ISSN 1816-0301

ИНФОРМАТИКА

2(46)

АПРЕЛЬ–ИЮНЬ
2015

Адрес редакции:
220012, Минск,
ул. Сурганова, 6, к. 305
тел. (017) 284-26-22
e-mail: rio@newman.bas-net.by
http://uiip.bas-net.by

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

ЕЖЕКВАРТАЛЬНЫЙ НАУЧНЫЙ ЖУРНАЛ

Издается с января 2004 г.

№ 2(46) • апрель-июнь 2015

СОДЕРЖАНИЕ

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

ЗАЩИТА ИНФОРМАЦИИ

Редактор Г.Б. Гончаренко
Корректор А.А. Михайлова
Компьютерная верстка О.Б. Бутевич

Сдано в набор 24.04.2015. Подписано в печать 07.06.2015.
Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография.
Усл. печ. л. 13,3. Уч.-изд. л. 13,0. Тираж 100 экз. Заказ 8.

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/274 от 04.04.2014.
ЛП № 02330/444 от 18.12.13.
Ул. Сурганова, 6, 220012, Минск.

THE UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

INFORMATICS

PUBLISHED QUATERLY

Issued since 2004

№ 2(46) • April-June 2015

CONTENTS

SIGNAL, IMAGE AND SPEECH PROCESSING

MATHEMATICAL MODELING

DESIGN AUTOMATION

LOGICAL DESIGN

INFORMATION SECURITY

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

УДК 004.932.4

В.В. Старовойтов

МЕТОДИКА ВЫБОРА ФИЛЬТРА ДЛЯ СГЛАЖИВАНИЯ СПЕКЛ-ШУМА РАДАРНЫХ ИЗОБРАЖЕНИЙ С СИНТЕЗИРОВАННОЙ АПЕРТУРОЙ

Предлагается методика сравнения результатов фильтрации радарных изображений с синтезированной апертурой разными типами фильтров. С ее помощью можно выбирать фильтры, которые лучше других сглаживают исследуемое изображение. Методика базируется на оценке параметра ENL, который, в отличие от известных подходов, вычисляется для автоматически выбираемых участков изображения.

Введение

Метод радиолокационного синтеза апертуры (РСА) позволяет получать радарные изображения земной поверхности и находящихся на ней объектов независимо от метеорологических условий и уровня солнечного освещения. Однако в силу особенностей получения таких изображений [1, 2] возникает специфический шум, который называется спекл-шумом. Он отсутствует на космических изображениях, полученных системами пассивной регистрации отраженных волн в оптическом и тепловом диапазонах.

В геоинформационных системах (например, в пакете ArcGIS) для фильтрации радарных изображений с синтезированной апертурой чаще всего используются фильтр Ли, улучшенный фильтр Ли, фильтр Фроста, фильтр Куана [3–6]. Эти фильтры сохраняют края и детали объектов, сокращая зернистость изображения. Как правило, их применяют локально, в окрестностях 3×3, 5×5, 7×7, 9×9 и 11×11 пикселов. Кроме этих фильтров для сглаживания изображений данного типа используются нелинейные ранговые фильтры, например медианные. Выбор того или иного типа фильтра предоставляется пользователю, однако сделать его непросто по двум причинам:

– данные о яркости в цифровых радарных изображениях имеют диапазон значений [0–65 535], а на мониторе компьютера отображаются не более 256 оттенков;

– размеры современных синтезированных радарных изображений огромны.

Например, немецкий спутник TerraSAR-X выполняет съемку поверхности Земли в *X*-диапазоне со средней длиной волны 31 мм в прожекторном режиме с разрешением 1–2 м, размером кадра (5–10)×10 км и шириной полосы обзора от 463 до 622 км. Одно из использованных в наших исследованиях цифровых изображений этого спутника имело размер 27 083×43 750 пикселов и диапазон значений яркости 16 битов/пиксел. Максимальное разрешение современных мониторов (например, WHUXGA) равно 7680×4800 пикселов. Таким образом, для визуального анализа исходного и преобразованного радарных изображений их следует преобразовать в новую шкалу значений яркостей и уменьшить в несколько раз либо просматривать по фрагментам.

В настоящее время методика автоматического выбора фильтра для радарных изображений с синтезированной апертурой в литературе не описана.

1. Модель спекл-шума

Мультипликативная модель шума применяется, когда полезный сигнал умножается на случайный сигнал. Спекл-шум при синтезе апертуры образуется в результате когерентной суперпозиции пространственно-случайных колебаний отраженного сигнала от разных источников рассеяния. Рассеянные волны накладываются друг на друга, вызывая тем самым появление спекл-шума на изображении, который увеличивает средний уровень серого в локальной окрестности пиксела.

Спекл-шум во всех сканирующих системах с когерентным формированием изображения вызван энергетическими помехами из-за беспорядочно распределенных отражателей сигнала, слишком мелких для того, чтобы их могла отобразить система. Он относится к классу шумов, зависящих от самого сигнала изображения, для объектов с низким разрешением обычно является мультипликативным. Спекл-шум возникает при получении цифровых изображений с помощью ультразвуковых медицинских сканеров и радаров, в результате изображение выглядит «зернистым», а на радарных изображениях появляется крестообразное повышение яркости вокруг некоторого объекта или его границ. Примеры изображений приведены в следующем разделе.

Теоретическая модель РСА-изображений, использующая экспоненциальную плотность распределения вероятности для описания мультипликативного шума, впервые была предложена Годманом [2]:

$$P_{Y/X}(y/x) = (L^L/(x^L\Gamma(L))) \times y^{L-1}e^{-Ly/x},$$

где Y – регистрируемое изображение;

X – неискаженное изображение;

$\Gamma(L)$ – гамма-функция;

L – число обзоров одного и того же участка поверхности Земли; если L=1, то $\Gamma(L)$=1.

Если в пределах некоторой окрестности оригинальное изображение X не изменяется, т. е. $E(X) = const$ (где E – символ математического ожидания), то плотность распределения вероятности отраженного излучения совпадает с плотностью распределения вероятности спекл-шума Z. Таким образом, если случайный процесс Z нормализован, т. е. $E(Z)$=1, то для спекл-шума имеем модель

$$P_Z(z) = (L^L/\Gamma(L)) \times z^{L-1}e^{-Lz}.$$

В итоге после нормализации выражения эффект добавления спекл-шума на изображении может быть описан мультипликативной моделью с помощью следующей формулы:

$$Y(i, j) = X(i, j) \times Z(i, j),$$

где (i, j) – координаты пиксела.

Кроме того, в радарных изображениях сигнал и шум статистически независимы друг от друга. Выборочное среднее и дисперсия одного пиксела равны среднему и дисперсии локальной окрестности с центром в этом пикселе.

2. Переквантование радарных изображений для отображения на мониторе

Для визуального анализа результатов обработки изображений рассматриваемого типа необходимо выполнить переквантование исходного диапазона уровней яркости в меньший диапазон [0–255].

Анализ гистограммы (рис. 1) показывает, что около 10,7 % пикселов имеют яркость, превышающую значение 240; около 9,2 % имеют яркость выше значения 255, которое является максимальным значением при восьмибитном представлении полутоновых изображений. Максимальное значение яркости данного изображения равняется 32 767.

Было исследовано несколько вариантов переквантования значений яркости. При равномерном (линейном) переквантовании яркостей из диапазона [0–65536] в диапазон [0–255] получается очень темное изображение с небольшими яркими вкраплениями, поскольку темные значения (0, 1, 2 и 3 соответственно) получают большинство оттенков яркости из основного диапазона [0–1023]. Визуально пикселы с такими значениями яркости на мониторе выглядят черными. В итоге изображение выглядит черным с вкраплениями отдельных светлых точек.

К полученному изображению применялось линейное растяжение контраста с отбрасыванием небольшого процента (около одного) самых темных и светлых пикселов. В итоге контраст изображения был существенно повышен, но светлые пикселы со значениями яркости, близкими к максимальным, в результате такого преобразования практически сливаются, а именно они

показывают отражение от металлических объектов. В то же время темные области изображения мало контрастны и присутствующие в них объекты малозаметны.

Таким образом, с учетом мультипликативной природы шума на радарных изображениях для яркостной нормализации изображений этого типа следует использовать нелинейные преобразования, в частности логарифмического типа. В этом случае значение яркости может быть представлено в виде суммы собственно отраженного сигнала и шумовой составляющей.

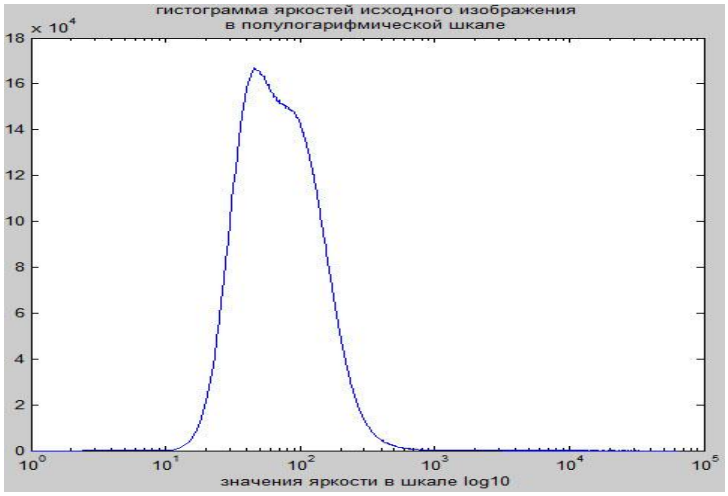


Рис. 1. Гистограмма яркостей фрагмента 5000×5000 пикселов радарного изображения в полулогарифмической шкале $\log_{10}$

Для нелинейного переквантования яркостей в диапазон [0–255] были исследованы различные варианты преобразования значений яркости. Опишем основные из них:

1. Преобразование яркости согласно функции $Y_{ij} = \log_2(I_{ij})^2$. Дает малоконтрастное изображение.
2. Преобразование $I_{1ij} = I_{ij}^{0,5}$. Дает темное изображение.
3. Преобразование $I_{2ij} = \log_2(I_{ij} - T) \times (255 - T)/16$, где T – пороговое значение, отделяющее зашумленные спекл-шумом пикселы от остальных. Дает очень светлое изображение.
4. Преобразование $a \times I_{1ij} + (1 - a) \times I_{2ij}$, где $0 < a < 1$, например $a = 0{,}5$. Дает малоконтрастный результат.
5. Преобразование $I_{1ij} = I_{ij}^{0,5}$ и линейное растяжение с отбрасыванием 1 % темных и ярких пикселов. Дает приемлемый контраст (рис. 2).

Последнее преобразование было выбрано в качестве основного для яркостной нормализации при отображении радарных изображений на данном этапе.

Рис. 2. Шестнадцатибитовое радарное изображение, нелинейно переквантованное в восьмибитовое представление

3. Методика выбора фильтра по результатам фильтрации

При обработке радарных изображений во избежание потери информации фильтруются 16-битовые представления изображений. Были исследованы ранговые фильтры, в частности медианный фильтр с разными вариантами окна, фильтры Ли, сигма-фильтр, фильтры Фроста, Куана, Винера, двунаправленный фильтр (bilateral), апостериорный гамма-фильтр и др. [1, 3–11]. Примеры фрагмента, обработанного двумя фильтрами и преобразованного для отображения на мониторе согласно пятому варианту, описанному выше, показаны на рис. 3 и 4. Основная трудность визуальной оценки заключается в том, что результаты могут быть очень похожи. Невозможно определить, какой фильтр лучше уменьшает спекл-шум.

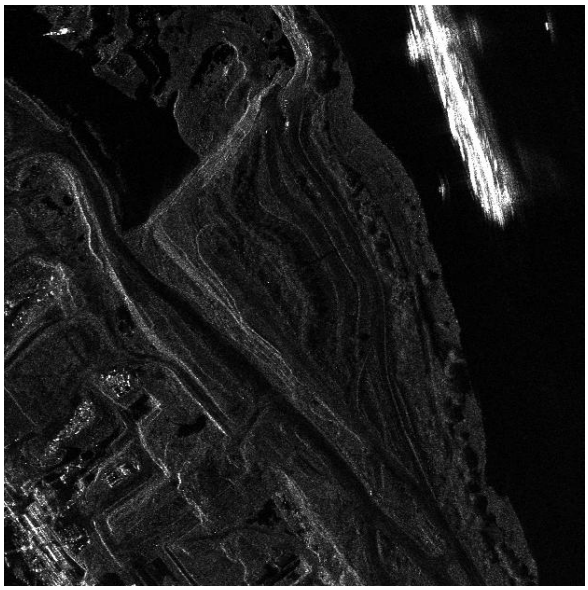


Рис. 3. Результат медианной фильтрации в окне 3×3

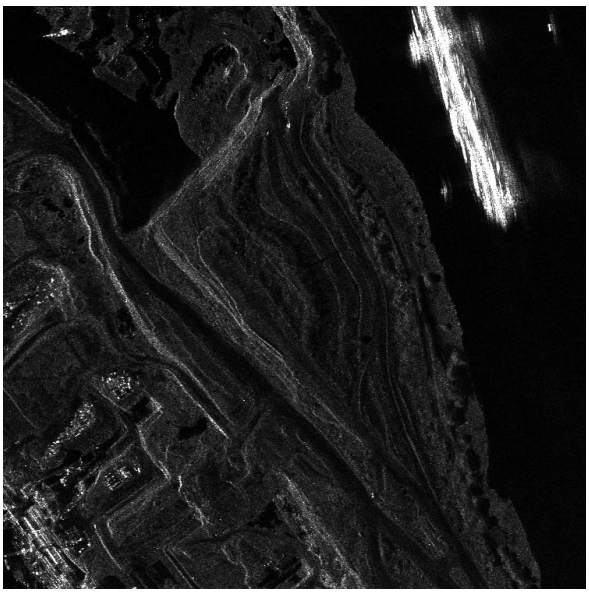
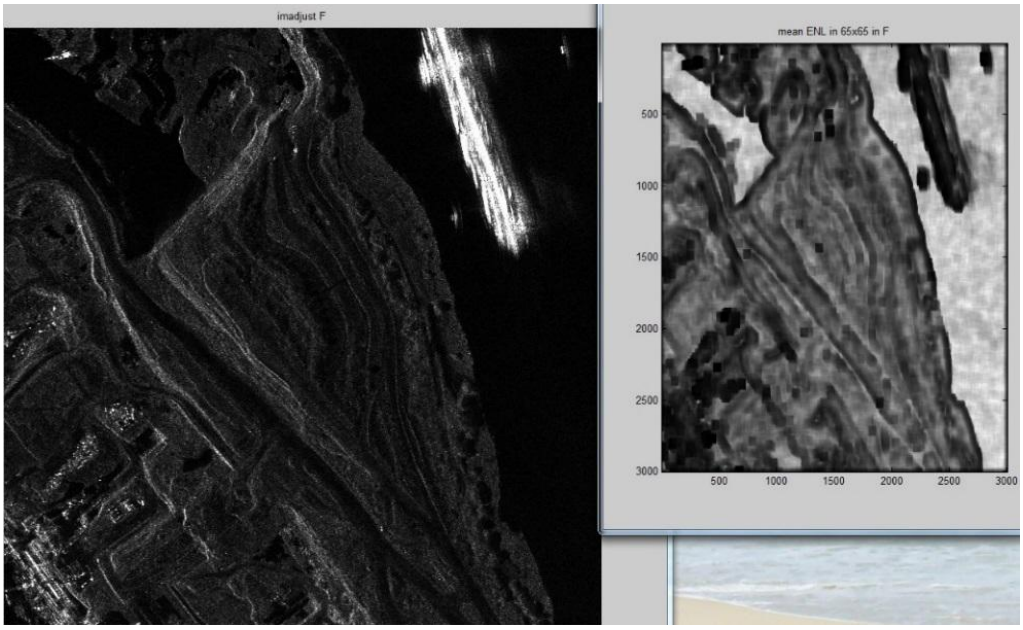


Рис. 4. Результат сглаживания фильтром Винера 3×3

В результате анализа литературы [1–11] по фильтрации изображений с синтезированной апертурой выбран рекомендуемый для сравнения результатов фильтрации параметр ENL (equal number of looks) – эквивалентное число наблюдений [12]. Этот параметр рекомендуется

вычислять на однородных участках земной поверхности, таких как поля, леса и т. п. Параметр равен отношению квадрата среднего значения к дисперсии или стандартному отклонению значений яркости пикселов. Основная проблема, связанная с применением этого параметра, состоит в том, что в литературе не описаны методы автоматического определения однородных участков для вычисления ENL.

Автором были исследованы различные варианты автоматического вычисления параметра ENL. Наиболее перспективным оказался вариант вычисления в окне фиксированного размера, который можно определить как функцию от разрешения снимка. На рис. 5 видно, что области воды наиболее однородны (выглядят наиболее темными на рис. 5, *а*), они дают наибольшие значения параметра ENL (выглядят наиболее светлыми на рис. 5, *б*). Поэтому они не могут служить в качестве однородных областей для вычисления ENL. Аналогично области с наибольшими значениями на исходном изображении (наиболее светлые на рис. 5, *б*) имеют меньшие значения параметра ENL (темные участки на рис. 5, *б*). Для представленного фрагмента максимальное значение ENL равно 98,3040, минимальное – 0,7912, а среднее – 40,0464.



а) *б)*

Рис. 5. Фрагмент (3000×3000 пикселов) исходного изображения (*а*);
карта значений параметра ENL, вычисленного в скользящем окне 65×65 пикселов (*б*)

В результате анализа различных вариантов вычисления параметра ENL предложено использовать медианное значение среди всех, вычисленных для обрабатываемого изображения в скользящем окне определенного размера, после чего определять координаты центра окна, соответствующего этому значению. Далее найденный квадратный фрагмент используется для сравнения результатов фильтрации различными функциями: чем выше значение ENL после фильтрации, тем качественнее результат обработки.

Для оценки результатов фильтрации изображения на рис. 5, *а* выбраны четыре квадратные области, координаты их центров представлены в табл. 1. Эти области имеют одинаковые значения параметра ENL, равные медианному, поскольку вычисляются в формате с плавающей точкой.

Таблица 1

Участок	ENL1	ENL2	ENL3	ENL4
y	753	2291	2587	1049
x	1071	1079	2437	2655

Таблица 2

Значения ENL в отобранных областях после применения разных фильтров

Фильтр	max(ENL)	min(ENL)	med(ENL)	ENL_1	ENL_2	ENL_3	ENL_4
Исходное изобр.	17,1670	0,0488	5,8312	5,8312	5,8312	5,8312	5,8312
Ли	38,8634	0,0673	8,8848	8,1970	7,9533	7,2671	10,3037
Сигма-Ли	17,1670	0,0500	5,8315	5,1169	5,7429	5,4332	6,9475
Фроста	85,3420	0,0831	12,6353	12,3868	10,5688	9,0764	14,1049
Куана	358,8283	0,0874	**24,9837**	**26,6505**	**19,119**	**12,5543**	**23,770**
Медианный, 3×3	32,5624	0,0623	7,8927	7,4157	7,3355	6,6998	9,1120
Винера, 3×3	38,7188	0,0492	8,7860	7,6425	7,9398	7,2622	10,3062
Bilateral	369,3838	0,1545	**24,4974**	**25,7966**	**18,5332**	**12,3430**	**23,3042**
L0 smoothing	$2{,}8431\times10^7$	0,0492	6,5068	5,2576	6,1008	6,5113	8,0718

Анализ табл. 2 показывает, что улучшенный фильтр Ли (названный в литературе сигма-Ли) практически не меняет значения ENL после фильтрации, а фильтры Куана и bilateral имеют наибольшие среди других фильтров показатели параметра ENL, поэтому они являются лучшими при фильтрации данного изображения.

Заключение

Описана новая методика автоматического выбора фильтра и его параметров для уменьшения влияния спекл-шума на радарных изображениях с синтезированной апертурой. С ее помощью можно выбирать фильтры, которые лучше других сгладят спекл-шум на обрабатываемом изображении. Методика заключается в вычислении параметра ENL в скользящем окне фиксированного размера для всего исходного изображения, поиске медианного значения ENL, определении участков изображения, соответствующего найденным медианным значениям, применении нескольких вариантов фильтрации к исходному изображению и выборе тех фильтров, которые дают максимальные значения ENL после обработки.

Для визуального контроля результатов предлагается вариант нелинейного переквантования яркостей радарных изображений, описываемых большим диапазоном значений: от 0 до $2^{16}-1$.

Список литературы

1. Радиолокационные системы землеобзора космического пространства / под ред. В.С. Вербы. – М. : Радиотехника, 2010. – 680 с.

2. Goodman, J.W. Some Fundamental Properties of Speckle / J.W. Goodman // Journal of the Optical Society of America. – 1976. – Vol. 66, № 11. – P. 1145–1150.

3. Lee, J.S. Digital Image Enhancement and Noise Filtering by Use of Local Statistics / J.S. Lee // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1980. – Vol. 2. – P. 165–168.

4. Improved sigma filter for speckle filtering of SAR imagery / J.S. Lee [et al.] // IEEE Transactions on Geoscience and Remote Sensing. – 2009. – Vol. 47, № 1. – P. 202–213.

5. Frost, V.S. A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise / V.S. Frost // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1982. – Vol. 4, № 2. – P. 157–166.

6. Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise / D.T. Kuan [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1985. – Vol. 7. – P. 165–177.

7. Azimi-Sadjadi, M.R. Two-Dimensional Adaptive Block Kalman Filtering of SAR Imagery / M.R. Azimi-Sadjadi, S. Bannour // IEEE Transactions on Geoscience and Remote Sensing. – 1991. – Vol. 29. – P. 742–753.

8. Gagnon, L. Speckle Filtering of SAR Images – A Comparative Study Between Complex-Wavelet-Based and Standard Filters / L. Gagnon, A. Jouan // SPIE Proc. – 1997. – Vol. 3169. – P. 80– 91.

9. Speckle Noise Reductionin SAR Imagery Usinga Local Adaptive Median Filter / F.J Qiu [et al.] // GIScience and Remote Sensing. – 2004. – Vol. 41, № 3.– P. 244–266.

10. Xiao, J. A detail-preserving and flexible adaptive filter for speckle suppression in SAR imagery / J. Xiao, J. Li, A. Moody // International Journal of Remote Sensing. – 2003. – Vol. 24, № 12. – P. 2451–2465.

11. Tomasi, C. Bilateral filtering for gray and color images / C. Tomasi, R. Manduchi // Proc. 6th Intern. Conf. on Computer Vision. – Bombay, 1998. – P. 839–846.

12. Bruniquel, J. Multi-variate optimal speckle reduction in SAR imagery / J. Bruniquel, A. Lopes // International Journal of Remote Sensing. – 1997. – Vol. 18, № 3. – P. 603–627.

Поступила 02.03.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: valerys@newman.bas-net.by

V.V. Starovoitov

METHOD OF FILTER SELECTION FOR SPECLE-NOISE SMOOTHING IN SAR IMAGES

A method for comparison of different SAR image filtration techniques is presented. It allows selecting the filters with better speckle noise smoothing effect. Unlike the known approaches, the presented method is based on *ENL* parameter calculation for automatically selected areas.

УДК 004.932.1

Н.Н. Кузьмицкий

ОБНАРУЖЕНИЕ ФРАГМЕНТОВ ТЕКСТА НА ИЗОБРАЖЕНИЯХ РЕАЛЬНЫХ СЦЕН НА БАЗЕ СВЕРТОЧНОЙ НЕЙРОСЕТЕВОЙ МОДЕЛИ

Рассматривается модель детектора текстовых образов на базе сверточной нейронной сети, способной синтезировать высокоуровневые признаки образов в режиме «черного ящика». Описывается методика применения детектора, основанная на алгоритмах мультимасштабного сканирования и локальной интерпретации откликов, позволяющая обнаруживать текстовые объекты на изображениях реальных сцен. Показываются преимущества разработок в сравнении с аналогами, выполняется оценка эффективности на примере известной базы данных.

Введение

Обнаружение текстовых объектов на изображениях реальных сцен является весьма распространенной задачей в практических приложениях: при поиске изображений по содержанию в больших коллекциях и глобальной сети, идентификации объектов в промышленности, помощи людям с ограниченными возможностями по зрению (просмотре цены товара), навигации в городе по информационным меткам на иностранном языке и др. [1]. Многие из предположений традиционных систем оптического распознавания (optical character recognition, OCR) относительно характеристик изображений в случае реальных сцен являются несостоятельными (например, черный текст на ярком фоне), при этом в ходе анализа приходится сталкиваться со следующими проблемами (рис. 1):

– текстовые образы могут иметь различную яркость и размер (в пределах одного слова), располагаться в произвольном месте сцены, содержащей текстуры и фоновые объекты;

– в то время как в документах используются известные машинописные шрифты, текст реальных сцен может содержать существенно стилизованные и искаженные образы, необходимые, например, для узнаваемости фирменного бренда;

– традиционные OCR-системы предназначены для обработки высококачественных изображений, полученных с помощью сканера, однако изображения реальных сцен могут быть синтезированы в условиях недостаточного освещения, различного ракурса, переносным устройством с низкими оптическими характеристиками камеры и т. п.

Рис. 1. Примеры изображений текстовых образов реальных сцен

Указанные особенности приводят к необходимости разработки новых подходов, лишенных таких недостатков OCR, как «ручной» выбор признаков, сложность оптимального подбора параметров, низкая универсальность и др. Перспективным направлением исследований является использование для решения поисковых задач методов машинного обучения.

1. Модель сверточного нейросетевого детектора текстовых образов

Сверточные нейронные сети (convolutional neural networks, CNN) – это многослойные иерархические модели, поводом к созданию которых послужили исследования зрительного аппарата кошек, проведенные в 1960-х гг. Их результатом стало открытие двух типов клеток, влияющих на зрительную восприимчивость: первые обладают свойством локальной чувстви-

тельности и предназначены для выделения элементарных признаков образов (например, ориентированных краев), вторые путем их комбинирования формируют высокоуровневые признаки.

Известны различные реализации сверточных нейронных сетей, отличающиеся топологией слоев, организацией процесса обучения и др. Исходя из результатов их применения при решении задач анализа изображений, а также возможности обучения без использования специализированного аппаратного обеспечения, в качестве базового для разработки детектора текстовых образов был выбран тип нейросетей, который разработал Y. LeCun в конце 1990-х гг. [2]. В их основе лежат три архитектурные идеи:

1) *локальные рецептивные поля* (нейроны получают сигнал от окрестностей нейронов предыдущего слоя, за счет чего сеть обучается двухмерной структуре входного образа);

2) *разделяемые веса* (нейроны слоя объединены картами, в которых они обладают общими весами, при этом карты формируют различные признаки и сокращают количество параметров, настраиваемых в ходе обучения);

3) *пространственные подвыборки* (локальное усреднение карт приводит к синтезу высокоуровневых признаков, повышая устойчивость к искажениям).

Обучение сверточной нейронной сети осуществляется модификацией алгоритма обратного распространения ошибки на основе метода Левенберга – Марквардта, обеспечивающей рост сходимости из-за индивидуальной настройки для весов каждого нейрона параметра η, что позволяет замедлять процесс на крутых областях весового пространства и ускорять на плоских [3]. Таким образом, сверточные нейронные сети служат эффективным средством решения задач обработки изображений. Их преимуществом является объединение в рамках одной модели экстрактора признаков и классификатора. При этом высокоуровневые признаки синтезируются в режиме «черного ящика» путем чередования процедур свертки с настраиваемыми фильтрами и подвыборки полученных откликов, а обучение классификатора полностью контролируется.

Для обнаружения текстовых образов на изображениях была разработана модель детектора в виде сверточной нейронной сети (рис. 2). Ее входной слой содержит 32×32 нейрона, которые получают сигнал в виде яркости полутонового изображения аналогичного размера. Первый из четырех скрытых слоев (C1) является сверточным с двенадцатью картами, содержащими по 28×28 нейронов, которые разделяют по одному фильтру размером 5×5 и параметру смещения (244 608 связей, 312 настраиваемых параметра). За ним следует подвыборочный слой (S2), усредняющий отклики нейронов предыдущего слоя по неперекрывающимся окрестностям размером 2×2, поэтому он содержит 12 карт по 14×14 нейронов, разделяющих по одному параметру весового усреднения и смещения (11 760 связей, 24 параметра).

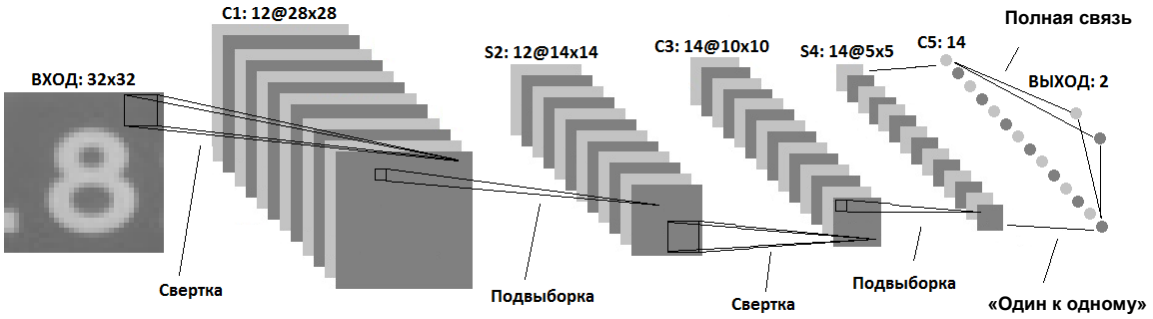


Рис. 2. Архитектура сверточного нейросетевого детектора

Следующий слой (C3) – сверточный, содержит 14 карт размером 10×10 нейронов, каждая соединена лишь с подмножеством карт слоя S2: первые восемь карт слоя C3 соединены с тремя из S2 (по одной из каждой четверки), остальные шесть связаны с половиной карт предыдущего. Для каждой пары настраивается свой фильтр размером 5×5 и один параметр смещения (151 400 связей, 1514 параметров). Целью разреженного соединения является ликвидация симметрии в топологии сети, что заставляет карты формировать различные признаки, так как они получают разный входной сигнал. Второй подвыборочный слой (S4) аналогичен S2 и уменьшает размер 14 карт до 5×5. Следующий слой (C5) содержит 14 нейронов, соединенных с одной

картой предыдущего слоя фильтром размером 5×5 и одним параметром смещения (364 связи, 364 параметра). Выходной слой представлен двумя нейронами, соединенными с каждым нейроном из C5 с помощью одного параметра веса и смещения (30 связей, 30 параметров). Всего представленная архитектура содержит 409 912 связей и 2272 настраиваемых параметра.

Первые четыре слоя сети являются экстрактором высокоуровневых признаков, в то время как последние два – классификатором в форме многослойного персептрона [4]. Выбор размера входного изображения и числа карт в слоях объясняется стремлением к достижению баланса между высокими обобщающими свойствами сети и эффективностью ее практического применения. Масштаб фильтров и окрестностей усреднения определяется так, чтобы в ходе послойного уменьшения размера карт количество нейронов в первом классифицирующем слое (C5) равнялось числу карт. В качестве функции активации нейронов был выбран гиперболический тангенс ввиду наличия горизонтальных асимптот, симметричности, простого вычисления производной и распространенности при решении задач, аналогичных рассматриваемой [2, 6].

Обучение детектора выполняется с использованием базы маркированных образов, разделенной на тренировочную и тестовую части. База должна содержать множество пар вида (x, y), где x – графический образ, y – номер класса (0 – текстовый, 1 – фоновый). Текстовым образом является полутоновое изображение масштаба 32×32 пиксела, в которое вписан символ с сохранением пропорций размеров. Выбор яркостных, текстурных и других признаков символов связан со свойствами сцен, на изображениях которых планируется выполнять детектирование. При этом эффективность обучения напрямую зависит от разнообразия образов, число которых определяется неравенством Вапника – Червоненкиса: обобщаемость сети прямо пропорциональна отношению объема тренировочной выборки к мере сложности модели (количеству параметров) [5].

Подготовка процесса обучения нейронной сети включает:

– присваивание весам случайных значений, равномерно распределенных на интервале $[-2{,}4; 2{,}4\ F_i]$, где F_i – число входных связей i-го нейрона;

– выбор количества эпох и методики изменения коэффициента η: обучение проводится за 34 эпохи с начальным значением $\eta = 0{,}000\ 85$, которое изменяется каждую эпоху путем умножения на коэффициент 0,85, в итоге конечное значение η составляет 0,000 004;

– настройку параметров искажений входных образов: величины поворота (например, в пределах ±5°), изменения масштаба (в пределах ±10 %).

Коррекция весов нейронов проводится после обработки каждого образа, при этом для ускорения обучения сети может применяться методика пропуска этапа обратного распространения ошибки в случае, если ее величина была меньше заданного значения ε. Также может быть выполнено дообучение нейросети, которое, по мнению ряда авторов, в частности С. Осовского, является весьма эффективным, так как позволяет выполнить «встряхивание весов» с минимальной вероятностью вывода поиска из сферы притяжения ранее найденного локального минимума, в отличие от обучения «с чистого листа». Сеть в такой ситуации должна проявить способности к усвоению наиболее характерных признаков и после кратковременной «амнезии» быстро восстановиться, а затем в большинстве случаев улучшить свои показатели [5].

Сравнительный анализ представленной модели и аналогичных показал, что в отличие от предложенной в [6] она обладает большей универсальностью, так как позволяет выполнять посимвольное, а не только построчное детектирование. При этом по сравнению с моделью, описанной в [7], данная модель не требует бесконтрольного обучения и содержит значительно меньше настраиваемых параметров (более чем в 40 раз), что повышает эффективность ее практического использования.

2. Алгоритм прохода детектора по изображению

Основной сферой применения детектора являются изображения реальных сцен, которые могут содержать произвольно распределенные текстовые объекты различного размера в отличие от объектов фиксированных размеров 32×32 пиксела для его входного слоя. С учетом указанных особенностей был разработан алгоритм применения детектора на основе мультимасштабного скользящего окна (рис. 3), состоящий из следующих шагов:

1) выполним масштабирование изображения к фиксированному размеру $H \times W$;

2) зададим диапазон изменения масштаба, например [–30, 30 %] с шагом 15 %;

3) выберем величину смещения скользящего окна по осям x (dx) и y (dy);

4) в цикле по каждому масштабу дополним изображение рамкой толщиной 16 пикселов с усредненной локальной яркостью края (для выделения текста, прилегающего к границам);

5) будем перемещать скользящее окно размером 32×32 пиксела слева направо и сверху вниз с шагами dx и dy так, чтобы центр окна (x, y) не попадал в рамку;

6) в текущей позиции окна выделим окрестность с координатами (x – 15, y – 15), (x + 16, y + 16) левого верхнего и правого нижнего угла, которую подадим на вход детектора;

7) сохраним отклики детектора в каждой позиции и каждом масштабе.

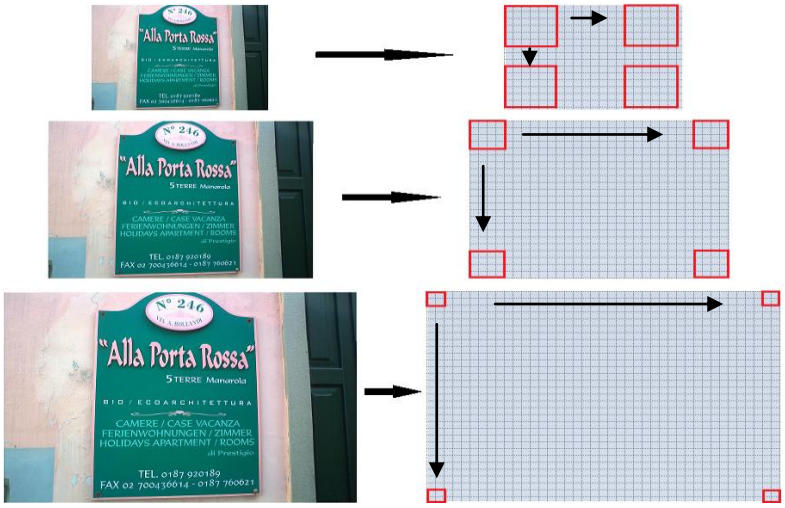


Рис. 3. Использование детектора в соответствии с алгоритмом мультимасштабного скользящего окна

Предложенный алгоритм позволяет обрабатывать текстовые образы различного размера, расположенные в произвольной части изображения, что значительно повышает универсальность детектора. Его недостатком являются значительные временные затраты ввиду необходимости выполнения многочисленных этапов прямой передачи сигнала в нейросети. Возможными путями их сокращения являются:

– использование информации о предполагаемых размерах текстовых образов, позволяющей уменьшить число масштабов (например, при обработке изображений сцен, полученных со стационарной камеры с фиксированными расстояниями до текстовых объектов);

– учет типичного пространственного распределения текстовых блоков (при анализе однотипных изображений с похожей композицией, например дорожных сцен);

– увеличение шага скользящего окна вдоль координатных осей, что приводит к уменьшению числа вызовов детектора;

– получение дополнительных данных о возможном размещении текстовой информации (например, исключение из обработки областей с недостаточной плотностью контуров).

Указанные направления оптимизации позволяют сократить время работы алгоритма на основе не всегда доступной априорной информации о структуре и свойствах сцены. Поэтому поиск путей дальнейшего усовершенствования был связан с анализом архитектуры сети и передачей сигналов в ней, который привел к следующему выводу: ввиду высокой вероятности пересечения окрестностей изображения, подаваемых на вход детектора в каждом проходе (при $dx < 32$ или $dy < 32$), целесообразным является вычисление его откликов не для отдельных, а для всех возможных окрестностей сразу. Другими словами, для нейронов каждого слоя можно последовательно сформировать входные сигналы и затем, группируя и передавая их, вычислить отклики детектора по всем позициям скользящего окна одновременно.

Преимущество данного подхода можно оценить на примере вычисления входных сигналов для нейронов сверточного слоя C1, содержащего 12 карт по 28×28 нейронов, которые разделяют по одному фильтру размером 5×5. Рассмотрим два локальных окна размером 32×32 пиксела с центрами в (x, y) и (x, y + 1), для которых необходимо выполнить свертку с не-

которым фильтром слоя. Пространственная фильтрация проводится путем вычисления сумм поэлементных произведений каждого подокна размером 5×5 с фильтром в 28×28 позиций с шагом 1. У двух рассматриваемых локальных окон результаты фильтрации будут отличаться лишь двумя крайними столбцами, поэтому более 90 % расчетов для первого окна могут быть использованы и для второго. Следовательно, для вычисления всех возможных входных сигналов для нейронов C1 нужно один раз выполнить свертку изображения с каждым из 12 фильтров и объединить результаты в соответствии со структурой слоя.

Аналогичным образом рассчитываются входные сигналы для остальных слоев, т. е. локальная интерпретация откликов детектора эффективно сочетается с глобальным характером вычислительной процедуры. Кроме того, расчеты можно сократить, учитывая величины смещений скользящего окна. Например, когда $dx = dy = 2$, вычисление входов для нейронов первого подвыборочного слоя S2 путем усреднения выходов C1 по окрестностям 2×2 может выполняться с шагом, равным 2, а не 1. Для сравнения, использование описанных оптимизаций при обработке изображения размером 640×480 пикселов позволяет сократить время расчетов более чем в 10 раз для оборудования следующей конфигурации: процессор Intel Core 2 Duo i3-530 2.93 GHz, ОЗУ DDR3 2048 Mb, ОС Windows 7 Ultimate, платформа .NET.

3. Алгоритм локализации текста на основе анализа откликов детектора

Результатом прохода нейросетевого детектора по изображению в одном масштабе является матрица откликов, каждый элемент которой представлен парой (t_1, t_2), где t_1 – выход фонового (первого) нейрона, t_2 – текстового. Если для соответствующего точке (x, y) элемента матрицы $t_2 > t_1$, то справедливо утверждение: окрестность изображения с координатами $(x - 15, y - 15)$ левого верхнего и $(x + 16, y + 16)$ правого нижнего угла содержит текстовый образ.

Функцией активации нейросетевого детектора является гиперболический тангенс, а обучение проводится так, чтобы $t_1 \to 1$ (–1), $t_2 \to -1$ (1) при подаче на вход фоновой (текстовой) области. Однако проведенные эксперименты показали, что для различных сигналов значения t_i могут быть произвольно распределены в диапазоне [–1, 1]. Поэтому для оценки решений детектора о наличии текстовых образов целесообразным является применение матрицы уверенностей (графическая интерпретация представлена на рис. 4, *а*), элементы которой в каждой точке (x, y) вычисляются по формуле

$$U(x, y) = max\ \{1 - (t_1 + 1) / (t_2 + 1), 0\}.$$

В основе предлагаемого алгоритма выделения текстовых областей лежит анализ пространственного распределения уверенностей откликов детектора. Как видно из рис. 4, *б*, окрестностям текстовых образов соответствуют высокие значения уверенностей (чем темнее, тем выше). При этом детектор может давать уверенный отклик и в окрестностях с неотцентрированным образом, а также на фоновых участках с текстурой, подобной текстовой. Таким образом, матрицу уверенностей преимущественно можно использовать для «поблочной» интерпретации распределения текстовой информации на изображении.

