургия глаза" им. акад. С. Н. Федорова Росмедтехнологии"

Информационное и программное обеспечение антимикробной фотодинамической терапии

Рассматриваются алгоритм и программное обеспечение, позволяющие осуществлять персонализированный расчет дозы лазерного облучения, объема и площади поверхности витреальной полости глаза. Излагаются алгоритм способа количественной оценки уровня флуоресценции и реализация данного алгоритма в виде программного обеспечения для анализа фотографий глазного дна пациента. Представлена разработанная трехмерная модель витреальной полости глаза человека, позволяющая изменять свои пропорции в соответствии с индивидуальными особенностями глазного яблока пациента.

Ключевые слова: программное обеспечение, алгоритм, информационное обеспечение, трехмерная модель витреальной полости глаза человека, антимикробная фотодинамическая терапия

Характеристика предметной области

В настоящее время в офтальмологическом центре Калужского филиала МНТК "Микрохирургия глаза" одним из успешно применяемых методов

лечения глазных заболеваний является антимикробная фотодинамическая терапия (АФДТ), в ходе которой осуществляются введение специальных лекарственных препаратов в глаза, а также облучение лазером витреальной полости глаза человека.

Изучение возможности проведения АФДТ невозможно без проведения точных расчетов дозы лазерного излучения. Необходимость безошибочного расчета дозы лазерного облучения при АФДТ обусловлена, с одной стороны, опасностью превышения необходимой мощности облучения и количества вводимого светочувствительного вещества, а с другой — представляет определенные трудности вследствие сложной геометрии витреальной полости и глаза в целом.

Необходимость введения в полость глаза веществ с различными физико-химическими свойствами, а также учет изменений нормальной анатомии глаза вследствие предшествующих вмешательств и травм делают необходимым персонализированный подход к расчету объема и площади витреальной полости глаза на основе индивидуальных параметров глаза. Индивидуальными параметрами глаза в данном случае приняты: размер наибольшего сечения, толщина хрусталика и расстояние от задней стенки хрусталика до заднего полюса глазного яблока.

На сегодняшний день в офтальмологии используются усредненные значения размера и объема витреальной полости [1].

Особенно актуально применение алгоритма расчета объема и площади витреальной полости в лечении заболеваний, требующих многоэтапных и неоднократных вмешательств [2].

Для уменьшения времени, затрачиваемого специалистом на принятие решения о состоянии пациента в процессе лечения заболевания в ходе фотодинамической терапии, разработан метод флуоресцентной ангиографии глазного дна, т. е. метод получения фотографий глазного дна пациента. Для этого пациенту заблаговременно вводится в глаза препарат, который имеет свойство

накапливаться в пораженных участках и впоследствии светиться под воздействием света. Такое свойство веществ называется флюоресценцией. Достоверность полученных результатов во многом зависит от квалификации специалиста, осуществляющего расшифровку флюоресцентных ангиограмм и, соответственно, носит субъективный характер. Поэтому актуальной является разработка методов количественного анализа данных флюоресцентной ангиографии, а именно — уровня флюоресценции, которые позволят специалисту сделать заключение о состоянии пациента, подкрепленное конкретными численными показателями [3].

Цели и задачи работы

Целями работы явились разработка алгоритмов определения уровня флюоресценции, а также расчет дозы лазерного облучения, объема и площади поверхности витреальной полости глаза человека.

Для реализации целей работы был выделен ряд задач, в числе которых:

- построение трехмерной модели витреальной полости глаза человека, позволяющей изменять свои пропорции в соответствии с индивидуальными особенностями глазного яблока пациента;
- разработка программного обеспечения, позволяющего осуществлять расчет дозы лазерного облучения, объема и площади поверхности витреальной полости глаза;
- разработка программного обеспечения для анализа фотографий глазного дна пациента и определения уровня флюоресценции.

Данная проблема ранее не рассматривалась. В доступной литературе авторам не удалось обнаружить решения подобных задач и достижения аналогичных целей.