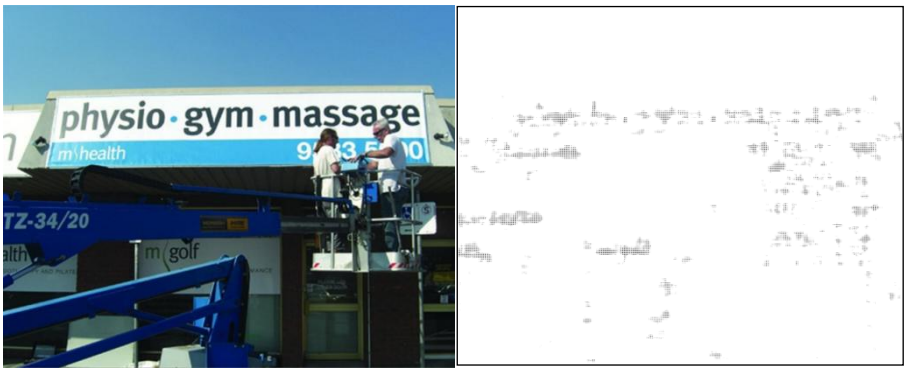
Основными этапами предлагаемого алгоритма локализации являются:

1) вычисление матриц уверенностей на разных масштабах изображения;
2) образование первичных текстовых блоков на отдельных масштабах;
3) формирование итоговых текстовых блоков путем объединения первичных.

Первый этап реализуется с помощью алгоритма мультимасштабного скользящего окна, описанного в разд. 2, для дискретного набора масштабов.

Образование первичных текстовых блоков. Детектор имеет уверенный отклик на фрагментах изображения (размером 32×32 пиксела), содержащих отцентрированный текстовый образ (центры таких фрагментов будем называть истинными прообразами). Следовательно, высота выделяемых символов не превышает 32 пиксела (точное значение определяется ее величиной у примеров выборки, используемой для обучения детектора), что позволяет ввести в рассмотрение следующие параметры: $max\_dy = 32$, $min\_dy = max\_dy / 3 \approx 10$ – максимальное и минималь-

ное расстояния по горизонтали в пикселах между символами. Кроме того, необходимо задать минимальный порог уверенности для истинного прообраза символа, например *min_U* = 0,8.



а) *б)*

в) *г)*

Рис. 4. Примеры изображений: *а)* сцены; *б)* матрицы уверенности детектора для изображения в одном масштабе; *в)* блоков различных масштабов изображения; *г)* итоговых текстовых блоков

Поиск истинных прообразов проведем с помощью *немаксимального подавления*:

– для текущей строки изображения сформируем список прообразов, уверенность которых больше порога *min_U*;

– обойдем список в порядке убывания уверенности элементов;

– отбросим прообраз, если в списке есть более уверенный и близкий к нему (расстояние по горизонтали не превышает *min_dy*) или же такой элемент находился в списке, но при этом его уверенность была больше не менее чем на величину $dU\_one\_max = min\_U / 10 = 0{,}08$.

В результате в списке останутся элементы, являющиеся локальными максимумами текущей строки матрицы уверенностей. Применение параметра *dU_one_max* позволяет корректно обработать случай, когда один прообраз мог удалить остальные, относящиеся к тому же символу, а сам он мог быть исключен прообразом другого, близко расположенного к нему. Используя полученный список, можно сформировать первичные текстовые блоки следующим образом:

– обойдем список в порядке убывания уверенности элементов; первый из свободных элементов, не отнесенных на текущий момент ни к одному текстовому блоку, образует новый;

– просмотрим список с начала, пока не встретим прообраз, уверенность которого превышает среднюю уверенность блока или меньше ее не более чем на величину $dU\_two\_max = 2 \cdot dU\_one\_max = 0{,}16$ и отдаленный не более чем на *max_dy* относительно любого элемента блока;

– если элемент был найден, добавим его к блоку, пересчитаем среднюю уверенность и повторим действия предыдущего пункта, иначе сохраним характеристики текущего блока, включая минимальный прямоугольник, текущий масштаб изображения, среднюю уверенность, а также данные о позиции и уверенности каждого элемента.

Процедуру немаксимального подавления и формирования блоков проведем для каждой строки изображения в каждом его масштабе из дискретного набора, зафиксированном на первом этапе. В результате получим первичную информацию о распределении текста.

Объединение первичных блоков в итоговые. Построенные блоки могут относиться к одному текстовому объекту ввиду того, что их формирование проходило на близких строках изображения. Поэтому необходимо выполнить объединение первичных блоков одного масштаба и пересчитать их характеристики:

– будем обрабатывать блоки в порядке убывания средней уверенности, пока не найдем два с достаточной площадью пересечения минимальных прямоугольников (более 50 % от площади любого);

– выполним их посимвольное слияние на базе более уверенного (первого), обходя элементы второго в порядке убывания уверенности:

- сравним координаты и уверенность текущего прообраза (x_2, y_2) второго блока со средней вертикальной координатой (x^1_{aver}) и уверенностью (U^1_{aver}) элементов первого; если $|x^1_{aver} - x_2| > max\_dx$ или $U^1_{aver} - U(x_2, y_2) > dU\_two\_max$ (где $max\_dx = 10$), отбросим этот прообраз;
- если для некоторого прообраза (x_1, y_1) первого блока справедливо $|y_1 - y_2| \le min\_dy$, считаем, что это прообразы одного символа, при этом, когда $|U(x_1, y_1) - U(x_2, y_2)| < dU\_one\_max$, усредним их координаты, иначе запомним характеристики более уверенного прообраза;
- в противном случае добавим текущий прообраз второго блока к первому;
- если изменились характеристики элементов, выполним проверку, аналогичную первым двум подпунктам для всех прообразов первого блока;

– пересчитаем характеристики первого блока и изменим очередность его обработки в соответствии с новым значением средней уверенности.

Примеры тренировочной выборки, используемые для создания детектора, содержат символы различной высоты. Одной из причин данного явления помимо естественной вариативности размеров является искажение образов на каждой эпохе обучения. В связи с этим блоки, сформированные на близких масштабах, так же, как и первичные блоки одного масштаба, могут представлять одинаковый текстовый объект. Справедливость данного замечания подтверждается представленными на рис. 4, *в* блоками, сформированными на разных масштабах изображения.

На заключительном этапе выполним следующие шаги:

– приведем координаты блоков и их образов к единому масштабу;

– объединим блоки с помощью описанной выше процедуры, наложив дополнительное ограничение: исходный масштаб каждого символа блока должен отличаться от среднего не более чем на величину $max\_dS = 0{,}5$;

– отбросим блоки, средняя уверенность которых меньше максимальной более чем на величину dU_one_max.

Изображение итоговых блоков представлено на рис. 4, *г*, при этом помимо координат минимальных прямоугольников результатом работы алгоритма являются также матрицы уверенностей и масштабы блоков.

4. Экспериментальная работа и анализ результатов

Для обучения детектора необходимы маркированные примеры двух классов: фоновые и текстовые, являющиеся полутоновыми изображениями размером 32×32 пиксела. Первые должны содержать фрагменты фонов с различным цветовым составом, текстурой, искусственными и естественными объектами, вторые – текстовый образ, вписанный в изображение с сохранением пропорций его размера. Так как символы слова могут располагаться плотно друг за другом, высока вероятность наличия в его квадратной окрестности сразу нескольких символов, поэтому подобные примеры также нужно включать в обучающее множество, при этом один из символов должен находиться в центре изображения. В целом разнообразие учебных примеров определяется композицией сцен, на которых планируется применять детектор, а также алфавитной принадлежностью искомого текста. Ниже демонстрируется пример создания детектора цифр и заглавных символов английского языка.

Способность детектора автоматически обучаться выделению признаков, по которым возможно разделение текстовых и фоновых объектов, осложняется необходимостью создания объемной базы примеров. Данный процесс связан с определением в множестве изображений сцен позиций минимальных объемлющих прямоугольников отдельных символов, что с гарантированно высоким качеством может быть выполнено только вручную. Некоторые авторы предлагают процедуры генерации искусственных образов путем наложения изображений символов на фиксированные фоны, что облегчает формирование базы [6]. Однако больший потенциал имеют детекторы, обученные на реальных данных, информативность которых значительно выше. В этой связи для выполнения экспериментальной работы была собрана обучающая выборка, включающая следующие базы:

ICDAR 2003 – одна из главных точек отсчета в исследованиях по детектированию, на основе которой проводится сравнение алгоритмов поиска и сегментации текста на изображениях реальных сцен, содержит 12 469 образов символов [8];

74K – представлена изображениями текстов на улицах индийских городов; помимо минимальных прямоугольников содержит маркеры символов, разделена на основную (7705 примеров) и зашумленную (4798 примеров) части [9];

KAIST – представлена изображениями сцен в корейских городах; как и в предыдущих базах, текстовые блоки описаны на уровне строк, слов и символов; разделена на экземпляры, полученные камерой с высоким и низким разрешением (камерой мобильного телефона); содержит 11 537 текстовых образов [10];

SVHN – создана путем сегментации текстовых образов на изображениях табличек номеров домов, содержит 99 289 примеров цифр [11];

CVL OCR DB – словенская база данных, включает 7014 изолированных текстовых образов, выделенных на изображениях рекламных плакатов, дорожных знаков, названий магазинов и др. [12].

Из перечисленных баз были выделены изображения квадратных окрестностей цифр и заглавных букв английского языка, которые масштабировались к размеру 32×32 пиксела. Из получившейся выборки были удалены образы, имеющие низкую уникальность, для оценки которой использовалось суммарное квадратичное отклонение. Кроме того, было сокращено представительство классов, значительно превышающее остальные (например, часто используемых символов 'A', 'C', 'S' и др.). При этом ввиду схожести начертания было усреднено число экземпляров символов 'I' и '1', 'O' и '0', а к классам с недостаточным количеством примеров добавлены их прописные образы в случае их схожести с заглавными, например 'w' и 'W', 'k' и 'K'. В итоге была получена выборка, содержащая 29 172 образа букв и 13 500 цифр, которая была разделена на тренировочную и тестовую части путем отнесения к первой 80 % образов каждого класса. Для получения контрпримеров были использованы фоновые фрагменты изображений сцен указанных баз, в результате общее число образов выборки составило 85 344. Создание детектора выполнялось на базе описанной выше модели в течение четырех циклов обучения: после первого точность классификации образов тренировочной части составила 94,97 %, тестовой – 93,56 %, после четвертого – 96,21 и 94,95 % соответственно. Анализ неверно классифицированных образов показал, что ошибки на текстовых примерах были вызваны их расфокусировкой и значительным искажением шрифта, фоновые же примеры содержали фрагменты, весьма похожие на символы.

Для оценки эффективности предложенной модели созданный детектор был применен для решения задачи автоматического обнаружения минимальных прямоугольников слов в рамках соревнования ICDAR 2013 Robust Reading Competition – Challenge 2: Reading Text in Scene Images – Task 1: Text Localization [13]. Качество обнаружения определялось по результату обработки 233 тестовых изображений реальных сцен (рис. 5), что в количественном измерении выражалось тремя параметрами: *recall* = (число верно локализованных слов) / (общее число слов в выборке), *precision* = (число верно локализованных слов) / (общее число выделенных слов), $F\text{-score} = 2 \cdot recall \cdot precision / (recall + precision)$. При этом использовались специальный способ определения корректности локализации слова с вероятностным выходом и система штрафов в ситуациях соответствия типа «один ко многим» и «многие ко многим» (подробную информацию можно найти в [14]).

Учитывая, что по условию задачи оценке подвергалось качество обнаружения минимальных прямоугольников слов, алгоритм локализации был дополнен следующей процедурой сегментации текстовых блоков:

– обрабатываются образы блока в порядке убывания уверенности;

– первый образ, не отнесенный на текущий момент ни к одному слову, образует новое;

– просматриваются образы блока с начала, пока не будет найден образ, который может быть добавлен к текущему слову при выполнении следующих условий:

- расстояние от образа до какого-либо элемента слова не должно превышать $min\{max\_dy, 1.5 \cdot aver\_dy\}$, где *aver_dy* – средняя удаленность образов слова;
- уверенность, масштаб и горизонтальная координата образа не должны отличаться от их усредненных значений для слова более чем на установленные выше пороговые величины.

Рис. 5. Примеры локализации текста, выполненной с помощью предложенной модели нейросетевого детектора, на изображениях базы ICDAR 2013

Используя дополненную версию алгоритма, была получена следующая оценка локализации слов: *recall* = 60,73, *precision* = 55,14, *F-score* = 57,80 (14-я в списке лучших). Данный результат подтверждает работоспособность созданного детектора, учитывая сложность композиции изображений и высокую стилистическую вариативность текстовых образов. При этом оценка локализации может быть существенно улучшена за счет совершенствования процедуры сегментации блока на слова (для уменьшения штрафов в ситуациях «один ко многим»), что не являлось главной задачей при разработке алгоритма интерпретации откликов детектора.

Качественное сравнение разработанного алгоритма с аналогичными из [6, 7] позволяет отметить, что он обладает большей универсальностью по следующим причинам:

– формирование итоговых блоков происходит путем объединения блоков-кандидатов с учетом индивидуальных характеристик символов, а не простым выбором наиболее уверенного кандидата, что, в частности, не накладывает завышенных требований к точности детектора, позволяя использовать в его основе значительно менее громоздкую нейросетевую архитектуру;

– алгоритм позволяет обнаруживать наклонные текстовые строки, причем максимальный уровень наклона может быть увеличен за счет изменения параметра *max_dx*;

– объединение результатов обработки на разных масштабах позволяет формировать блоки с регулируемым уровнем отличия размеров символов (*max_dS*), что необходимо для учета отличия регистра символов и вариативности их стилистического оформления.

Заключение

В статье представлена модель текстового детектора на базе сверточной нейросети авторской архитектуры, преимуществом которой, по сравнению с аналогичными, является сочетание высокой обобщающей способности и низкой вычислительной стоимости обучения и применения. Предложен алгоритм прохода детектора по изображению, позволяющий обнаруживать текстовые образы произвольного размера и пространственного распределения. Описаны как универсальные пути его оптимизации, так и основанные на априорной контекстной информации. Разработан алгоритм локализации текстовых блоков на изображениях реальных сцен путем интерпретации откликов детектора. В отличие от аналогичных он не предъявляет завышенных требований к точности детектора, позволяет применять емкие нейросетевые архитектуры и обнаруживать наклонные текстовые объекты, обладает большей универсальностью за счет объединения откликов детектора, полученных на различных масштабах входного изображения.

Список литературы

1. Sumathi, C.P. A Survey on various approaches of text extraction in images / C.P. Sumathi, T. Santhanam, G. Gayathri // International Journal of Computer Science & Engineering Survey. – 2012. – Vol. 3, № 4. – P. 27–42.

2. LeCun, Y. Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou // Proceedings of the IEEE. – 1998. – Vol. 86, № 11. – P. 2278–2324.

3. Кузьмицкий, Н.Н. Сверточная нейросетевая модель в задаче классификации изображений изолированных цифр / Н.Н. Кузьмицкий // Доклады БГУИР. – Минск, 2012. – № 7. – С. 64–70.

4. Головко, В.А. Нейронные сети: обучение, организация и применение : учеб. пособие / В.А. Головко. – М. : ИПРЖР, 2001. – Кн. 4. – 256 с.

5. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2002. – 344 с.

6. Delakis, M. Text detection with convolutional neural networks / M. Delakis, Cr. Garcia // Intern. Conf. on Computer Vision Theory and Applications. – Cambridge, 2008. – P. 290–294.

7. Wang, K. End-to-end scene text recognition / K. Wang, B. Babenko, S. Belongie // IEEE Intern. Conf. on Computer Vision (ICCV). – Barcelona, 2011. – P. 1457–1464.

8. ICDAR 2003 robust reading competitions / S.M. Lucas [et al.] // Proc. of Seventh Intern. Conf. on Document Analysis and Recognition. – Edinburgh, 2003. – P. 682–687.

9. Campos, T.E. Character Recognition in Natural Images / T.E. Campos, B.R. Babu // VISAPP. – 2009. – Vol. 2. – P. 273–280.

10. Touch TT : Scene text extractor using touchscreen interface / J. Jung [et al.] // ETRI Journal. – 2011. – Vol. 33, № 1. – P. 78–88.

11. The Street View House Numbers (SVHN) Dataset [Electronic resource]. – 2011. – Mode of access : http://ufldl.stanford.edu/housenumbers. – Date of access : 03.07.2014.

12. Ikica, A. An improved edge profile based method for text detection in images of natural scenes / A. Ikica, P. Peer // Intern. Conf. on Computer as a Tool (EUROCON). – Lisbon, 2011. – P. 1–4.

13. ICDAR 2013 Robust Reading Competitio / D. Karatzas [et al.] // Proc. 12th Intern. Conf. of Document Analysis and Recognition, IEEE CPS. – Washington, 2013. – P. 1115–1124.

14. Wolf, C. Object Count Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms / C. Wolf, J.M. Jolion // International Journal of Document Analysis. – 2006. – Vol. 8, № 4. – P. 280–296.

Поступила 07.04.2015

Брестский государственный
технический университет,
Брест, ул. Московская, 267
e-mail: knnbrest@yandex.ru

N.N. Kuzmitsky

DETECTION OF TEXT OBJECTS IN IMAGES OF REAL SCENES BASED ON CONVOLUTIONAL NEURAL NETWORK MODEL

A model of text image detector based on a convolutional neural network architecture is presented, capable of synthesizing high-level features of images in the «black box» mode. An implementation of the detector application, based on algorithms of multi-scale scanning and local responses interpretation is described, allowing to find out text samples on images of real scenes. Advantages in comparison with analogs are shown and efficiency evaluation on an example of a known database is conducted.

УДК 004

Б.А. Залесский, Э.Н. Середин, Н.В. Ядловский

ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ НА ОСНОВЕ СРАВНЕНИЯ ГИСТОГРАММ ЦВЕТА

Предлагаются три версии гистограммного алгоритма отслеживания объектов на видеопоследовательностях, снятых нестабилизированной камерой. Версии основываются на сравнении с помощью критерия Баттачариа гистограмм цвета областей кадров либо гистограмм цвета пар ближайших пикселов областей кадров, которые могут рассматриваться как частный случай матриц встречаемости. Использование технологии программирования видеокарты CUDA позволяет добиться выполнения версий в режиме реального времени. Проводится сравнение модифицированного алгоритма с хорошо известным алгоритмом среднего сдвига, основанным на линейном приближении критерия Баттачариа. На основе экспериментов показывается, что модификации алгоритма являются более точными и надежными по сравнению с алгоритмом среднего сдвига, хотя и требуют большего объема вычислений. Проводится сравнение предложенных версий с корреляционными алгоритмами.

Введение

В последние годы алгоритмы отслеживания объектов широко используются для решения прикладных задач в различных областях народного хозяйства, например для обеспечения безопасности дорожного движения, сохранности имущества, в навигации и т. д.

Для отслеживания объектов на видеопоследовательностях, снятых в разных условиях, используются различные подходы. Рассмотрим случай, когда видеопоследовательность снята камерой, установленной над объектом, например на борту летательного аппарата. Задача заключается в отслеживании в реальном времени объекта интереса, наблюдаемого камерой стандартного разрешения. Особенность данной задачи состоит в относительно невысоком качестве получаемого видео, часто нестабилизированного, кадры которого обычно искажены шумами различных типов.

В настоящее время известны несколько подходов к решению сформулированной задачи, например основанные на вычислении среднего сдвига, сравнении корреляций, гистограмм яркостей и ориентированных градиентов, оптических потоков, текстур, активных контуров и др. [1–7]. Использование сложных алгоритмов, требующих большого объема вычислений, ограничено в данном случае малым допустимым временем (25÷40 мс), необходимым для обработки каждого кадра.

Алгоритм среднего сдвига (mean shift) был предложен для преодоления упомянутых трудностей. Использование линейного приближения произведения гистограмм выделенных областей начального и текущего кадров, которое реализует критерий Баттачариа, дает возможность уменьшить время вычисления так называемого среднего сдвига текущего кадра [1] настолько, что процессорные версии алгоритма выполняются на современных персональных компьютерах всего за несколько миллисекунд. Однако проведенное тестирование алгоритма на видеопоследовательностях невысокого качества показало его меньшую точность и надежность по сравнению с другими алгоритмами, например корреляционными.

В статье предложены три версии алгоритма отслеживания объектов на видеопоследовательностях, в которых, как и в алгоритме среднего сдвига, используется критерий Баттачариа для сравнения многомерных гистограмм цвета (инвариантных или почти инвариантных к повороту изображения) при выполнении отслеживания объекта интереса. Однако при этом, в отличие от известного алгоритма, сам критерий не заменяется на его линейное приближение. Во второй и третьей версиях алгоритма используются гистограммы яркостей соседних пикселов, которые могут рассматриваться как один из вариантов матриц встречаемости [8]. Приведены результаты сравнения точности и надежности разработанных версий с известными алгоритмами среднего сдвига и корреляционного поиска.

Использование исходного представления критерия Баттачариа в виде произведения нормированных квантизованных гистограмм позволило повысить точность и надежность оценки координат отслеживаемого объекта, хотя и привело к увеличению объема выполняемых вычислений. Возникшие трудности с программированием работающих в режиме реального времени версий были преодолены с помощью технологии программирования видеокарты CUDA. Использование видеокарты позволило сократить время поиска объекта на каждом кадре до 10–15 мс.

1. Описание алгоритма отслеживания объектов на основе критерия Баттачариа

Предложенные версии алгоритма отслеживания объектов могут быть использованы для полутоновых и *RGB*-изображений, но, в связи с тем что случай цветных изображений более труден для реализации из-за большего объема вычислений, здесь будут рассмотрены только цветные видеопоследовательности.

Обозначим через $\mathbf{I}$ *RGB*-изображение с пикселами $p = (x, y)$, образующими регулярную решетку $S = \{0,\ldots, M-1\}\times\{0,\ldots, N-1\}$, и будем считать, что координаты цвета пикселов $I(p) = (I_R(p), I_G(p), I_B(p))$ изменяются в диапазоне $0,\ldots, L-1$. Кадры видеопоследовательности в моменты времени $t = \{0, 1,\ldots\}$ будем обозначать $\mathbf{I}_t$. Все три версии предложенного алгоритма отслеживания основаны на сравнении оконных цветовых гистограмм различных типов с помощью критерия Баттачариа. Гистограммы строятся по квантизованным изображениям $\mathbf{J}$, яркости цветовых каналов которых изменяются в меньшем диапазоне $0,\ldots, l-1$ $(0 < l \le L)$ и задаются равенствами

$$J_R(p)=\left[\frac{(l-1)I_R(p)}{(L-1)}\right],\quad J_G(p)=\left[\frac{(l-1)I_G(p)}{(L-1)}\right],\quad J_B(p)=\left[\frac{(l-1)I_B(p)}{(L-1)}\right],$$

где $[a]$ – целая часть числа a. Квантизация изображений производится по двум причинам. Во-первых, использование меньшего диапазона яркостей уменьшает объем требуемых вычислений и, следовательно, сокращает время обработки кадров. Во-вторых, квантизация снижает влияние на результаты нелинейных изменений яркостей кадров, присущих в той или иной мере всем камерам. На реальных 24-битных *RGB*-видеопоследовательностях ($L = 256$) значения $l = 8,\ldots, 32$ позволяют получать достаточно точные и надежные результаты. При $l > 32$ точность и надежность результатов практически не повышаются, но время вычислений становится недопустимо большим. При $l < 8$ ухудшается точность.

Пусть окно $O(p)$ – квадрат с центром в пикселе p, содержащий $m\times m$ пикселов, $O(p_0)$ – окно, содержащее объект интереса, выделенный на исходном кадре $\mathbf{I}_0$. В первой версии алгоритма используются традиционные для метода средних сдвигов сглаженные оконные гистограммы $\mathbf{h}_{t,p}$ квантизованных кадров $\mathbf{J}_t$, которые представляют собой 3D-матрицы, индексированные квантизованным цветом:

$$\mathbf{h}_{t,p}=\left\{\mathrm{h}_{t,p}(n_R, n_G, n_B)\right\}_{0\le n_R,\, n_G,\, n_B\le l-1},$$

вида

$$\mathrm{h}_{t,p}(n_R, n_G, n_B) = \frac{1}{C}\sum_{q\in O(p)} w(q-p)\mathbf{1}_{(J_t(q)=(n_R, n_G, n_B))}, \qquad (1)$$

где нормирующая константа C выбирается так, чтобы сумма всех элементов гистограммы была равна 1; w – сглаживающая функция, а $\mathbf{1}_{(J_t(q)=(n_R, n_G, n_B))}$ – индикатор, равный 1, если $J(q) = (n_R,n_G,n_B)$, и равный 0 в противном случае. В качестве сглаживающей функции во всех трех версиях использовалась либо экспонента

$$w(p) = w_r(p) = exp\left(-\tfrac{1}{r}\|p\|^2\right),\quad r > 0,$$

либо ядро Епанечникова

$$w(p) = w_r(p) = \begin{cases} 1 - \frac{1}{r}\|p\|^2, & \text{если } \|p\|^2 < r, \\ 0 & \text{в противном случае} \end{cases}$$

с параметром масштаба r.

В качестве меры близости двух гистограмм применялся непосредственно критерий Бат-тачариа, который задается равенством

$$\alpha(\mathbf{h}', \mathbf{h}'') = \sum_{n_R=0, n_G=0, n_B=0}^{l-1, l-1, l-1} \sqrt{h'(n_R, n_G, n_B) h''(n_R, n_G, n_B)}. \quad (2)$$

(Напомним, что в классическом алгоритме среднего сдвига вместо $\alpha(\mathbf{h}', \mathbf{h}'')$ используется его линейное приближение.)

Были исследованы две модификации алгоритма поиска на основе критерия α. В одной из модификаций оценка $\hat{p}_t$ положения окна, содержащего объект интереса на текущем кадре $\mathbf{J}_t$, выполнялась на основе глобального поиска по всему кадру:

$$\hat{p}_t = \underset{q \in S}{\operatorname{argmax}} \left\{ \alpha \left(\mathbf{h}_{0,p_0}, \mathbf{h}_{t,q} \right) \right\}.$$

Во второй модификации первой версии алгоритма использовался локальный поиск в окне $W\left(\hat{p}_{t-1}\right)$ с центром $\hat{p}_{t-1}$, равным оценке положения объекта интереса на предыдущем кадре, и размером, зависящим от оценки максимальной скорости движения объекта относительно кадра:

$$\hat{p}_t = \underset{q \in W(\hat{p}_{t-1})}{\operatorname{argmax}} \left\{ \alpha \left(\mathbf{h}_{0,p_0}, \mathbf{h}_{t,q} \right) \right\}.$$

Каждая из модификаций имеет свои преимущества и недостатки, которые будут описаны в разд. 2.

Вторая версия предложенного алгоритма основана на сравнении сглаженных оконных гистограмм $\tilde{\mathbf{h}}_{t,p}$ совместных яркостей квантизованых кадров $\mathbf{J}_t$ пар ближайших пикселов, которые представляют собой 6D-матрицы, индексированные парами квантизованных цветов ближайших пикселов:

$$\tilde{\mathbf{h}}_{t,p} = \left\{ \tilde{\mathrm{h}}_{t,p}\left(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}\right) \right\}_{0 \le n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{1,B} \le l-1}.$$

Сглаженная оконная гистограмма $\tilde{\mathbf{h}}_p$ для изображения $\mathbf{J}$ может быть представлена в виде суммы, аналогичной сумме (1), однако представление получается весьма громоздким, поэтому приведем конструктивный способ ее описания. Для пиксела q обозначим через q' его ближайшего правого соседа, а через q'' его ближайшего верхнего соседа. Нетрудно заметить, что $q' = q + (1,0)$, а $q'' = q - (0,1)$. Алгоритм построения $\tilde{\mathbf{h}}_p$ состоит из следующих шагов:

Шаг 1. Полагаем все $\tilde{\mathrm{h}}_{t,p}\left(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}\right) = 0$.

Шаг 2. Просматриваем все пикселы q окна $O(p)$. Для каждого пиксела q и переменной $F \in \{R, G, B\}$ вычисляем

$$n_{1,F} = \min\left(J_F(q), J_F(q')\right), \; n_{2,F} = \max\left(J_F(q), J_F(q')\right),$$

а затем находим

$$\tilde{\mathrm{h}}_{t,p}\left(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}\right) := \tilde{\mathrm{h}}_{t,p}\left(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}\right) + w(q - p).$$

Шаг 3. Просматриваем все пикселы q окна $O(p)$. Для каждого пиксела q и переменной $F \in \{R, G, B\}$ вычисляем

$$n_{1,F} = \min\left(J_F(q), J_F(q'')\right), \quad n_{2,F} = \max\left(J_F(q), J_F(q'')\right),$$

а затем находим

$$\tilde{\mathrm{h}}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) := \tilde{\mathrm{h}}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) + w(q-p).$$

Шаг 4. Нормируем гистограмму $\tilde{\mathbf{h}}_{t,p}$ так, чтобы сумма ее элементов была равна единице.

Теоретически построенная нижняя треугольная 6D-матрица $\tilde{\mathbf{h}}$ является инвариантной только к поворотам на углы от $\omega \in \left(\frac{\pi}{2}j - \frac{\pi}{6}, \frac{\pi}{2}j + \frac{\pi}{6}\right)$, $j = 0, 1, 2, 3$, но практически она лишь незначительно меняется при других углах поворота, что позволяет применять ее для реального сравнения областей изображений, повернутых на произвольные углы, о чем свидетельствуют проведенные эксперименты.

Во второй версии алгоритма для поиска объекта интереса использовался критерий Баттачариа

$$\beta(\tilde{\mathbf{h}}', \tilde{\mathbf{h}}'') = \sum_{\substack{n_{1,R}=0, n_{2,R}=0, n_{1,G}=0 \\ n_{2,G}=0, n_{1,B}=0, n_{2,B}=0}}^{l-1, l-1, l-1, l-1, l-1, l-1} \sqrt{\tilde{\mathrm{h}}'(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B})\, \tilde{\mathrm{h}}''(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B})}. \qquad (3)$$

В первой модификации второй версии алгоритма по аналогии с первой модификацией первой версии осуществляется поиск по всему текущему кадру:

$$\hat{p}_t = \underset{q \in S}{\operatorname{argmax}} \left\{ \beta\left(\tilde{\mathbf{h}}_{0,p_0}, \tilde{\mathbf{h}}_{t,q}\right) \right\}.$$

Во второй модификации второй версии алгоритма осуществляется локальный поиск на текущем кадре в окне $W\left(\hat{p}_{t-1}\right)$, центр которого полагается равным оценке координат объекта на предыдущем кадре $\hat{p}_{t-1}$:

$$\hat{p}_t = \underset{q \in W(\hat{p}_{t-1})}{\operatorname{argmax}} \left\{ \beta\left(\tilde{\mathbf{h}}_{0,p_0}, \tilde{\mathbf{h}}_{t,q}\right) \right\}.$$

Вторая версия обеспечивает наиболее точное и надежное отслеживание объекта среди всех предложенных в данной статье, однако она требует наибольшего числа вычислений, которые достаточно трудно осуществить в режиме реального времени.

Для случая ограниченных вычислительных ресурсов предлагается третья версия алгоритма отслеживания объектов, в которой используются сглаженные оконные гистограммы ${}_R\bar{\mathbf{h}}_{t,p}$, ${}_G\bar{\mathbf{h}}_{t,p}$, ${}_B\bar{\mathbf{h}}_{t,p}$ совместных яркостей RGB-каналов квантизованных кадров $\mathbf{J}_t$, представляющие собой 3D-матрицы, индексированные парами квантизованных цветов ближайших пикселов:

$${}_A\bar{\mathbf{h}}_{t,p} = \left\{ {}_A\mathrm{h}_{t,p}(n_1, n_2) \right\}_{0 \le n_1, n_2 \le l-1}, \quad A \in \{R, G, B\}.$$

Напомним, что для пиксела q его ближайший правый сосед обозначен буквой q', а ближайший верхний сосед – буквой q'' так, что $q' = q + (1, 0)$, а $q'' = q - (0, 1)$.

Конструктивный способ построения ${}_R\bar{\mathbf{h}}_{t,p}$, ${}_G\bar{\mathbf{h}}_{t,p}$, ${}_B\bar{\mathbf{h}}_{t,p}$ состоит из следующих шагов:

Шаг 1. Полагаем все ${}_A\bar{\mathbf{h}}_{t,p} = 0$, $A \in \{R, G, B\}$.

Шаг 2. Просматриваем все пикселы q окна $O(p)$. Для каждого пиксела q и переменной $A \in \{R, G, B\}$ вычисляем

$$n_{1,A} = \min\left(J_A(q), J_A(q')\right), \; n_{2,A} = \max\left(J_A(q), J_A(q')\right),$$

а затем находим

$${}_A\bar{h}_{t,p}(n_{1,A}, n_{2,A}) := {}_A\bar{h}_{t,p}(n_{1,A}, n_{2,A}) + w(q-p).$$

Шаг 3. Просматриваем все пикселы q окна $O(p)$. Для каждого пиксела q и переменной $A \in \{R, G, B\}$ вычисляем

$$n_{1,A} = \min\left(J_A(q), J_A(q'')\right), \;\; n_{2,A} = \max\left(J_A(q), J_A(q'')\right),$$

а затем находим

$${}_A\bar{h}_{t,p}(n_{1,A}, n_{2,A}) := {}_A\bar{h}_{t,p}(n_{1,A}, n_{2,A}) + w(q-p).$$

Шаг 4. Нормируем гистограмму ${}_A\bar{\mathbf{h}}_{t,p}$ так, чтобы сумма ее элементов была равна 1.

В третьей версии алгоритма для поиска объекта интереса используется критерий Баттачариа

$$\gamma\left({}_R\bar{\mathbf{h}}', {}_G\bar{\mathbf{h}}', {}_B\bar{\mathbf{h}}', {}_R\bar{\mathbf{h}}'', {}_G\bar{\mathbf{h}}'', {}_B\bar{\mathbf{h}}''\right) = \sum_{\substack{n_{1,R}=0, n_{2,R}=0, n_{1,G}=0, \\ n_{2,G}=0, n_{1,B}=0, n_{2,B}=0}}^{l-1,l-1,l-1,l-1,l-1,l-1} \sqrt{{}_R\bar{h}'(n_{1,R}, n_{2,R})\, {}_R\bar{h}''(n_{1,R}, n_{2,R})} \times$$
$$\times \sqrt{{}_G\bar{h}'(n_{1,G}, n_{2,G})\, {}_G\bar{h}''(n_{1,G}, n_{2,G})} \times \sqrt{{}_B\bar{h}'(n_{1,B}, n_{2,B})\, {}_B\bar{h}''(n_{1,B}, n_{2,B})}. \quad (4)$$

В первой модификации третьей версии алгоритма по аналогии с первой модификацией первой версии осуществляется поиск по всему текущему кадру

$$\hat{p}_t = \underset{q \in S}{\operatorname{argmax}} \left\{\gamma\left({}_R\bar{\mathbf{h}}_{0,p_0}, {}_G\bar{\mathbf{h}}_{0,p_0}, {}_B\bar{\mathbf{h}}_{0,p_0}, {}_R\bar{\mathbf{h}}_{t,p}, {}_G\bar{\mathbf{h}}_{t,p}, {}_B\bar{\mathbf{h}}_{t,p}\right)\right\}.$$

Во второй модификации третьей версии алгоритма осуществляется локальный поиск на текущем кадре в окне $W\left(\hat{p}_{t-1}\right)$, центр которого полагается равным оценке координат объекта на предыдущем кадре $\hat{p}_{t-1}$:

$$\hat{p}_t = \underset{q \in W(\hat{p}_{t-1})}{\operatorname{argmax}} \left\{\gamma\left({}_R\bar{\mathbf{h}}_{0,p_0}, {}_G\bar{\mathbf{h}}_{0,p_0}, {}_B\bar{\mathbf{h}}_{0,p_0}, {}_R\bar{\mathbf{h}}_{t,p}, {}_G\bar{\mathbf{h}}_{t,p}, {}_B\bar{\mathbf{h}}_{t,p}\right)\right\}.$$

Третья версия алгоритма требует меньших объемов памяти и вычислений по сравнению со второй, но бóльших по сравнению с первой. В разд. 2 приведены результаты вычислительных экспериментов с описанными версиями алгоритма.

2. Особенности программной реализации алгоритмов

Как было отмечено выше, представленные версии алгоритма были предложены для повышения точности и надежности классической версии алгоритма среднего сдвига, в которой

вместо критерия Баттачариа использовалось его линейное приближение. Применение линеализации позволило Чену [1], а затем Команечи, Рамешу и Мейеру [9] разработать итерационный метод поиска приближенного максимума критерия Баттачариа, являющийся, по сути, разновидностью метода градиентного спуска. Разработанный метод обладает всеми преимуществами и недостатками градиентных методов. Он достаточно быстр, чтобы обеспечить отслеживание объектов на видеопоследовательностях в режиме реального времени с помощью бюджетного персонального компьютера, но в силу сходимости оценок к локальному экстремуму нередко приводит к потере сопровождаемого объекта.

Для повышения точности и надежности сопровождения объектов на основе критерия Баттачариа вместо линеализованного приближения были использованы его исходные представления (2)–(4). Это дало возможность находить глобальный максимум критерия вместо одного из локальных и осуществлять поиск объекта по всему кадру (или по всей разрешенной области), однако значительно увеличило объем вычислений. Поэтому для обеспечения выполнения вычислений в режиме реального времени пришлось разработать параллельные программные реализации алгоритма, в том числе и с применением технологии программирования видеокарты CUDA.

Оптимизированные процессорные программные реализации первой версии алгоритма позволили выполнять отслеживание объектов в режиме реального времени, в то время как для достижения режима реального времени для второй и третьей версий алгоритма нужно было реализовать их на CUDA.

Проведенные исследования показали, что все три версии оказались более точными и надежными по сравнению с классическим алгоритмом среднего сдвига, а вторая версия оказалась на 5–10 % точнее первой из представленных в статье версии.

3. Сравнение характеристик предложенных алгоритмов на основе экспериментов

Для исследования точности и надежности версий алгоритма были использованы видеопоследовательности, полученные видеокамерами с разрешением 720×480 пикселов с борта беспилотного летательного аппарата (БЛА) (рисунок, *а*), а также снятые нестабилизированной камерой в помещении (рисунок, *б*). Каждая из последовательностей содержала примерно 500 кадров, представляющих собой 24-битные *RGB*-изображения (в данном случае $L = 256$).

а) *б)*

Примеры кадров видеопоследовательностей:
а) полученных с борта БЛА; *б)* полученных в помещении

Сначала предложенные версии сравнивались с известным алгоритмом среднего сдвига [1]. Алгоритм среднего сдвига (основанный на локальном поиске) в отличие от предложенных версий нередко безвозвратно теряет сопровождаемый объект, что вызывает трудности с автоматическим тестированием, а на корректно обработанных кадрах дает примерно в 1,5 раза менее точный результат, поэтому ниже приведены результаты тестирования только предложенных модификаций.

В табл. 1 приведены результаты тестирования предложенных версий алгоритма на видеопоследовательности, снятой с борта БЛА. Тестирование проводилось на PC Intel® Core TM i5 CPU 750 2,67 GHz (реальная частота 3,6 GHz) с видеокартой NVIDIA GeForce GTX 650Ti. Для

выделения объекта был выбран размер окна 20×20 пикселов. В обоих случаях размер окна выбирался экспериментально таким образом, чтобы получить наилучшие точность и надежность результатов. Оказалось, что оптимальный размер окна совпадает с размером объекта. Экспериментально было установлено, что число градаций квантизованого цвета l = 8,…, 32 обеспечивает наилучшее соотношение точности и скорости вычислений.

Таблица 1

Версии алгоритма	Критерий	Окно объекта	Кол-во потерь	Среднеквадр. ошибка, пикс.	t_{CPU}, с/кадр	t_{GPU}, с/кадр
Версия 1. Поиск по всему кадру	α	20×20	7	4,88	0,502	0,081
Поиск в окне 100×100	α	20×20	3	4,92	0,015	0,007
Версия 2. Поиск по всему кадру	β	20×20	2	4,36	4,536	1,364
Поиск в окне 100×100	β	20×20	1	4,36	0,141	0,043
Версия 3. Поиск по всему кадру	γ	20×20	14	4,67	1,732	0,258
Поиск в окне 100×100	γ	20×20	6	4,71	0,072	0,017

Объект считался потерянным, если ошибка в определении его координат была больше, чем половина размера окна, т. е. больше 10 пикселов (хотя при этом окно могло частично «накрывать» объект).

Результаты тестирования алгоритма на второй видеопоследовательности приведены в табл. 2.

Таблица 2

Версии алгоритма	Критерий	Окно объекта	Кол-во потерь	Среднеквадр. ошибка, пикс.	t_{CPU}, с/кадр	t_{GPU}, с/кадр
Версия 1. Поиск по всему кадру	α	50×50	7	4,45	2,28	0,358
Поиск в окне 100×100	α	50×50	0	7,12	0,083	0,022
Версия 2. Поиск по всему кадру	β	50×50	12	4,27	48,766	11,601
Поиск в окне 100×100	β	50×50	1	6,44	1,898	0,828
Версия 3. Поиск по всему кадру	γ	50×50	29	7,89	9,399	1,389
Поиск в окне 100×100	γ	50×50	3	10,27	0,526	0,119

При тестировании использовался тот же критерий потери объекта: он считался потерянным, если ошибка в определении его координат была больше, чем половина размера окна, т. е. больше 25 пикселов. Первая и вторая версии алгоритма обеспечили устойчивое сопровождение объектов размером 10×10 пикселов.

Было проведено сравнение точности и надежности разработанных версий алгоритмов с алгоритмами отслеживания объектов корреляционного типа, основанных на сравнении корреляции и ковариации Пирсона [10, 11] на исходных цветных видеопоследовательностях, а также на их полутоновых копиях (были протестированы четыре версии корреляционных алгоритмов).

Корреляционные алгоритмы в отличие от предложенных модификаций алгоритма среднего сдвига оказались неустойчивыми при отслеживании объектов (см. рисунок, *а*): все они теряли объект примерно на половине просмотренных кадров. В данном случае непосредственное использование корреляционных алгоритмов без дополнительных средств стабилизации решения дало бы очень плохие результаты.

На рисунке, *б* видно, что использование корреляции Пирсона на самих изображениях и их полутоновых копиях сделало возможным получение чуть более точных результатов по сравнению с предложенными версиями алгоритма (в среднем на 1,8 пиксела точнее при размере сопровождаемого объекта 100 пикселов), в то время как ковариация дала менее точные и значительно менее устойчивые решения.

Таким образом, при тестировании предложенные модификации оказались более устойчивыми к качеству видеопоследовательностей по сравнению с корреляционными алгоритмами, которые считаются одними из самых надежных при решении задач отслеживания. Кроме того, первая и третья модификации предложенного алгоритма превосходят по быстродействию корреляционные алгоритмы в несколько раз, поэтому легче реализовать их программные версии для работы на CPU и GPU.

Полученные результаты подтверждают возможность применения рассмотренных алгоритмов для решения задач отслеживания объектов на видеопоследовательностях.

Заключение

Предложены три версии алгоритма отслеживания объектов, снятых нестабилизированными видеокамерами, которые основаны на сравнении локальных оконных гистограмм яркостей областей изображений. В качестве меры близости гистограмм в предложенных версиях в отличие от известного алгоритма среднего сдвига использован критерий Баттачариа в его исходном нелинейном виде. Вместе с глобальным поиском это позволяет находить вместо ближайшего локального экстремума критерия его гарантированный глобальный максимум, что ведет к повышению точности и надежности отслеживания.

Проведенные эксперименты показали бóльшую точность и надежность предложенных версий по сравнению не только с алгоритмом среднего сдвига, но и с алгоритмами, основанными на применении корреляции и ковариации Пирсона.

Программные реализации версий, выполненные с помощью технологии программирования видеокарты CUDA, оказались настолько быстрыми, что благодаря им можно выполнять отслеживание объектов в режиме реального времени.

Список литературы

1. Cheng, Y. Mean shift, mode seeking, and clustering / Y. Cheng // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 1998. – № 17(8). – P. 790–799.

2. Yilmaz, A. Object tracking: A survey / A. Yilmaz, O. Javed, M. Shah // ACM Computing Surveys. – 2006. – Vol. 38, № 4. – 45 p.

3. Marimon, D. Orientation histogram-based matching for region tracking / D. Marimon, T. Ebrahimi // Proc. 8th Intern. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS. – Santorini, 2007. – P. 8–12.

4. Lowe, D. Object recognition from local scale invariant features / D. Lowe // Proc. Intern. Conf. on Computer Vision ICCV. – Corfu, 1999. – P. 1150–1157.

5. Bay, H. Surf: Speeded up robust features / H. Bay, T. Tuytelaars, L. Van Gool // Proc. 9th Europ. Conf. on Computer Vision ECCV. – Graz, 2006. – P. 404–417.

6. Altmann, J. A Fast Correlation Method for Scale-and Translation-Invariant Pattern Recognition / J. Altmann, H.J. Reitböck // IEEE Trans. Pattern Anal. Mach. Intell. – 1984. – Vol. 6, № 1. – P. 46–57.

7. Object Tracking by Particle Filtering Techniques in Video Sequences / L. Mihaylova [et al.] // Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series. – Netherlands : IOS Press, 2007. – P. 260–268.

8. Haralick, R.M. Textural Features for Image Classification / R.M. Haralick, K. Shanmugam, I. Dinstein // IEEE Transactions on Systems, Man, and Cybernetics. – 1973. – № 6. – P. 610–621.

9. Comaniciu, D. Real-Time Tracking of Non-Rigid Objects using Mean Shift // D. Comaniciu, V. Ramesh, P. Meer // Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR. – Hilton Head Island, 2000. – Vol. 2. – P. 142–149.

10. Афифи, А. Статистический анализ / А. Афифи, С. Эйзен. – М. : Мир, 1982. – 488 с.

11. Zalesky, B.A. Real Time Object Tracking Algorithm / B.A. Zalesky, E.N. Seredin // Intern. Congress on Computer Science: Information Systems and Technologies, CSIST 2013. – Minsk, 2013. – P. 500–504.

Поступила 11.05.2015

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: zalesky@newman.bas-net.by
eduard.seredin@tut.by

B.A. Zalesky, E.N. Seredin, M.V. Yadlouski

OBJECT TRACKING VIA COMPARISON OF COLOR HISTOGRAMS

Three versions of a histogram algorithm for tracking objects on video sequences made by an unstable camera are presented. Local color 1D-histograms of pixels and local color 2D-histograms of pairs of adjacent pixels are used in all versions as region features. The histograms are compared by the Bhattacharia criterion. A parallel computing platform CUDA, developed to program GPUs, allows creation of real time or near-real time program realizations of the offered versions. Results of comparison of the versions with the known mean shift algorithm and correlation type algorithms are also presented. It is shown by experiments that the versions are more accurate and reliable than the mean shift algorithm, which estimates similarity of linear approximations of local histograms, and more robust with respect to the video quality than the correlation algorithms.

УДК 004.9

В.А. Ковалев

ВЛИЯНИЕ ИСКАЖЕНИЙ И ФРАГМЕНТАЦИИ ИЗОБРАЖЕНИЙ-ОБРАЗЦОВ НА КАЧЕСТВО ПОИСКА ЦВЕТНЫХ ИЗОБРАЖЕНИЙ ПО СОДЕРЖАНИЮ

Рассматривается проблема поиска по образцу цветных изображений в больших базах данных с использованием дескрипторов, построенных на матрицах совместной встречаемости цветов. Известно, что качество функционирования соответствующих систем поиска изображений зависит не только от используемых методов и алгоритмов, но и от широкого круга различных «шумовых» факторов. Приводятся результаты экспериментального исследования влияния на качество поиска таких факторов, как изменение размеров изображений и их пространственное искажение, а также использование фрагментов исходных изображений в качестве образца поиска.

Введение

Появившиеся в 1990-х гг. методы и программные средства поиска цветных изображений по содержанию стали достаточно эффективным ответом на вызовы наступающей «цифровой эры» в области съемки, накопления, хранения, поиска, обработки и анализа цифровых изображений и видеоданных [1, 2]. Известные трудности в формировании запроса на поиск в неиндексированных базах изображений без использования словесных описаний, ключевых слов и каких-либо количественных признаков привели к тому, что одной из популярных парадигм в технологии поиска стал так называемый поиск по образцу, когда запрос формулируется по принципу «найди изображения, наиболее похожие на это».

Целью данной работы является экспериментальное исследование влияния на качество поиска цветных изображений таких факторов, как изменение размеров изображений и пространственные искажения, а также использование фрагментов (частей) изображений в качестве изображений-образцов для поиска. Работа имеет практическую направленность и не претендует на полноту охвата всего спектра возможных причин и параметров, оказывающих влияние на результаты поиска цветных изображений в базах данных по содержанию. Однако предполагается, что полученные в результате проведенных вычислительных экспериментов зависимости и количественные оценки могут оказаться полезными для инженеров и программистов, разрабатывающих соответствующие прикладные программные комплексы.

1. Материалы и методы исследования

1.1. Тестовая база изображений

Технические характеристики. Тестовая база состояла из 10 000 цветных (RGB) изображений относительно небольших размеров, имеющих прямоугольную форму с горизонтальной или вертикальной ориентацией. Горизонтальные размеры варьировали в пределах от 222 до 512 пикселов, а вертикальные – в пределах от 164 до 382. Средний размер (площадь) изображений был эквивалентен изображению квадратной формы размером 350×350 пикселов.

Содержание изображений. Используемая база изображений создавалась сотрудниками Фраунхоферовского института компьютерной графики (Германия) с целью тестирования эффективности разрабатываемых методов, алгоритмов и программных средств поиска изображений по содержанию общего характера. Для обеспечения полноценного тестирования содержание изображений не ограничивалось какой-либо предметной областью или тематикой. Кроме того, тестовые изображения не имели никакой специфики относительно предпочтительной цветовой гаммы, наличия только реальных или только искусственных объектов, использования стандартизованных условий съемки, доминирования определенных текстурных свойств, объектов какой-либо определенной формы и т. п. Используя бытовую лексику, содержание тестовой базы может быть охарактеризовано как «изображения всего чего угодно».

1.2. Способ описания содержания изображений

Для описания содержания изображений использовались матрицы совместной встречаемости цветов. Основываясь на общей концепции матриц совместной встречаемости, предложенной в работах [3, 4], и следуя методике конструирования конкретных их вариантов, изложенной в [4, 5], в качестве дескрипторов изображений были выбраны трехмерные матрицы совместной встречаемости пар цветов W_C типа «цвет – цвет – расстояние», которые могут быть определены как

$$W_C = || \, w_c(c(i), c(k), d_{i,k}) \, ||,$$
$$w_c(c(i), c(k), d_{i,k}) = card\{ \, i, k \in R^2 \mid i \neq k, \, d_{i,k} = round(d(i, k)), \, d_{i,k} \leq D,$$
$$x_k = (x_i + \Delta x), \, y_k = (y_i + \Delta y), \, -D \leq \Delta x \leq D, \;\; 0 \leq \Delta y \leq D, \, \Delta y(D+1) + \Delta x > 0 \, \}.$$

В приведенном определении через (i, k) обозначена произвольная пара пикселов, идентифицируемая с помощью их индексов (номеров) i и k, которые расположены на расстоянии $d_{i,k}$ друг от друга. Параметры $c(i)$ и $c(k)$ соответствуют цветам этих пикселов, а w_c представляет собой количество (частоту встречаемости) пикселов с указанными цветами на расстоянии $d_{i,k}$. Весь диапазон RGB-цветов квантуется путем равномерного разбиения на заданное количество интервалов, а межпиксельное расстояние, округляемое до ближайшего целого значения, изменяется в пределах от 1 до D. Соответственно *card* обозначает количество, а *round* – операцию округления до ближайшего целого. Неравенства, представленные в последней строке определения, формализуют порядок перебора соседей k текущего пиксела i по принципу вперед и вниз от текущего. Дальнейшие детали об алгоритме перебора можно найти в работе [5].

В силу того что при традиционных восьми битах на каждый из RGB-каналов цветовое пространство имеет весьма большие размеры, на практике матрицы встречаемости цветов являются сильно разряженными, с большим количеством нулевых элементов. Это позволяет использовать достаточно грубые схемы квантизации с относительно небольшим количеством интервалов цветов. В данной работе использовалась схема с тремя битами на каждый из трех цветовых каналов, т. е. количество интервалов (квантов) цветов равнялось 512. Адаптивные схемы редуцирования цветового пространства не применялись, поскольку используемые изображения не имели доминирующих цветов. Таким образом, элементы матрицы w_c описывают «элементарные» цветовые сегменты, их границы и, опосредованно, пространственное расположение сегментов на изображении. На примере базы данных из 20 000 изображений было экспериментально показано [6], что такое представление является достаточно гибким и одинаково хорошо подходит для описания широкого класса сцен, начиная от простых комбинаций цветных объектов на однородном фоне и заканчивая высокочастотными цветными текстурами натурального происхождения.

Важно отметить, что при вычислении матриц совместной встречаемости рассматриваются все возможные пары в локальной окрестности каждого текущего пиксела. При этом пары с одними и теми же цветами, находящиеся на одних и тех же расстояниях, считаются идентичными независимо от порядка встречаемости цветов при переборе пикселов изображения. Фактически это означает, что для каждого из расстояний соответствующий слой результирующего трехмерного массива частот встречаемости представляет собой нижнюю треугольную матрицу. Завершающим шагом вычисления матриц является их нормализация, выполняемая путем деления всех элементов на их сумму. Перечисленные особенности алгоритма вычисления матриц приводят к тому, что дескрипторы совместной встречаемости обладают следующими важными свойствами:

а) инвариантность по отношению к таким преобразованиям изображений, как поворот на угол, кратный 90°, и зеркальное отражение (из-за перебора всех возможных пар пикселов);

б) сдвиг объектов изображений по однородному фону (из-за отсутствия координатной привязки);

в) относительная независимость от размера, т. е. возможность сравнивать содержание изображений разного размера и формы (из-за нормализации матриц);

г) возможность поиска по фрагментам изображений, что обусловлено такими факторами, как нормализация матриц и использование метрики L1, в которой вычисляется разность одноименных элементов матриц, т. е. относительных частот встречаемости некоторых цветов, представленных как на материнском изображении, так и на его фрагментах.

На заключительном этапе вычисления дескриптора каждого изображения набор ненулевых элементов соответствующей матрицы совместной встречаемости представляется в виде списка (<номер элемента матрицы>, <значение>) и записывается в файл дескрипторов в порядке возрастания номеров элементов, что обеспечивает их быстрый перебор при сравнении. Следует отметить, что на практике обычно игнорируются не только элементы матрицы, которые строго равны нулю, но и все «шумовые» элементы, значение которых меньше некоторого (достаточно малого) порога. Данный прием позволяет существенно сократить количество элементов, включаемых в дескриптор, что значительно уменьшает его размер и повышает скорость работы системы поиска изображений при условии обеспечения практически тех же самых результатов поиска. Так, например, при установке минимального значения порога, эквивалентного 0,2 % от площади изображения, среднее значение количества элементов матрицы, включаемых в дескриптор, равно 66 [6].

1.3. Сравнение дескрипторов изображений

Дескрипторы изображений, т. е. множества ненулевых элементов дескриптора изображения-образца поиска и каждого из дескрипторов базы данных, сравнивались с помощью метрики L1 [7], т. е. степень различия между изображениями R вычислялась как нормализованная сумма абсолютных значений разностей однотипных элементов сравниваемых дескрипторов. При отсутствии какого-либо элемента в дескрипторе его значение полагалось равным нулю. Соответственно степень близости изображений в пространстве признаков (степень сходства S) вычислялась как $S = 1 - R$. Следуя сложившейся традиции, для количественной оценки результатов поиска выводились и использовались только первые $N = 20$ изображений, наиболее близких к заданному образцу.

2. Поиск изображений при различных условиях

2.1. Типичные результаты поиска

На рис. 1 показаны примеры образцов поиска и $N = 3$ наиболее близких результатов, найденных в тестовой базе из 10 000 изображений. Последняя колонка иллюстрирует инвариантность результатов поиска по отношению к таким преобразованиям, как зеркальное отражение и поворот.

Рис. 1. Изображения-образцы (*а*) и $N = 3$ наиболее похожих результата поиска для каждого из них вместе с соответствующей количественной мерой сходства S (*б*)

Следует особо подчеркнуть, что при визуальной оценке результатов поиска похожих изображений в базах данных пользователи подобных сервисов интуитивно ожидают увидеть среди N наиболее близких изображений действительно «очень похожие» (в соответствии с их индивидуальными и, надо сказать, весьма различными представлениями о «похожести»). Вместе с тем очевидно, что в результате поиска в N наиболее близких будут включены только изображения из числа фактически присутствующих в используемой базе изображений. Соответственно в общем случае (и особенно в случае небольших баз изображений) совсем не обязательно, что они будут действительно «очень похожи» на заданный образец и будут отвечать ожиданиям пользователя.

2.2. Влияние пространственных искажений на результаты поиска

В качестве типичного примера пространственных искажений изображений в соответствующей серии экспериментов была использована операция их редуцирования к некоторому «стандартизованному» размеру: до небольших, в нашем случае 128×128 пикселов, миниатюр (preview), которые часто используются в интерфейсных целях в современных компьютеризированных сервисах (рис. 2). При этом информация об исходной портретной или пейзажной форме исходных изображений во внимание не принималась. Кроме приведенного варианта пространственных искажений в литературе описываются и такие важные случаи, как искажения изображений, связанные с различиями в освещении сцены [8], и некоторые другие. Однако, по мнению автора, в данной работе рассматривается гораздо более массовый и понятный способ пространственного искажения изображений из числа встречающихся на практике.

Рис. 2. Примеры исходных изображений и их искаженных (уменьшенных до 128×128 пикселов) версий

Схема вычислительных экспериментов по оценке влияния пространственных искажений на результаты поиска представлена на рис. 3, а их результаты – на рис. 4. Процесс тестирования включал серию из трех независимых экспериментов, описанных ниже.

1. *Раздельный поиск*. Каждое изображение исходной базы данных подавалось на вход программы в качестве образца поиска, и идентификаторы получаемых $N = 20$ наиболее близких результатов поиска запоминались в качестве «правильных». Та же самая процедура независимо выполнялась над искаженными версиями изображений, и результаты поиска по «искаженной» базе данных сравнивались с результатами, полученными при поиске исходных изображений. Количественная оценка воспроизводимости результатов среди ближайших 4, 8, 12, 16 и 20 наиболее похожих ответов показана на рис. 4 с помощью кривой синего цвета.

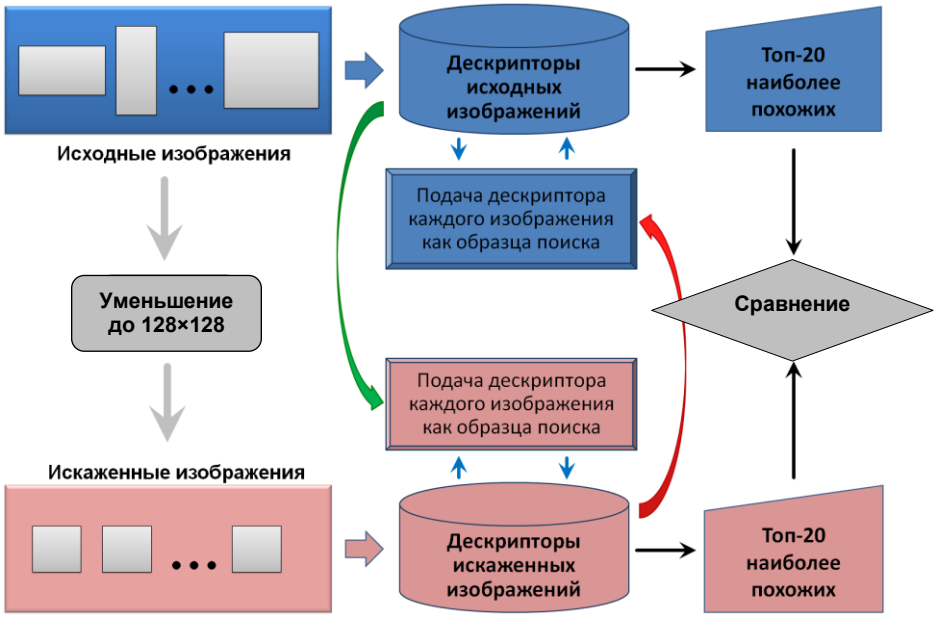


Рис. 3. Общая схема экспериментов по оценке влияния пространственных искажений на качество поиска

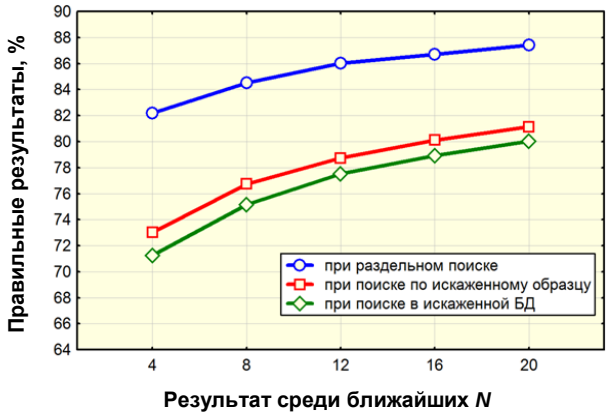


Рис. 4. Графики правильных результатов поиска (в процентах) среди N = 4, 8, 12, 16 и 20 изображений, ближайших к заданному изображению-образцу

2. *Поиск по искаженному образцу*. В качестве образца поиска использовался дескриптор искаженного изображения, в то время как поиск наиболее похожих осуществлялся среди исходных, неискаженных изображений (см. красную стрелку на рис. 3). Результаты эксперимента показаны на рис. 4 кривой красного цвета.

3. *Поиск в искаженной базе*. Данный эксперимент являлся противоположным эксперименту 2, т. е. в качестве образца поиска использовался дескриптор исходного изображения, а поиск осуществлялся среди искаженных изображений (см. стрелку зеленого цвета на рис. 3 и зеленую кривую на рис. 4, представляющую точность получаемых результатов).

Проведенные эксперименты показали, что поиск по искаженному образцу среди искаженных же изображений обеспечивает более высокую воспроизводимость результатов (в среднем на 7–9 % лучше) по сравнению с гибридными схемами, использующими исходные и искаженные изображения одновременно. В частности, для 10 000 запросов на поиск по образцу среднее количество корректных результатов среди N = 20 наиболее похожих изображений достигает 87,4 %.

2.3. Поиск по фрагментам изображений

На практике довольно часто встречаются ситуации, когда используется не полное исходное цифровое изображение, а тот или иной его фрагмент. Более того, в сети Интернет нередко

можно наблюдать случаи, когда некоторая удачная иллюстрация или ее часть заимствуются различными авторами без ссылки и без разрешения ее автора (правообладателя). Поэтому возможность автоматического поиска изображений по их фрагментам представляется весьма полезной вне зависимости от того, идет ли речь об обнаружении некорректных заимствований или об обычном использовании свободных мультимедийных ресурсов.

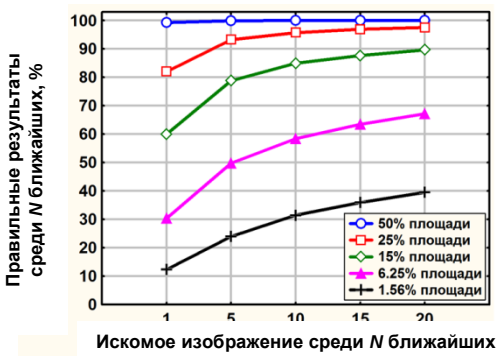
Следует подчеркнуть, что в отличие от известных работ по проблеме поиска изображений по фрагментам (см., например, [9, 10]), которые посвящены повышению качества поиска за счет использования некоторых специфических фрагментов изображений, в данной работе проблема выбора каких бы то ни было «характерных» областей (участков, фрагментов) не рассматривается. Вместо этого фрагменты различного размера выбираются вслепую, случайным образом, без оценки их информативности и важности относительно других.

Таким образом, данный подраздел посвящен экспериментальной оценке эффективности решения задачи поиска исходных изображений по образцу на основе дескрипторов совместной встречаемости в условиях, когда в качестве образца поиска выступает некоторый фрагмент исходного изображения. Естественно, что при этом дается только базовая количественная оценка потенциальной возможности решения подобных задач без рассмотрения всей технологии поиска дубликатов в архивах изображений или в сети Интернет, а также других родственных задач.

В рассматриваемом случае экспериментальное исследование потребовало довольно значительных вычислительных затрат и состояло из следующих основных этапов:

1. *Подготовка фрагментов изображений.* Размеры фрагментов изображений варьировались в широком диапазоне от 50 до 1,56 % от площади исходного изображения, включая следующие пять значений: 50; 25; 15; 6,25 и 1,56 %. Для каждого размера и каждого изображения базы данных в позиции, выбираемой на изображении случайным образом, вырезались 10 фрагментов, т. е. брались всего 50 фрагментов с каждого изображения или в общей сложности 500 000 фрагментов из 10 000 тестовых изображений используемой базы данных. Типичные примеры фрагментов показаны на рис. 5, *а*.

а)



б)

в)

Рис. 5. Фрагменты изображений различных размеров (*а*); количество случаев (в процентах), когда искомое изображение было найдено среди N ближайших (*б*) либо среди всех $N = 20$ (*в*)

2. *Вычисление дескрипторов изображений.* Как уже упоминалось выше, все дескрипторы (трехмерные гистограммы частоты совместной встречаемости) нормализовывались, чтобы избежать зависимости от размеров изображений.

3. *Собственно проведение поиска по фрагментам.* Данный этап выполнялся в виде пяти серий отдельных экспериментов, по одной серии для каждого размера фрагментов. Каждый из 100 000 фрагментов подавался в качестве образца поиска и ближайшие $N = 20$ результатов запоминались. Таким образом, в каждой из пяти серий задача сравнения дескриптора фрагмента-образца с дескрипторами 10 000 исходных изображений решалась 100 000 раз путем выполнения 10^9 сравнений.

4. *Анализ результатов.* Исходное изображение считалось найденным по заданному фрагменту, если оно появлялось на каком-либо месте среди $N = 20$ изображений, наиболее близких к заданному образцу. Графики, представленные на рис. 5, *б*, иллюстрируют процент случаев, когда искомое исходное изображение находилось на первом месте, а также среди ближайших 5, 10, 15 или 20 изображений, являющихся результатом поиска. Кроме того, для удобства процент корректных результатов поиска среди $N = 20$ ближайших изображений как функция размера фрагмента-образца показан отдельно на рис. 5, *в*. Следует отметить, что в представленных данных вероятность случайного появления искомого изображения среди наиболее близких результатов поиска не учитывалась в силу ее малости.

Как и следовало ожидать, доля корректных результатов поиска исходного изображения зависит от размера фрагмента, используемого в качестве образца. Так, например, при размере фрагмента, составляющем 50 % площади материнского изображения, оно находится в 100 % случаев, причем в подавляющем большинстве из них материнское изображение оказывается на первом месте (см. графики синего цвета на рис. 5). С другой стороны, что довольно неожиданно, использование в качестве образца фрагментов весьма малого размера, составляющих лишь 1,56 % от площади исходного изображения, тем не менее позволяет найти исходное изображение среди $N = 20$ результатов в 39,5 % случаев. При этом в 12 % случаев оно оказывается на первом месте (см. кривую черного цвета на рис. 5).

Заключение

Экспериментальные результаты, полученные на тестовой базе из 10 000 цветных изображений общего характера, позволяют сделать следующие выводы.

Пространственное искажение цветных изображений путем приведения их к миниатюрам стандартизованного размера в 128×128 пикселов приводит к тому, что результирующий набор из $N = 20$ изображений, ближайших к заданному образцу поиска, может содержать порядка 13–20 % ошибок. В частности, наихудший результат с 20 % ошибок (соответственно точность 80 %) получается при «смешанной схеме», т. е. при поиске по неискаженному образцу в искаженной базе изображений. Противоположная смешанная схема, т. е. поиск по искаженному образцу в исходной (неискаженной) базе изображений, занимает среднее, второе место с показателем качества 18,9 % ошибок (точность 81,1 %). Наилучшие результаты (12,6 % ошибок и, соответственно, точность 87,4 %) получаются при «однородной» схеме, т. е. при поиске по искаженному изображению-образцу в искаженной базе изображений.

При поиске по фрагменту изображения его оригинала результаты в существенной степени зависят от размера используемого фрагмента. При размере фрагмента, составляющем 50 % от площади исходного изображения, материнское изображение находится среди $N = 20$ наиболее похожих всегда (100 %), причем в подавляющем большинстве случаев (99,3 %) оно оказывается на первом месте. При уменьшении фрагмента-образца поиска точность поиска падает. Однако даже при использовании фрагментов, составляющих менее 2 % от площади исходного изображения, рассматриваемый метод обеспечивает автоматический поиск исходного изображения почти в 40 % случаев.

Несмотря на то что рассматриваемые дескрипторы в принципе могут использоваться для описания структуры и поиска в базах данных полутоновых изображений, качество их поиска будет сравнительно невысоким. Причина этого заключается в том, что для покрытия всего цветового пространства рассматриваемые матрицы совместной встречаемости используют достаточно

грубую схему его разбивки на 8*8*8 = 512 цветов. Учитывая, что гамме серого цвета в пространстве RGB соответствуют цвета с равными цветовыми компонентами $r = g = b$, максимальное количество различимых уровней серого будет равно восьми. Этого может быть недостаточно для целого ряда задач хранения и поиска полутоновых изображений по содержанию. Более подробные сведения о дескрипторах полутоновых изображений можно найти в [4].

Автор выражает глубокую благодарность доктору Штефану Волмеру и профессору Георгиусу Сакасу за предоставленную базу изображений и плодотворные дискуссии по проблеме поиска изображений по содержанию.

Список литературы

1. Content-based image retrieval at the end of the early years / A.W.M. Smeulders [et al.] // IEEE Transactions on PAMI. – 2000. – Vol. 22(2). – P. 1349–1380.

2. A survey of content-based image retrieval with high-level semantics / Y. Liu [et al.] // Pattern Recognition. – 2007. – Vol. 40(1). – P. 262–282.

3. Kovalev, V.A. Feature extraction and visualization methods based on image class comparison / V.A. Kovalev // Medical Imaging 1994 Intern. Symp., Image Processing. – 1994. – Vol. 2167. – P. 691–701.

4. Kovalev, V.A. Multidimensional co-occurrence matrices for object recognition and matching / V.A. Kovalev, M. Petrou // Graphical Models and Image Processing. – 1996. – Vol. 58, № 3. – P. 187–197.

5. Ковалев, В.А. Анализ текстуры трехмерных медицинских изображений / В.А. Ковалев. – Минск : Белорус. наука, 2008. − 264 с.

6. Kovalev, V.A. Color co-occurrence descriptors for querying-by-example / V.A. Kovalev, S. Volmer // Intern. Conf. on Multimedia Modelling. – Lausanne, 1998. – P. 32–38.

7. Ковалев, В.А. Влияние мер близости в пространстве признаков на качество поиска медицинских изображений по содержанию / В.А. Ковалев, А.А. Дмитрук // Информатика. – 2011. – № 2(30). – С. 5–11.

8. Liao, C.-J. Complementary retrieval for distorted images / C.-J. Liao, S.-Y. Chen // Pattern Recognition. – 2002. – Vol. 35. – P. 1705–1722.

9. A scheme of fragment-based faceted image search / T. Komamizu [et al.] // Database and Expert Sytems Applications. – 2012. – Vol. 7447. – P. 450–457.

10. Li, J. IRM: Integrated region matching for image retrieval / J. Li, J.Z. Wang, G. Wiederhold // Proc. of the 2000 ACM Multimedia Conf. – Los Angeles, 2000. – P. 83–86.

Поступила 08.04.2015

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: vassili.kovalev@gmail.com

V.A. Kovalev

THE EFFECT OF DISTORTIONS AND FRAGMENTATION ON RESULTS OF CONTENT-BASED RETRIEVAL OF COLOR IMAGES

This paper is dealing with the problem of content-based image retrieval using color co-occurrence matrices. It is well-known that there are a number of factors influencing the results of image retrieval. The specific contribution of this paper lies on quantitative assessment of the effect of spatial image distortions as well as on estimating the accuracy of retrieving original images using its fragments of different size as query examples. The presented study is capitalized on database containing 10 000 color RGB images of very different content. The necessary experimental results are provided.

УДК 004.432.45

Е.Г. Лутцев

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ, ПРИБЛИЖЕННЫХ К ЕСТЕСТВЕННОМУ: ОБЗОР ЛИТЕРАТУРЫ

Рассматривается ряд научных статей, посвященных вопросам программирования на языках, приближенных к естественному. Дается описание классических подходов к созданию естественных языков программирования и новых подходов, которые сделали разработку таких языков практичнее. Приводится сравнение одного из языков высокого уровня – CLIPS с естественными языками программирования. Анализируется монография на русском языке, посвященная естественно-языковым интерфейсам.

Введение

Естественный язык (ЕЯ) – вербальный язык общения людей между собой. Попытки программирования на ЕЯ делались и несколько десятилетий назад, но современные исследователи сходятся на том, что только сейчас появились необходимые предпосылки для успешного развития этого направления.

В научной среде известен феномен, когда именитый ученый, высказываясь негативно о каком-то направлении исследований, фактически уничтожает его, прерывая поток публикаций на несколько десятилетий. Подобный эффект произвела статья Э.В. Дейкстра «О глупости программирования на естественном языке» [1]. В ней, как кажется автору, необоснованно проведена полная аналогия между программированием и математикой, хотя программирование многогранно и в значительной степени пересекается с психологией. Ранние подходы к созданию языка, напоминающего естественный, страдали и от противоположного недостатка – полного игнорирования математических абстракций, удобной нотации и сокращений. В результате были созданы переусложненные языки, например такие, как КОБОЛ, где сложение двух чисел выражается инструкцией ADD A B GIVING ANS. Современные программисты часто не используют знания математики, выходящие за пределы начальной школы.

В Интернете встречается значительное количество ресурсов со злоупотреблением термином «программирование на естественном языке», поэтому при выборе материала для чтения следует проявлять осмотрительность.

В настоящей статье рассматривается письменная речь, т. е. жесты, эмоциональный тон и прочие невербальные элементы не учитываются.

1. Языки программирования, использующие классические технологии обработки ЕЯ

В связи с новизной рассматриваемой темы ей пока не посвящена ни одна монография, существует только небольшое количество статей. Программы, описанные во всех рассмотренных источниках, осуществляют ввод-вывод в текстовом виде.

В настоящее время развитие систем программирования на ЕЯ происходит преимущественно по двум направлениям: создание универсальных систем, но с ограниченным словарем и онтологической моделью и создание узких языков для конкретных проблемных областей. Пример первого направления – язык программирования Inform 7 [2]. Важной его характеристикой является сильный уклон в сторону декларативного, основанного на правилах стиля программирования и способности выводить типы и свойства объектов на основе того, как они используются. Например, высказывание «Джон носит шляпу» создает «лицо» по имени Джон (так как только люди способны носить вещи), создает «вещь» со свойством «носимая» (поскольку только объекты с пометкой «носимые» можно носить) и задает Джону свойство «носит шляпу».

Еще одним существенным аспектом языка Inform 7 выступает прямая поддержка отношений, которые позволяют отслеживать связи между объектами. Это заданные системой отно-

шения, например, когда один объект содержит другой, «носимый», объект, но разработчик может добавить и свои собственные отношения (например, любви или ненависти между людьми) либо отслеживать, какие персонажи встречались друг с другом.

На переднем крае исследований находятся разработки MIT Media Lab, в частности система Metafor [3]. Metafor может перевести фразу «Pacman is a yellow character who eats dots» в заготовку исполняемого кода, в котором есть объект Pacman со свойством color=yellow класса character и метод класса eat (dot). С помощью Metafor можно создать игру, способную «понять» следующий текст:

There is a brown room with lights in every corner, and a treasure chest in the center. If someone opens the treasure chest, the lights will all turn off.

На сайте [4] предлагается подход, заключающийся в использовании только «правильного» подмножества ЕЯ, которое называется sEnglish.

Следующая теорема – это код на sEnglish, который компилируется в исполняемый код на MATLAB:

Pythagoras' Theorem

Let T be a 'triangle'. Let T have 'side lengths' denoted by a, b, c. Let T have 'angles' denoted by alpha, beta, gamma. If gamma of T is equal to ~90°, then $c^2 = a^2 + b^2$ within measurement tolerances.

Определение ЕЯ в данной статье в зависимости от целей разработчиков языков, приближенных к ЕЯ, будет либо слишком узким, либо слишком широким. Приведенную теорему Пифагора поймет почти любой, владеющий английским, и признает, что это язык общения обычных людей между собой, а не специалистов-математиков. Если исключим из нее сокращения и спецсимволы, то придем к пресловутому КОБОЛу, а если будем использовать их по максимуму, то превратим текст в слабочитаемый и текстом на ЕЯ он уже считаться не сможет.

Иногда программа на sEnglish становится излишне многословной:

Attributes of a phase plot.

A 'phase plot' has the following properties: its 'highest frequency' that is a number array, its 'lowest frequency' that is a number array and its 'plot object handle' that is a number array.

А иногда является образцом того, как должны выглядеть программы на ЕЯ:

If U_ is 'smc01-control', then do the following. Define surface weights Alpha as "[0.5, 0.5]". Initialise matrix Phi as a 'unit matrix'. Define J as the 'inertia matrix' of Spc01. Compute matrix J2 as the inverse of J. Compute position velocity error Ve and angular velocity error Oe from dynamical state X, guidance reference Xnow. Define the joint sliding surface G2 from the position velocity error Ve and angular velocity error Oe using the surface weights Alpha. Compute the smoothed sign function SG2 from the joint sliding surface G2 with sign threshold 0.01. Compute special dynamical force F from dynamical state X and surface weights Alpha. Execute "Diffdot=zeros(6,1);". Compute control torque T and control force U from matrix J2, surface weights Alpha, special dynamical force F, smoothed sign function SG2 and Diffdot. Finish conditional actions.

Данный отрывок, безусловно, можно считать текстом на ЕЯ, только естественен он для ограниченного круга людей – физиков и инженеров. Если бы их попросили максимально понятно описать последовательность действий для своего коллеги, они использовали бы именно этот язык. Таким образом, ЕЯ включает в себя множество различных языков – не только языков разных народов, но и языков различных специальностей.

sEnglish обладает той важной особенностью, что в него можно включать отрывки кода на системе более низкого уровня – MATLAB. Это позволяет иногда сократить запись.

В статье [5] рассматривается возможность перевода описаний на ЕЯ в регулярные выражения.

Регулярные выражения (регэкспы) зарекомендовали себя как чрезвычайно мощный и универсальный формализм, который проник во все типы программ: от электронных таблиц до баз данных. Однако даже многие программисты не до конца понимают, как работают регэкспы. Таким образом, способность автоматически генерировать регулярные выражения из естественного языка была бы полезна во многих ситуациях.

Когда Н. Кушман представил доклад в соавторстве с Р. Барзилэй, он попросил группу информатиков записать регулярное выражение, соответствующее достаточно простому поиску в тексте. Когда он показал ответ и попросил поднять руки тех, кто ответил правильно, подня-

лось только несколько рук. Таким образом, система может быть полезна для зрелых программистов, но она также может позволить обычным пользователям, скажем, электронных таблиц и текстовых процессоров задавать сложный поиск с использованием ЕЯ.

Парсеры осуществляют синтаксический анализ текста. На вход парсера подается текст, на выходе строится дерево разбора. В статье [6] представлен метод для автоматической генерации парсеров из спецификаций форматов входных файлов на английском языке. Парсеры входных данных предназначены для выяснения, какие части файла содержат определенные типы данных. Без такого парсера файл – это просто случайный набор нулей и единиц. Для отображения релевантных явлений естественного языка и перевода спецификации на английском языке в дерево спецификации, которое затем транслируется в парсер на C++, используется порождающая модель Байеса.

Исследователи Массачусетского технологического института проверили генератор парсеров более чем на ста примерах, отобранных из текстов задач олимпиад по программированию ACM, которые включали спецификации файлов для каждой задачи. Генератор был в состоянии производить рабочие парсеры примерно для 80 % спецификаций, и в остальных случаях изменение слова или двух в спецификации обычно порождало работающий парсер.

Значительным достижением представляется работа с ЕЯ в системе Wolfram Alpha, описанная на сайте [7]. К примеру, данная система способна интерпретировать следующие предложения: draw a translucent tiffany blue sphere and a red cone, 20 random integers between –5 and 5, positions of negative numbers in t. К сожалению, она является платной и с закрытым исходным кодом.

В статье [8] рассматривается попытка программирования пространственных алгоритмов на ЕЯ. Входными данными для предлагаемой системы служит естественно-языковое описание алгоритма обработки пространственных данных, а выходными – объектно-ориентированный программный код, который должен быть скомпилирован и выполнен. Предложены и оценены два подхода. Первый основан на текстовом сопоставлении с образцом: для каждого предложения выбирается наилучшим образом подходящий шаблон, а объекты и методы создаются в соответствии с этим шаблоном. Второй подход преобразует текст в логическую форму, подвергающуюся ряду преобразований для получения результирующего кода. Для выполнения этих преобразований на каждом шаге используется ряд эвристических правил. Затем требуется дополнительный проход для обработки ссылок, чтобы найти одинаковые объекты, методы и переменные среди инструкций кода. Результаты предварительного исследования показывают, что с помощью системы программирования на ЕЯ в интерактивном режиме, где пользователь вручную редактирует сгенерированный код, можно значительно повысить производительность кодирования. Тем не менее в настоящее время точность полностью автоматизированного, т. е. неинтерактивного, режима генерации кода по-прежнему слишком низка, чтобы быть полезной.

2. Новые подходы к разработке естественных языков программирования

В статье [9] рассматривается применение теории онтологий, концептуальных графов, а также теории языков программирования к разработке теоретических основ программирования на ЕЯ, которое было в последние годы использовано для создания документов на ЕЯ для интеллектуальных агентов и читателей-людей. Проведенный анализ показывает три преимущества программирования на ЕЯ. Во-первых, это концептуализированное программирование, которое позволяет разработчикам писать программы с меньшим количеством ошибок благодаря ясности представления кода и вынужденному структурированию данных. Во-вторых, программирование на ЕЯ может помочь программированию всех важных абстракций для роботов: события, действия и модели мира могут быть созданы с помощью предложений. В-третьих, программирование на ЕЯ может использоваться для публикации исследователями документов на ЕЯ, т. е. документов по теории и процедурам управления роботами. Данная теоретическая работа также определяет большой класс интеллектуальных агентов, которые могут читать такие документы. Все это позволяет пользователям-людям и агентам иметь общее понимание того, как работают прикладные системы.