В результате работы по решению задачи разработки программного обеспечения для анализа фотографий глазного дна пациента и определения уровня флюоресценции был получен патент на изобретение № 2387471, дата регистрации 27.04.2010 [4].

Основные этапы работы

Рассмотрим поэтапно реализацию решения задач с описанием перечня экспериментальных данных, оценки их значимости, способов получения.

На *первом этапе* исследования была проведена компьютерная S-Томография (томограф *Ultra Z Markoni Medical Systems* (США)) десяти глаз человека. В результате были получены серии томографических срезов с шагом 1,0 мм. На основе литературных данных: глаз имеет не совсем правильную шаровидную форму, длина его сагитальной оси в среднем равна 24 мм, горизонтальной — 23,6 мм, вертикальной — 23,3 мм [1]. Таким образом, можно

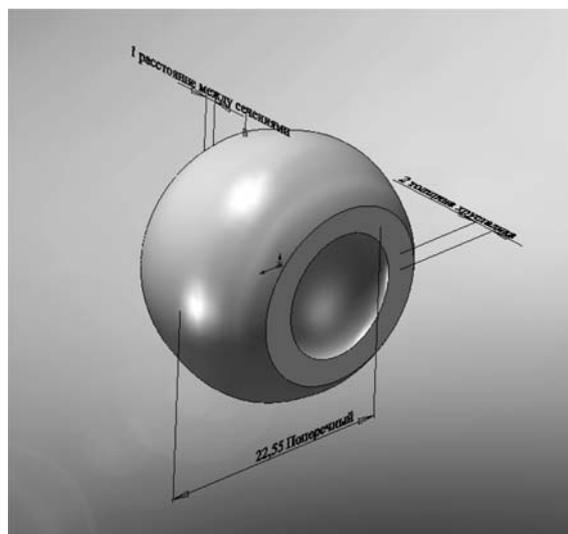


Рис. 1. Трехмерная модель витреальной полости глазного яблока человека

сделать вывод, что сечение глазного яблока является эллипсом, имеющим горизонтальный и вертикальный габаритные размеры. Определение размеров проводилось с помощью программного обеспечения, входящего в состав томографа *UltraZ*.

В ходе *второго этапа* была построена трехмерная модель витреальной полости глаза (рис. 1). Модель построена таким образом, что позволяет принимать пропорции в соответствии с индивидуальными размерами глазного блока пациента. Все стадии построения модели проводились в системе автоматизированного проектирования *SolidWorks*.

Изучение модели витреальной полости позволило разработать алгоритм расчета объема и площади боковой поверхности витреальной полости [5, 6]:

$$V = \sum_{i=1}^{17} \frac{\pi h_i}{3(a_{i1} - a_{i2})} (a_{i1}^2 b_{i1} - a_{i2}^2 b_{i2}); \quad (1)$$

$$S_{\text{бок}} = \sum_{i=1}^{17} \pi \sqrt{h_i^2 + \left(\frac{a_{i1} + b_{i1}}{2} - \frac{a_{i2} + b_{i2}}{2}\right)^2} \times \left(\frac{a_{i1} + b_{i1}}{2} + \frac{a_{i2} + b_{i2}}{2}\right), \quad (2)$$

где a_{i1} , b_{i1} и a_{i2} , b_{i2} — полуоси оснований усеченных эллиптических конусов; h_i — высота усеченного эллиптического конуса.

На основе полученных результатов расчета появляется возможность найти значения достаточно важных величин, таких как плотность энергии лазерного излучения и ее распределение по времени.

Третий этап заключался в создании программного обеспечения на основе разработанного алгоритма расчета объема и площади боковой поверхности витреальной полости.

Основная рабочая область (рис. 2) представляет собой поля для введения исходных данных. Она максимально адаптирована и упрощена для возмож-

ности успешной работы врача с различным уровнем подготовки обращения с компьютером.

Для более удобного восприятия и оценки исходные данные и полученные результаты консолидируются в таблице статистики (рис. 3).

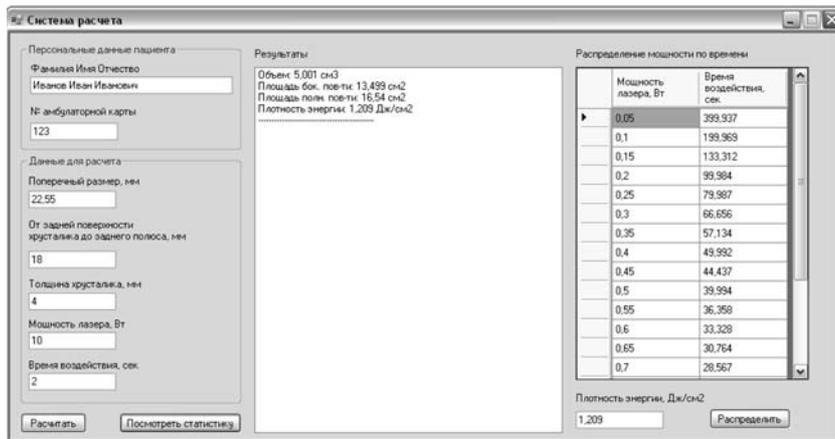


Рис. 2. Основная рабочая область

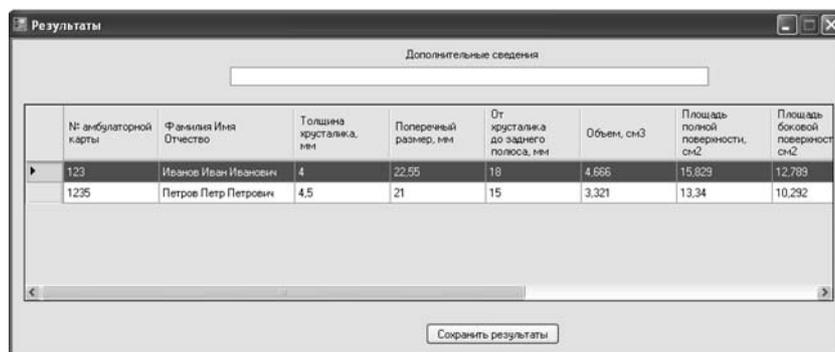


Рис. 3. Таблица результатов



Рис. 4. Основное рабочее поле модуля анализа флюоресценции

Если по мнению специалиста-пользователя результаты представляют какой-либо интерес, то для дальнейшего использования информацию можно сохранить в файле одного из трех самых популярных форматов doc, xls, html.

Разработанное на данном этапе программное обеспечение выполняет все возложенные на него операции, времени на обучение пользования основными функциями затрачивается очень мало, полученные результаты можно использовать в любое время после окончания научных изысканий благодаря сохранению для пользователя форме.

На *четвертом этапе* разрабатывался алгоритм для решения задачи нахождения уровня флюоресценции и создавалось программное обеспечение, нацеленное на работу с графическими изображениями (рис. 4). Его основным назначением является количественный анализ уровня яркости интересующих врача областей на снимках глазного дна пациента.

Уровень яркости (флюоресценции) вычисляется по плоским черно-белым цифровым изображениям объемного глазного дна, полученным в формате bmp.

Для расчетов уровень флюоресценции принимается как конкретная безразмерная величина, выражаемая в числовом эквиваленте.

Анализ выделенных областей заключается в попиксельной детализации выбранного участка с помощью метода, использующего следующее цифровое соотношение: 0 — черный цвет, 1 — белый цвет. Соответственно на данных снимках существующие серые цвета представляют собой так называемую градацию от белого цвета к черному и имеют уже дробное числовое выражение. Таким образом, получая и суммируя вычисленную яркость каждого пикселя, находим общую яркость интересующего нас фрагмента изображения.