В статье [10] рассматривается модель для формализованного представления знаний «концептуальные графы». Человеческие знания можно смоделировать с помощью наборов концепту-

альных графов, которые могут иметь модальности прошлого, настоящего и будущего, или косвенной речи. Концептуальные графы могут быть и абстракцией более сложной концептуальной модели, описанной графами. Некоторые концептуальные графы основаны на восприятии окружающего физического мира и являются прямыми абстракциями процессов восприятия.

ЕЯ обладает богатой семантикой, но, к сожалению, его неформальные выразительные возможности часто ошибочно принимаются за простую неточность. Поскольку полные парсеры английского языка еще недоступны, люди не представляют возможным использовать английский язык непосредственно как средство программирования компьютеров. Тем не менее в статье [11] показывается, что описания процедур на английском языке часто содержат семантику программирования – языковые особенности, которые могут быть легко отображены в конструкции языка программирования. Некоторые лингвистические особенности могут даже вдохновить на новые способы мышления о задании программы. Далекие от того, чтобы быть безнадежно неоднозначными, ЕЯ основываются на важных принципах коммуникации, которые могут быть использованы, чтобы сделать взаимодействие человека с компьютером более естественным.

Для демонстрации возможности программирования на ЕЯ в статье [12] анализируются случаи из числа самых трудных: шаги и циклы. Рассматривается корпус описаний на английском языке, используемых в качестве заданий по программированию, и разрабатываются некоторые методы для отображения лингвистических конструкций на программные структуры, которые являются программной семантикой.

В статье [13] высказывается мнение, что улучшенные технологии обработки языка, диалог со смешанной инициативой и программирование на примерах могут сделать возможным программирование на ЕЯ сейчас, хотя это не было возможным в прошлом.

Улучшенные технологии обработки языка. В то время как полное понимание ЕЯ попрежнему остается вне досягаемости, есть шанс, что недавние улучшения в надежном грамматическом разборе с широкой областью применения, семантически-управляемом синтаксическом разборе, поверхностном парсинге (chunking) и успешном развертывании управляющих систем на ЕЯ могут обеспечить частичное понимание, достаточное для начала разработки практической системы.

Диалог со смешанной инициативой. Принцип «пользователь будет просто читать код вслух» заменяется на «пользователь и система должны разговаривать о программе». Система должна пробовать интерпретировать то, что пользователь говорит о программе, а затем спросить пользователя о том, что она не понимает, предоставить недостающую информацию и устранить недоразумения.

Программирование на примерах. Принимается методология «покажи и расскажи», которая объединяет описания на естественном языке с конкретными демонстрациями на примере. Иногда легче продемонстрировать то, что вы хотите, чем описать словесно. Пользователь может сказать системе: «Вот то, что я хочу». Система, в свою очередь, может проверить свое понимание: «Это то, что вы имеете в виду?» Это сделает систему более отказоустойчивой в тех случаях, когда язык не может быть непосредственно понят, а в случае полной неудачи более сложных методов пользователь может ввести код на низкоуровневом языке программирования.

MOOIDE [14] является интерфейсом, позволяющим начинающим пользователям запрограммировать среду МОО, используя ЕЯ. Программирование МОО включает разнообразные задачи, такие как создание объектов и их состояний, назначение глаголов-действий объектам и программирование поведения, которое изменяет состояния объектов и генерирует сообщения. После того как МОО запрограммирован, другие пользователи могут взаимодействовать с объектами в развлекательных или образовательных целях.

Для того чтобы сделать программирование МОО проще и доступнее для начинающих программистов, интерфейс на ЕЯ позволяет пользователям описывать различные задачи программирования МОО на английском языке, а именно добавление объектов, свойств объектов, состояний и отношений между объектами, а также глаголы, с помощью которых осуществляется доступ к поведению объектов в МОО. Английский язык может использоваться для описания инструкций принятия решений, циклов, условий и других типичных конструкций программирования.

Более ранние системы были сосредоточены на решении проблем синтаксического анализа в программировании, однако этим системам не хватало общеизвестных знаний. MOOIDE

привносит в программирование на ЕЯ оперирование на основе здравого смысла в дополнение к синтаксическому разбору. Рассуждения на основе здравого смысла позволяют MOOIDE автоматически поддерживать типичные свойства объектов, а также допустимые эффекты глаголов. Подобное дополнение в программирование на ЕЯ с помощью рассуждений поможет сделать его значительно более интуитивным для начинающих программистов.

В статье [15], где описывается Ubiquity – эксперимент Mozilla Labs по разработке естественно-языкового интерфейса для браузера, высказывается важная мысль: *«Если синтаксис языка является слишком ограничительным или, что еще хуже, конфликтует с природной интуицией пользователя о его языке, язык моментально перестает быть естественным независимо от того, насколько похожи ключевые слова или грамматика на таковые в естественном языке».*

3. Язык CLIPS

Рассмотрим язык CLIPS, который является декларативным, т. е. должен облегчать программирование некоторых задач по сравнению с императивными языками. Основными элементами CLIPS являются факты, структуры, правила и функции.

Пример определения структуры:

```
(deftemplate personal-data
  (slot name)
  (slot age)
  (slot weight)
  (multislot date_of_birth)
)
```

где slot – это поле, принимающее одно значение, а multislot – поле, принимающее список значений.

Определение фактов:

```
(deffacts trouble_shooting
  (car_problem (name ignition_key) (status on))
  (car_problem (name engine) (status wont_start))
  (car_problem (name headlights) (status work))
)
```

В этих фактах используется структура car_problem с полями name и status. Ясно, что синтаксис совершенно не похож на синтаксис ЕЯ. Факты можно добавлять, отображать и удалять.

Правило для псевдокода

```
IF the animal is a duck
THEN the sound made is quack
```

выглядит следующим образом:

```
(defrule duck
  (animal-is duck)
=>
  (assert (sound-is quack)))
```

И опять код достаточно далек от псевдокода.

Еще пример правила:

```
(defrule not-yellow-red
   (light ?color&~red&~yellow)
=>
   (printout t "Go, since light is "
            ?color crlf))
```

Пример функции:

```
(deffunction hypotenuse (?a ?b)
  (sqrt (+ (* ?a ?a) (* ?b ?b))))
```

CLIPS использует синтаксис Лиспа, поэтому подобные функции сложно воспринимать даже программистам, использующим другие языки. Учитывая, что семантика CLIPS так же бедна, как и синтаксис, можно сделать вывод, что он не менее далек от программирования на ЕЯ, чем любой другой современный язык программирования высокого уровня.

4. Русскоязычная литература

На русском языке также имеется литература, близкая по тематике к программированию на ЕЯ. В частности, следует отметить монографию О.Е. Елисеевой «Естественно-языковой интерфейс интеллектуальных систем» [16]. Она состоит из трех частей:

1. Обобщенное рассмотрение языка и естественно-языкового интерфейса.
2. Конкретные синтаксические и семантические особенности русского языка, а также свойства, общие для всех языков.
3. Рассмотрение реализации конкретной системы.

Первая часть не представляет особого интереса для специалиста, неплохо владеющего базовыми знаниями в области обработки ЕЯ и желающего лишь ознакомиться с новыми исследованиями по программированию на ЕЯ. В ней изложены общие вопросы лингвистики и психологии, а также темы, которые находятся на слишком высоком уровне абстракции по сравнению с уровнем компьютерных языков программирования. К примеру, рассматривается «поэтическая, или эстетическая, функция языка».

Во второй части теория становится ближе к практике. Данная часть включает в себя следующие темы:

– уровни изучения текста от отдельных морфем до полной картины сведений о мире;
– алгоритмы анализа и синтеза естественно-языковых текстов от морфологического анализа до построения базы знаний;
– морфология, т. е. классификация слов по частям речи и деление слов на морфемы;
– синтаксическая структура предложения;
– семантический язык и его связь с синтаксическим;
– семантический анализ.

Все темы изложены настолько подробно, что вполне могут послужить основой для разработки алгоритма на псевдокоде. Достаточное внимание ко всем рассмотренным вопросам позволит создать полноценный язык программирования.

В третьей части кратко представлены компоненты конкретных систем общения с пользователем на ЕЯ, а также принципы построения интерфейса интеллектуальных систем.

Заключение

Программирование на ЕЯ находится на том этапе, когда написаны первые прототипы систем программирования и открываются перспективные пути исследований. Полученные положительные результаты свидетельствуют о том, что вскоре должен произойти переход к следующему этапу – разработке систем промышленного программирования на языке, приближенном к естественному. На первом этапе главным был вопрос «Насколько реализуемы данные системы?», на втором таким вопросом будет «Насколько они практичны?». Разработка языка – не только научная, но и инженерная, и психологическая задача. Написание программ на языке должно быть удобным для программиста и давать положительный экономический эффект.

Список литературы

1. Dijkstra, E.W. On the foolishness of «natural language programming» / E.W. Dijkstra [Electronic resource]. – 1978. – Mode of access : http:// http://cs.utexas.edu/users/EWD/transcriptians/EWD06XX/EWD667. – Date of access : 02.02.2015.

2. Nelson, G. Natural language, semantic analysis and interactive fiction / G. Nelson // Inform7 web site [Electronic resource]. – 2005. – Mode of access : http://inform7.com/learn/documents/WhitePaper.pdf. – Date of access : 15.02.2015.

3. Liu, H. Metafor : Visualizing Stories as Code / H. Liu, H. Lieberman // Proc. of the 10th Intern. Conf. on Intelligent User Interfaces. – NY, 2005. – P. 305–307.

4. Veres, S.M. Sysbrain: natural language programming / S.M. Veres, A. Tsourdos // Sysbrain [Electronic resource]. – 2008. – Mode of access : http://sysbrain.org. – Date of access : 08.02.2015.

5. Kushman, N. Using Semantic Unification to Generate Regular Expressions from Natural Language / N. Kushman, R. Barzilay // Human Language Technologies : Conference of the North American Chapter of the Association of Computational Linguistics, Proc. – Cambridge, 2013. – P. 826–836.

6. From Natural Language Specifications to Program Input Parsers / T. Lei [et al.] // The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013). – Sofia, 2013. – P. 1294–1303.

7. Programming with natural language is actually going to work [Electronic resource]. – 2010. – Mode of access : http://blog.wolfram.com/2010/11/16/programming-with-natural-language-is-actually-going-to-work. – Date of access : 03.02.2015.

8. Galitsky, B. Programming Spatial Algorithms in Natural Language / B. Galitsky, D. Usikov // AAAI Workshop Technical Report WS-08-11. – Palo Alto, 2008. – P. 16–24.

9. Veres, S.M. Theoretical foundations of natural language programming and publishing for intelligent agents and robots / S.M. Veres // TAROS 2010 Proc. – Southampton, 2010. – P. 292–299.

10. Veres, S.M. Documents for intelligent agents in English / S.M. Veres, L. Molnar // Artificial Intelligence and Applications 2010 Conf. Proc. – Innsbruck, 2010. – P. 10.

11. Liu, H. Toward a Programmatic Semantics of Natural Language / H. Liu, H. Lieberman // IEEE Symposium on Visual Languages and Human-Centric Computing 2004 Proc. – Cambridge, 2004. – P. 281–282.

12. Mihalcea, R. NLP (Natural Language Processing) for NLP (Natural Language Programming) / R. Mihalcea, H. Liu, H. Lieberman // Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science. – 2006. – Vol. 3878. – P. 319–330.

13. Lieberman, H. Feasibility studies for programming in natural language / H. Lieberman, H. Liu // Kluwer Academic Publishers. – Dordrecht, 2005. – 16 p.

14. Ahmad, M. MOOIDE : Natural Language Interface for Programming MOO Environment / M. Ahmad. – Massachusetts Institute of Technology, 2008. – 70 p.

15. How natural should a natural interface be? Michael Yoshitaka Erlewine blog [Electronic resource]. – 2009. – Mode of access : http://mitcho.com/blog/projects/how-natural-should-a-natural-interface-be. – Date of access : 25.01.2015.

16. Елисеева, О.Е. Естественно-языковой интерфейс интеллектуальных систем : учеб. пособие / О.Е. Елисеева; под ред. В.В. Голенкова. – Минск : БГУИР, 2009. – 151 с.

Поступила 20.02.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: dsblizzard@gmail.com

E.G. Luttsev

PROGRAMMING IN NATURAL LANGUAGE: PUBLICATIONS REVIEW

Paper addresses a number of scientific papers devoted to the issues of programming languages close to natural languages. Description of classical approaches to the design of natural programming languages and new approaches that have made the development of these languages practical is given. One of the high-level languages – CLIPS – is compared with natural programming languages. A monograph in Russian devoted to the natural language interfaces is reviewed.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 537.8:517. 958

А.И. Куц, Г.Ч. Шушкевич

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ РАССЕЯНИЯ ПОЛЯ ЭЛЕКТРИЧЕСКОГО ДИПОЛЯ НА БИИЗОТРОПНОМ ШАРЕ

Дается аналитическое решение осесимметричной граничной задачи, описывающей процесс рассеяния электромагнитного поля электрического диполя на биизотропном шаре. Решение задачи сводится к решению системы линейных алгебраических уравнений. Приводится формула для вычисления диаграммы направленности электрического поля. Численно исследуется влияние материальных параметров биизотропного шара на диаграмму направленности отраженного электрического поля.

Введение

В середине 80-х гг. XX в. в электродинамике СВЧ возрос интерес к исследованию сложных электромагнитных сред. Примером такой среды является киральная среда, которая моделируется совокупностью проводящих зеркально-асимметричных микроэлементов, равномерно распределенных в изотропной магнитодиэлектрической среде [1]. В качестве киральных микроэлементов могут использоваться право- и левовинтовые металлические спирали, кольца с ортогональными прямолинейными концами, сферы со спиральной проводимостью, цилиндры с проводимостью вдоль винтовых линий, частицы в виде греческой буквы Ω. Более подробная классификация киральных сред приводится в работах [1–4]. Биизотропные среды являются обобщением киральных сред. Кроме киральности, данные среды обладают также свойством невзаимности, что делает их перспективными в прикладном отношении [5–7].

Интерес к изучению рассеяния электромагнитных волн на биизотропных средах обусловлен способностью этих сред как усиливать, так и поглощать электромагнитные поля. Свойство усиления может быть использовано для создания различных эффективных антенных систем СВЧ-диапазона. Свойство поглощения перспективно для создания маскирующих и малоотражающих покрытий в СВЧ-диапазоне [8–10].

Рассмотрим некоторые научные работы, относящиеся к данной теме. В работе [11] проводится исследование влияния киральности среды на электромагнитное поле электрического диполя. Излучение системы источников в киральной среде рассматривается в [12–14]. Аналитическое решение задачи дифракции плоской электромагнитной волны на биизотропном шаре предложено в работах [15, 16]. В [17] дается аналитическое решение задачи дифракции плоской электромагнитной волны на плоском слое из композитного материала. Проникновение электромагнитных полей электрического и магнитного диполей через плоский биизтропный слой рассматривается в [18]. В работах [19, 20] исследуется отражение электромагнитных волн от плоских киральных структур. Методом частичных областей в [21] решается задача рассеяния плоской электромагнитной волны на металлическом цилиндре, покрытом киральным слоем.

В настоящей работе построено точное осесимметричное решение задачи о рассеянии электромагнитного поля электрического диполя на биизотропном шаре, проведен вычислительный эксперимент для некоторых геометрических параметров задачи и различных электромагнитных параметров материала биизотропного шара.

1. Постановка задачи

Пусть пространство R^3 разделено сферой S радиуса a с центром в точке O на две области: $D_0(r > a)$ и $D_1(0 \le r < a)$. Область D_0 заполнена средой с диэлектрической проницаемостью ε_0

и магнитной проницаемостью μ_0, область D_1 – однородной биизотропной средой, материал которой характеризуется параметрами ε, μ, G, Z.

На расстоянии $h(h > a)$ от точки O расположен электрический диполь Герца, колеблющийся с круговой частотой ω. Будем полагать, что на поверхности S отсутствуют поверхностные токи и заряды, а электрический диполь ориентирован вдоль оси шара (рис. 1).

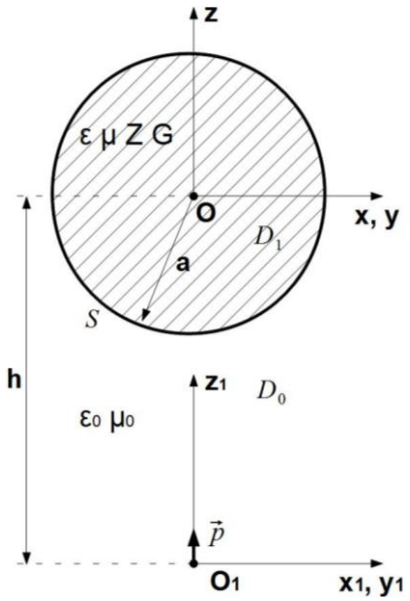


Рис. 1. Геометрия задачи

Для решения задачи свяжем с точками O и O_1 сферические координаты. Сферическая оболочка S описывается следующим образом:

$$S = \{r = a,\ 0 \le \theta \le 2\pi,\ 0 \le \varphi \le 2\pi\}.$$

Обозначим через $\vec{E}_e$, $\vec{H}_e$ векторы напряженности электрического и магнитного полей диполя соответственно. В результате взаимодействия электромагнитного поля диполя с биизотропным шаром образуются вторичные поля. Пусть $\vec{E}_0$, $\vec{H}_0$ – вторичное поле, отраженное от границы S в области D_0; $\vec{E}_1$, $\vec{H}_1$ – вторичное поле в области D_1.

Реальное электромагнитное поле определяется с помощью формул

$$\vec{\mathbb{E}}_j = \mathrm{Re}\left(\vec{E}_j\, e^{-i\omega t}\right), \quad \vec{\mathbb{H}}_j = \mathrm{Re}\left(\vec{H}_j\, e^{-i\omega t}\right),$$

где $j = 0, 1$; i – мнимая единица.

Постановка задачи. Требуется определить вторичные электромагнитные поля $\vec{E}_0$, $\vec{H}_0 \in C^1(D_0) \cap C(\bar{D}_0)$, $\vec{E}_1, \vec{H}_1 \in C(D_1) \cap C(\bar{D}_1)$, которые удовлетворяют:

– уравнениям Максвелла

$$\mathrm{rot}\,\vec{E}_0 = i\omega\mu_0 \vec{H}_0, \qquad \mathrm{rot}\,\vec{H}_0 = -i\omega\varepsilon_0 \vec{E}_0; \tag{1}$$

$$\mathrm{rot}\,\vec{E}_1 = i\omega\left(\mu\vec{H}_1 + Z\vec{E}_1\right), \qquad \mathrm{rot}\,\vec{H}_1 = -i\omega\left(\varepsilon\vec{E}_1 + G\vec{H}_1\right), \tag{2}$$

где $G = (\tau + i\kappa)\sqrt{\varepsilon_0\mu_0}$, $Z = (\tau - i\kappa)\sqrt{\varepsilon_0\mu_0}$, κ – параметр киральности, τ – параметр Теллегена;

– граничным условиям на поверхности сферы S

$$\left[\vec{n},\, \vec{E}_e + \vec{E}_0\right]\Big|_S = \left[\vec{n},\, \vec{E}_1\right]\Big|_S, \quad \left[\vec{n},\, \vec{H}_e + \vec{H}_0\right]\Big|_S = \left[\vec{n},\, \vec{H}_1\right]\Big|_S, \tag{3}$$

где $\vec{n}$ – единичная нормаль к поверхности S;

– условию излучения на бесконечности [22, 23]

$$\lim_{r\to\infty} r\left(\frac{\partial \vec{E}_0}{\partial r} - ik_0\vec{E}_0\right) = 0, \quad \lim_{r\to\infty} r\left(\frac{\partial \vec{H}_0}{\partial r} - ik_0\vec{H}_0\right) = 0, \tag{4}$$

где $k_0 = \omega\sqrt{\varepsilon_0\mu_0}$ – действительное волновое число.

2. Представление электромагнитных полей

Первичное поле ориентированного вдоль оси Oz электрического диполя Герца представим через векторные сферические волновые функции [23]:

$$\vec{E}_e = E_0\tilde{\tilde{n}}_{01}(r_1,\theta_1,k_0), \quad \vec{H}_e = H_0\tilde{\tilde{m}}_{01}(r_1,\theta_1,k_0), \tag{5}$$

где $E_0 = \dfrac{ik_0^3}{4\pi\varepsilon_0}p, \quad H_0 = \dfrac{E_0k_0}{i\omega\mu_0},$

$$\tilde{\tilde{n}}_{0n}(r,\theta,k) = \frac{n(n+1)}{kr}h_n^{(1)}(kr)P_n(\cos\theta)\vec{e}_r + g_n^{(1)}(kr)P_n^1(\cos\theta)\vec{e}_\theta,$$

$$\tilde{\tilde{m}}_{0n}(r,\theta,k) = -h_n^{(1)}(kr)P_n^1(\cos\theta)\vec{e}_\varphi,$$

$$g_n^{(1)}(x) = \frac{1}{x}\frac{d}{dx}\left(xh_n^{(1)}(x)\right) = \frac{1}{2n+1}\left((n+1)h_{n-1}^{(1)}(x) - nh_{n+1}^{(1)}(x)\right), \quad n = 1, 2, ...,$$

$\vec{p} = p\vec{e}_z$ – электрический момент диполя, $P_n(x)$ – полиномы Лежандра, $P_n^1(\cos\theta)$ – присоединенные функции Лежандра первого рода, $h_n^{(1)}(x)$ – сферические функции Ханкеля первого рода [24].

Отраженное от границы S электромагнитное поле представим в виде суперпозиции векторных сферических волновых функций, которые удовлетворяют уравнениям (1) и условию на бесконечности (4):

$$\begin{aligned}
\vec{E}_0 &= E_0\sum_{n=1}^{\infty}\left[a_n^{(2)}\tilde{\tilde{m}}_{0n}(r,\theta,k_0) + b_n^{(2)}\tilde{\tilde{n}}_{0n}(r,\theta,k_0)\right], \\
\vec{H}_0 &= H_0\sum_{n=1}^{\infty}\left[a_n^{(2)}\tilde{\tilde{n}}_{0n}(r,\theta,k_0) + b_n^{(2)}\tilde{\tilde{m}}_{0n}(r,\theta,k_0)\right].
\end{aligned} \tag{6}$$

Вторичное электромагнитное поле в области D_1 представим в виде суперпозии векторных сферических функций в композитных средах, которые удовлетворяют уравнениям (2):

$$\begin{aligned}
\vec{E}_1 &= E_0\sum_{n=1}^{\infty}\left[a_n^{(1)}\vec{K}_{0n}^{(1)}(r,\theta,k_1) + b_n^{(1)}\vec{K}_{0n}^{(2)}(r,\theta,k_2)\right], \\
\vec{H}_1 &= E_0\sum_{n=1}^{\infty}\left[a_n^{(1)}p_1\vec{K}_{0n}^{(1)}(r,\theta,k_1) + b_n^{(1)}p_2\vec{K}_{0n}^{(2)}(r,\theta,k_2)\right],
\end{aligned} \tag{7}$$

где $\vec{K}_{0n}^{(j)}\left(r,\theta,k_j\right)=\vec{n}_{0n}\left(r,\theta,k_j\right)-q_j\vec{m}_{0n}\left(r,\theta,k_j\right)$,

$$\vec{n}_{0n}\left(r,\theta,k\right)=\frac{n(n+1)}{kr}\,j_n\left(kr\right)P_n\left(\cos\theta\right)\vec{e}_r+g_n\left(kr\right)P_n^1\left(\cos\theta\right)\vec{e}_\theta,$$

$$\vec{m}_{0n}\left(r,\theta,k\right)=-j_n\left(kr\right)P_n^1\left(\cos\theta\right)\vec{e}_\varphi,$$

$$g_n\left(x\right)=\frac{1}{x}\frac{d}{dx}\left(xj_n\left(x\right)\right)=\frac{1}{2n+1}\left((n+1)j_{n-1}\left(x\right)-nj_{n+1}\left(x\right)\right),\ n=1,\,2,...,$$

$$k_j=\sqrt{g+0{,}5a^2+a\,f_j},\quad 0\le\arg k_j<\pi,\qquad g=\omega^2\left(\varepsilon\mu-ZG\right),\qquad f_j=(-1)^j\,f_0,$$

$$f_0=\sqrt{\omega^2\,\varepsilon\mu-b^2},\quad 0\le\arg f_0<\pi,\quad b=0{,}5\omega(G+Z),\qquad a=i\omega(G-Z),$$

$$g_j=f_j-0{,}5a,\ q_j=\frac{g}{k_jg_j},\ p_j=\frac{1}{\mu}\left(\frac{ig}{\omega g_j}-Z\right),$$

$j_n\left(x\right)$ – сферические функции Бесселя первого рода [24].

Неизвестные коэффициенты $a_n^{(j)}, b_n^{(j)},\ j$ = 0, 1, определены в работе [23, с. 275].

3. Выполнение граничных условий

Для выполнения граничных условий (3) представим функции (5) через векторные сферические волновые функции в системе координат с началом в точке O с помощью следующих теорем сложения [23]:

$$\tilde{\vec{n}}_{0n}\left(r_1,\theta_1,k_0\right)=\sum_{s=1}^{\infty}\tilde{A}_s^n\left(k_0h,0\right)\vec{n}_{0s}\left(r,\theta,k_0\right),\ 0\le r<h,$$

$$\tilde{\vec{m}}_{0n}\left(r_1,\theta_1,k_0\right)=\sum_{s=1}^{\infty}\tilde{A}_s^n\left(k_0h,0\right)\vec{m}_{0s}\left(r,\theta,k_0\right),\ 0\le r<h,$$

где $\tilde{A}_s^n\left(k_0h,\alpha\right)=k_0h\cos\alpha\left[\frac{1}{(2s+3)}\tilde{C}_{s+1}^n+\frac{1}{(2s-1)}\tilde{C}_{s-1}^n\right]+\tilde{C}_s^n,$

$$\tilde{C}_s^n=(2s+1)\sum_{\sigma=|s-n|}^{s+n}i^{\sigma+s-n}b_\sigma^{(n0s0)}h_\sigma^{(1)}\left(k_0h\right)P_\sigma\left(\cos\alpha\right),\quad b_\sigma^{(n0q0)}=(nq00\,|\,\sigma0)^2,$$

$(nq00\,|\,\sigma0)$ – коэффициенты Клебша – Гордона [22].

Тогда

$$\vec{E}_e=E_0\sum_{n=1}^{\infty}\tilde{A}_n^1\left(k_0h,0\right)\vec{n}_{0n}\left(r,\theta,k_0\right),\quad \vec{H}_e=H_0\sum_{n=1}^{\infty}\tilde{A}_n^1\left(k_0h,0\right)\vec{m}_{0n}\left(r,\theta,k_0\right).\qquad(8)$$

Принимая во внимание представления (6)–(8), выполняя граничные условия (3) и учитывая ортогональность присоединенных функций Лежандра на отрезке $[0, \pi]$, получим систему линейных алгебраических уравнений вида

$$M(n)\cdot V(n)=F(n), \tag{9}$$

где $M(n)=\begin{pmatrix} m_{11}(n) & m_{12}(n) & m_{13}(n) & m_{14}(n)\\ m_{21}(n) & m_{22}(n) & m_{23}(n) & m_{24}(n)\\ m_{31}(n) & m_{32}(n) & m_{33}(n) & m_{34}(n)\\ m_{41}(n) & m_{42}(n) & m_{43}(n) & m_{44}(n)\end{pmatrix}$, $V(n)=\begin{pmatrix} a_n^{(1)}\\ a_n^{(2)}\\ b_n^{(1)}\\ b_n^{(2)}\end{pmatrix}$, $F(n)=\begin{pmatrix} f_1(n)\\ f_2(n)\\ f_3(n)\\ f_4(n)\end{pmatrix}$,

$m_{11}(n)=g_n(\xi_1)$, $m_{12}(n)=0$, $m_{13}(n)=g_n(\xi_2)$, $a_{14}=-g_n^{(1)}(\xi_0)$,

$m_{21}(n)=q_1 j_n(\xi_1)$, $m_{22}(n)=h_n^{(1)}(\xi_0)$, $m_{23}(n)=q_2 j_n(\xi_2)$, $m_{24}(n)=0$,

$m_{31}(n)=\overline{p}_1 g_n(\xi_1)$, $m_{32}(n)=-g_n^{(1)}(\xi_0)$, $m_{33}(n)=\overline{p}_2 g_n(\xi_2)$, $m_{34}(n)=0$,

$m_{41}(n)=q_1\overline{p}_1 j_n(\xi_1)$, $m_{42}(n)=0$, $m_{43}(n)=q_2\overline{p}_2 j_n(\xi_2)$, $m_{44}(n)=h_n^{(1)}(\xi_0)$,

$f_1(n)=\tilde{A}_n^1(k_0h,0)g_n(\xi_0)$, $f_2(n)=0$, $f_3(n)=0$, $f_4(n)=-\tilde{A}_n^1(k_0h,0)\,j_n(\xi_0)$,

$\overline{p}_j=i\omega\mu_0 p_j/k_0$, $j=1,2$, $\xi_0=k_0a$, $\xi_1=k_1a$, $\xi_2=k_2a$.

4. Диаграмма направленности электромагнитного поля

Используя асимптотические формулы [23]

$$\tilde{\tilde{n}}_{0n}(r,\theta,k_0)\approx(-i)^n\frac{e^{ik_0r}}{k_0r}P_n^1(\cos\theta)\vec{e}_\theta,\qquad \tilde{\tilde{m}}_{0n}(r,\theta,k_0)\approx-(-i)^{n+1}\frac{e^{ik_0r}}{k_0r}P_n^1(\cos\theta)\vec{e}_\varphi,$$

получим асимптотическое представление для вектора электрического поля $\vec{E}_0$:

$$\vec{E}_0\approx E_0\frac{e^{ik_0r}}{k_0r}\vec{\Psi}(\theta),\quad r\to\infty,$$

где $\vec{\Psi}(\theta)=\Psi_1(\theta)\vec{e}_\theta+\Psi_2(\theta)\vec{e}_\varphi$,

$$\Psi_1(\theta)=\sum_{n=1}^{\infty}(-i)^n P_n^1(\cos\theta)b_n^{(2)},\qquad \Psi_2(\theta)=-\sum_{n=1}^{\infty}(-i)^{n+1}P_n^1(\cos\theta)a_n^{(2)}. \tag{10}$$

Кривая $D(\theta)=\left|\vec{\Psi}(\theta)\right|^2$ является диаграммой направленности электрического поля $\vec{E}_0$ и характеризует величину электромагнитной энергии в направлении θ:

$$D(\theta)=\left|\Psi_1(\theta)\right|^2+\left|\Psi_2(\theta)\right|^2.$$

Решая систему (9), находим представления для коэффициентов $a_n^{(2)}$, $b_n^{(2)}$:

$$a_n^{(2)} = \left|M_2(n)\right| / \left|M(n)\right|, \quad b_n^{(2)} = \left|M_4(n)\right| / \left|M(n)\right|, \tag{11}$$

где $\left|M(n)\right|$ – определитель матрицы $M(n)$; $\left|M_j(n)\right|$ – определитель матрицы $M_j(n)$; $M_j(n)$ – матрица $M(n)$, в которой *j*-й столбец заменен на вектор-столбец $F(n)$, $j = 2,4$.

5. Вычислительный эксперимент

Для проведения вычислительного эксперимента была использована система компьютерной математики Mathcad [25].

Специальные функции $j_n(x)$, $h_n^{(1)}(x) = j_n(x) + iy_n(x)$ ($y_n(x)$ – сферическая функция Бесселя второго рода), полиномы Лежандра $P_n(x)$ и присоединенная функция Лежандра

$$P_n^1(\cos\theta) = \left(nxP_n(x) - nP_{n-1}(x)\right) / \sqrt{1-x^2}\ , \ x \in (-1,1),$$

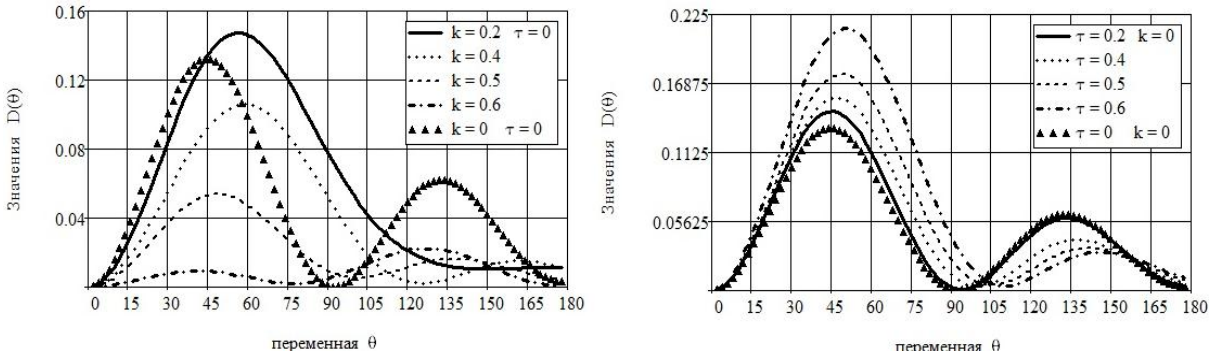
вычислялись с помощью встроенных функций [25].

Производные сферических функций вычислялись с помощью рекуррентных формул [28]

$$\frac{d}{dx} f_n(x) = nf_n(x)/x - f_{n+1}(x), \quad n = 0, 1, 2, \ \ldots \ .$$

Все сходящиеся бесконечные суммы в выражениях (10) вычислялись с точностью 10^{-5}.

На рис. 2 изображены диаграммы направленности $D(\theta)$ электрического поля $\vec{E}_0$ для возрастающих значений параметра киральности k = 0,2, 0,4, 0,5, 0,6 при $\tau = 0$ (рис. 2, *а*) и параметра Телленга τ = 0,2, 0,4, 0,5, 0,6 при $k = 0$ (рис. 2, *б*). В обоих случаях a = 0,2 м, h = 0,5 м, частота исходного поля $f = 5 \cdot 10^9$ Гц. Область D_1 заполнена материалом с относительной магнитной проницаемостью μ_r = 1,01 и относительной диэлектрической проницаемостью ε_r = 2,5. Для сравнения представлена диаграмма направленности $D(\theta)$ электрического поля для проницаемого шара с параметрами μ_r = 1,01, ε_r = 2,5 (обозначена маленькими треугольниками).

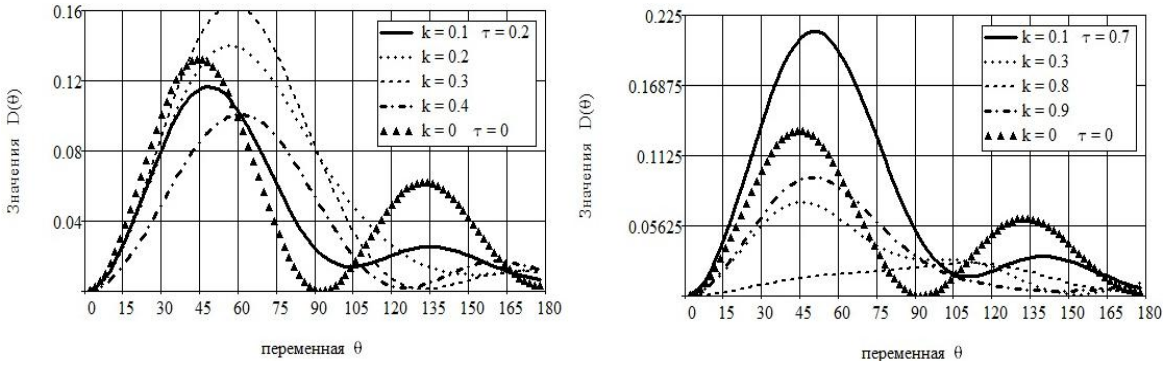


а) *б)*

Рис. 2. Диаграммы направленности $D(\theta)$ для некоторых значений параметра киральности k при τ=0 (*а*) и параметра Телленга τ при k=0 (*б*)

Анализ графиков на рис. 2 показывает, что амплитуда всех лепестков диаграммы направленности поля уменьшается с увеличением значения параметра киральности k (рис. 2, *а*), амплитуда основного лепестка диаграммы направленности возрастает с увеличением значения параметра Телленга τ (рис. 2, *б*), а боковые лепестки уменьшаются. В обоих случаях наблюдается небольшое смещение диаграммы направленности вправо по сравнению с диаграммой направленности для проницаемого шара.

На рис. 3 изображены диаграммы направленности $D(\theta)$ электрического поля $\vec{E}_0$ для возрастающих значений параметра киральности k и значений параметра Телленга $\tau = 0{,}2$ (рис. 3, *а*), $\tau = 0{,}7$ (рис. 3, *б*). Остальные параметры расчетов – прежние.

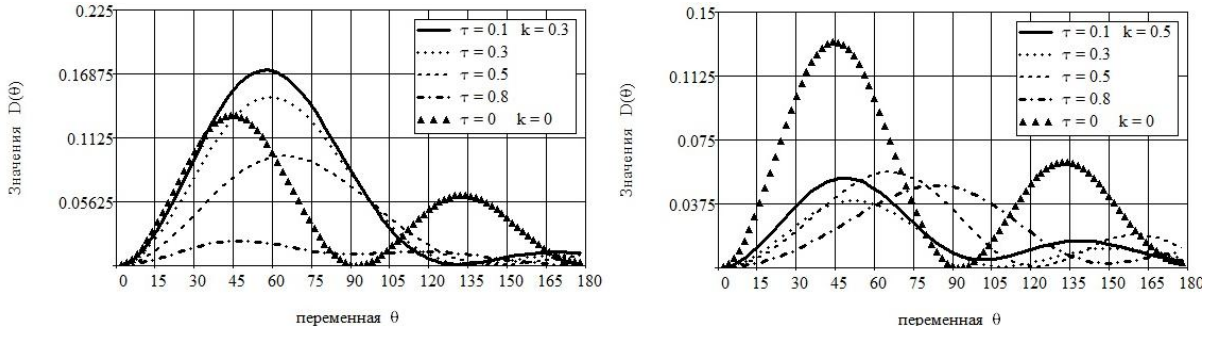


а) *б)*

Рис. 3. Диаграммы направленности $D(\theta)$ для некоторых значений параметра киральности k и параметра Телленга $\tau = 0{,}2$ (*а*), $\tau = 0{,}7$ (*б*)

Анализ графиков показывает, что с увеличением значений параметра киральности k (при фиксированном значении $\tau > 0$) амплитуда диаграммы направленности может как увеличиваться, так и уменьшаться в зависимости от значения параметра Телленга τ. При значении параметра $\tau = 0{,}2$ наблюдается возрастание амплитуды диаграммы направленности для $k = 0{,}1,\ 0{,}2,\ 0{,}3$ и уменьшение для $k = 0{,}4$ (рис. 3, *а*). При $\tau = 0{,}7$ наблюдается уменьшение амплитуды диаграммы направленности для $k = 0{,}1,\ 0{,}3,\ 0{,}8$ и возрастание для $k = 0{,}9$ (рис. 3, *б*). Как и на рис. 2, наблюдается смещение диаграмм направленности.

На рис. 4 изображены диаграммы направленности $D(\theta)$ электрического поля $\vec{E}_0$ для возрастающих значений параметра Телленга τ при фиксированных значениях параметра киральности $k = 0{,}3$ (рис. 4, *а*) и $k = 0{,}5$ (рис. 4, *б*). Остальные параметры расчетов – прежние.



а) *б)*

Рис. 4. Диаграммы направленности $D(\theta)$ для некоторых значений параметра Телленга τ и параметра киральности $k = 0{,}3$ (*а*), $k = 0{,}5$ (*б*)

Характер изменения диаграммы направленности показывает, что при увеличении параметра Телленга τ амплитуда диаграммы направленности уменьшается при значении параметра киральности $k = 0{,}3$ (рис. 4, *а*), а при значении параметра киральности $k = 0{,}5$ максимум амплитуды практически не изменяется, но происходит его смещение в сторону увеличения угла распространения поля (рис. 4, *б*).

На рис. 5 изображены диаграммы направленности $D(\theta)$ электрического поля $\vec{E}_0$ для следующих значений частоты f первичного поля: $4\cdot10^8$, $8\cdot10^8$, 10^9, $2\cdot10^9$ Гц и $a = 0{,}5$ м, $h = 1{,}5$ м. Область D_1 заполнена киральным материалом с параметрами $\mu_r = 2$, $\varepsilon_r = -4$, $\kappa = 3$, $\tau = 0$.

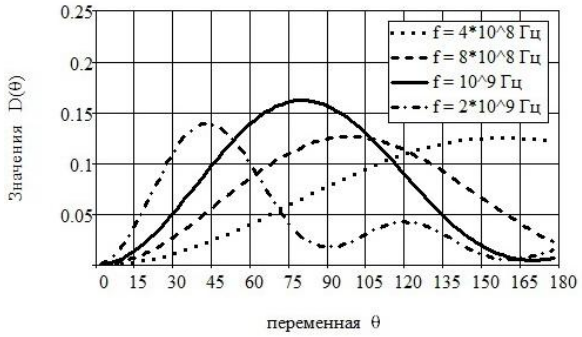


Рис. 5. Диаграммы направленности $D(\theta)$ для некоторых значений частоты f первичного поля

Из рис. 5 видно, что увеличение частоты исходного поля приводит к смещению максимума диаграммы направленности в сторону меньших значений угла распространения.