Можно выбрать одну, две интересные области или участок, заключенный между областями. Причем на одном и том же изображении можно осуществить анализ всех приведенных выше областей.

Результаты анализа изображений представляются в виде диаграммы, по оси абсцисс которой располагаются псевдонаименования изображений и числовой результат для проанализированных областей, по оси ординат — соответственно числовое значение, полученное в результате расчетов (рис. 5).

Полностью статистику можно посмотреть в сводной таблице, где уже участвуют реальные имена фотографий (файлов). Данная таблица имеет следующую структуру: название фотографии, результат анализа первой, второй, вложенной областей, а также отношений числовых результатов первого участка ко второму и второго к первому (рис. 6).

Все результаты, полученные в ходе исследования, сохраняются в файле самых распространенных форматов (doc, xls, html) с пометками, предварительно сделанными исследователем еще в таблице статистики. В дальнейшем эти файлы могут подвергаться редактированию в программных пакетах, поддерживающих работу с этими форматами.

Результаты работы

Итогами работы являются:

- разработка трехмерной модели витреальной полости глаза человека, позволяющая изменять свои пропорции в соответствии с индивидуальными особенностями глазного яблока пациента;
- создание программного обеспечения, позволяющего осуществлять персонализированный расчет дозы лазерного облучения, объема и площади поверхности витреальной полости глаза;
- внедрение программного обеспечения для анализа фотографий глазного дна пациента и определения уровня флюоресценции, в основу которого лег патент, где описан способ количественной оценки эффективности фотодинамической терапии хориоидальной неоваскуляризации.

Рассматривая массив современных медицинских информационных систем, можно определить, что основная часть направлена на перевод в элек-

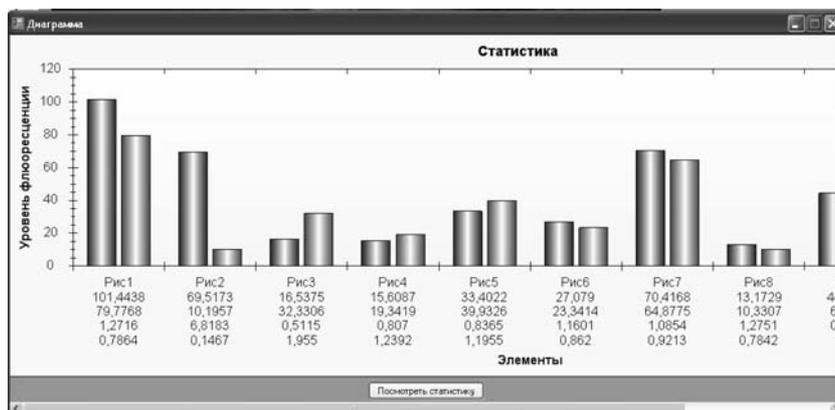


Рис. 5. Диаграмма результатов анализа уровня флюоресценции

Название изображения	Уровень флюоресценции 1-й области	Уровень флюоресценции 2-й области	Отношение уровня флюоресценции 1-й обл. ко 2-й	Отношение уровня флюоресценции 2-й обл. к 1-й
01	101,4438	79,7768	1,2716	0,7864
02	69,5173	10,1957	6,8183	0,1467
03	16,5375	32,3306	0,5115	1,955
04	15,6087	19,3419	0,807	1,2392
05	33,4022	39,9326	0,8365	1,1955
06	27,079	23,3414	1,1601	0,862
07	70,4168	64,8775	1,0854	0,9213
08	13,1729	10,3307	1,2751	0,7842
09	44,4203	61,388	0,7236	1,382
10	16,3788	18,5706	0,882	1,1338

Рис. 6. Статистическая таблица результатов анализа флюоресценции

тронный вид историй болезни. Также немалая часть нацелена на учет финансовых потоков в случае оказания платных медицинских услуг. Мало внимания уделяется применению возможностей информационных технологий в исследовательской деятельности, обработке экспериментальной информации в медицине. Особенно это ощутимо в офтальмологии ввиду сложной специализации этой области медицины. Разработанное программное обеспечение позволяет применять методы и средства информационных технологий в очень сложной и специфической области медицины — офтальмологии. Благодаря всестороннему анализу предметной области, интерпретации офтальмологических задач и желаемых результатов на язык информационных технологий данное программное обеспечение учитывает и выполняет все тре-

бования, необходимые для достижения максимального положительного эффекта от внедрения разработанной системы в лечебный процесс.