Заключение

В статье разработан аналитико-численный алгоритм решения осесимметричной задачи дифракции электромагнитного поля диполя на биизотропном шаре с радиусом, соизмеримым с длиной волны. Вычислена диаграмма направленности отраженного электрического поля. Проведен анализ влияния параметров киральности и Телленга, а также частоты поля на значения диаграммы направленности. Показано, что параметры биизотропности могут быть использованы для концентрации электромагнитного излучения энергии в направлении основного лепестка диаграммы. Для сравнительного анализа приведены диаграммы направленности для шара из кирального метаматериала, когда диэлетрическая проницаемость отрицательная, и графики для шара из обычного магнитодиэлектрического материала.

Разработанный алгоритм и программное обеспечение могут найти практическое применение при моделировании рассеяния электромагнитного поля на биизотропном шаре.

Работа выполнена при поддержке проекта «Trans-Atlantic Micromechanics Evolving Research: Materials containing inhomogeneities of diverse physical properties, shapes and orientations» № IRSES-GA-2013-610547.

Список литературы

1. Неганов, В.А. Электродинамика отражающих и волноведущих структур с искусственными киральными слоями / В.А. Неганов, О.В. Осипов // Успехи современной радиоэлектроники. – 2005. – № 8. – С. 20–45.

2. Киральные электродинамические объекты / Б.З. Каценеленбаум [и др.] // Успехи физических наук. – 1997. – Т. 167, № 11. – С. 1201–1212.

3. Cui, T.J. Metamaterials. Theory, Design and Applications / T.J. Cui, D.R. Smith, R. Lui. – Springer, 2009. – 367 p.

4. Магнитоэлектрические материалы / М.И. Бичурин [и др.]. – М. : Акад. естествознания, 2006. – 296 с.

5. Костин, М.В. К теории киральной среды на основе сферических спирально проводящих частиц / М.В. Костин, В.В. Шевченко // Радиотехника и электроника. – 1998. – Т. 43, № 8. – С. 921–926.

6. Санников, Д.Г. Кроссполяризация света на границе раздела «диэлектрик – биизотропная среда» / Д.Г. Санников // Письма в ЖТФ. – 2009. – Т. 35, вып 8. – С. 14–21.

7. Фисанов, В.В. О материальных параметрах и инвариантах изотропной киральной среды / В.В. Фисанов // Доклады ТУСУРа. – 2011.– № 2. – С. 193–196.

8. Неганов, В.А. Отражающие, волноведущие и излучающие структуры с киральными элементами / В.А. Неганов, О.В. Осипов. – М. : Радио и связь, 2006. – 280 с.

9. Иванов, О.В. Распространение элетромагнитных волн в анизотропных и бианизотропных слоистых структурах / О.В. Иванов. – Ульяновск : УлГТУ, 2010. – 262 с.

10. Проникновение электромагнитных волн через композитные экраны, содержащие идеально проводящие спирали / В.Т. Ерофеенко [и др.] // Инженерно-физический журнал. – 2011. – Т. 84, № 4. – С. 740–746.

11. Шорохова, Е.А. Исследование влияния киральности среды на излучение вертикального электрического диполя / Е.А. Шорохова, М.С. Манахова // Труды XIV науч. конф. по радиофизике. – Нижний Новгород : ННГУ, 2010. – С. 29–31.

12. Шорохова, Е.А. Излучение элементарных источников в киральной среде / Е.А. Шорохова // Радиотехника и электроника. – 2009. – Т. 54, № 6. – С. 680–688.

13. Фисанов, В.В. Об излучении источников в изотропной киральной среде / В.В. Фисанов // Изв. вузов. Физика. – 2006. – № 9. – С. 87–90.

14. Демидчик, В.И. Излучение произвольной системы источников в киральной среде / В.И. Демидчик // Вестник БГУ. Сер. 1. – 2013. – № 2. – С. 44–47.

15. Капшай, В.Н. Рассеяние электромагнитных волн на биизотропном шаре в биизотропной среде / В.Н. Капшай, В.В. Кондратюк // Проблемы физики, математики и техники. – 2010. – № 3. – С. 7–21.

16. Беличенко, В.И. Рассеяние электромагнитных волн биизотропной сферой / В.И. Беличенко, В.В. Фисанов // Изв. вузов. Физика. – 1994. – № 10. – С. 108–112.

17. Ерофеенко, В.Т. Дифракция плоской электромагнитной волны на плоском слое из биизотропного материала / В.Т. Ерофеенко, С.В. Малый // Вестник БГУ. Сер. 1. – 2010. – № 2. – С. 11–16.

18. Ерофеенко, В.Т. Численное исследование взаимодействия электромагнитных полей электрического и магнитного диполей с композитным экраном / В.Т. Ерофеенко, В.Ф. Бондаренко // Изв. НАН Беларуси. Сер. физ.-техн. наук. – 2013. – № 4. – С. 113–120.

19. Неганов, В.А. Отражение электромагнитных волн от плоских киральных структур / В.А. Неганов, О.В. Осипов // Изв. вузов. Радиофизика. – 1999. – Т. 42, № 9. – С. 870–878.

20. Неганов, В.А. Особенности отражения электромагнитных волн от плоских киральных структур / В.А. Неганов, О.В. Осипов // Физика волновых процессов и радиотехнические системы. – 1999. – Т. 2, № 1. – С. 5–11.

21. Неганов, В.А. Рассеяние плоских электромагнитных волн на кирально-металлическом цилиндре / В.А. Неганов, О.В. Осипов // Письма в ЖТФ. – 2000. – Т. 26, вып. 1. – С. 77–83.

22. Иванов, Е.А. Дифракция электромагнитных волн на двух телах / Е.А. Иванов. – Минск : Наука и техника, 1968. – 584 с.

23. Ерофеенко, В.Т. Аналитическое моделирование в электродинамике / В.Т. Ерофеенко, И.С. Козловская. – М. : КД «Либроком», 2014. – 304 с.

24. Справочник по специальным функциям с формулами, графиками и таблицами / под ред. М. Абрамовица, И. Стиган. – М. : Наука, 1979. – 830 с.

25. Шушкевич, Г.Ч. Компьютерные технологии в математике. Система Mathcad 14 / Г.Ч. Шушкевич, С.В. Шушкевич. – Минск : Изд-во Гревцова, 2010. – Ч. 1. – 287 с.

Поступила 10.04.2015

Гродненский государственный
университет им. Янки Купалы,
Гродно, Ожешко, 22
e-mail: g_shu@tut.by

A.I. Kuts, G.Ch. Shushkevich

NUMERICAL STUDY OF A FIELD SCATTERING OF AN ELECTRICAL DIPOLE ON A BI-ISOTROPIC BALL

An analytical solution of the boundary problem describing scattering of an electromagnetic field of the electric dipole on the bi-isotropic ball is constructed. An influence of some parameters of the problem on the value of the directivity pattern of the electric field is studied by a numerical simulation.

УДК 517. 958:537.876.23:621.3

В.Т. Ерофеенко[1], В.Ф. Бондаренко[2]

ВЫЧИСЛЕНИЕ ЭФФЕКТИВНЫХ ЭЛЕКТРОДИНАМИЧЕСКИХ ПАРАМЕТРОВ КОМПОЗИТА С ЧАСТИЦАМИ ИЗ СПЕЦИАЛЬНЫХ МАТЕРИАЛОВ

Рассматривается аналитический алгоритм вычисления эффективных материальных параметров матричного композита, состоящего из магнитодиэлектрической матрицы и случайно распределенной системы сферических частиц. Исследуются эффективные параметры композитов с частицами из материалов различных типов: биизотропных, проводящих, наноразмерных, с запаздыванием по времени и релаксацией среды. Представлен графический материал расчетов.

Введение

В настоящее время одним из приоритетных направлений научных исследований является изучение электродинамических свойств и построение математических моделей композитов [1–5]. Композитные материалы представляют собой структурно неоднородную среду с большим числом частиц, случайно распределенных в однородной среде, называемой матрицей. Структурные элементы композита различаются геометрией, химическим составом и линейными размерами по отношению к длине электромагнитной волны, воздействующей на композит. К композитам относятся метаматериалы, киральные среды, квадрупольные материалы, наноструктуры и др. Один из основных методов моделирования композитов, упрощающих численное исследование, сводится к эквивалентной замене структурно неоднородных материалов однородными средами. В литературе описан ряд подходов к определению эффективных параметров композитов, которые ориентированы на учет геометрических особенностей структурных неоднородностей и степень их взаимного влияния. При этом используются различные принципы моделирования.

Настоящая работа посвящена исследованию матричных композитов, содержащих сферические частицы из материалов различных типов. Разработан алгоритм для вычисления эффективных параметров композитов, который основывается на методе, описанном в [6, 7]. В основе метода лежит принцип однократного рассеяния электромагнитного поля между частицами композита с длиной волны, значительно превосходящей размеры частиц. В статье представлены графики эффективных параметров в зависимости от концентрации частиц в композите и частоты поля, воздействующего на композит.

1. Структура композитов

В матрице, заполненной средой с диэлектрической и магнитной проницаемостями $\varepsilon_M = \varepsilon_M^r \varepsilon_0$, $\mu_M = \mu_M^r \mu_0$, случайным образом размещено большое число биизотропных сферических частиц радиуса R, характеризуемых параметрами $\varepsilon = \varepsilon_r \varepsilon_0$, $\mu = \mu_r \mu_0$, $G = G_r / c$, $Z = Z_r / c$, где c – скорость света. Для описания структуры композита введем обозначения: ν – концентрация частиц (число частиц в единице объема матрицы), $\tau = \nu V_R$ – объемный коэффициент заполнения матрицы, D_S – область внутри частицы с номером s, D_0 – область между частицами; $\vec{E}, \vec{H}$ – электромагнитное поле в области D_0; $\vec{E}_S, \vec{H}_S$ – поле в частице D_S.

Поля в композите с биизотропными частицами подчиняются уравнениям [7]

$$\begin{gathered} \operatorname{rot}\vec{E} = i\omega\mu_M \vec{H}, \quad \operatorname{rot}\vec{H} = -i\omega\varepsilon_M \vec{E} \ \text{ в } D_0, \\ \operatorname{rot}\vec{E}_S = i\omega\left(\mu\vec{H}_S + Z\vec{E}_S\right), \quad \operatorname{rot}\vec{H}_S = -i\omega\left(\varepsilon\vec{E}_S + G\vec{H}_S\right) \ \text{ в } D_S, \end{gathered} \tag{1}$$

где ω – круговая частота поля с длиной волны, значительно большей диаметра частицы. Неоднородный композит из матрицы с частицами заменим на эквивалентную биизотропную однородную среду с эффективными параметрами $\varepsilon_{эф} = \varepsilon_э \varepsilon_0$, $\mu_{эф} = \mu_э \mu_0$, $G_{эф} = G_э / c$, $Z_{эф} = Z_э / c$. Поле в эффективной среде удовлетворяет уравнениям [7]

$$\operatorname{rot} \vec{E} = i\omega\left(\mu_{эф}\vec{H} + Z_{эф}\vec{E}\right), \quad \operatorname{rot} \vec{H} = -i\omega\left(\varepsilon_{эф}\vec{E} + G_{эф}\vec{H}\right) \text{ в } R^3. \tag{2}$$

Таким образом, неоднородный композит, описываемый уравнениями (1), заменяется на эквивалентный однородный биизотропный композит, который соответствует уравнениям (2). Рассмотрим следующие уравнения как частные случаи уравнений (1).

Для проводящих частиц имеем уравнения

$$\operatorname{rot} \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t}, \quad \operatorname{rot} \mathbf{H} = \varepsilon_0 \varepsilon' \frac{\partial \mathbf{E}}{\partial t} + \gamma \mathbf{E},$$

тогда монохроматические поля с временной зависимостью $\exp(-i\omega t)$ описываются уравнениями для комплексных амплитуд

$$\operatorname{rot} \vec{E} = i\omega\mu\vec{H}, \quad \operatorname{rot} \vec{H} = -i\omega\varepsilon_{П}(\omega)\vec{E} \text{ в } D_S,$$

где

$$\varepsilon_{П}(\omega) = \varepsilon_0 \varepsilon_r(\omega) = \varepsilon_0\left(\varepsilon' + i\frac{\gamma}{\varepsilon_0\omega}\right); \tag{3}$$

γ – удельная электрическая проводимость.

Электродинамику частиц с запаздыванием электрической и магнитной поляризации опишем с помощью уравнений [8, с. 22]

$$\begin{gathered} \operatorname{rot} \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad \operatorname{rot} \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t}, \\ \mathbf{D} = \varepsilon_0\left(\mathbf{E}(t) + \kappa\mathbf{E}(t - t_e)\right), \ \mathbf{B} = \mu_0\left(\mathbf{H}(t) + \chi\mathbf{H}(t - t_m)\right), \end{gathered} \tag{4}$$

где t_e, t_m – времена запаздывания; κ – диэлектрическая восприимчивость среды, χ – магнитная восприимчивость среды.

Из уравнений (4) следует, что монохроматические поля в частицах подчиняются уравнениям

$$\operatorname{rot} \vec{E} = i\omega\mu_3\vec{H}, \quad \operatorname{rot} \vec{H} = -i\omega\varepsilon_3(\omega)\vec{E} \text{ в } D_S,$$

где

$$\begin{aligned} \varepsilon_3 &= \varepsilon_0 \varepsilon_r(\omega) = \varepsilon_0\left(1 + \kappa e^{i\omega t_e}\right), \\ \mu_3 &= \mu_0 \mu_r(\omega) = \mu_0\left(1 + \chi e^{i\omega t_m}\right). \end{aligned} \tag{5}$$

Электродинамику частиц, состоящих из сред с релаксацией, определим уравнениями индукций [8, с. 78]

$$\mathbf{D} = \varepsilon_0\left(\mathbf{E} + \kappa\int_0^\infty f(\eta)\mathbf{E}(t - \eta)d\eta\right), \quad \mathbf{B} = \mu_0\left(\mathbf{H} + \chi\int_0^\infty g(\eta)\mathbf{H}(t - \eta)d\eta\right), \tag{6}$$

где $f(\eta)=\frac{1}{\tau_e}\exp\left(-\frac{\eta}{\tau_e}\right)$, $g(\eta)=\frac{1}{\tau_m}\exp\left(-\frac{\eta}{\tau_m}\right)$, τ_e – время электрической релаксации, τ_m – время магнитной релаксации.

Для монохроматических полей в частицах с релаксацией из (6) получим уравнения

$$\operatorname{rot}\vec{E}=i\omega\mu_R(\omega)\vec{H},\quad \operatorname{rot}\vec{H}=-i\omega\varepsilon_R(\omega)\vec{E}\ \text{ в } D_S,$$

где

$$\varepsilon_R(\omega)=\varepsilon_0\varepsilon_r(\omega)=\varepsilon_0\left(1+\frac{\kappa}{1-i\tau_e\omega}\right),\qquad \mu_R(\omega)=\mu_0\mu_r(\omega)=\mu_0\left(1+\frac{\chi}{1-i\tau_m\omega}\right). \tag{7}$$

По аналогии могут быть рассмотрены частицы из сред других типов. В частности, в работе [7] производился расчет композитов с частицами из биизотропных фокусирующих материалов.

2. Алгоритм расчета эффективных материальных параметров композита

Алгоритм вычисления эффективных параметров опишем следующими процедурами:

1. Ввод исходных данных:

f – частота;

R – радиус шара;

τ – коэффициент заполнения, $0<\tau<0{,}5$;

ε_r – относительная диэлектрическая проницаемость (комплексная);

μ_r – относительная магнитная проницаемость (комплексная);

G_r, Z_r – параметры киральности (комплексные);

ε_M^r – относительная диэлектрическая проницаемость матрицы (комплексная);

μ_M^r – относительная магнитная проницаемость матрицы (комплексная);

$\varepsilon_0=8{,}854\cdot10^{-12}$, $\mu_0=4\pi\cdot10^{-7}$.

2. Вычисление вспомогательных величин:

$\omega=2\pi f$ – круговая частота,

$\kappa_0=\frac{\omega}{c}$, $Z_0=\sqrt{\frac{\mu_0}{\varepsilon_0}}$,

$V_R=\frac{4\pi}{3}R^3$ – объем шара,

$$\overline{\kappa}_M=\sqrt{\varepsilon_M^r\mu_M^r},\quad 0\le\arg\overline{\kappa}_M<\pi,\quad \kappa_M=\kappa_0\overline{\kappa}_M,\quad h_M=\frac{\overline{\kappa}_M}{iZ_0\mu_M^r},\ \nu=\frac{\tau}{V_R},\quad \overline{\tau}=1-\tau,$$

$$a=i(G_r-Z_r),\quad g=\varepsilon_r\mu_r-Z_rG_r,\quad f_0=\sqrt{\varepsilon_r\mu_r-\frac{1}{4}(G_r+Z_r)^2},\quad 0\le\arg f_0<\pi,$$

$$f_j=(-1)^j f_0,\quad j=1,2,\quad g_j=f_j-\frac{1}{2}a,\quad \overline{\kappa}_j=\sqrt{g+\frac{1}{2}a^2+af_j},\quad 0\le\arg\overline{\kappa}_j<\pi,$$

$$\kappa_j=\kappa_0\overline{\kappa}_j,\quad \xi_j=\kappa_jR,\quad \xi_M=\kappa_MR,\quad q_j=\frac{g}{\overline{\kappa}_jg_j},$$

$$\bar{p}_j = -\frac{\mu_M^r}{\bar{\kappa}_M \mu_r}\left(\frac{g}{g_j} + iZ_r\right), \quad p_j = \frac{1}{\mu_r}\left(i\frac{g}{g_j} - Z_r\right),$$

$$F_0 = \frac{V_R}{\xi_M} + \frac{4\pi i}{3\kappa_M^3 h_1^{(1)}(\xi_M)}, \quad p_0 = i\frac{2}{3}\xi_M^2\left(\bar{\tau} + 3\nu F_0 j_1(\xi_M)\right),$$

$$F_j = \frac{V_R}{\xi_j} j_1(\xi_j), \qquad F_j^{(1)} = F_j\varepsilon_0\left(\varepsilon_r + p_j G_r\right), \quad F_j^{(2)} = \frac{i}{c}F_j\frac{g}{g_j}, \quad j = 1, 2,$$

$$j_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x}, \quad h_1^{(1)}(x) = -\left(\frac{1}{x} + \frac{i}{x^2}\right)e^{ix},$$

$$g_1^{(1)}(x) = \frac{1}{x}\left(\frac{i}{x^2} + \frac{1}{x} - i\right)e^{ix}, \quad g_1(x) = \frac{1}{x}\left(\frac{\cos x}{x} + \left(1 - \frac{1}{x^2}\right)\sin x\right).$$

3. Вычисление матрицы:

$$\hat{N} = \begin{pmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{pmatrix}, \quad \hat{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix},$$

$$n_{11} = \varepsilon_0\varepsilon_M^r\left(p_0 p_{21} + 2\nu F_0 q_{21}\right) + 2\nu F_1^{(1)}, \quad n_{12} = \varepsilon_0\varepsilon_M^r\left(p_0 p_{22} + 2\nu F_0 q_{22}\right) + 2\nu F_2^{(1)},$$

$$n_{21} = \mu_0\mu_M^r h_M\left(p_0 p_{11} + 2\nu F_0 q_{11}\right) + 2\nu F_1^{(2)}, \quad n_{22} = \mu_0\mu_M^r h_M\left(p_0 p_{12} + 2\nu F_0 q_{12}\right) + 2\nu F_2^{(2)};$$

$$m_{11} = h_M\left(p_0 p_{12} + 2\nu F_0 q_{12}\right) + 2\nu\frac{p_2}{Z_0}F_2, \quad m_{12} = -\left(p_0 p_{22} + 2\nu F_0 q_{22} + 2\nu F_2\right),$$

$$m_{21} = -h_M\left(p_0 p_{11} + 2\nu F_0 q_{11}\right) - 2\nu\frac{p_1}{Z_0}F_1, \quad m_{22} = p_0 p_{21} + 2\nu F_0 q_{21} + 2\nu F_1;$$

$$d = m_{11}m_{22} - m_{12}m_{21}, \quad q_{11} = q_1 j_1(\xi_1), \quad q_{12} = q_2 j_1(\xi_2), \quad q_{21} = \bar{p}_1 q_1 j_1(\xi_1), \quad q_{22} = \bar{p}_2 q_2 j_1(\xi_2),$$

$$p_{11} = \bar{p}_1 g_1(\xi_1) h_1^{(1)}(\xi_M) + q_1 j_1(\xi_1) g_1^{(1)}(\xi_M), \quad p_{12} = \bar{p}_2 g_1(\xi_2) h_1^{(1)}(\xi_M) + q_2 j_1(\xi_2) g_1^{(1)}(\xi_M),$$

$$p_{21} = g_1(\xi_1) h_1^{(1)}(\xi_M) + \bar{p}_1 q_1 j_1(\xi_1) g_1^{(1)}(\xi_M), \quad p_{22} = g_1(\xi_2) h_1^{(1)}(\xi_M) + \bar{p}_2 q_2 j_1(\xi_2) g_1^{(1)}(\xi_M);$$

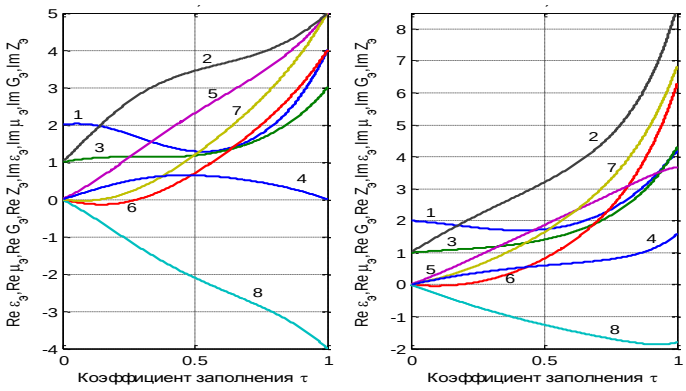
$$\hat{C} = \frac{1}{d}\hat{N}\hat{M} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}.$$

4. Вычисление эффективных параметров композита:

$$\varepsilon_э = c_{11}/\varepsilon_0, \quad \mu_э = c_{22}/\mu_0, \quad G_э = cc_{12}, \quad Z_э = cc_{21}. \tag{8}$$

3. Вычислительный эксперимент

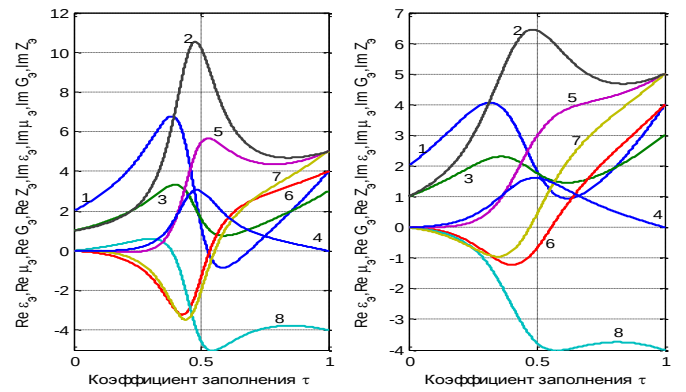
Используя разработанный алгоритм, построим графики эффективных параметров (8) для ряда композитов: для биизотропных материалов (рис. 1–3) и для обычных сред $G = 0$, $Z = 0$ (рис. 4–6).



а) *б)*

Рис. 1. Эффективные параметры композита с биизотропными частицами для высоких частот:
а) $f = 10$ МГц; *б)* $f = 4$ ТГц
$1 - \mathrm{Re}\,\varepsilon_э,\ 2 - \mathrm{Im}\,\varepsilon_э,\ 3 - \mathrm{Re}\mu_э,\ 4 - \mathrm{Im}\mu_э,\ 5 - \mathrm{Re}\,G_э,\ 6 - \mathrm{Im}G_э,\ 7 - \mathrm{Re}\,Z_э,\ 8 - \mathrm{Im}\,Z_э,$
$R = 4\cdot 10^{-6},\ \varepsilon_r = 4 + 5i,\ \mu_r = 3,\ G_r = 5 + 4i,\ Z_r = 5 - 4i,\ \varepsilon_M^r = 2 + i,\ \mu_M^r = 1$

На рис. 1 представлены параметры композита из киральных частиц, характеризуемых комплексно сопряженными параметрами биизотропности $G_r = \nu + i\kappa,\ Z_r = \nu - i\kappa,$ где ν – параметр Теллегена, κ – параметр киральности $(\nu = 5, \kappa = 4)$. Сравнение графиков показывает, что эффективные параметры биизотропности G, Z в композите не являются комплексно сопряженными. Это, по всей видимости, связано с тем, что параметры матрицы имеют разные фазы. На рисунках представлены графики с аргументом в пределах $0 < \tau < 1$. Модель же разработана для значений $0 < \tau < 0{,}524$. При $\tau = 0{,}524$ частицы композита соприкасаются. Для параметров $0{,}524 < \tau < 1$ предполагается, что между крупными частицами радиуса R располагаются частицы меньших радиусов, но в модели это не учтено. Из рис. 1, *а* следует, что при $\tau = 1$ эффективные параметры принимают значения материальных параметров частицы. Это согласуется с фактом, что при $\tau = 1$ композит полностью состоит из материала частицы. При высоких частотах (рис. 1, *б*) такое не наблюдается. Значит, при низких частотах алгоритм достаточно точно моделирует эффективные параметры композита. Для высоких частот $f > 4\cdot$ ТГц имеем $\dfrac{R}{\lambda} > 5{,}3\cdot 10^{-2}$, т. е. нарушается условие модели $\dfrac{R}{\lambda} < \dfrac{1}{30}$ [6].

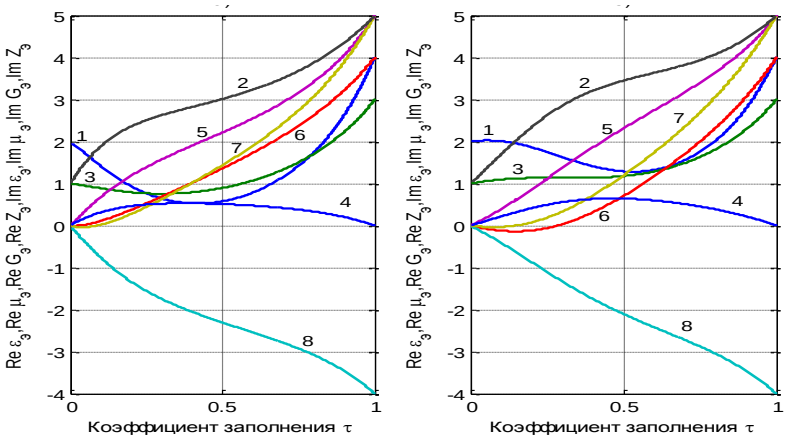


а) *б)*

Рис. 2. Эффективные параметры композита с биизотропными частицами для низких частот:
а) $f = 10$ Гц; *б)* $f = 1$ кГц
$1 - \mathrm{Re}\,\varepsilon_э,\ 2 - \mathrm{Im}\,\varepsilon_э,\ 3 - \mathrm{Re}\mu_э,\ 4 - \mathrm{Im}\mu_э,\ 5 - \mathrm{Re}\,G_э,\ 6 - \mathrm{Im}G_э,\ 7 - \mathrm{Re}\,Z_э,\ 8 - \mathrm{Im}\,Z_э,$
$R = 4\cdot 10^{-6},\ \varepsilon_r = 4 + 5i,\ \mu_r = 3,\ G_r = 5 + 4i,\ Z_r = 5 - 4i,\ \varepsilon_M^r = 2 + i,\ \mu_M^r = 1$

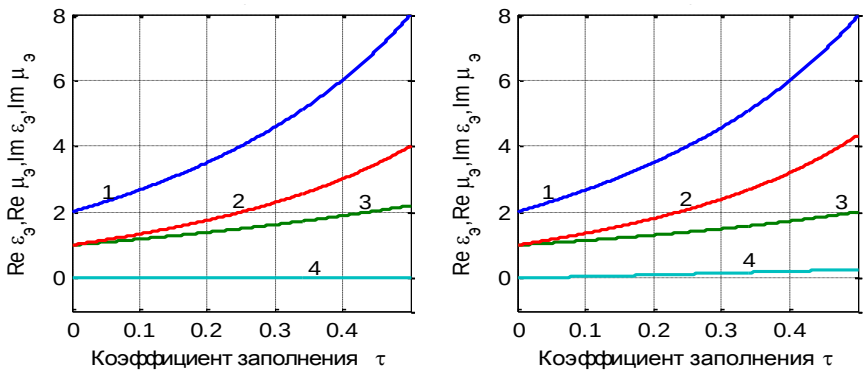
На рис. 2 представлены эффективные параметры композита, аналогичные параметрам рис. 1, но при низких частотах (f = 10 Гц, f = 1 кГц). Показано, что при плотной упаковке композита частицами $(\tau \to 0{,}5)$ зависимость эффективных параметров от величины τ сильно изменяется: на кривых появляются экстремумы, тогда как на рис. 1 параметры изменяются монотонно.

На рис. 3 исследуется композит с материальными параметрами, аналогичными параметрам рис. 1, но рассматриваются частицы меньших размеров (наноразмерные). В этом случае отношение радиуса частиц и длины волны поля $\frac{R}{\lambda} \approx 5{,}3 \cdot 10^{-4}$. Надо отметить, что графики эффективных параметров при $\tau = 1$ согласуются с физическими значениями, когда композит полностью заполнен материалом частиц.



а) *б)*

Рис. 3. Эффективные параметры композита с биизотропными наночастицами:
а) $f = 10$ МГц; *б)* $f = 4$ ТГц
$1 - \mathrm{Re}\,\varepsilon_э$, $2 - \mathrm{Im}\,\varepsilon_э$, $3 - \mathrm{Re}\,\mu_э$, $4 - \mathrm{Im}\,\mu_э$, $5 - \mathrm{Re}\,G_э$, $6 - \mathrm{Im}\,G_э$, $7 - \mathrm{Re}\,Z_э$, $8 - \mathrm{Im}\,Z_э$,
$R = 4 \cdot 10^{-8}$, $\varepsilon_r = 4 + 5i$, $\mu_r = 3$, $G_r = 5 + 4i$, $Z_r = 5 - 4i$, $\varepsilon_M^r = 2 + i$, $\mu_M^r = 1$



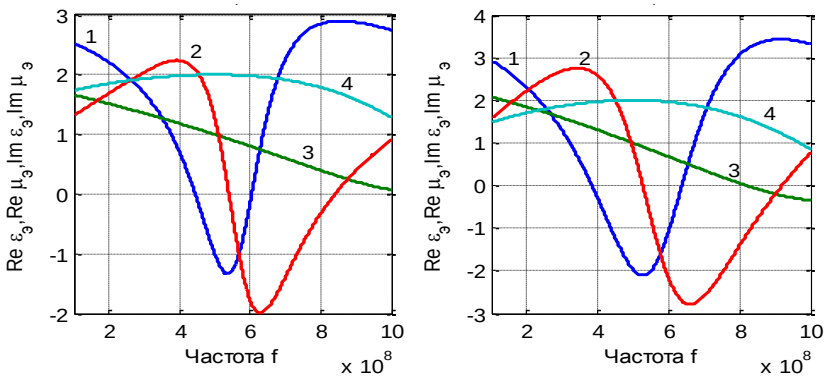
а) *б)*

Рис. 4. Эффективные параметры композита с проводящими частицами:
а) $f = 1$ кГц; *б)* $f = 1$ ТГц
$1 - \mathrm{Re}\,\varepsilon_э$, $2 - \mathrm{Im}\,\varepsilon_э$, $3 - \mathrm{Re}\,\mu_э$, $4 - \mathrm{Im}\,\mu_э$, $R = 4 \cdot 10^{-6}$, $\gamma = 3 \cdot 10^4$,
$\mu_r = 5$, $G_r = 0$, $Z_r = 0$, $\varepsilon_M^r = 2 + i$, $\mu_M^r = 1$

Из рис. 4 следует, что композит из частиц с удельной электрической проводимостью $\gamma = 3 \times 10^4 \frac{\text{См}}{\text{м}}$ и $\varepsilon' = 1$ (3) при низкой частоте f = 1 кГц обладает слабой эффективной проводи-

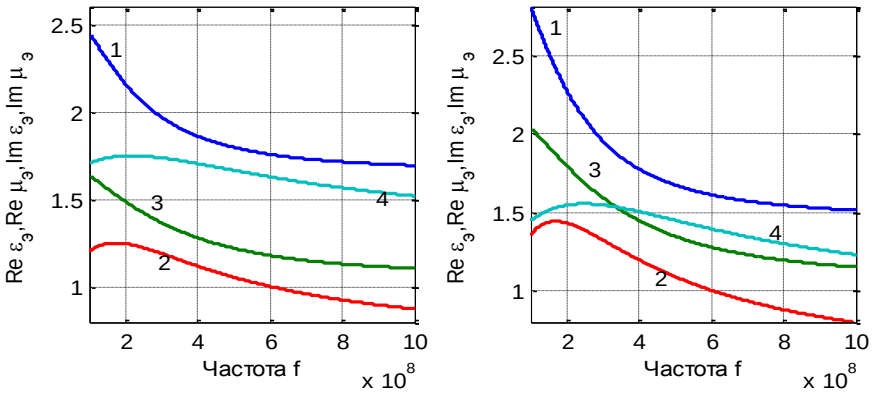
мостью $\gamma_э = \omega\varepsilon_0 \mathrm{Im}(\varepsilon_э) \approx 2{,}2\times10^{-7}\ \frac{\text{См}}{\text{м}}$ Это связано с тем, что в модели не учитывается контакт частиц. При увеличении частоты эффективная проводимость пропорционально повышается: $\gamma_э \approx 2{,}2\cdot10^{2}$ при частоте f = 1 ТГц. Заметим, что увеличение проводимости γ до 10^{7} практически не сказывается на динамике изменения рассчитанных параметров от частоты.

На рис. 5 показаны графики зависимости эффективных параметров композита в диапазоне частот, для которых период колебаний поля согласуется с временами запаздывания. Для расчетов используются формулы (5). При частоте $f_0 = 5\cdot10^{8}$, расположенной в середине диапазона, период $T = 2\cdot10^{-9}$. Следует $t_e = T/2,\ t_m = 0{,}05T$. В окрестности указанной частоты мнимая и действительная части эффективной диэлектрической проницаемости принимают отрицательные значения, т. е. на данной частоте композит является метаматериалом.



а) *б)*

Рис. 5. Эффективные параметры композита из частиц с запаздыванием:
а) $\tau = 0{,}3$; *б)* $\tau = 0{,}5$
$1 - \mathrm{Re}\,\varepsilon_э,\ 2 - \mathrm{Im}\,\varepsilon_э,\ 3 - \mathrm{Re}\mu_э,\ 4 - \mathrm{Im}\mu_э$
$R = 4\cdot10^{-6},\ t_e = 10^{-9},\ t_m = 5\cdot10^{-10},\ \kappa = 4,\ \chi = 2,\ G_r = 0,\ Z_r = 0,\ \varepsilon_M^r = 2 + i,\ \mu_M^r = 1 + 2i$



а) *б)*

Рис. 6. Эффективные параметры композита из частиц с релаксацией:
а) $\tau = 0{,}3$; *б)* $\tau = 0{,}5$
$1 - \mathrm{Re}\,\varepsilon_э,\ 2 - \mathrm{Im}\,\varepsilon_э,\ 3 - \mathrm{Re}\mu_э,\ 4 - \mathrm{Im}\mu_э$
$R = 4\cdot10^{-6},\ t_e = 10^{-9},\ t_m = 5\cdot10^{-10},\ \kappa = 4,\ \chi = 2,\ G_r = 0,\ Z_r = 0,\ \varepsilon_M^r = 2 + i,\ \mu_M^r = 1 + 2i$

Для построения графиков на рис. 6 использовались формулы (7). Получены пологие графики, в то время как в композите с запаздыванием (см. рис. 5) происходят существенные изменения параметров в зависимости от частоты поля.

Заключение

В широком диапазоне частот численно исследуется алгоритм определения эффективных материальных параметров матричного композита из магнитодиэлектрической матрицы и сферических частиц в случае, когда частицы заполняют композит вплоть до соприкосновения частиц. При этом учитывается одно переотражение электромагнитного поля между частицами. Рассмотрен также вариант аналитического продолжения алгоритма на случай $(0{,}5 < \tau \le 1)$, когда материал частиц полностью заполняет матрицу. Важно, что этот вариант при низких частотах согласуется с физическими значениями материальных параметров.

Алгоритм применен к исследованию композитов с частицами из различных материалов, перечень которых может быть значительно расширен.

Работа выполнена по заданию ГПНИ «Информатика и космос».

Список литературы

1. Виноградов, А.П. Электродинамика композитных материалов / А.П. Виноградов. – М. : Эдиториал УРСС, 2001. – 206 с.

2. Виноградов, А.П. К вопросу об эффективных параметрах метаматериалов /А.П. Виноградов, А.В Дорофеенко, С. Зухди // Успехи физических наук. – 2008. – Т. 178, № 5. – С. 514–518.

3. Памятных, Е.А. Основы электродинамики материальных сред в переменных и неоднородных полях / Е.А. Памятных, Е.А. Туров. – М. : Наука. Физматлит, 2000. – 240 с.

4. Витязь, П.А. Основы нанотехнологий и наноматериалов / П.А. Витязь, Н.А. Свидунович. – Минск : Вышэйшая школа, 2010. – 304 с.

5. Исимару, А. Распространение и рассеяние волн в случайно неоднородных средах. Т. 1. Однократное рассеяние / А. Исимару. – М. : Мир, 1981. – 280 с.

6. Ерофеенко, В.Т. Электродинамическая модель расчета эффективных параметров композитов из сферических биизотропных частиц / В.Т. Ерофеенко // Информатика. – 2014. – № 1. – С. 45 – 58.

7. Ерофеенко, В.Т. Экранирование электромагнитных полей экранами из матричных композитов, содержащих биизотропные частицы / В.Т. Ерофеенко, В.Ф. Бондаренко // Информатика. – 2014. – № 3.– С. 28–43.

8. Ерофеенко, В.Т. Аналитическое моделирование в электродинамике / В.Т. Ерофеенко, И.С. Козловская. – М. : КД «Либроком», 2014. – 304 с.

Поступила 19.04.2015

[1]*Учреждение БГУ «НИИ прикладных проблем математики и информатики»,*
Минск, пр. Независимости, 4
e-mail: bsu_erofeenko@tut.by

[2]*Высший государственный колледж связи,*
Минск, ул. Ф. Скорины, 8, корп. 2
e-mail: valbandarenka@yandex.ru

V.T. Erofeenko, V.F. Bondarenko

CALCULATION OF EFFECTIVE ELECTRODYNAMICAL PARAMETERS FOR COMPOSITES WITH PARTICLES FROM SPECIAL MEDIUMS

A analytical algorithm for calculation of the effective material parameters of matrix composites consisting of the magnetodielectric matrix and a randomly distributed system of spherical particles is considered. The effective parameters of composites of different types materials such as biisotropical, conducting, nanodimensianal, with lag time and relaxation factor of medium are investigated. Graphical material of computational experiments is presented.

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ

УДК 004.33.054

В.Н. Ярмолик[1], В.А. Леванцевич[1], И. Мрозек[2]

МНОГОКРАТНЫЕ УПРАВЛЯЕМЫЕ ВЕРОЯТНОСТНЫЕ ТЕСТЫ

Рассматриваются однократные управляемые вероятностные тесты, методы их формирования, а также их применение для тестирования средств вычислительных систем. Показываются основные недостатки построения однократных вероятностных тестов. Предлагается метод построения многократных управляемых вероятностных тестов на базе исходного однократного теста. Анализируются различные численные метрики для построения как однократных, так и многократных управляемых вероятностных тестов.

Введение

Вероятностные тесты (Random Tests) и их многочисленные модификации, основанные на принципе черного ящика, являются эффективным средством для тестирования современных вычислительных систем [1–6]. Существующие модификации классического метода построения вероятностных тестов [1, 2] объединяются принципом управления процедурой формирования очередного тестового набора [3]. Действительно, такие виды формирования модифицированных вероятностных тестов, как антивероятностные тесты (Antirandom Tests) [7], быстрые антивероятностные тесты (FastAntirandomTests) [8], адаптивные вероятностные тесты (Adaptive RandomTests) [9], эффективные вероятностные тесты (Good Random Tests) [10], зеркальные вероятностные тесты (Mirror Random Tests) [11], упорядоченные вероятностные тесты (Orderly Random Tests) [12], эволюционные вероятностные тесты (Evolutionary Random Tests) [13], управляемые вероятностные тесты (Controlled Random Tests) [3] и др., основаны на вычислении некоторых характеристик для управляемого формирования очередного случайного тестового набора [1, 3, 7–13]. Приведенные разновидности вероятностных тестов и множество других их модификаций получили общее название «управляемые вероятностные тесты» [3, 14].