СПИСОК ЛИТЕРАТУРЫ

1. Даниличев В. Ф. Современная офтальмология. СПб.: Питер, 2000. 672 с.
2. Плахотный М. А. Интравитриальное применение антибактериальной фотодинамической терапии в ходе витротамии при лечении экзогенного бактериального эндофтальмита (экспериментальное клиническое исследование). Дисс. ... на соискание ученой степени канд. мед. наук, 2009. 145 с.
3. Володин П. Л. Фотодинамическая терапия с фотосенсибилизатором хлоринового ряда в офтальмологии (экспериментальное клиническое исследование). Дисс. ... на соискание ученой степени канд. мед. наук, 2008. 355 с.
4. Белый Ю. А., Терещенко А. В., Володин П. Л., Кучеров А. А., Шаулов В. В. Способ количественной оценки эффективности фотодинамической терапии хориоидальной неоваскуляризации. Патент на изобретение № 2387471 от 27.04.2010.
5. Плахотный М. А., Белый Ю. А., Терещенко А. В., Кучеров А. А. Математическая модель витреальной полости глаза // Актуальные проблемы офтальмологии. Матер. III Всероссийской науч. конф. молодых ученых с участием иностранных специалистов. М.: Медицина, 2008. С. 209—210.
6. Белый Ю. А., Терещенко А. В., Плахотный М. А., Кучеров А. А. Автоматизированная модель витреальной полости глаза. Российский общенациональный офтальмологический форум // Сб. трудов научно-практич. конф. с международным участием / Под ред. В. В. Нероева. 2008. С. 44—46.
7. Советов Б. Я., Цихановский В. В. Информационные технологии. М.: Высшая школа, 2006. 263 с.
8. Тахчиди Х. Л., Бессарабов А. Л., Пантелеев Е. Л. Параметризованный схематический стандартный глаз для решения вычислительных задач офтальмологии. Ч. 1 // Офтальмохирургия. 2006. № 4. С. 57—63.
9. Тахчиди Х. Л., Бессарабов А. Л., Пантелеев Е. Л. Параметризованный схематический стандартный глаз для решения вычислительных задач офтальмологии. Ч. 2 // Офтальмохирургия. 2007. № 1. С. 59—69.
10. Куприянов А. В., Ильясова Н. Ю., Ананьин М. А. Оценка диагностических параметров сосудов на изображениях глазного дна в области диска зрительного нерва // Компьютерная оптика. 2006. № 29. С. 146—150.

CONTENTS

Hodashinsky I. A., Sinkov D. S. Parameters Identification of Fuzzy Systems Based on Adaptive Particle Swarm Optimization Algorithm	2
-----------------------------------------------------------------------------------------------------------------------------------------------------	---

In this work, improvements of the standard particle swarm optimization algorithm are considered. We describe adaptive particle swarm optimization algorithm for the parameters identification of fuzzy systems. The proposed algorithm is tested with promising results.

Keywords: fuzzy system identification, metaheuristics, adaptive particle swarm optimization algorithm

Maslikov V. I. The Senior Lecturer of Faculty "Publishing and Journalism"	6
--------------------------------------------------------------------------------------------	---

The full basis of judgements syllogistics types A, E, I, O, a, e, i, o is developed. Communication of all types of judgements with Boolean algebra, Euler-Wenn diagrams and treelike structures is certain. The received system of logic attitudes between judgements can be used for development in areas of knowledge bases, neural and intellectual systems.

Keywords: universum, logic attitudes, Boolean algebra, syllogism, syllogistics, an artificial intellect, neural networks

Sadykov S. S., Savicheva S. V. Identification of the Actual Flat Objects on the Basis of a Single Trait Points of Their Outer Contours.	13
----------------------------------------------------------------------------------------------------------------------------------------------------------	----

An algorithm for identification of the real plane objects that uses only the values of the curvature points of the outer contours of their binary images. The algorithm is illustrated by examples.

Keywords: identification of real objects, the value of curvature

Bobkov V. A., Ronshin Yu. I. GPU Implementation of Depth Map Algorithm	17
-----------------------------------------------------------------------------------------	----

The algorithm for construction of depth maps from multiple images and its implementation on GPU with CUDA is presented. It is based on application of space-sweep strategy and adaptive window-based technique. The evaluation of two versions of algorithm on CPU and GPU is described.

Keywords: 3D reconstruction, depth map, image matching, CUDA

Gulakov V. K., Trubakov A. O. *An Effective Algorithm for Converting Full-Color Image to an Image in the Palette for Content Based Image Retrieval System* 22

The paper analyzes the algorithms for palettes in different color spaces such as RGB, HSV, L*a*b, L*C*h. Identified requirements for the palette in content based image retrieval systems. Also proposed a new method for generating a palette and an algorithm for converting full-color images to an image in the palette by using the map transformation.

Keywords: image processing, color palette, content-based image retrieval, CBIR

Belyakov S. L., Belykova M. L., Samojlov D. S. *Geoinformation Service of the Situational Center* 29

In work the variant of construction of the geoinformation service focused on granting of maps and charts taking into account experience of their use at decision-making is analyzed. Features of statement and realization of the optimizing search problems realized at construction of new decisions are considered.

Keywords: situational center, geoinformation systems, services, network communities

Struchenkov V. I., Kozlov A. N., Egunov A. S. *Dynamic Programming of Linear Structures Route Design* . . . 33

Flat spline approximation is specified by discrete sequence points, which is not a single schedule. Elements of the approximate number of curves, arcs and are straight. Approximation curve must satisfy the constraints and minimize the value adopted optimality criterion. The challenge is to develop linear structures route CAD. Describes an algorithm for dynamic programming method.

Keywords: approximation, restricts, dynamic programming

Kobzareno D. N. *Application Features for Some Algorithms of Computing Geometry in Geoinformation Technologies* 37

Features of two algorithms: definition for two lines crossing and definition for a point is inside a polygon are considered. These tasks seem to be simple ones, but have some features during developing soft for geoinformation systems, need to be taken into account. Such features aren't described in a literature, but to ignore it in some cases may leads to mistakes and incorrect results.

Keywords: geoinformation technologies, computer graphics

Lvovich Ya. E., Lvovich I. Ya., Preobrazhenskiy A. P., Golovinov S. O. *The Research of the Method of Trace of Beams for Designing of Wireless Communication Systems* 40

In the given paper the description of algorithm of calculation of quantity of beams of distribution of electromagnetic waves in the conditions of city building is carried out. By results of research of the offered algorithm reduction of quantity of considered beams in the point of reception with use of a method of trace of beams is shown.

Keywords: a method of trace of beams, city buildings, correlation factors

Moschevikin A. P., Galov A. S., Volkov A. S. *Positioning in NanoLOC (IEEE 802.15.4a) Wireless Network* 43

The paper presents the description of nanoLOCTM (IEEE 802.15.4a) technology for mobile target positioning, the principles of positioning calculation and the results of the experiments on location accuracy determination. The recommendations on accuracy calculation methods are given.

Keywords: wireless sensor network, indoor positioning systems, location accuracy, nanoLOCTM, IEEE 802.15.4a

Gilmanova N. A. *A Multi-Method Algorithm for Three-Dimensional Packing with Controllable Objects Overhanging Inside a Container* 48

The paper presents a multi-method algorithm for solving a three-dimensional packing problem with some additional constraints derived from practical cargo storage and transportation. A computational experiment ascertains the way the quality of solutions obtained depends on various algorithm parameters. It is demonstrated how heuristics combining allows to get rational solutions in short time.