Существенный недостаток управляемых вероятностных тестов – необходимость перебора потенциальных кандидатов в тестовые наборы и вычисления для них характеристик, являющихся критериями для включения либо невключения их в вероятностный тест, что увеличивает вычислительную сложность формирования подобных тестов [7−14].

С целью уменьшения вычислительной сложности формирования управляемых вероятностных тестов широко обсуждаются и используются итеративные вероятностные тесты (Iterative Random Tests) [1, 15, 16], исчерпывающие и почти псевдоисчерпывающие вероятностные тесты (Combinatorial Tests) [1, 15–18], вероятностные тесты с малым числом наборов [3, 19], квазивероятностные тесты (Quasi-Random Tests) [4, 5], а также многократные тесты (Multi-runTests) для запоминающих устройств [20, 21]. Основное достоинство указанных разновидностей вероятностных тестов заключается в использовании некоторой обобщающей характеристики для теста в целом, а не для тестового набора в отдельности, что позволяет значительно уменьшить вычислительную сложность построения подобных тестов. Кроме того, в ряде случаев построение многократных тестов, в частности многократных маршевых тестов запоминающих устройств, не требует вычисления каких-либо характеристик, а основывается на реализации предварительно определенной процедуры [20, 21].

Целью настоящей статьи является разработка методики построения многократных управляемых вероятностных тестов, которая основана на использовании исходного управляемого вероятностного теста меньшей длины, построенного по известным методологиям [1, 3, 7–13]. Последующие тесты многократного теста строятся на основании исходного как простейшие модификации, не требующие дополнительного анализа и каких-либо вычислительных затрат. В результате многократный управляемый вероятностный тест может быть интерпретирован как единый вероятностный и использован для периодического тестирования в приложениях с ограничением временного ресурса на процедуры тестирования.

1. Анализ управляемых вероятностных тестов

Под *управляемыми вероятностными тестами* в дальнейшем будем понимать вероятностные тестовые последовательности, в которых очередной тестовый набор формируется с учетом ранее сгенерированных наборов и которые соответствуют следующему определению [3].

О п р е д е л е н и е 1. Управляемым вероятностным тестом $CRT=\{T_0, T_1, T_2, \ldots, T_{q-1}\}$ является тест, состоящий из m-разрядных, сгенерированных случайным образом тестовых наборов $T_i = t_{i,m-1}\, t_{i,m-2} \ldots t_{i,2}t_{i,1}t_{i,0}$, где $t_{i,l} \in \{0,1\}$, $i \in \{0, 1, 2, \ldots, q-1\}$, таких, что очередной тестовый набор T_i удовлетворяет некоторым критериям, численные значения которых получаются на основании предыдущих тестовых наборов $T_0, T_1, T_2, \ldots, T_{i-1}$.

Ключевой особенностью управляемого генерирования вероятностных тестовых наборов является информация, которая извлекается в виде некоторых характеристик (метрик) из ранее сгенерированных тестовых наборов и используется для формирования очередного тестового набора [7–13].

Основная идея управляемых вероятностных тестов заключается в том, что очередной тестовый набор T_i генерируемого теста формируется в терминах предварительно определенных и обоснованных характеристик (мер) максимально отличным от ранее сгенерированных наборов. В данном случае принимается гипотеза, что для двух тестовых наборов, имеющих минимальное различие, количество обнаруживаемых неисправностей (ошибок) будет минимальным и, наоборот, для максимально различных тестовых наборов обнаруживающая способность будет максимальной [1, 3, 7–13]. В качестве меры отличия тестового набора T_i от предыдущих наборов $T_0, T_1, T_2, \ldots, T_{i-1}$ чаще всего используются расстояние Хемминга и расстояние Евклида [3, 7–13]. Данные характеристики определяются для двоичных тестовых наборов T_i и T_j, для которых расстояние Хемминга $HD(T_i,T_j)$ вычисляется как вес $w(T_i \oplus T_j)$ вектора $T_i \oplus T_j$ согласно соотношению

$$HD(T_i, T_j) = w(T_i \oplus T_j) = \sum_{l=0}^{m-1} (t_{i,l} \oplus t_{j,l}). \quad (1)$$

Отметим, что для двоичного случая $\min HD(T_i,T_j) = 0$ при $T_i = T_j$, а $\max HD(T_i,T_j) = m$ достигается при $T_j = \overline{T_i}$. Расстояние Евклида $ED(T_i,T_j)$ определяется в соответствии с выражением

$$ED(T_i,T_j) = \sqrt{\sum_{l=0}^{m-1} (t_{i,l} - t_{j,l})^2} = \sqrt{\sum_{l=0}^{m-1} (t_{i,l} \oplus t_{j,l})} = \sqrt{HD(T_i,T_j)}. \quad (2)$$

Очевидно, что $\min ED(T_i,T_j) = 0$ при $T_i=T_j$, а $\max ED(T_i,T_j) = \sqrt{m}$ достигается при $T_j=\overline{T_i}$.

При формировании набора T_i, когда $i > 2$, применяются суммарные значения расстояний для T_i по отношению к предыдущим наборам $T_0, T_1, T_2, \ldots, T_{i-1}$ [3]. Тогда для очередного набора T_i суммарное значение расстояний (1) и (2) относительно $T_0, T_1, T_2, \ldots, T_{i-1}$ вычисляется как

$$THD(T_i) = \sum_{j=0}^{i-1} HD(T_i, T_j), \quad TED(T_i) = \sum_{j=0}^{i-1} ED(T_i, T_j). \quad (3)$$

Значения $THD(T_i)$ и $TED(T_i)$ представляют собой суммарное расстояние Хемминга и суммарное расстояние Евклида [3]. Новый тестовый набор T_i согласно рассмотренным методам формирования управляемых вероятностных тестов выбирается таким образом, чтобы метрики различия (3) принимали максимальное значение [1, 3]. Отметим, что метрики различия (3) характеризуются заметной вычислительной сложностью, которая возрастает с ростом индекса i тестового набора T_i. Кроме того, следует отметить возрастание вычислительной сложности за счет увеличения количества кандидатов в тесты с ростом индекса i для поиска тестового набора, удовлетворяющего пороговым значениям метрик различия [1, 3, 7, 8, 10–14]. Это в основном связано с уменьшением пространства поиска нового тестового набора T_i, которое сокращается за счет предыдущих процедур поиска предшествующих тестовых наборов $T_0, T_1, T_2, \ldots, T_{i-1}$.

Основываясь на методологии однократных управляемых вероятностных тестов, дадим определение многократного управляемого теста.

О п р е д е л е н и е 2. Многократным управляемым вероятностным тестом $MCRT_r$ является множество, состоящее из r однократных управляемых вероятностных тестов CRT_0, CRT_1, CRT_2,…, CRT_{r-1}, каждый из которых включает q тестовых наборов, где CRT_0 удовлетворяет определению 1, а последующие тесты CRT_j, $j \in \{1, 2, 3, ..., r-1\}$, формируются согласно некоторым алгоритмам таким образом, чтобы эти тесты удовлетворяли определенному критерию либо критериям, полученным на основании предыдущих тестов CRT_0, CRT_1, CRT_2, …, CRT_{j-1} и теста CRT_j.

По аналогии с (1) и (2) рассмотрим расстояние Хемминга и расстояние Евклида для двух тестов CRT_k и CRT_l. Отметим, что расстояние Хемминга $HD(CRT_k, CRT_l)$ (4), которое равняется числу несовпадающих компонентов $T_{k,i}$ и $T_{l,i}$ исходного теста CRT_k и формируемого CRT_l, может рассматриваться как необходимое условие, которому должен удовлетворять тест CRT_l:

$$HD(CRT_k, CRT_l) = \sum_{i=0}^{q-1} f(T_{k,i}, T_{l,i}), \quad f(T_{k,i}, T_{l,i}) = \begin{cases} 1, \text{если } T_{k,i} \neq T_{l,i}; \\ 0, \text{если } T_{k,i} = T_{l,i}. \end{cases} \tag{4}$$

Очевидно, что с точки зрения максимального различия требованием, которому должны соответствовать CRT_k и CRT_l, является отсутствие у них совпадающих компонентов $T_{k,i}$ и $T_{l,i}$, что эквивалентно выполнению неравенства $T_{l,i} \neq T_{k,i}$, $i \in \{0, 1, 2, ..., q-1\}$.

Расстояние Евклида для CRT_k и CRT_l определяется как

$$ED(CRT_k, CRT_l) = \sqrt{\sum_{i=0}^{q-1} (T_{i,k} - T_{i,l})^2}. \tag{5}$$

Подобно суммарным значениям расстояний $THD(T_i)$ и $TED(T_i)$ (3) для тестового набора T_i по отношению к предыдущим наборам T_0, T_1, T_2, …, T_{i-1} управляемого вероятностного теста CRT введем аналогичную меру расстояния для многократных управляемых вероятностных тестов CRT_0, CRT_1, CRT_2, …, CRT_{r-1}:

$$THD(CRT_j) = \sum_{i=0}^{j-1} HD(CRT_i, CRT_j); \quad TED(CRT_j) = \sum_{i=0}^{j-1} ED(CRT_i, CRT_j). \tag{6}$$

Пример 1. В качестве примера теста, формирующего p-ичные ($p = 2^4$) данные, используем управляемый вероятностный тест CRT_0= {0010(2), 0111(7), 1101(13), 0100(4), 1001(9)}, состоящий из $q = 5$ тестовых наборов, разрядность m каждого из которых равняется 4. Здесь в скобках указаны десятичные значения тестовых данных. Отметим, что тест CRT_0 может быть получен на основании одного из известных методов построения подобных тестов [1, 7–13]. Для формирования теста CRT_1 будем использовать исходный тест CRT_0 и, например, операцию инвертирования для получения наборов теста $T_{1,i}$ на основании $T_{0,i}$. Так, инвертируя только младший бит тестовых наборов теста CRT_0, получим управляемый вероятностный тест CRT_1 = {0011(3), 0110(6), 1100(12), 0101(5), 1000(8)}. Значения покомпонентных расстояний для тестов CRT_0 и CRT_1 приведены в табл. 1.

Таблица 1

Значения покомпонентных расстояний для CRT_0 и CRT_1

i	$T_{0,i}=t_{0,3}t_{0,2}t_{0,1}t_{0,0}$	$T_{1,i}=t_{1,3}t_{1,2}t_{1,1}t_{1,0}$	$HD(T_{0,i}, T_{1,i})$	$(T_{0,i}-T_{1,i})^2$
0	0010	0011	1	1
1	0111	0110	1	1
2	1101	1100	1	1
3	0100	0101	1	1
4	1001	1000	1	1

Так как $HD(CRT_k, CRT_l)$ равняется числу несовпадающих компонентов, то на основании (4) $HD(CRT_0, CRT_1) = 5$, т. е. принимает минимально возможное значение. Расстояние Евклида определяется в соответствии с (5) как $ED(CRT_0, CRT_1) = (1^2+1^2+1^2+1^2+1^2)^{1/2}=2{,}23$. Совместное применение двух тестов CRT_0 и CRT_1 для формирования входных тестовых данных показывает неравномерность покрытия входного пространства p-ичных ($p = 2^4$) данных (табл. 2).

Таблица 2

Оценка покрывающей способности тестов CRT_0 и CRT_1

Входные данные	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CRT_0			+		+			+		+				+		
CRT_1				+		+	+		+				+			
CRT_0&CRT_1			+	+	+	+	+	+	+	+			+	+		

Используя тот же исходный управляемый вероятностный тест CRT_0 = {0010 (2), 0111 (7), 1101 (13), 0100 (4), 1001 (9)} и применив ту же операцию отрицания, но уже старшего разряда тестовых наборов CRT_0, получим CRT_2 = {1010 (10), 1111 (15), 0101 (5), 1100 (12), 0001 (1)}.

На основании (4), так же как и в предыдущем случае, $HD(CRT_0, CRT_2) = 5$, т. е. принимает минимально возможное значение. Расстояние Евклида определяется в соответствии с (5) как $ED(CRT_0, CRT_2) = (8^2+8^2+8^2+8^2+8^2)^{1/2} = 17{,}88$ (табл. 3). Совместное применение двух тестов CRT_0 и CRT_2 для формирования входных тестовых данных показывает заметно лучшую равномерность покрытия входного пространства p-ичных ($p = 2^4$) данных (табл. 4).

Таблица 3

Значения покомпонентных расстояний для CRT_0 и CRT_2

i	$T_{0,i}=t_{0,3}t_{0,2}t_{0,1}t_{0,0}$	$T_{1,i}=t_{1,3}t_{1,2}t_{1,1}t_{1,0}$	$HD(T_{0,i}, T_{1,i})$	$(T_{0,i}-T_{1,i})^2$
0	0010	1010	1	8
1	0111	1111	1	8
2	1101	0101	1	8
3	0100	1100	1	8
4	1001	0001	1	8

Таблица 4

Оценка покрывающей способности тестов CRT_0 и CRT_2

Входные данные	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CRT_0			+		+			+		+				+		
CRT_2		+				+					+		+			+
CRT_0&CRT_2		+	+		+	+		+		+	+		+	+		+

Сравнивая примеры тестов CRT_1 и CRT_2, полученных на основании одного и того же исходного теста CRT_0, можно отметить, что во втором случае (CRT_2) достигается существенно большая равномерность покрытия входного пространства тестовыми данными, формируемыми тестами CRT_0 и CRT_2, по сравнению с CRT_0 и CRT_1 (см. табл. 4 и табл. 2).

Отметим, что равномерность входных данных является ключевым критерием реализации классических алгоритмов метода Монте-Карло [22]. В рамках метода Монте-Карло для достижения меньших вычислительных погрешностей и более быстрой сходимости (меньшей вычислительной сложности) на практике часто используют последовательности неслучайных данных, называемых квазислучайными [4, 22]. Именно такие последовательности используются на практике для различных задач так называемого метода квази-Монте-Карло, что позволяет достичь меньших вычислительных погрешностей и меньшей вычислительной сложности для различных прикладных задач [22]. В случае управляемых вероятностных тестов также формируются не случайные, а квазислучайные тестовые наборы [4, 22, 23].

В большом количестве работ по методам квази-Монте-Карло дано определение степени равномерности данных и введены численные статистические метрики для ее оценки [3, 4, 22, 23]. В подавляющем числе случаев в качестве основной характеристики равномерности используется значение дискрепанса (discrepancy) [22]. Последовательности с малым дискрепансом называют $ЛП_\tau$-последовательностями, что интерпретируется как «*любой последовательный* участок хорошо распределен» (более равномерно по сравнению с псевдослучайными последовательностями) [4, 22, 23]. Численные оценки дискрепанса требуют сложных вычислений и неоднозначной интерпретации, поэтому в области построения управляемых вероятностных тестов степень равномерности данных (тестовых наборов), т. е. их качество, принято оценивать более простыми метриками, такими как расстояние Евклида либо расстояние Хемминга. Существуют различные модификации дискрепанса, в том числе и характеристика, определенная в [3] и представленная в настоящей статье (см. разд. 4, определение 3) как интегральная мера оценки эффективности теста. Так же, как и дискрепанс, данная характеристика имеет значительную вычислительную сложность, возрастающую экспоненциально в зависимости от размера тестового набора [3]. Применение дискрепанса или его модификаций для построения управляемых вероятностных тестов в силу чрезвычайно большой вычислительной сложности по сравнению с расстоянием Евклида весьма затруднительно.

Анализируя тесты CRT_1 и CRT_2, сделаем вывод, что формальным признаком степени равномерности может служить расстояние Евклида. Действительно, тест CRT_1 находится на расстоянии Евклида $ED(CRT_0, CRT_1) = 2{,}23$ от исходного теста CRT_0, а тест CRT_2 – на расстоянии $ED(CRT_0, CRT_2) = 17{,}88$. Приведенный пример позволяет предположить, что чем больше расстояние Евклида между двумя вероятностными тестами, тем больше равномерность формируемых ими входных данных и соответственно больше эффективность процедуры тестирования [4].

2. Метод генерирования многократных управляемых вероятностных тестов

В качестве основной операции, используемой для формирования многократных вероятностных тестов, применим операцию отрицания, что позволит обеспечить минимальную вычислительную сложность при получении многократных вероятностных тестов $MCRT_r$. Действительно, все последующие тесты CRT_1, CRT_2, …, CRT_{r-1} могут быть легко сформированы на базе CRT_0 путем инвертирования определенных разрядов его тестовых наборов. Примером процедуры генерирования подобных тестов являются тесты CRT_1 = {0011, 0110, 1100, 0101, 1000} и CRT_2= {1010, 1111, 0101, 1100, 0001}, полученные на основании CRT_0 = {0010, 0111, 1101, 0100, 1001} путем инвертирования только младшего разряда тестовых наборов CRT_0 в случае CRT_1 и только старшего разряда в случае CRT_2.

В качестве алгоритма формирования многократных тестов используем хорошо зарекомендовавший себя метод, основанный на применении масок в виде двоичного вектора $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0 \neq 0\,0\,\ldots\,0\,0$, единичные значения которого определяют наличие инверсий разрядов тестовых наборов исходного базового теста CRT_k по отношению к формируемому новому тесту CRT_l [24].

Предположив, что исходный тест CRT_k состоит из тестовых наборов $T_{k,i} = t_{k,m-1}t_{k,m-2}\ldots t_{k,2}t_{k,1}t_{k,0}$, где $t_{k,j}\in\{0, 1\}$ для $j\in\{0, 1, 2, \ldots, m-1\}$, выражение для наборов $T_{l,i} \neq T_{k,i}$ теста CRT_l будет иметь вид [24]

$$T_{l,i} = t_{k,m-1}^{\lambda_{m-1}} t_{k,m-2}^{\lambda_{m-2}} \ldots t_{k,1}^{\lambda_1} t_{k,0}^{\lambda_0} = (\lambda_{m-1} \oplus t_{k,m-1})(\lambda_{m-2} \oplus t_{k,m-2})\ldots(\lambda_1 \oplus t_{k,1})\,(\lambda_0 \oplus t_{k,0}), \qquad (7)$$

где при λ_j=1 отрицание над $t_{k,j}$ присутствует, а при λ_j=0 отсутствует. Отметим, что в качестве исходного кода $T_{k,i}$ может выступать любая m-разрядная двоичная комбинация. Отличие $T_{l,i}$ от кода $T_{k,i}$ определяется двоичным вектором $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$.

Отметим, что максимальное расстояние Хемминга $HD(CRT_k, CRT_l)$, которое равняется числу несовпадающих компонентов $T_{k,i}$ и $T_{l,i}$ исходного теста CRT_k и формируемого CRT_l, может рассматриваться как необходимое условие, которому должен удовлетворять тест CRT_l. Очевидно,

что требованием, которому должны соответствовать CRT_k и CRT_l, является отсутствие у них совпадающих компонентов $T_{k,i}$ и $T_{l,i}$, что эквивалентно выполнению неравенства $T_{l,i} \neq T_{k,i}$, $i \in \{0, 1, 2, ..., q-1\}$, и обеспечивает равенство $HD(CRT_k, CRT_l) = q$. В рамках предложенного алгоритма (7) выполнение неравенства $T_{l,i} \neq T_{k,i}$ достигается с помощью неравенства $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0 \neq 0\,0\,\ldots\,0\,0$. Формально это условие максимизирует расстояние Хемминга $HD(CRT_k, CRT_l)$, которое в этом случае равняется q.

Основой для вычисления характеристик различия, приведенных в разд. 2, является соотношение пар тестовых наборов $T_{k,i}$ и $T_{l,i}$, $i \in \{0, 1, 2, ..., q-1\}$, для двух управляемых вероятностных тестов: исходного CRT_k и формируемого CRT_l. Чем более различными (несовпадающими) являются коды наборов $T_{k,i}$ и $T_{l,i}$, тем, очевидно, более эффективным будет использование тестов CRT_k и CRT_l при реализации многократного теста $MCRT_r$, состоящего из r однократных тестов.

Для произвольной пары тестовых наборов $T_{k,i}$ и $T_{l,i}$ из двух управляемых вероятностных тестов CRT_k и CRT_l справедлива следующая теорема [21].

Теорема 1. *Значение* $T_{k,i}-T_{l,i}$ *для* $T_{k,i} = t_{k,m-1}t_{k,m-2}\ldots t_{k,1}t_{k,0}$, *где* $t_{k,j} \in \{0,1\}$, *при* $j \in \{0, 1, 2, \ldots, m-1\}$ и $T_{l,i} = t_{k,m-1}^{\lambda_{m-1}} t_{k,m-2}^{\lambda_{m-2}} \ldots t_{k,1}^{\lambda_1} t_{k,0}^{\lambda_0} = (\lambda_{m-1} \oplus t_{k,m-1})\,(\lambda_{m-2} \oplus t_{k,m-2})\,\ldots\,(\lambda_1 \oplus t_{k,1})\,(\lambda_0 \oplus t_{k,0})$, *где* g *значений* $\lambda_\alpha, \lambda_\beta, \ldots, \lambda_\chi, \lambda_\delta$, $(\alpha > \beta > \ldots > \chi > \delta)$ *вектора масок* $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$ *равняются* 1, *а остальные* $m-g$ *значения* λ_k *для* $k \neq \alpha \neq \beta \neq \chi \neq \ldots \neq \delta$, *где* $k, \alpha, \beta, \chi, \ldots, \delta \in \{0, 1, 2, \ldots, m-1\}$, *равняются* 0, *вычисляется по формуле*

$$T_{k,i} - T_{l,i} = \sum_{c \in \{\alpha,\beta,\ldots,\chi,\delta\}} (t_{k,c} - |t_{k,c} - 1|)2^c. \tag{8}$$

Пример 2. Для тестовых наборов $T_{0,2} = 1101$ и $T_{1,2} = 0101$, приведенных в тестах CRT_0 и CRT_2 примера 1, учитывая, что $\lambda_3\lambda_2\lambda_1\lambda_0 = 1\,0\,0\,0$, получим, что $T_{0,2} - T_{1,2} = (t_{0,3} - |t_{0,3} - 1|)2^3 = 2^3 = 8$ (см. табл. 3).

Теорема 1 позволяет сформулировать следствие для случая, когда все тестовые наборы $T_{l,0}, T_{l,1}, T_{l,2}, \ldots, T_{l,q-1}$ теста CRT_l для $q=2^m$ сформированы из тестовых наборов $T_{k,0}, T_{k,1}, T_{k,2}, \ldots, T_{k,q-1}$ на основании соотношения (7). При получении теста CRT_l используется один и тот же двоичный вектор, а для $ED(CRT_k, CRT_l)$ справедливо следующее утверждение [21].

Утверждение 1. *Расстояние Евклида* $ED(CRT_k, CRT_l)$ *для тестов* CRT_k *и* CRT_l, *где* $CRT_k = \{T_{k,0}, T_{k,1}, T_{k,2}, \ldots, T_{k,q-1}\}$ *и включает* $q=2^m$ m*-разрядных, неповторяющихся, сгенерированных случайным образом тестовых наборов* $T_{k,i}$, *а тестовые наборы* $T_{l,i}$ *получены согласно* (7) *на основании вектора отрицаний* $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$, *для которого* g *значений* $\lambda_\alpha, \lambda_\beta, \ldots, \lambda_\chi, \lambda_\delta$ $(\alpha > \beta > \ldots > \chi > \delta)$ *равняются* 1, *вычисляется согласно выражению*

$$ED(CRT_k, CRT_l) = \sqrt{2^{m-g} \sum_{t_{k,\alpha}\ldots t_{k,\chi}t_{k,\delta}=00\ldots00}^{1\ldots11} \left[(t_{k,\alpha} - \overline{t_{k,\alpha}})2^\alpha + \ldots + (t_{k,\chi} - \overline{t_{k,\chi}})2^\chi + (t_{k,\delta} - \overline{t_{k,\delta}})2^\delta\right]^2}. \tag{9}$$

Приведенное соотношение (9) для $ED(CRT_k, CRT_l)$ может быть заметно упрощено на основе следующей теоремы.

Теорема 2. *Расстояние Евклида* $ED(CRT_k, CRT_l)$ *для тестов* CRT_k и CRT_l, *где* $CRT_k = \{T_{k,0}, T_{k,1}, T_{k,2}, \ldots, T_{k,q-1}\}$ *включает* $q = 2^m$ m*-разрядных, неповторяющихся, сгенерированных случайным образом тестовых наборов* $T_{k,i}$, *а тестовые наборы* $T_{l,i}$ *получены согласно* (7) *на основании вектора отрицаний* $\lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$, *для которого* g *значений* $\lambda_\alpha, \lambda_\beta, \ldots, \lambda_\chi, \lambda_\delta$ $(\alpha > \beta > \ldots > \chi > \delta)$ *равняются* 1, *вычисляется как*

$$ED(CRT_k, CRT_l) = \sqrt{2^m \left(2^{2\alpha} + 2^{2\beta} + \ldots + 2^{2\chi} + 2^{2\delta}\right)}. \tag{10}$$

Доказательство. Выражение

$$\left[(t_{k,\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}+(t_{k,\delta}-\overline{t_{k,\delta}})2^{\delta}\right]^2$$

под знаком суммирования в соотношении (9) можно представить в виде двух слагаемых a и b, возведенных в квадрат, т. е. $(a+b)^2$. Например, в качестве слагаемого b рассмотрим компонент с индексами k и δ, тогда $b=(t_{k,\delta}-\overline{t_{k,\delta}})2^{\delta}$. Оставшаяся часть соотношения (9) будет представлять собой слагаемое a. Отметим, что для случая $q = 2^m$ под знаком суммирования будут использованы всевозможные m-разрядные комбинации, соответственно двоичная переменная $t_{k,\delta}$ в половине случаев примет значение 0, а для второй половины – значение 1. Отсюда следует, что слагаемое b будет иметь знак плюс для $t_{k,\delta} = 1$ и знак минус для $t_{k,\delta} = 0$. Тогда

$$\sum_{t_{k,\alpha}t_{k,\beta}\ldots t_{k,\chi}t_{k,\delta}=00\ldots00}^{11\ldots11}\left[(t_{k,\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}+(t_{k,\delta}-\overline{t_{k,\delta}})2^{\delta}\right]^2=$$

$$=\sum_{t_{k,\alpha}t_{k,\beta}\ldots t_{k,\chi}=00\ldots0}^{11\ldots1}\{\left[(t_{k,\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}+2^{\delta}\right]^2+$$

$$+\left[(t_{\mathrm{k},\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}-2^{\delta}\right]^2\}.$$

Используя соотношение $(a + b)^2 + (a - b)^2 = 2(a^2 + b^2)$, получим

$$2\sum_{t_{k,\alpha}t_{k,\beta}\ldots t_{k,\chi}=00\ldots0}^{11\ldots1}\{\left[(t_{k,\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}\right]^2+2^{2\delta}\}=2^{g}2^{2\delta}+$$

$$+2\sum_{t_{k,\alpha}t_{k,\beta}\ldots t_{k,\chi}=00\ldots0}^{11\ldots1}\left[(t_{k,\alpha}-\overline{t_{k,\alpha}})2^{\alpha}+(t_{k,\beta}-\overline{t_{k,\beta}})2^{\beta}+\ldots+(t_{k,\chi}-\overline{t_{k,\chi}})2^{\chi}\right]^2.$$

Таким образом, последовательно рассматривая оставшиеся слагаемые под знаком суммы, окончательно получим

$$ED(CRT_k, CRT_l)=\sqrt{2^m\left(2^{2\alpha}+2^{2\beta}+\ldots+2^{2\chi}+2^{2\delta}\right)},$$

что и требовалось доказать. ■ Для теоремы 2 справедливы следующие следствия.

Следствие 1. *Численное значение $ED(CRT_k, CRT_l)$ в соответствии с теоремой* 2 *для двоичного вектора $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_{i+1}\lambda_i\lambda_{i-1}\ldots\lambda_1\lambda_0 = 00\ldots011\ldots11$ и при использовании всевозможных m-разрядных двоичных комбинаций в тестах CRT_l и CRT_k в соответствии с* (7) *принимает следующий вид:*

$$ED(CRT_k, CRT_l)=\sqrt{2^m\left(2^{2i}+2^{2i-2}+\ldots+2^2+2^0\right)}=\sqrt{\frac{2^m\left(2^{2(i+1)}-1\right)}{3}}. \quad (11)$$

Максимальное значение $ED(CRT_k, CRT_l)$ в соответствии со следствием 1 достигается для двоичного вектора $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0 = 11\ldots11$ и принимает вид

$$ED(CRT_k, CRT_l)=\sqrt{\frac{2^m\left(2^{2m}-1\right)}{3}}. \quad (12)$$

Отметим, что в случае максимального значения $ED(CRT_k, CRT_l)$ второй тест CRT_l представляет собой инверсные значения тестовых наборов исходного теста CRT_k. Например, для случая $m = 3$, когда $CRT_k = t_{k,2}t_{k,1}t_{k,0} = \{111, 110, 101, 100, 011, 010, 001, 000\}$, а $CRT_l = \overline{t_{l.2}}\,\overline{t_{l,1}}\,\overline{t_{l,0}} = \{000, 001, 010, 011, 100, 101, 110, 111\}$ и количество тестовых наборов в CRT_k и CRT_1 равняется $q = 2^m = 2^3 = 8$, из табл. 5 и соотношения (12) следует, что $ED(CRT_k, CRT_l) = \sqrt{168}$. Применив соотношения (5), получим аналогичный результат $ED(CRT_k, CRT_l) = ((7-0)^2+(6-1)^2+ (5-2)^2+ +(4-3)^2+(3-4)^2+(2-5)^2+(1-6)^2+(0-7)^2)^{1/2} = \sqrt{168}$.

Таблица 5

Значения расстояния Евклида для $m = 3$

$CRT_k = t_{k,2}t_{k,1}t_{k,0}$	CRT_l						
	$t_{l,2}t_{l,1}\overline{t_{l,0}}$	$t_{l,2}\overline{t_{l,1}}t_{l,0}$	$t_{l,2}\overline{t_{l,1}}\,\overline{t_{l,0}}$	$\overline{t_{l,2}}t_{l.1}t_{l,0}$	$\overline{t_{l,2}}t_{l,1}\overline{t_{l,0}}$	$\overline{t_{l,2}}\,\overline{t_{l,1}}t_{l,0}$	$\overline{t_{l.2}}\,\overline{t_{l,1}}\,\overline{t_{l,0}}$
$ED(CRT_k, CRT_l)$	$\sqrt{8}$	$\sqrt{32}$	$\sqrt{40}$	$\sqrt{128}$	$\sqrt{136}$	$\sqrt{160}$	$\sqrt{168}$

Следствие 2. *Численное значение* $ED(CRT_k, CRT_l)$ *в соответствии с теоремой* 2 *для двоичного вектора масок* $\lambda_{m-1}\lambda_{m-2} \ldots \lambda_{i+1}\lambda_i\lambda_{i-1} \ldots \lambda_1\lambda_0 = 00\ldots010\ldots00$ *и при использовании всевозможных m-разрядных двоичных комбинаций в тестах* CRT_l *и* CRT_k *в соответствии с* (7) *принимает следующий вид:*

$$ED(CRT_k, CRT_l) = \sqrt{2^{2i+m}}. \tag{13}$$

Минимальное значение $ED(CRT_k, CRT_l)$ в соответствии с теоремой 2 достигается для двоичного вектора масок $\lambda_{m-1}\lambda_{m-2} \ldots \lambda_1\lambda_0 = 00\ldots01$ и равняется $\sqrt{2^m}$. Для соотношения (13) в отличие от (11) и (12) справедливо обобщение для произвольного значения $q \le 2^m$

$$ED(CRT_k, CRT_l) = \sqrt{q2^{2i}}. \tag{14}$$

Действительно, для тестов CRT_0, CRT_1 и CRT_2 согласно (14) получим соответственно $ED(CRT_0, CRT_1) = \sqrt{q2^{2i}} = \sqrt{5\times 2^{2\times 0}} = \sqrt{5}$ и $ED(CRT_0, CRT_2) = \sqrt{q2^{2i}} = \sqrt{5\times 2^{2\times 3}} = \sqrt{320}$.

Отметим, что выражения (10) − (13) справедливы для $q = 2^m$ и могут быть использованы как среднее значение для произвольного значения $q < 2^m$.

3. Многократные управляемые вероятностные тесты

Одним из первых примеров многократных тестов являются многократные маршевые тесты запоминающих устройств [19–21, 24]. При реализации многократных маршевых тестов запоминающих устройств в качестве меры отличия адресных последовательностей использовалось арифметическое расстояние, позволяющее определить набор адресных последовательностей (тестов), имеющих максимальные отличия между собой [21].

Показано, что чем старше инверсный бит адресной последовательности, тем больше арифметическое расстояние и, соответственно, больше полнота покрытия теста [21]. В то же время отмечается, что при равенстве значений арифметического расстояния наблюдается незначительное, но все-таки отличие покрывающей способности теста (его эффективности), которая однозначно зависит от другой метрики, а именно расстояния Евклида $ED(CRT_0, CRT_1)$. Так, при использовании для второго теста CRT_1 двоичного вектора $\lambda_5\lambda_4\lambda_3\lambda_2\lambda_1\lambda_0 = 011111$ суммарное значение полноты покрытия сложных неисправностей равняется 12,07 %. Использование вектора $\lambda_5\lambda_4\lambda_3\lambda_2\lambda_1\lambda_0 = 010000$ характеризуется полнотой покрытия 10,67 %. Для первого и второго случаев арифметическое расстояние принимает одинаковое значение, равное 2^{6+4}, а $ED(CRT_0, CRT_1)$ принимает различные значения: в первом случае 147,73, а во втором – 128,00.

Как видно из полученных в [21] экспериментальных результатов, для случая многократных маршевых тестов метрика $ED(CRT_0, CRT_1)$ позволяет определять вид двоичного вектора $\lambda_{m-1}\lambda_{m-2}\ldots\lambda_1\lambda_0$, при котором достигается максимальная эффективность многократных маршевых тестов запоминающих устройств.

Соотношение (7), применяемое для формирования многократных тестов, основано на использовании двоичных векторов масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$; $\lambda_i \in \{0, 1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$. Так, если для получения теста CRT_1 на основании теста CRT_0 используется вектор $\Lambda(0, 1) = \lambda_{m-1}(0, 1), \lambda_{m-2}(0, 1), \ldots, \lambda_1(0, 1), \lambda_0(0, 1)$, то для получения CRT_2 на основании CRT_0 будет использоваться другой вектор масок $\Lambda(0, 2) = \lambda_{m-1}(0, 2), \lambda_{m-2}(0, 2), \ldots, \lambda_1(0, 2), \lambda_0(0, 2)$. Таким образом, для всех последующих тестов многократного теста $MCRT_r = \{CRT_0, CRT_1, CRT_2, \ldots, CRT_{r-1}\}$ применяется множество масок $\{\Lambda(0, 1), \Lambda(0, 2), \Lambda(0, 3), \ldots, \Lambda(0, r-1)\}$.

Используя свойство $\overline{\overline{\lambda_i}} = \lambda_i$ двоичной операции отрицания, применяемой для метода формирования многократных тестов (7), сформулируем ряд утверждений.

У т в е р ж д е н и е 2. *Если тест CRT_l получен из исходного теста CRT_k на основании двоичного вектора масок $\Lambda(k, l) = \lambda_{m-1}(k, l), \lambda_{m-2}(k, l), \ldots, \lambda_1(k, l), \lambda_0(k, l)$ в соответствии с* (7), *то, используя этот же вектор масок $\Lambda(k, l)$, а в качестве исходного теста CRT_l, на основании того же алгоритма* (7) *будет получен тест CRT_k. Формально это утверждение описывается равенством $\Lambda(k, l) = \Lambda(l, k)$.*

У т в е р ж д е н и е 3. *Если тест CRT_l сформирован из исходного теста CRT_k на основании двоичного вектора масок $\Lambda(k, l) = \lambda_{m-1}(k, l), \lambda_{m-2}(k, l), \ldots, \lambda_1(k, l), \lambda_0(k, l)$, а тест CRT_j получен из CRT_k на основании маски $\Lambda(k, j) = \lambda_{m-1}(k, j), \lambda_{m-2}(k, j), \ldots, \lambda_1(k, j), \lambda_0(k, j)$ в соответствии с* (7), *то тест CRT_j может быть получен на основании теста CRT_l с помощью двоичного вектора масок $\Lambda(j, l) = \Lambda(k, j) \oplus \Lambda(k, l) = \lambda_{m-1}(k, j) \oplus \lambda_{m-1}(k, l), \lambda_{m-2}(k, j) \oplus \lambda_{m-2}(k, l), \ldots, \lambda_1(k, j) \oplus \lambda_1(k, l), \lambda_0(k, j) \oplus \lambda_1(k, l)$.*

Пример 2. Тест CRT_1 в примере 1 получен на основании теста CRT_0 и вектора маски $\Lambda(0, 1) = \lambda_3\lambda_2\lambda_1\lambda_0 = 0\ 0\ 0\ 1$, а тест CRT_2 – на базе CRT_0 с использованием вектора маски $\Lambda(0,2) = \lambda_3\lambda_2\lambda_1\lambda_0 = 1\ 0\ 0\ 0$. Согласно утверждению 2 тест CRT_2 на основании CRT_1 может быть получен в соответствии с (7) при использовании вектора $\Lambda(1, 2) = \lambda_3\lambda_2\lambda_1\lambda_0 = 1 \oplus 0, 0 \oplus 0, 0 \oplus 0, 0 \oplus 1 = 1\ 0\ 0\ 1$. Справедливо и обратное (см. утверждение 2): тест CRT_1 на основании CRT_2 может быть получен при использовании $\Lambda(2,1) = \lambda_3\lambda_2\ldots\lambda_1\lambda_0 = 0 \oplus 1, 0 \oplus 0, 0 \oplus 0, 1 \oplus 0 = 1\ 0\ 0\ 1$.

Следующее утверждение сформулируем в виде теоремы.

Теорема 3. *Если тест CRT_l сформирован из исходного теста CRT_k на основании двоичного вектора масок $\Lambda(k, l)$, а тест CRT_j получен из CRT_k на основании $\Lambda(k, j) > \Lambda(k, l)$ в соответствии с* (7), *то выполняется неравенство $ED(CRT_k, CRT_j) > ED(CRT_k, CRT_l)$.*

Д о к а з а т е л ь с т в о. Двоичный унитарный вектор масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$, $\lambda_i \in \{0, 1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$, применяемый в методе генерирования управляемых вероятностных тестов (7), можно интерпретировать как числовое значение, представленное в позиционном коде как $\Lambda = \lambda_{m-1}\times 2^{m-1} + \lambda_{m-2}\times 2^{m-2} + \ldots + \lambda_1\times 2^1 + \lambda_0\times 2^0$. Тогда $\Lambda(k, l) = \lambda_{m-1}(k, l), \lambda_{m-2}(k, l), \ldots, \lambda_1(k, l), \lambda_0(k, l)$ можно записать в виде числа $\Lambda(k, l) = \lambda_{m-1}(k, l) \times 2^{m-1} + \lambda_{m-2}(k, l) \times 2^{m-2} + \ldots + \lambda_{i-1}(k, l) \times 2^{i-1} + \lambda_i(k, l) \times 2^i + \lambda_{i+1}(k, l) \times 2^{i+1} + \ldots + \lambda_1(k, l) \times 2^1 + \lambda_0(k, l) \times 2^0$, а $\Lambda(k, j) = \lambda_{m-1}(k, j), \lambda_{m-2}(k, j), \ldots, \lambda_1(k, j), \lambda_0(k, j)$ как число $\Lambda(k, j) = \lambda_{m-1}(k, j) \times 2^{m-1} + \lambda_{m-2}(k, j) \times 2^{m-2} + \ldots + \lambda_{i-1}(k, j) \times 2^{i-1} + \lambda_i(k, j) \times 2^i + \lambda_{i+1}(k, j) \times 2^{i+1} + \ldots + \lambda_1(k, j) \times 2^1 + \lambda_0(k, j) \times 2^0$. Отметим, что $\Lambda(k, l)$ используется в (7) для получения CRT_l на базе исходного теста CRT_k, а $\Lambda(k, j)$ – для получения теста CRT_j при использовании того же исходного теста CRT_k. На основании неравенства $\Lambda(k, j) > \Lambda(k, l)$, можно заключить, что всегда существует такое $i \in \{0, 1, 2, \ldots, m-1\}$, для которого $\lambda_i(k, j) = 1$, а $\lambda_i(k, l) = 0$; кроме того, $\lambda_c(k, j) = \lambda_c(k, l)$ для $c \in \{i + 1, i + 2, \ldots, m-1\}$. Остальные разряды $\lambda_e(k, l)$ и $\lambda_e(k, j)$, $e \in \{0, 1, 2, \ldots, i-1\}$, масок $\Lambda(k, l)$ и $\Lambda(k, j)$ принимают произвольные значения. Максимально близкое числовое значение $\Lambda(k, l)$ к значению $\Lambda(k, j)$ достигается для случая, когда $\lambda_e(k, j) = 0$, а $\lambda_e(k, l) = 1$ для всех e. Тогда разность $\Lambda(k, j) - \Lambda(k, l)$ принимает минимальное значение, равное 1.

Примером данного случая могут быть значения масок $\Lambda(k,j) = \lambda_6,\lambda_5,\lambda_4,\lambda_3,\lambda_2,\lambda_1,\lambda_0 = 0101000$ и $\Lambda(k, l) = \lambda_6,\lambda_5,\lambda_4,\lambda_3,\lambda_2,\lambda_1,\lambda_0 = 0100111$, где $i = 3$, так как $\lambda_3(k, j) = 1$, $\lambda_3(k, l) = 0$, $\lambda_c(k, j) = \lambda_c(k, l)$ для $c \in \{4, 5, 6\}$ и $\lambda_e(k, j) = 0$, а $\lambda_e(k, l) = 1$ для $e \in \{0, 1, 2\}$.

Определим расстояния Евклида $ED(CRT_k, CRT_j)$ и $ED(CRT_k, CRT_l)$ для тестов CRT_j и CRT_l по отношению к CRT_k. Рассмотрим случай, когда векторы масок $\Lambda(k, j)$ и $\Lambda(k, l)$, для которых выполняется неравенство $\Lambda(k, j) > \Lambda(k, l)$, принимают максимально близкие численные значения, т. е. когда $\lambda_c(k, j) = \lambda_c(k, l)$ для $c \in \{i+1, i+2, \ldots, m-1\}$, $\lambda_i(k, j) = 1$, $\lambda_i(k, l) = 0$ и $\lambda_i(k, j) = 1$, а $\lambda_i(k, l) = 0$ для $e \in \{0, 1, 2, \ldots, i-1\}$. Предположим, что среди $m - i - 1$ старших разрядов векторов масок $\Lambda(k, j)$ и $\Lambda(k, l)$ только $\lambda_\alpha, \lambda_\beta, \ldots, \lambda_\delta$ ($\alpha > \beta > \ldots > \delta$) принимают единичные значения, тогда согласно (10) получим

$$ED(CRT_k, CRT_j) = \sqrt{2^m\left(2^{2\alpha} + 2^{2\beta} + \ldots + 2^{2\delta} + 2^{2i}\right)},$$

$$ED(CRT_k, CRT_l) = \sqrt{2^m\left(2^{2\alpha} + 2^{2\beta} + \ldots + 2^{2\delta} + \sum_{n=0}^{i-1} 2^{2n}\right)}. \quad (15)$$

Учитывая соотношение $\sum_{n=0}^{i-1} 2^{2n} = \frac{2^{2i}-1}{3} < 2^{2i}$, использованное при получении (11), можно заключить, что для приведенного наихудшего соотношения векторов масок $\Lambda(k, j)$ и $\Lambda(k, l)$, когда $\Lambda(k, j) = \Lambda(k, l) + 1$, выполняется неравенство $ED(CRT_k, CRT_j) > ED(CRT_k, CRT_l)$. В остальных случаях, когда $\Lambda(k, j)$ более чем на единицу превышает численное значение $\Lambda(k, l)$, аналогичным образом показывается, что $ED(CRT_k, CRT_j) > ED(CRT_k, CRT_l)$. ■

Численные значения $ED(CRT_k, CRT_j)$ для $m = 4$ и других m подтверждают результаты теоремы 3. Действительно, чем больше значение $\Lambda(k, l) = \lambda_3\lambda_2\lambda_1\lambda_0 = 1111$, использованное для получения CRT_l согласно (7), тем больше расстояние Евклида $ED(CRT_k,CRT_l) = 36{,}68$.

Важным следствием теоремы 3 является возможность использования в качестве метрики, позволяющей строить многократные управляемые вероятностные тесты с использованием соотношения (7), значений вектора масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$.

Рассмотрим последовательно многократные управляемые вероятностные тесты $MCRT_r$ различной кратности, начиная с двукратных тестов $MCRT_2$, состоящих из CRT_0 и CRT_1, где CRT_1 формируется на основании CRT_0 согласно (7).

В случае двукратных тестов максимальное значение $\Lambda(0,1)$ достигается для двоичного вектора масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0 = 11\ldots11$, представляющего собой максимальное числовое значение m-разрядного двоичного вектора масок. Отметим, что в данном случае второй тест CRT_1 представляет собой инверсные значения тестовых наборов исходного теста CRT_0. Например, для $m = 4$ $\Lambda(0, 1) = 15$ и, соответственно, $ED(CRT_0, CRT_1) = 36{,}68$, что равно максимальному значению.

В случае трехкратного вероятностного теста $MCRT_3$ для получения второго CRT_1 и третьего CRT_2 теста на основании первого теста CRT_0 необходимо использовать оптимальные сочетания двоичных векторов $\Lambda=\lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$, руководствуясь теоремой 3. Используя $\Lambda(0, 1) = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0 = 11\ldots11$ для получения CRT_1 на основании CRT_0, получим максимально возможное расстояние $ED(CRT_0, CRT_1)$ между первым и вторым тестами. Для формирования управляемого теста CRT_2 на основании CRT_0 согласно (7), руководствуясь теоремой 3, необходимо максимизировать расстояние теста CRT_2 от CRT_0 и CRT_1. Это означает, что вектор маски $\Lambda(0, 1) = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$, используемый для получения CRT_1 на основании CRT_0, и векторы масок $\Lambda(0, 2)$ и $\Lambda(1, 2)$, определяющие расстояния CRT_2 от CRT_0 и CRT_1, одновременно принимали максимально возможные значения. Отметим, что согласно утверждению 3 $\Lambda(1, 2) =\Lambda(0, 1) \oplus \Lambda(0, 2)$. С учетом того что $\Lambda(0,1) = 11\ldots11$, возможны два варианта значений для $\Lambda(0, 2)$, а именно $\Lambda(0, 2) = \lambda_{m-1},\lambda_{m-2},\ldots,\lambda_1,\lambda_0 = 01\ldots11$ либо $\Lambda(0, 2) = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0 =$

= 10…00. В первом случае $\Lambda(1, 2) = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0 = 10\ldots00$, а во втором $\Lambda(1, 2) = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0 = 01\ldots11$. Для примера, когда $m = 4$, $\{\Lambda(0, 1), \Lambda(0, 2), \Lambda(1, 2)\} = \{1111, 1000, 0111\} = \{15, 8, 7\}$ и, соответственно, минимальное численное значение элементов множества $\{\Lambda(0, 1), \Lambda(0, 2), \Lambda(1, 2)\}$ принимает максимально возможное значение, равное 0111 (7). Анализ результатов для случая m=4 показывает, что для $\Lambda(0,1) = 1111$ и $\Lambda(0,2) = 1000$ соотношения значений расстояний Евклида $ED(CRT_0, CRT_1) = 36{,}68$, $ED(CRT_0, CRT_2) = 32$ и $ED(CRT_1, CRT_2) = 18{,}33$ обеспечивают максимальную удаленность каждого из тестов CRT_0, CRT_1 и CRT_2 друг от друга, причем минимальная удаленность равна 18,33. При других сочетаниях значений $\Lambda(0, 1)$ и $\Lambda(0, 2)$, кроме $\Lambda(0, 1) = 1111$ и $\Lambda(0, 2) = 0111$, минимальное значение расстояния Евклида всегда будет меньше чем 18,33.

Проведенный анализ для случая $MCRT_3$ позволяет заключить, что оптимальным сочетанием векторов $\Lambda(0, 1)$ и $\Lambda(0, 2)$ для получения CRT_1 и CRT_2 согласно (7) являются два возможных набора (111…111, 011…111) либо (111…111, 100…000).

Для случая четырехкратных вероятностных тестов $MCRT_4$ набор двоичных векторов $\{\Lambda(0, 1), \Lambda(0, 2), \Lambda(0, 3)\}$, обеспечивающих максимальные значения минимального расстояния Евклида $ED(CRT_k, CRT_l)$ между любыми двумя тестами из четырех CRT_0, CRT_1, CRT_2 и CRT_3, имеет вид {111…111, 011…111, 100…000}. Так, для примера, когда $m = 0$, $\{\Lambda(0, 1), \Lambda(0, 2), \Lambda(0, 3), \Lambda(1, 2), \Lambda(1, 3), \Lambda(2, 3)\} = \{1111, 0111, 1000, 1000, 0111, 1111\}$, а соответствующие расстояния Евклида больше значения 18,33 либо равны ему.

Для общего случая многократных управляемых вероятностных тестов $MCRT_r$ оптимальным сочетанием набора двоичных векторов масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$ для тестов CRT_0, CRT_1, CRT_2, …, CRT_r будут векторы, сформированные следующим образом. Первые старшие двоичные значения векторов масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$, состоящие из $\lceil \log_2 r \rceil$ бит вектора, соответствуют двоичному коду индекса i, $i \in \{1, 2, 3, \ldots, r-1\}$, теста CRT_i, а остальные повторяют значение младшего бита кода индекса i. Например, для многократного теста $MCRT_{11}$, включающего вероятностные тесты CRT_0, CRT_1, CRT_2, CRT_3, …, CRT_{10}, их двоичные векторы принимают вид **0000**00…000, **0001**11…111, **0010**00…000, **0011**11…111, …, **1010**00…000. В данном случае $\lceil \log_2 r \rceil = \lceil \log_2 11 \rceil = 4$.

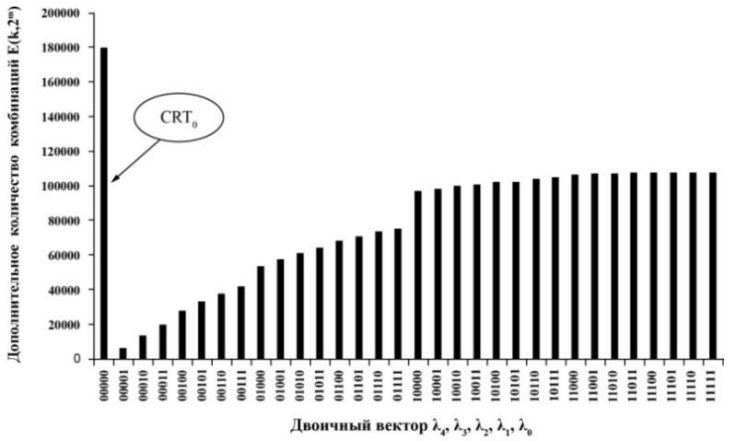
4. Экспериментальный анализ многократных управляемых тестов

В качестве меры эффективности многократных управляемых вероятностных тестов $MCRT_r$ используем метрику $E(k,2^m)$, введенную в [3], для формирования очередных тестовых наборов для однократного управляемого вероятностного теста. В случае многократных тестов подобная характеристика применяется для очередного теста CRT_j и может быть определена следующим образом.

О п р е д е л е н и е 3. Мерой эффективности $E(k, 2^m)$ для очередного управляемого теста CRT_j является дополнительное количество двоичных комбинаций на всевозможных k из 2^m разрядов, генерируемых тестовыми наборами теста CRT_j по отношению к множеству двоичных k из 2^m комбинаций, сгенерированных предыдущими тестами CRT_0, CRT_1, CRT_2, …, CRT_{j-1} многократного теста $MCRT_r$.

Очевидно, что чем больше значение данной метрики, тем более эффективным является очередной управляемый тест CRT_j, который в совокупности с предыдущими тестами позволяет достичь максимальной эффективности. В предыдущих разделах было показано, что для достижения максимальной эффективности многократных управляемых вероятностных тестов $MCRT_r$ необходимо, чтобы расстояние Евклида для теста CRT_j было максимальным по отношению к ранее сформированным тестам CRT_0, CRT_1, CRT_2, …, CRT_{j-1}. Это обеспечивается максимизацией двоичного вектора масок $\Lambda = \lambda_{m-1}, \lambda_{m-2}, \ldots, \lambda_1, \lambda_0$.

В графическом виде данное утверждение подтверждается для запоминающего устройства емкостью 32 бита. На рисунке показано значение метрики $E(k, 2^m)$ для двукратного теста.



Количество дополнительных двоичных комбинаций $E(k, 2^4)$, формируемых тестом CRT_1, на произвольных k=5 из 32 ячеек запоминающего устройства

Как видно из приведенного рисунка, максимальная эффективность двукратного теста, состоящего из CRT_0 и CRT_1, достигается для двоичного вектора $\Lambda(0, 1) = \lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0 = 11111$. Отметим, что в этом случае и расстояние Евклида $ED(CRT_0, CRT_1)$ принимает максимальное значение, равное $\sqrt{1360}$. В то же время для вектора $\Lambda(0, 1) = \lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0 = 00001$ имеем минимальную эффективность двукратного теста, о чем и свидетельствует минимальное значение $ED(CRT_0, CRT_1)$, равное $\sqrt{32}$. Графически видно существенное отличие эффективности двукратного теста для двух различных векторов $\Lambda(0,1) = \lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0 = 01111$ и $\Lambda(0, 1) = \lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0 =$ = 10 000. Этот факт подтверждается значениями расстояния Евклида $ED(CRT_0, CRT_1)$. Действительно, в первом случае при $\Lambda(0, 1) = \lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0 = 01111$ имеем $ED(CRT_0, CRT_1) = \sqrt{336}$, а во втором $ED(CRT_0, CRT_1) = \sqrt{1024}$, что соответствует результатам, приведенным на рисунке.

Заключение

В работе проведен анализ управляемых вероятностных тестов, а также исследованы применяемые численные характеристики, используемые для формирования их тестовых наборов. Показаны основные недостатки классических управляемых вероятностных тестов, подчеркивается значительная вычислительная сложность их построения. Как альтернатива классическим тестам в статье рассмотрена концепция многократных управляемых вероятностных тестов, основанная на применении исходного управляемого вероятностного теста меньших размеров. Показано, что, используя исходный тест, без существенных вычислительных затрат можно сформировать его модификации как последующие тесты многократного теста. Подтверждена применимость расстояния Евклида для целей построения многократных тестов. Эффективность предложенных авторами решений согласуется с экспериментальными результатами, приведенными в заключительной части статьи для случая многократных тестов запоминающих устройств.

Список литературы

1. An Orchestrated Survey on Automated Software Test Case Generation / S. Anand [et al.] // Journal of Systems and Software. – 2014. – Vol. C-39, № 4. – P. 582–586.

2. Malaiya, Y.K. The coverage problem for random testing / Y.K. Malaiya, S. Yang // Proc. of ITC. – Philadelphia, 1984. – P. 237–242.

3. Ярмолик, С.В. Управляемые вероятностные тесты / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2012. – № 10. – С. 142–155.

4. Ярмолик, С.В. Квазислучайное тестирование вычислительных систем / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2013. – № 3(39). – С. 92–103.

5. Chen, T.Y. Quasi-Random Testing / T.Y. Chen, R. Merkel // IEEE Trans. on Reliability.– 2007. – Vol. 56, № 3. – P. 562–568.

6. Shahbazi, A. Centroidal Voronoi Tessellation – a New Approach to Random Testing / A. Shahbazi, A.F. Tappenden, J. Miller // IEEE Trans. on Soft. Eng. – 2013. – Vol. 39, № 2. – P. 163–183.

7. Antirandom Testing: A Distance-Based Approach / S.H. Wu [et al.] // VLSI Design. – 2008.– № 2. – P. 1−9.

8. Fast Antirandom (FAR) Test Generation /A. Mayrhause [et al.] // Proc. Third IEEE Intern. High-Assurance System Eng. Symp. – Boulder, 1998. − P. 262−269.

9. Zhou, Z.Q. Using Coverage Information to Guide Test Case Selection in Adaptive Random Testing // Proc. 34th IEEE Comp. Soft and Applications Conf. – Seoul, 2010. – P. 208−213.

10. Chan, K.P. Good Random Testing / K.P. Chan, T.Y. Chen, D. Towey // Proc. 9th Ada-Europe Intern. Conf. on Reliable Software Technologies (LNCS). – York, 2004. − P. 200−212.

11. Kuo, F.C. An in-depth study of mirror adaptive random testing // Proc. 14th European Conf. on Soft Quality. – Los Alamitos, 2009. − P. 51−58.

12. Shiyi, Xu. Orderly Random Testing for Both Hardware and Software / Xu. Shiyi // Proc. Pacific Rim Intern. Symp. on Dependable Computing. – Shanghai, 2008. − P. 160−167.

13. Tappenden, A.F. A Novel Evolutionary Approach for Adaptive Random Testing / A.F. Tappenden, J. Miller // IEEE Trans. on Reliability. − 2009. − Vol. 58, № 4. − P. 619−632.

14. Ярмолик, С.В. Управляемое случайное тестирование / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2011. – № 1(29). – С. 79–88.

15. Ярмолик, С.В. Итеративные почти псевдоисчерпывающие вероятностные тесты / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2010. – № 2(26). – С. 66–75.

16. Mrozek, I. Iterative Antirandom Testing / I. Mrozek, V.N. Yarmolik // Journal of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, № 3. – P. 251–266.

17. Das, D. Exhaustive and Near-Exhaustive Memory Testing Techniques and their BIST Implementations / D. Das, M.G. Karpovsky // Journal of Electronic Testing. − 1997. – Vol. 10. – P. 215−229.

18. Segall, I. Using binary decision diagrams for combinatorial test design / I. Segall, R. Tzoref-Brill, E. Farchi // Proc. of the Intern. Symp. Software Testing and Analysis (ISSTA'11). – NY, 2011. − P. 254–264.

19. Ярмолик, С.В. Синтез вероятностных тестов с малым числом наборов / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и вычислительная техника. – 2011. – № 3. – С. 19–30.

20. Ярмолик, С.В. Обнаружение кодочувствительных неисправностей запоминающих устройств с многократным использованием маршевых тестов / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2006. – № 1(9). – С. 104–129.

21. Ярмолик, С.В. Многократные неразрушающие маршевые тесты с изменяемыми адресными последовательностями / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2007. – № 4. – С. 126–137.

22. Соболь, И.М. Точки, равномерно заполняющие многомерный куб / И.М. Соболь. – М. : Знание, 1985. – 32 с.

23. Ярмолик, В.Н. Генерирование модифицированных последовательностей Соболя для многократных маршевых тестов ОЗУ / В.Н. Ярмолик, С.В. Ярмолик //Автоматика и вычислительная техника. – 2013. – № 5. – С. 25–33.

24. Ярмолик, С.В. Маршевые тесты для самотестирования ОЗУ / С.В. Ярмолик, А.П. Занкович, А.А. Иванюк. – Минск : Издательский центр БГУ, 2009. – 270 с.

Поступила 09.04.2015

[1]*Белорусский государственный университет информатики и радиоэлектроники, Минск, П. Бровки, 6*
e-mail: yarmolik10ru@yahoo.com, lvn@bsuir.by

[2]*Белостокский технический университет, Польша, Белосток, ул. Вейска, 45А, 15–351*
e-mail: i.mrozek@pb.edu.pl

V.N. Yarmolik, B.A. Levantsevich, I. Mrozek

MULTIPLE CONTROLLED RANDOM TESTS

Controlled Random Tests and methods for their generation have been analyzed and investigated. The similarities of all known controlled random testing approaches are shown. A new method and algorithm for Multiple Controlled Random Tests have been proposed and analyzed.

УДК 658.512.22.011.56:004(076.5)

М.А. Мирзаванд[1], А.В. Бородуля[1], А.Н. Соловьев[2], В.В. Напрасников[1]

ПАРАМЕТРИЧЕСКАЯ КОНЕЧНО-ЭЛЕМЕНТНАЯ МОДЕЛЬ КЕССОННОЙ КОНСТРУКЦИИ

Рассматривается создание параметрической геометрической модели конструкции бокса сухой сварки для ремонта остова нефтедобывающей платформы с использованием специальных встроенных языков систем конечно-элементного моделирования, а также конечно-элементной модели этой конструкции с учетом эксплуатационных нагрузок. Описывается подготовка оптимизационной модели конструкции, выполняются оптимизационные расчеты и даются рекомендации по выбору рациональных параметров конструкции.

Введение

При выполнении работ по проектированию, установке, эксплуатационному ремонту и демонтажу соответствующего оборудования для разведки и добычи углеводородов на морском шельфе возникает ряд задач, решение которых предполагает необходимость разработки специальных моделей, позволяющих ответить на вопросы о работоспособности, долговечности, ремонтопригодности сложных технических систем добычи и транспортировки углеводородов с учетом современных требований экологии. Такие работы ведутся в настоящее время в Исламской Республике Иран, с которой Республика Беларусь поддерживает и развивает отношения долгосрочного партнерства и сотрудничества [1–6].

Целью настоящего исследования является разработка конечно-элементных моделей, позволяющих принимать обоснованные решения по рациональному проектированию компонентов конструкций для ремонта морских платформ с учетом эксплуатационных нагрузок.

1. Основные задачи исследования

Для достижения поставленной цели потребовалось решить следующие основные задачи:

– создать параметрическую геометрическую модель компонентов конструкций для ремонта морских платформ с использованием специальных встроенных языков систем конечно-элементного моделирования;

– исследовать возможности выполнения поиска конструкции бокса сухой сварки для ремонта остова нефтедобывающей платформы с использованием параметрической конечно-элементной модели;

– создать конечно-элементную модель этой конструкции с учетом эксплуатационных нагрузок;

– подготовить оптимизационную модель конструкции;

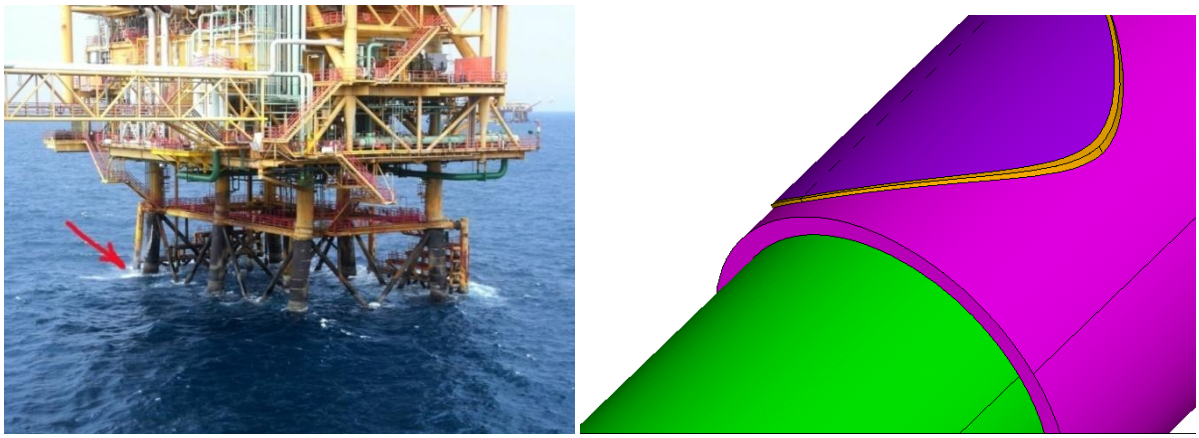
– выполнить оптимизационные расчеты и выработать рекомендации по выбору рациональных параметров конструкции.

2. Моделирование и решение задач

Рассмотрим примеры реализации перечисленных выше задач для ситуации, когда необходимо дать ответ на вопрос о возможности проведения ремонта в подводной части конструкции морской нефтедобывающей платформы [2–4].

Из рис. 1 видно, что конструкция частично ослаблена. Ремонт предполагает замену поврежденной части конструкции (рис. 2) путем вырезания дефектного участка с помощью сварки и приваривания накладки.

Рис. 1. Пример дефекта в подводной части конструкции морской платформы



а) *б)*

Рис. 2. Расположение поврежденной части морской платформы рядом с причальным устройством *(а)* и ее параметрическая геометрическая модель с учетом сварного шва и накладки *(б)*

Максимальное напряжение, возникающее в конструкции с вырезанным дефектным участком при рабочем нагружении, составляет 34,2 МПа и не превышает допустимое для материала конструкции. Таким образом, ремонт, предполагающий вырезание поврежденной части конструкции и приваривание упрочняющей накладки, возможен.

Ремонтные работы для восстановления конструкции предусматривают выполнение сухой сварки. Для этого необходимо спроектировать погружаемый бокс, заполняемый воздухом, внутри которого должна быть расположена часть конструкции, подвергающаяся ремонту. Виды геометрии конструкций двух возможных вариантов таких боксов, подготовленных с использованием соответствующих параметрических моделей, представлены на следующих рисунках.

В первом варианте бокса (рис. 3) предусмотрен каркас из уголков. Основные геометрические параметры модели – это высота уровней обвязки уголками и размеры пластин, указанные на рис. 3. На рис. 4 представлены конечно-элементная расчетная схема и распределение перемещений в материале конструкции при погружении на глубину расположения дефекта.

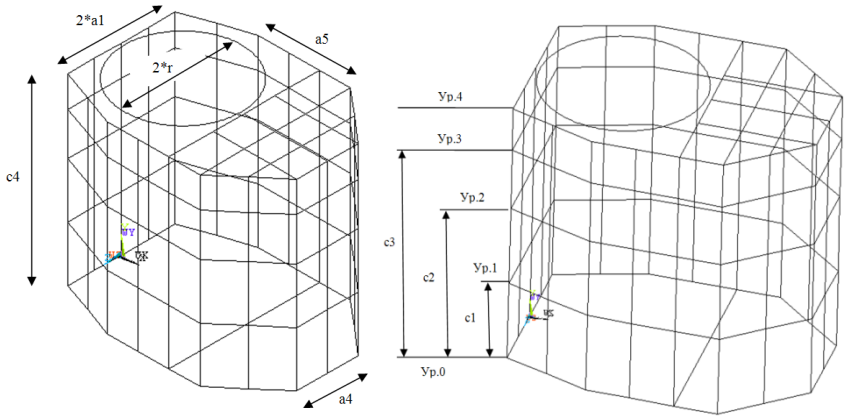
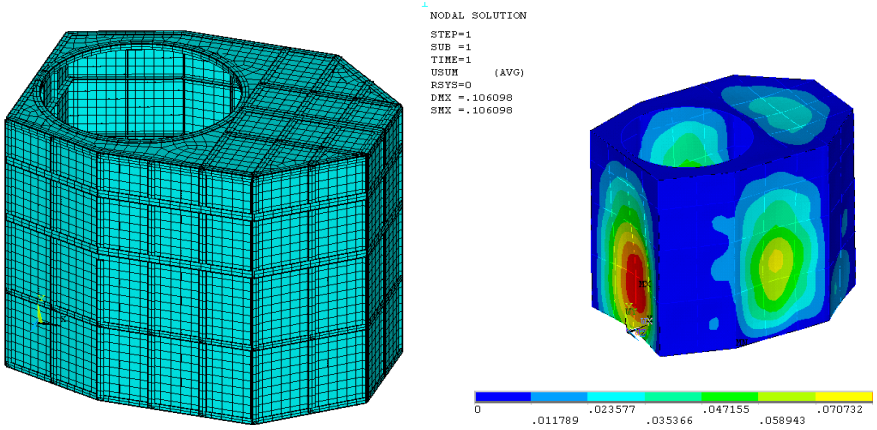


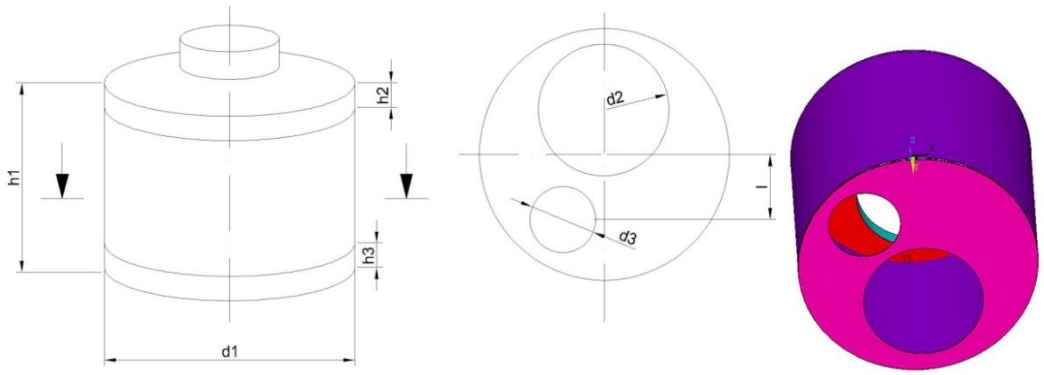
Рис. 3. Основные геометрические параметры модели первого варианта бокса



а) *б)*

Рис. 4. Геометрия конструкции первого варианта кессона *(а)* и суммарные перемещения *(б)* при погружении на рабочую глубину

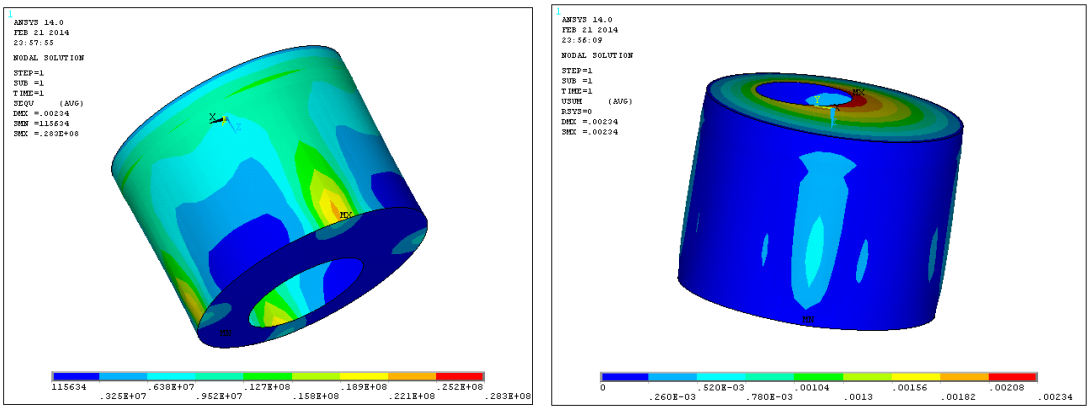
Во втором варианте бокса (рис. 5) каркас отсутствует. Для этой модели параметрами являются геометрические размеры, указанные на рис. 5, и толщина цилиндрической поверхности кессона. Основание и крышка представлены достаточно мощными стальными пластинами, боковая стенка представляет собой относительно тонкий стальной лист. Закрепление кессона осуществляется в нескольких местах. Сверху он плотно прикрепляется к опорной колонне с помощью манжет, снизу прикреплен четырьмя тросами ко дну.



а) *б)*

Рис. 5. Геометрия конструкции второго варианта кессона *(а)* и напряжения по Мизесу в материале его конструкции *(б)* при погружении на рабочую глубину

Результаты расчета напряжений в материале конструкции по теории прочности Мизеса для варианта со значениями параметров $d1$ = 4 м, $d2$ = 2 м, $d3$ = 1,5 м, $h1$ = 3 м, $h2$ = 0,1 м, $h3$ = 0,1 м и толщиной стенки $h4$ = 0,004 м представлены на рис. 6.



а) *б)*

Рис. 6. Распределение напряжений по Мизесу в материале конструкции кессона при погружении на рабочую глубину *(а)* и суммарных перемещений *(б)*

Анализ напряженно-деформированного состояния кессона показывает, что напряжения, возникающие в материале конструкции, составляют 23 МПа и не превосходят соответствующих допустимых значений для материала конструкции. При этом максимальные суммарные перемещения возникают на нижней пластине в районе водолазного входа и на боковой поверхности и не превышают 0,0046 м.

Таким образом, предложенные модели позволяют выбрать второй вариант кессона [2, 3], поскольку технологически его изготовление значительно проще, чем для первого варианта. Далее встает задача выбора рационального варианта конструкции.

3. Оптимизация

Поставим задачу параметрической оптимизации. Оптимизация позволяет расчетным путем найти наиболее эффективное сочетание параметров изделия прежде, чем начинать изготовление опытных экземпляров.

Общая задача многокритериальной минимизации с m независимыми переменными, n целями, p ограничениями в виде неравенств и q ограничениями в виде равенств [7–9] выглядит следующим образом:

$$\begin{cases} f(x) \to min, \\ g(x) \geq 0, \\ h(x) = 0, \end{cases} \quad (1)$$

где $x = (x_1,\ldots,x_m) \in X$, x – вектор решений (независимых переменных);

X – пространство параметров;

$f(x)^T = [f_1(x),\ldots, f_n(x)]$ – цели;

$g(x)^T = [g_1(x),\ldots, g_p(x)]$ – ограничения в виде неравенств;

$h(x)^T = [h_1(x),\ldots, h_q(x)]$ – ограничения в виде равенств.

В рассматриваемом случае компонентами вектора X являются толщина верхней крышки $h2$ (изменяется от 0,05 до 0,2 м), толщина нижней крышки $h3$ (изменяется от 0,05 до 0,2 м) и толщина боковой стенки $h4$ (изменяется от 0,002 до 0,06 м).

Переменной состояния является напряжение по Мизесу в материале конструкции (не должно превышать 200 МПа, что соответствует второму из ограничений вида (1)), целевой функцией $f(x)$ является объем материала конструкции.

При решении задачи оптимизации в среде ANSYS использовался метод первого порядка, который применяет информацию о производных зависимых переменных относительно переменных проекта. Этот метод очень точен и хорошо решает задачи с большими диапазонами изменения зависимых переменных, однако требует значительных вычислительных ресурсов [10].

Задача формулируется в виде

$$Q(x,q) = \frac{f}{f_0} + \sum_{i=1}^{n} P_x(x_i)\left(\sum_{i=1}^{m_1} P_g(g_i) + \sum_{i=1}^{m_2} P_h(h_i) + \sum_{i=1}^{m_3} P_w(w_i) \right), \quad (2)$$

где Q – безразмерная целевая функция; P_x, P_g, P_h, P_w – штрафы, которые зависят от переменных состояния; f_0 – начальное значение целевой функции.

При использовании метода первого порядка программа преобразует оптимизационную задачу с ограничениями в задачу без ограничений (2), а затем на каждой итерации вычисляет градиент целевой функции по переменным проекта. Для вычисления каждой частной производной программа присваивает небольшое приращение соответствующей переменной проекта, оставляя значения других переменных проекта прежними, и производит расчет конструкции с данным набором параметров.

После вычисления всех частных производных определяется направление поиска экстремума на данной итерации. Следует отметить, что в общем случае поиск осуществляется не в направлении градиента. Для определения направления поиска используется более сложная зависимость. Затем осуществляется линейный поиск экстремума целевой функции по данному направлению.

Пользователь может указать приращения переменных проекта, используемые для вычисления градиента, а также предельное значение шага линейного поиска. Таким образом, каждая итерация разделяется на набор субитераций, который включает поиск направления и вычисление градиента. В связи с этим одна оптимизационная итерация для метода первого порядка включает в себя несколько циклов анализа.

Найденный таким образом экстремум используется в качестве исходной точки для следующей итерации и т. д. Итерации продолжаются до тех пор, пока не будет достигнута сходимость или условия прерывания процесса оптимизации. Задача считается сошедшейся, если текущий, предыдущий и наилучший проекты (наборы параметров) таковы, что выполняется одно из следующих условий:

– разность значений целевой функции между лучшим проектом и текущим проектом меньше погрешности сходимости целевой функции;

– разность значений целевой функции между предыдущим проектом и текущим проектом меньше погрешности сходимости целевой функции.

Результаты использования данного метода применительно к описанной задаче представлены на рис. 7.

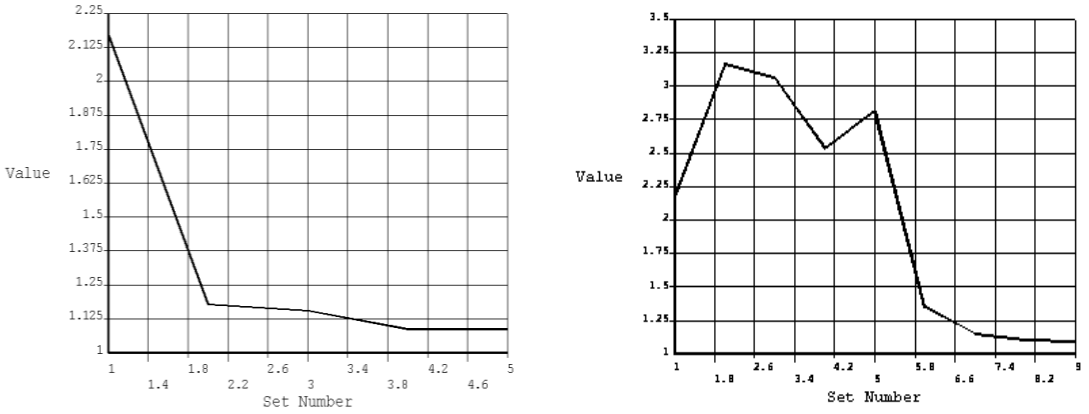
```
LIST OPTIMIZATION SETS FROM SET   1 TO SET   5 AND SHOW
ONLY OPTIMIZATION PARAMETERS. (A "*" SYMBOL IS USED TO
INDICATE THE BEST LISTED SET)

                 SET  1        SET  2        SET  3       *SET  4*
               (FEASIBLE)    (FEASIBLE)    (FEASIBLE)    (FEASIBLE)
STRESS_MAX(SV)    0.23261E+08   0.76816E+08   0.76824E+08   0.78308E+08
H2      (DV)    0.10000       0.50000E-01   0.50000E-01   0.50000E-01
H3      (DV)    0.10000       0.52450E-01   0.50000E-01   0.50000E-01
H4      (DV)    0.40000E-02   0.38252E-02   0.38183E-02   0.20000E-02
TOTAL_VOL(OBJ)    2.1732        1.1785        1.1551        1.0866

                 SET  5
               (FEASIBLE)
STRESS_MAX(SV)    0.78308E+08
H2      (DV)    0.50000E-01
H3      (DV)    0.50000E-01
H4      (DV)    0.20000E-02
TOTAL_VOL(OBJ)    1.0866
```

Рис. 7. Результаты оптимизации (символом ‘*’ отмечен наилучший набор параметров)

Динамика изменения объема материала в зависимости от номера итерации показана на рис. 8, *а*.



а) *б)*

Рис. 8. Динамика изменения объема материала в зависимости от номера итерации: *а)* метод первого порядка; *б)* метод квадратичной аппроксимации (с отсутствием перекрестных членов)

Для того чтобы исключить возможность скатывания в локальный максимум, оптимизационные вычисления были выполнены и другими методами и средствами оптимизации. В частности, использовался метод квадратичной аппроксимации с отсутствием перекрестных членов.

Целевая функция может быть записана в квадратичном виде (с перекрестными членами):

$$\hat{f} = a_0 + \sum_i^n a_i x_i + \sum_i^n \sum_j^n b_{ij} x_i x_j .$$

Норма наименьшей взвешенной квадратичной ошибки для целевой функции определяется формулой

$$E^2 = \sum_{j=1}^{n_d} \varphi^{(j)} \left(f^{(j)} - \hat{f}^{(j)} \right)^2 ,$$

где $\varphi^{(j)}$ – весовой коэффициент, связанный с набором параметров; n_d – число наборов параметров.

Результаты использования метода квадратичной аппроксимации с отсутствием перекрестных членов показаны на рис. 8, *б*. При этом потребовалось девять итераций, минимальный объем конструкции составил 1,0968 м^3.

В дополнение к двум методам оптимизации в программе ANSYS доступны пять различных средств оптимизации, которые применяются для определения области варьирования параметров проекта. Они обеспечивают не оптимизацию целевой функции, а автоматическое получение нескольких наборов параметров проекта при определенном законе изменения переменных проекта. Для использования этих средств не требуется наличие целевой функции, однако переменные проекта должны быть определены.

Сканирование области варьирования параметров создает заданное количество наборов параметров, поочередно варьируя каждую переменную проекта в исходном наборе параметров через весь диапазон ее изменения. Значения других переменных проекта при этом остаются неизменными. При использовании этого средства после 16 итераций минимальный объем конструкции составил 1,6332 м^3.

Наконец, используем средство «случайное варьирование». Сгенерируем в пространстве переменных проекта 20 точек. При этом найденный минимальный объем конструкции составил 1,3092 м^3. Результаты оптимизационных вычислений сведены в таблицу.

Результаты оптимизации	Методы и средства оптимизации			
	Случайное варьирование	Квадратичная аппроксимация (с отсутствием перекрестных членов)	Сканирование области варьирования параметров	Метод первого порядка
Количество итераций	21	9	31	5
$h2$, м $\cdot 10^{-1}$	0,55363	0,50493	0,50000	0,50000
$h3$, м $\cdot 10^{-1}$	0,57355	0,50445	0,50000	0,50000
$h4$, м $\cdot 10^{-2}$	0,45299	0,20190	0,40000	0,20000
Эквивалентное напряжение по Мизесу, МПа	62,27	76,71	76,39	78,23
Объем материала конструкции, $м^3$	1,3092	1,0968	1,1620	1,0866
Время расчета, с	157,344	72,188	242,547	198,391

Заключение

Выполненные исследования позволили получить следующие результаты:

– с использованием языка APDL системы конечно-элементного моделирования ANSYS создана параметрическая геометрическая модель конструкции бокса сухой сварки для ремонта остова нефтедобывающей платформы, а также конечно-элементная модель этой конструкции с учетом эксплуатационных нагрузок;

– представлен спектр конечно-элементных моделей, который позволяет обоснованно выбирать рациональные варианты проектов сложных технических систем добычи и транспортировки углеводородов на морском шельфе;

– с помощью построенных моделей выполнен поиск рациональных параметров водолазного кессона для сухой сварки. При этом удалось снизить объем материала конструкции до 1,0866 $м^3$ против 2,1732 $м^3$ в исходном проектном варианте;

– подготовлена оптимизационная модель конструкции;

– выполнены оптимизационные расчеты и выработаны рекомендации по выбору рациональных параметров конструкции;

– проведена проверка оптимального варианта на устойчивость;

– получена экономия материала (стали).