Keywords: three-dimensional packing, multi-method technology, heuristics combining

Kovalevich A. A., Yakimov A. I., Albkeirat D. M. *Research of Optimization Stochastic Algorithms for Application in Simulations of Systems* 55

The comparative analysis of optimization stochastic algorithms is done: particle swarm, simulated annealing and differential evolution. For check of algorithms functions *Branin RCOS*, modified version *RCOS*, *Easom* and *Goldstein-Price* were used. It is shown, that the algorithm of differential evolution is the best algorithm as stably finds an optimum of function for minimal time.

Keywords: optimization stochastic algorithm, particle swarm, simulated annealing, differential evolution, simulation

Pashchenko V. N., Martinuyk V. A., Monikov A. S., Pashchenko V. V. *Research of Influence of Finite Elements Type on Duration and Precision of Calculation of Temperature Stress in the ANSYSWorkbench* 61

The technique of solving the problem of engineering analysis of thermal deformations of complex size-dependent structures with fine bindings included in the small spacecraft (ICA). Studied the effect of the finite element type in the results of the decision. Constructed finite element model using a one-, two- and three-dimensional finite element. The results of numerical experiments that demonstrate the advantages of this approach.

Keywords: one-dimensional finite elements, line body, spot welding, thermal-stress calculation, steady-state thermal, static structural, selecting finite elements, technical systems with exact dimensional bindings, calculation of the aerial systems, calculation in the ANSYSWorkbench

Asadov H. H., Alieva Y. N. *On Possibility of Application of Method of Multi-Level Optimization for Optimal Information Assessment of Species Diversity in Ecological Systems* 64

The new method of multi-level optimization is suggested. The method provides for two levels of optimization. In the first level of optimization the conditional variation optimization task to be solved, and in the second level, the task of to be solved, and in the second level the task of linear programming to be solved, which make it possible to decrease the number of optimal solutions obtained in the first level as far as two ones. On the basis of suggested method the method for assessment of diversity of specimens in ecological systems is developed.

Keywords: ecological systems, optimization, variation computation, linear programming, ecological diversity

Kuznetsov A. A. *System Communication Between Processes of a Heart Rhythm and Systoles Potential Dynamic* 69

According to processing 58 twenty minute electrocardiograms registration of a different sex conditionally healthy young people a system interrelation search between processes of a heart rhythm and systole potential dynamics is carried out. Application of an integrated method of the analysis and the two-parametrical histogram has allowed to allocate nonlinear communication in rather narrow interval of R-R intervals values distribution and to treat the electrocardiograms data from positions of the offered interrelation mechanisms. Characteristics of these mechanisms are given. The model of system interrelation is offered. It is revealed, that the cardio signal structure is formed in cardio interval frameworks and set by a rhythm, and rhythm management and regulation is made by means of a systole potential changing.

Keywords: the electrocardiogram, a heart rhythm, a systole potential, system communication, the two-parametrical histogram

Galkin V. A., Belij Yu. A., Kucherov A. A. *Antibacterial Photodynamic Therapy Dataware and Software* 74

This article gives consideration to the algorithm and software, which permits a possibility for personalized calculations of the laser illumination dose, and vitreal cavity square and volume calculations. It describes an algorithm of the way of the fluorescence level quantitative assessment, software realization of this algorithm for the analysis of the patient's fundus of eye photos. It also presents a developed 3D model of the human vitreal cavity, which is able to change its proportions according to the patient's eye-bulb peculiarities.

Keywords: software, algorithm, dataware, 3D model of the human vitreal cavity, antibacterial photodynamic therapy

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.
Корректор *Т.В. Пчелкина*.

Сдано в набор 07.06.2011. Подписано в печать 20.07.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 11,25. Заказ 545. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15