Список литературы

1. Информационный программно-технический комплекс для дистанционного решения сложных прикладных задач на основе использования суперкомпьютерных систем / В.А. Кочуров [и др.] // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. – 2010. – № 2. – С. 86–96.

2. Построение спектра конечно-элементных моделей для принятия рациональных инженерных решений при ремонте морских платформ / А.В. Бородуля [и др.] // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. – 2013. – № 4. – С. 101–107.

3. Выбор рациональных параметров конструкции каркаса морской платформы / А.В. Бородуля [и др.] // Информатика. – 2014. – № 3(43). – С. 89–95.

4. Mirzavand, M. Construction of the spectrum of finite element models for the rational design of complex technical production systems and transportation of hydrocarbons offshore / M. Mirzavand, H. Izadneshan // Pensee Journal. – 2014. – Vol. 76, № 2. – P. 348–356.

5. Zhu, S. Numerical calculation of forces induced by shortcrested waves on a vertical cylinder of arbitrary cross-section / S. Zhu, G. Moule // Ocean Engineering. – 1994. – Vol. 21, № 7. – P. 645–662.

6. Dong, P. A structural stress definition and numerical implementation for fatigue analysis of welded joints / P. Dong // International Journal of Fatigue. – 2001. – Vol. 23. – P. 865–876.

7. Соболь, И.М. Точки, равномерно заполняющие многомерный куб / И.М. Соболь. – М. : Знание, 1985. – С. 32.

8. Соболь, И.М. Выбор оптимальных параметров в задачах со многими критериями / И.М. Соболь, Р.Б. Статников. – М. : Наука, 1981. – С. 193.

9. Методы оптимизации / под ред. В.С. Зарубин, А.П. Крищенко. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2003. – 440 с.

10. Tang, W.H. Uncertainties in offshore axial pile capacity / W.H. Tang // Foundation Engineering: Current Principles and Practices. – NY, 1989. – P. 833–847.

Поступила 18.03.2015

[1]*Белорусский национальный*
технический университет,
Минск, пр. Независимости, 65
e-mail: n_v_v@tut.by,
mohsen.mirzavand@yahoo.co.uk

[2]*Донской государственный*
технологический университет,
Ростов-на-Дону, пл. Гагарина, 1
e-mail: solovievarc@gmail.com

M.A. Mirzavand[1], A.V. Borodulia[1], A.N. Soloveev[2], V.V. Naprasnikov[1]

PARAMETRIC FINITE ELEMENT MODEL OF A CAISSON CONSTRUCTION

This research is devoted to the creation of a finite element model of a dry welding box for repairing core oil platform taking into account operational loads.

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 681.325

Н.А. Авдеев, П.Н. Бибило

ПРИМЕНЕНИЕ ОЦЕНОК СЛОЖНОСТИ ДИАГРАММ ДВОИЧНОГО ВЫБОРА ПРИ СИНТЕЗЕ ЛОГИЧЕСКИХ СХЕМ

Предлагается формула для оценки площади логической схемы, построенной в заданной библиотеке логических элементов по BDD-представлению (диаграмме двоичного выбора) системы булевых функций. Описываются результаты синтеза комбинационных логических схем по минимизированным BDD-представлениям в библиотеке проектирования заказных КМОП СБИС.

Введение

Синтез логических схем в заданном базисе (библиотеке) элементов традиционно разбивается на два больших этапа: технологически независимую оптимизацию реализуемых систем булевых функций и технологическое отображение – покрытие оптимизированных представлений описаниями библиотечных логических элементов. Решающее влияние на основные параметры (сложность, быстродействие, энергопотребление) логических схем оказывает первый этап. На данном этапе в качестве главных методов оптимизации до недавнего времени использовались методы раздельной и совместной минимизации систем булевых функций в классе дизъюнктивных нормальных форм (ДНФ). В последнее время к ним добавились методы оптимизации многоуровневых представлений на основе разложения Шеннона – диаграмм двоичного выбора (англ. Binary Decision Diagram, BDD) [1, 2].

В настоящей работе предлагается формула для оценки площади логической схемы, построенной по BDD-представлению системы булевых функций. Данная оценка выведена согласно непосредственной реализации формул многоуровневого представления системы булевых функций логическими элементами. Она может применяться для определения максимальной площади проектируемой комбинационной логической схемы в заданной библиотеке КМОП-элементов. Формула проверена экспериментально на потоке практических примеров минимизированных BDD-представлений, позволяет быстро оценивать схемные решения, получаемые синтезаторами логических схем, и различные варианты при выборе лучшей схемы в экспертных системах логического проектирования [3].

1. Диаграмма двоичного выбора для системы булевых функций и оценки ее сложности

Разложением Шеннона полностью определенной булевой функции $f = f(x_1,...,x_n)$ по переменной x_i называется представление ее в виде

$$f = \bar{x}_i f(x_1,...,x_{i-1}, 0, x_{i+1},...,x_n) \vee x_i f(x_1,...,x_{i-1}, 1, x_{i+1},...,x_n). \quad (1)$$

Функции в правой части (1) называются коэффициентами разложения. Они получаются из функции $f(x_1,...,x_n)$ подстановкой вместо переменной x_i констант 0 и 1 соответственно. Каждый из коэффициентов $f_0 = f(x_1,...,x_{i-1}, 0, x_{i+1},...,x_n)$ и $f_1 = f(x_1,...,x_{i-1}, 1, x_{i+1},...,x_n)$ может быть разложен по одной из переменных множества $\{x_1,..., x_{i-1}, x_{i+1},..., x_n\}$. Процесс разложения коэффициентов f_0, f_1 заканчивается, когда все n переменных будут использованы для раз-

ложения. В процессе разложения на каждом шаге некоторые из коэффициентов могут вырождаться до констант, на последнем шаге разложения все полученные коэффициенты являются константами 0, 1. На каждом шаге разложения выполняется сравнение на равенство полученных коэффициентов и оставляется один из нескольких попарно равных коэффициентов. Если же коэффициенты разложения по переменной x_i равны, то переменная x_i называется несущественной переменной и $f = f(x_1,...,x_{i-1},x_{i+1},...,x_n)$.

Под *диаграммой двоичного выбора*, т. е. под BDD, понимается ориентированный ациклический граф, задающий последовательные разложения Шеннона булевой функции $f(x_1,...,x_n)$ по всем ее переменным $x_1,...,x_n$ при заданном порядке (перестановке) переменных, по которым проводятся разложения. BDD одной полностью определенной булевой функции содержит три вида вершин: функциональные вершины, соответствующие разлагаемым функциям либо коэффициентам; вершины-переменные и листовые вершины, соответствующие константам 0, 1. Функциональная вершина, соответствующая функции f, называется корнем. Если коэффициенты разложения (1) равны, то граф BDD упрощается, так как вершина x_i, из которой исходит одна дуга, удаляется из графа. BDD, представляющая систему *m* полностью определенных булевых функций, имеет *m* корневых и две листовые вершины 0, 1, которые обычно дублируются для упрощения изображения графа. Ориентация дуг не показывается, так как при изображении графа BDD все дуги ориентируются сверху вниз.

Совместными BDD (далее BDD), представляющими систему функций F, будем называть такие BDD, которые построены по общей (для всех функций системы) перестановке переменных. В литературе [4] эти BDD называют сокращенными упорядоченными BDD (англ. Reduced Ordered BDD, ROBDD). Методы построения BDD для систем булевых функций хорошо известны [1, 2].

Проиллюстрируем построение BDD на примере системы функций (табл. 1) схемы сумматора, предназначенного для сложения чисел, заданных двоичными кодами: $(a_1,b_1)+(a_2,b_2)=$ $=(c^2,s^2,s^1)$, где a_1,a_2 – старшие разряды складываемых чисел; b_1,b_2 – младшие разряды складываемых чисел. Функциями системы F являются: c^2 – перенос в третий разряд; s^2 – старший разряд суммы; s^1 – младший разряд суммы.

Таблица 1

Таблица истинности функций двухразрядного сумматора

a_1	b_1	a_2	b_2	c^2	s^2	s^1
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

На рис. 1 показана BDD, построенная по общей для всех функций перестановке $< b_2, b_1, a_1, a_2 >$. Многоуровневое представление F_{BDD}, соответствующее BDD, имеет вид

$$c^2 = \bar{b}_2 r^5 \vee b_2 r^1;\ s^2 = \bar{b}_2 r^7 \vee b_2 r^2;\ s^1 = \bar{b}_2 r^3 \vee b_2 r^4;$$

$$r^1 = \bar{b}_1 r^5 \vee b_1 r^6;\ \ r^2 = \bar{b}_1 r^7 \vee b_1 r^8;\ r^3 = b_1;\ r^4 = \bar{b}_1;$$

$$r^5 = a_1 r^9;\ \ r^6 = \bar{a}_1 r^9 \vee a_1;\ r^7 = \bar{a}_1 r^9 \vee a_1 r^{10};\ r^8 = \bar{a}_1 r^{10} \vee a_1 r^9;$$

$$r^9 = a_2;\ r^{10} = \bar{a}_2.$$

Под *сложностью* E^{node} BDD будем понимать число функциональных вершин BDD без учета функциональных вершин, реализующих переменные либо инверсии переменных. На рис. 1 BDD имеет 13 вершин, из них четыре вершины $r^3 = b_1$, $r^4 = \bar{b}_1$, $r^9 = a_2$, $r^{10} = \bar{a}_2$ реализуют переменные (либо инверсии переменных); следовательно, E^{node}= 9 (функциональных вершин). В литературе [4, 5] обычно используется оценка сложности BDD по числу всех функциональных вершин. В работе [6] предлагались эмпирические формулы для оценки числа вершин BDD, реализующей одну булеву функцию, по числу аргументов функции и числу конъюнкций в совершенной ДНФ, представляющей функцию.

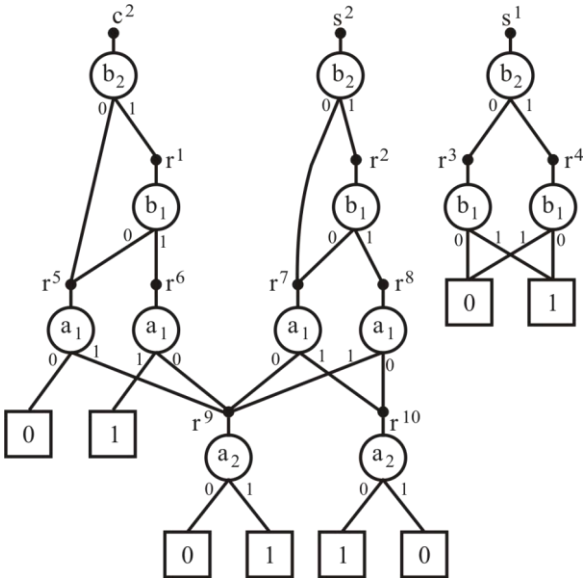


Рис. 1. Диаграмма двоичного выбора

Число операций дизъюнкции $\vee$ в многоуровневом представлении F_{BDD} системы функций F будем обозначать через $E^{\vee}$. Аналогично число операций конъюнкции & в многоуровневом представлении F_{BDD} системы функций F будем обозначать через $E^{\&}$. В примере $E^{\vee} = 8$, $E^{\&} = 15$.

Чтобы подсчитать суммарную площадь $S_{\text{нр}}(F_{BDD})$ всех элементов схемы, реализующей уравнения многоуровневого представления F_{BDD}, воспользуемся простым способом технологического отображения – будем реализовывать каждое полное уравнение вида (1) многоуровневого представления единообразно в виде подсхемы из двух последовательно соединенных логических элементов. Рассмотрим выражение

$$f = AB \vee CD. \qquad (2)$$

Схема, реализующая формулу (2) в рассматриваемом библиотечном базисе КМОП-элементов [7], будет состоять из двух последовательно соединенных элементов: элемента NOAA и инвертора N. Функция элемента NOAA выражается формулой $f_{NOAA} = \overline{AB \vee CD}$, а инвертор реализует функцию отрицания. В схеме выход элемента NOAA подается на вход инвертора N. Площадь S_{NOAA} элемента NOAA равна 485,46, площадь S_N инвертора N равна 223,2 (условных единиц площади).

Чтобы реализовать BDD-представление системы функций в виде схемы из КМОП-элементов, каждую из входных переменных схемы будем инвертировать только один раз и считать, что инверсия каждой из входных переменных в схеме является доступной, тогда каждое выражение (1) можно заменить при синтезе парой элементов NOAA, N: на вход *A* элемента NOAA подается инверсия $\bar{x}_i$ переменной x_i, на вход *B* – коэффициент f_0, на вход *C* – переменная x_i, на вход *D* – коэффициент f_1. Каждая вершина BDD, описываемая логическим выражением, имеющим в своем составе операцию дизъюнкции, будет при таком подходе реализована элементом NOAA, выход которого соединен с входом инвертора N.

Легко подсчитать в многоуровневом представлении системы функций число выражений вида

$$\varphi = x^{\alpha} \,\&\, f_{\beta} = x^{\alpha} f_{\beta}, \qquad (3)$$

где x^{α} – переменная x_i ($\alpha = 0$) либо инверсия $\bar{x}_i$ ($\alpha = 0$); f_{β} – коэффициент f_0 ($\beta = 0$) либо f_1 ($\beta = 1$). Число выражений вида (3) равно $E^{node} - E^{\vee}$. Выражения вида (3) будем реализовывать элементом *A*2 (двухвходовым конъюнктором), площадь которого $S_{A2} = 435{,}24$.

Для построения схемы по формулам многоуровневого BDD-представления исключим (подставим в другие выражения) переменные, являющиеся переименованием переменных либо их инверсий. В рассматриваемом примере $E^{node} = 9$ и получаются девять формул:

$$c^2 = \bar{b}_2 r^5 \vee b_2 r^1;\ s^2 = \bar{b}_2 r^7 \vee b_2 r^2;\ s^1 = \bar{b}_2 b_1 \vee b_2 \bar{b}_1;$$
$$r^1 = \bar{b}_1 r^5 \vee b_1 r^6;\ r^2 = \bar{b}_1 r^7 \vee b_1 r^8;$$
$$r^5 = a_1 a_2;\ r^6 = \bar{a}_1 a_2;\ r^7 = \bar{a}_1 a_2 \vee a_1 \bar{a}_2;\ r^8 = \bar{a}_1 \bar{a}_2 \vee a_1 a_2.$$

Для нахождения оценки $S_{\text{нр}}(F_{BDD})$ площади схемы будем реализовывать выражения вида (3) элементами *A*2, а выражения вида (2) – элементами NOAA. Таким образом, значение $S_{\text{нр}}(F_{BDD})$ площади каскадной схемы будем вычислять по формуле

$$S_{\text{нр}}(F_{BDD}) = [E^{\vee} \times (S_{NOAA} + S_N)] + (S_N \times n) + [S_{A2} \times (E^{node} - E^{\vee})] =$$
$$= 485{,}46\, E^{\vee} + 223{,}2\, n + 435{,}24(E^{node} - E^{\vee}). \qquad (4)$$

Логическая схема, построенная по полученным формулам, иллюстрирует предлагаемый способ нахождения оценки сложности схемы, а не ориентирована на минимизацию сложности сумматора (рис. 2). Применяемая в литературе оценка сложности логической схемы по числу E^{node} вершин BDD (уравнений) не учитывает разницу в аппаратной сложности уравнений вида (2) и (3) и не исключает аппаратную сложность функциональных вершин BDD, помеченных входными переменными.

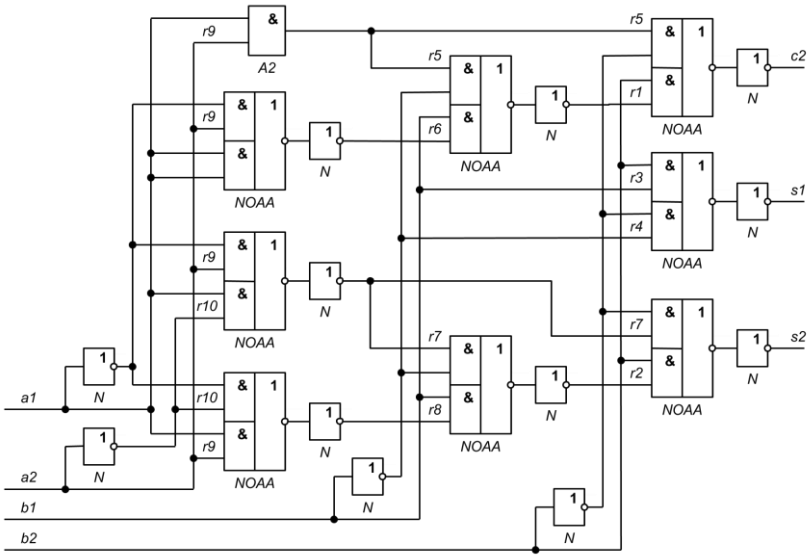


Рис. 2. Логическая схема

2. Эксперимент по схемной реализации BDD-представлений систем булевых функций

Проведенный эксперимент заключался в схемной реализации систем ДНФ булевых функций F логическими схемами из библиотечных логических элементов и состоял из двух этапов:

1. По исходной системе ДНФ строилось оптимизированное BDD-представление функций. Для совместной BDD-минимизации систем булевых функций использовалась программа TIE_BDD [2], реализующая алгоритм минимизации многоуровневых представлений системы булевых функций на основе разложения Шеннона. При проведении экспериментов данная программа строила BDD не более чем по 5000 случайно выбираемым перестановкам переменных и выбирала из рассмотренных вариантов BDD наименьшей сложности.

2. Синтезировались схемы по минимизированным BDD-представлениям систем функций.

Исходными данными явились 60 примеров систем ДНФ полностью определенных булевых функций из набора примеров [8]. В качестве промышленной системы синтеза логических схем во всех экспериментах использовался синтезатор LeonardoSpectrum (версия 2011a.4), при этом BDD-представления конвертировались в соответствующие VHDL-описания многоуровневого представления системы функций. В качестве целевой библиотеки синтеза выступала отечественная библиотека [7] проектирования заказных цифровых КМОП СБИС.

Результирующими данными для каждой из схем явилась площадь $S(F_{BDD})$ логической схемы, реализующей BDD-представление системы F. Площадь схемы, полученной синтезатором LeonardoSpectrum, сравнивалась с площадью $S_{\text{нр}}(F_{BDD})$ схемы, полученной непосредственной реализацией формул многоуровневого представления.

3. Результаты эксперимента

Результаты эксперимента приведены в табл. 2, где приняты следующие обозначения:

n – число аргументов (число входных полюсов схемы);

m – число функций (число выходных полюсов схемы);

k – число общих элементарных конъюнкций, входящих в систему ДНФ булевых функций;

E^{node} – оценка сложности BDD;

$S(F_{BDD})$ – суммарная площадь всех элементов схемы, построенной синтезатором LeonardoSpectrum по многоуровневому BDD-представлению F_{BDD} системы функций F;

$S_{\text{нр}}(F_{BDD})$ – оценка суммарной площади всех элементов схемы по формуле (4);

$\rho_1 = S(F_{BDD}) / S_{\text{нр}}(F_{BDD})$ – отношение площади схемы, полученной синтезатором LeonardoSpectrum, к оценке площади схемы по формуле (4).

Проведенный эксперимент, как и другие подобные эксперименты [3], показал, что схемные реализации различных представлений одной и той же системы булевых функций при синтезе в промышленном синтезаторе LeonardoSpectrum могут иметь различную площадь, так как данный синтезатор чувствителен к форме задания исходных данных. В синтезатор LeonardoSpectrum включены собственные модули оптимизации, поэтому для небольших размерностей задач результаты синтеза обычно не зависят от формы задания функций. Для примеров схем небольших размерностей синтезатор не принимал во внимание результаты предварительной оптимизации в классе BDD-представлений, а осуществлял собственную оптимизацию функций.

По результатам эксперимента можно также сделать вывод о том, что в среднем реальная суммарная площадь элементов схемы на четверть больше, чем площадь, вычисляемая по формуле (4). Однако имеется ряд примеров (LIFE, MLP4 и др.), для которых площадь, вычисленная по формуле (4), оказывается меньше, чем площадь схемы, построенной синтезатором LeonardoSpectrum. В этом случае целесообразно использование компилятивного метода синтеза (см. схему на рис. 2), заключающегося в покрытии уравнений логическими элементами и дополненного процедурой введения повторителей сигналов для удовлетворения нагрузочных способностей логических элементов.

Приведем пример использования оценки (4). Пусть требуется реализовать систему булевых функций для приближения тригонометрической функции синуса $y = \sin(x)$ на интервале $[0,\ \pi/2]$ значений аргумента x. Исходное задание на синтез представляет собой таблица истинности системы функций $F = \{ f^1(x_1, x_2, ..., x_{14}), ..., f^{14}(x_1, x_2, ..., x_{14}) \}$, содержащая 2^{14} строки. Оптимизация заключается в выборе лучшего BDD-представления путем перебора 500 случайно выбранных перестановок переменных.

Таблица 2

Имя схемы	n	m	k	$S(F_{BDD})$	E^{node}	$E^{\vee}$	$S_{\text{нр}}(F_{BDD})$	ρ_1
ADD6	12	7	1092	20 222	58	52	30 533,76	0,66
ADDM4	9	8	512	91 372	187	153	91 082,34	1,00
ADR4	8	5	256	7661	27	21	14 591,7	0,53
ALU1	12	8	19	7109	16	11	10 194,66	0,70
B12	15	9	431	17 002	54	39	28 809,54	0,59
B2	16	17	110	217 464	558	410	267 025,3	0,81
B9	16	5	123	25 104	69	61	36 666,18	0,68
BR1	12	8	34	30 562	76	32	37 363,68	0,82
BR2	12	8	35	22 973	70	33	34 802,46	0,66
CLPL	11	5	20	2929	14	9	9000,54	0,33
CO14	14	1	47	12 996	25	13	14 658,66	0,89
DC2	8	7	58	22 956	60	35	29 657,7	0,77
DIST	8	5	256	69 884	144	123	70 637,22	0,99
EX7	16	5	123	25 104	69	61	36 666,18	0,68
F51M	8	8	256	26 717	62	52	31 381,92	0,85
gary	15	11	442	102 192	301	186	143 696,2	0,71
IN1	16	17	110	217 464	558	410	267 025,3	0,81
IN2	19	10	137	95 669	255	173	123 915,1	0,77
INTB	15	7	664	239 527	615	507	296 482,1	0,81
LIFE	9	1	512	14 977	24	18	13 358,52	**1,12**
LOG8MOD	8	5	47	26 778	60	49	30 360,78	0,88
M1	6	12	32	18 899	55	40	27 286,2	0,69
M181	15	9	430	18 124	45	30	24 440,4	0,74
M2	8	16	96	63 679	127	104	62 283,96	**1,02**

Окончание табл. 2

Имя схемы	n	m	k	$S(F_{BDD})$	E^{node}	$E^{\vee}$	$S_{нр}(F_{BDD})$	ρ_1
M3	8	16	128	62 859	139	101	67 356,18	0,93
M4	8	16	256	84 637	204	161	98 659,98	0,86
MAX1024	10	6	1024	146 430	295	257	143 534,3	**1,02**
MAX46	9	1	46	39 802	72	51	35 907,3	**1,11**
MAX512	9	6	512	73 221	173	138	84 235,68	0,87
MLP4	8	8	256	73 143	134	115	65 883,06	**1,11**
MP2D	14	14	123	18 827	69	45	35 416,26	0,53
NEWAPLA	12	10	17	10 189	21	4	12 019,32	0,85
NEWAPLA1	12	7	10	7650	18	0	10 512,72	0,73
NEWBYTE	5	8	8	5692	19	0	9385,56	0,61
NEWCOND	11	2	31	13 961	27	18	15 110,64	0,92
NEWCPLA1	9	16	38	28 006	80	41	38 887,02	0,72
NEWCPLA2	7	10	19	17 806	58	37	28 664,46	0,62
NEWILL	8	1	8	5312	14	8	8280,72	0,64
NEWTAG	8	1	8	2126	8	6	5568,84	0,38
NEWTPLA	15	5	23	14 419	11	2	8236,08	**1,75**
NEWTPLA1	10	2	4	3800	28	14	15 121,8	0,25
NEWTPLA2	10	4	9	7354	50	18	24 897,96	0,30
P82	5	14	24	19 971	57	30	27 431,28	0,73
RADD	8	5	120	8465	27	21	14 591,7	0,58
RD53	5	3	32	10 055	21	17	11 109,78	0,91
RD73	7	3	147	15 925	41	37	21 265,38	0,75
ROOT	8	5	256	27 381	73	49	36 018,9	0,76
RYY6	16	1	112	4224	15	11	10 652,22	0,40
SEX	9	14	23	13 928	45	24	22 799,88	0,61
soar	83	94	529	172 913	646	402	319 879,1	0,54
SQN	7	3	96	24 329	48	39	24 412,5	0,996
SQR6	6	12	64	27 158	68	50	33 446,52	0,81
SYM10	10	1	837	21 143	36	32	19 507,68	**1,08**
T3	12	8	152	20 116	54	19	27 135,54	0,74
TIAL	14	8	640	313 339	706	613	341 189,1	0,92
vtx1	27	6	110	21 952	85	52	45 633,24	0,48
x9dn	27	7	120	22 342	89	57	47 625,3	0,47
Z4	7	4	128	6992	24	19	12 962,34	0,54
Z5XP1	7	10	128	28 374	65	49	32 313,78	0,88
Z9SYM	9	1	420	18 191	31	27	16 857,18	**1,08**
							Среднее	0,77

Полученная BDD содержит E^{node}= 6017 вершин, а соответствующее многоуровневое представление состоит из 6017 логических выражений, в которых насчитывается $E^{\vee}$ = 5839 операторов дизъюнкции.

Синтез схемы из библиотечных элементов в синтезаторе LeonardoSpectrum занял 6 мин, полученная схема имела площадь 3 535 817 условных единиц. Оценка площади схемы по формуле (4) при значениях E^{node}= 6017, $E^{\vee}$ = 5839, n = 14 проводится следующим образом:

$$S_{нр}(F_{BDD}) = 485{,}46\times5839 + 223{,}2\times14 + 435{,}24\times(6017 - 5839) =$$
$$= 2\,834\,600{,}94 + 3124{,}8 + 77\,472{,}72 = 2\,837\,725{,}74.$$

Площадь согласно оценке оказалась значительно меньше площади схемы, полученной промышленным синтезатором. Однако следует учесть тот факт, что удовлетворение электрических характеристик сигналов (учет нагрузочных способностей элементов) немного снижает этот выигрыш, так как требуется ввести в схему дополнительные элементы – повторители сигналов.

Использование формулы (4) позволяет оценить качество схем, получаемых промышленными синтезаторами и имеющими собственные программно реализованные алгоритмы оптимизации. Проведенные эксперименты показывают, что при реализации функциональных блоков большой размерности целесообразно осуществить предварительную глобальную BDD-оптимизацию и оценить площадь схемы по формуле (4). Не исключены варианты, что в процессе процедуры синтеза, использующей мощную глобальную минимизацию BDD-представлений и простую процедуру технологического отображения, могут быть получены логические схемы меньшей площади, чем схемы, получаемые промышленными синтезаторами. Промышленные синтезаторы ориентируются на схемную реализацию высокоуровневых описаний, представленных на языках VHDL, Verilog [9], и имеют в своем составе собственные алгоритмы локальной минимизации функциональных описаний логических схем.

Заключение

В литературе сложность BDD оценивается по числу вершин BDD, однако предложенная оценка сложности BDD является более практичной и может быть использована для оценки площади логических схем, получаемых промышленными синтезаторами. Как показали эксперименты, при больших размерностях задачи синтеза комбинационной логики не исключены случаи, когда целесообразно реализовать логическую схему непосредственно по минимизированному BDD-представлению.

Список литературы

1. Meinel, C. Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications / C. Meinel, T. Theobald. – Berlin, Heidelberg : Springer-Verlag, 1998. – 267 p.

2. Бибило, П.Н. Алгоритм построения диаграммы двоичного выбора для системы полностью определенных булевых функций / П.Н. Бибило, П.В. Леончик // Управляющие системы и машины. – 2009. – № 6. – С. 42–49.

3. Бибило, П.Н. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний / П.Н. Бибило, В.И. Романов. – Минск : Беларус. навука, 2011. – 279 с.

4. Кнут, Д.Э. Искусство программирования / Д.Э. Кнут. – М. : Вильямс, 2013. – Т. 4, А : Комбинаторные алгоритмы, ч. 1. – 960 с.

5. Ishiura, N. Minimization of Binary Decision Diagrams Based on Exchanges of Variables / N. Ishiura, H. Sawada, S. Yajima // IEEE Intern. Conf. Computer-Aided Design (ICCAD–1991). – USA, 1991. – P. 472–475.

6. Raseen, M. An efficient estimation of the ROBDDs complexity / M. Raseen, P.W. Chandana Prasad, A. Assi // Integration, the VLSI Journal. – 2006. – Vol. 39, № 3. – P. 211–228.

7. Бибило, П.Н. Оценка энергопотребления логических КМОП-схем по их переключательной активности / П.Н. Бибило, Н.А. Кириенко // Микроэлектроника. – 2012. – № 1. – С. 65 – 77.

8. Espresso examples [Electronic resource]. – Mode of access : http://www1.cs.columbia.edu/~cs6861/sis/espresso-examples/ex. – Date of access : 25.03.2015.

9. Поляков, А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры / А.К. Поляков. – М. : СОЛОН-Пресс, 2003. – 320 с.

Поступила 24.04.2015

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: bibilo@newman.bas-net.by

N.A. Avdeev, P.N. Bibilo

EMPLOYING COMPLEXITY ESTIMATES OF BINARY DECISION DIAGRAMS IN THE SYNTHESIS OF LOGICAL CIRCUITS

A formula is suggested to evaluate the area of a logical circuit that is built in a given library of logical elements according to the BDD (Binary Decision Diagram) representation of a system of Boolean functions. The experimental results of synthesis of combinational logical circuits from the minimized BDD representations in the design library of custom CMOS VLSI circuits are described.

УДК 519.7

Ю.В. Поттосин

ЭНЕРГОСБЕРЕГАЮЩЕЕ ПРОТИВОГОНОЧНОЕ КОДИРОВАНИЕ СОСТОЯНИЙ АСИНХРОННОГО АВТОМАТА

Рассматривается задача противогоночного кодирования состояний асинхронного автомата, которая сводится к задаче нахождения минимального взвешенного покрытия. Предлагается метод кодирования состояний, который наряду с устранением опасных состязаний элементов памяти в реализующей схеме обеспечивает минимизацию их числа и минимизацию интенсивности их переключений.

Введение

В настоящее время при проектировании дискретных устройств управления на основе сверхбольших интегральных схем большое внимание уделяется проблеме снижения энергопотребления проектируемой схемы. Это обусловлено стремлением, с одной стороны, увеличить время действия источника энергии в портативных приборах, а с другой – снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Поэтому одним из основных критериев оптимизации при проектировании дискретных устройств является величина потребляемой схемой энергии.

Как отмечено в работах [1, 2], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна интенсивности переключений логических элементов и элементов памяти, что дает возможность частично решить проблему снижения энергопотребления на уровне логического проектирования. В частности, снижения энергопотребления можно добиться при кодировании состояний автомата [3–5]. Очевидно, кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти.

1. Модель поведения асинхронного автомата

Моделью поведения логической схемы с памятью является конечный автомат, представляющий собой пятерку (A, B, Q, Ψ, Φ), где A, B и Q – соответственно множества входных сигналов, выходных сигналов и состояний автомата, а Ψ и Φ – функции $\Psi : A \times Q \to Q$ и $\Phi : A \times Q \to B$, называемые соответственно *функцией переходов* и *функцией выходов*. Для состояний $q_i, q_j \in Q$ и входного сигнала $a \in A$ состояние $q_j = \Psi(a, q_i)$ является тем состоянием, в которое автомат переходит из состояния q_i под воздействием входного сигнала a. Конечный автомат функционирует в дискретном времени, т. е. время разбивается на конечные промежутки, называемые *тактами*, в течение каждого из которых автомат может перейти из состояния в состояние и выдать соответствующий выходной сигнал. Рассматриваемая задача позволяет игнорировать функцию выходов Ф. Поэтому в дальнейшем она не будет упоминаться.

В настоящей работе рассматривается асинхронная реализация конечного автомата, называемая *асинхронным автоматом*, которая в отличие от синхронной реализации не имеет внешнего источника тактирующих сигналов. Переход от такта к такту происходит в момент изменения входного сигнала. При действии любого входного сигнала асинхронный автомат приходит в некоторое устойчивое состояние, из которого он не выходит до конца действия данного сигнала. При этом должно выполняться требование прямого перехода, которое формально выражается следующим образом: если $\Psi(a, q_i) = q_j$ для фиксированного входного сигнала a и некоторых состояний q_i и q_j, то $\Psi(a, q_j) = q_j$.

Задача кодирования состояний автомата заключается в присвоении каждому состоянию определенного булева вектора $(z_1, z_2, \ldots, z_k)$, называемого *кодом состояния*, который соответствует набору состояний двоичных элементов памяти (триггеров) в логической схеме, где каждый переход из состояния в состояние представляется переключением одного или нескольких

триггеров. Естественно, что в реальной электронной схеме такое переключение не может происходить мгновенно и одновременно. Явление неодновременного переключения элементов памяти называется *состязаниями* или *гонками* элементов памяти [6]. Принято называть состязания *неопасными*, если все промежуточные состояния, в которых автомат может оказаться при переходе из одного состояния в другое под воздействием некоторого входного сигнала a, являются неустойчивыми для сигнала a, т. е. при любом порядке переключений элементов памяти автомат из некоторого состояния q_i под воздействием входного сигнала a переходит всегда в состояние $q_j = \Psi(a, q_i)$. Если же при этом автомат может оказаться в некотором устойчивом состоянии q_k, отличном от q_j, то состязания называются *опасными*.

Кодирование состояний, обеспечивающее отсутствие опасных состязаний (гонок), называется *противогоночным*. Естественно, здесь возникает задача минимизации длины кода состояния, приводящая к наименьшему числу элементов памяти в реальной схеме.

Другим критерием оптимизации схемы, как указано выше, является величина потребляемой энергии. Проблеме энергосберегающего кодирования состояний синхронного автомата посвящено довольно много работ, одной из которых является, например, работа [5], где процесс кодирования состояний синхронного автомата представляется как размещение состояний в булевом пространстве внутренних переменных. Асинхронным автоматам давно стало уделяться большое внимание [7–9], и некоторые преимущества в определенных условиях асинхронных схем над синхронными схемами отмечены, например, в работе [7, с. 45]. Задача энергосбережения для асинхронных автоматов также может быть частично сведена к уменьшению переключательной активности элементов схемы. В предлагаемой вниманию работе рассматривается возможность учета энергосбережения при противогоночном кодировании состояний асинхронного автомата.

2. Условия отсутствия опасных состязаний

Существование опасных состязаний для пары переходов $q_i \to q_j$, $q_k \to q_l$ $(q_j \neq q_l)$ при одном и том же входном сигнале a может привести к тому, что автомат вместо перехода в состояние q_j из состояния q_i может оказаться в состоянии q_l, которое также является устойчивым при входном сигнале a. Условие отсутствия опасных состязаний для этой пары можно выразить троичным вектором, в котором компоненты i и j соответствуют состояниям автомата и имеют одно значение (0 или 1), а компоненты k и l – противоположное значение [10]. Остальным компонентам приписывается значение «–». В схеме, реализующей заданный автомат, это условие выполняется триггером, который в процессе одного из переходов рассматриваемой пары хранит состояние 0, а в процессе другого перехода – состояние 1.

Пусть, например, табл. 1 представляет функцию переходов $\Psi(a, q)$ заданного автомата, т. е. является его таблицей переходов. Ее строкам соответствуют состояния автомата, а столбцам – входные сигналы. В клетках таблицы даны значения функции $\Psi(a, q)$ при соответствующих значениях аргументов a и q. Устойчивые состояния для каждого входного сигнала выделены.

Таблица 1

	a_1	a_2	a_3	a_4
q_1	$\boldsymbol{q_1}$	q_2	q_3	$\boldsymbol{q_1}$
q_2	$\boldsymbol{q_2}$	$\boldsymbol{q_2}$	q_8	q_4
q_3	q_1	q_2	$\boldsymbol{q_3}$	q_4
q_4	q_2	q_2	q_5	$\boldsymbol{q_4}$
q_5	$\boldsymbol{q_5}$	$\boldsymbol{q_5}$	$\boldsymbol{q_5}$	q_6
q_6	$\boldsymbol{q_6}$	$\boldsymbol{q_6}$	q_3	$\boldsymbol{q_6}$
q_7	$\boldsymbol{q_7}$	q_8	$\boldsymbol{q_7}$	$\boldsymbol{q_7}$
q_8	q_7	$\boldsymbol{q_8}$	$\boldsymbol{q_8}$	q_1

Условие отсутствия опасных состязаний для пары переходов $q_3 \to q_1$, $q_4 \to q_2$ при входном сигнале a_1 выражается вектором (0 1 0 1 – –) либо покомпонентной инверсией этого вектора (1 0 1 0 – –).

На множестве векторов, представляющих условия отсутствия опасных состязаний, имеется отношение импликации: троичный вектор *a имплицирует* троичный вектор *b*, если *b* получается из *a* заменой некоторых нулей или единиц значением «–» и, возможно, инвертированием полученного результата. Например, вектор (1 0 – – 1 0 1) имплицирует вектор (1 0 – – – 0 1), а также вектор (0 1 – – – 1 –). Смысл этого отношения в том, что условие, представленное вектором *b*, автоматически выполняется при соблюдении условия, представленного вектором *a*.

Все условия отсутствия опасных состязаний в виде описанных векторов составляют троичную матрицу, в которой отсутствуют имплицируемые строки. Эта матрица называется *матрицей условий* [10]. Для автомата, таблицей переходов которого является табл. 1, при рассмотрении пар переходов число строк этой матрицы равно 31.

Чтобы избежать громоздких вычислений, для ускорения процесса противогоночного кодирования состояний асинхронного автомата вместо пар переходов рассматривают пары так называемых *K-множеств* [6], каждое из которых является множеством состояний асинхронного автомата, переходы из которых при некотором фиксированном входном сигнале ведут в одно и то же устойчивое состояние (также принадлежащее данному множеству). Например, для входного сигнала a_2 автомата, поведение которого задает табл. 1, *K*-множествами являются $\{q_1, q_2, q_3, q_4\}$, $\{q_5\}$, $\{q_6\}$ и $\{q_7, q_8\}$. Фактически *K*-множество представляет множество переходов в одно и то же устойчивое состояние при одном и том же входном сигнале.

Два различных *K*-множества, построенных для одного и того же входного сигнала, образуют *пару K-множеств*. Так же как и для пары переходов, для каждой пары *K*-множеств строится троичный вектор, компоненты которого соответствуют состояниям автомата. Компоненты, соответствующие состояниям, которые принадлежат одному из этих *K*-множеств, имеют значение 0; компоненты, соответствующие состояниям из другого *K*-множества, – значение 1, а компоненты, соответствующие состояниям, не принадлежащим ни одному из них, – значение «–». Так же как и для пар переходов, множество этих векторов для всех пар *K*-множеств, из которого удалены имплицируемые векторы, образует матрицу условий. Для автомата, поведение которого описано в табл. 1, матрица условий имеет следующий вид:

q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	
0	1	0	1	–	–	–	–	1
–	–	–	–	0	–	1	1	2
–	–	–	–	–	0	1	1	3
0	0	0	0	1	–	–	–	4
0	0	0	0	–	1	–	–	5
0	0	0	0	–	–	1	1	6
0	1	0	–	–	0	–	1	7
0	–	0	1	1	0	–	–	8
0	–	0	–	–	0	1	–	9
–	0	–	1	1	–	–	0	10
–	0	–	–	–	–	1	0	11
0	1	1	1	–	–	–	0	12
0	–	–	–	1	1	–	0	13
0	–	–	–	–	–	1	0	14
–	0	0	0	1	1	–	–	15
–	–	–	0	0	0	1	–	16

3. Минимизация длины кода состояния

Троичная матрица R *имплицирует* троичную матрицу S, если для каждой строки матрицы S в матрице R найдется имплицирующая ее строка. Задача противогоночного кодирования с минимизацией длины кода состояния сводится к нахождению матрицы с минимальным числом строк, имплицирующей матрицу условий и называемой *кратчайшей имплицирующей формой* матрицы условий. Столбцы этой матрицы будут представлять искомые коды состояний, а получаемая в результате ее транспонирования матрица называется *матрицей кодирования*. Строкам матрицы кодирования соответствуют состояния автомата, а столбцам – внутренние переменные. Строки этой матрицы представляют коды соответствующих состояний.

Кратчайшая имплицирующая форма матрицы условий находится следующим образом. Множество строк матрицы условий называется *совместимым*, если существует вектор, имплицирующий каждую строку этого множества. Совместимое множество называется *максимальным*, если оно не является собственным подмножеством другого совместимого множества. Теперь необходимо найти кратчайшее покрытие множества строк матрицы условий максимальными совместимыми множествами. Каждому совместимому множеству соответствует вектор, имплицирующий все строки, принадлежащие этому множеству. Указанные векторы, соответствующие элементам полученного покрытия, в качестве строк составят кратчайшую имплицирующую форму заданной матрицы условий. Необходимо заметить, что при рассмотрении K-множеств не всегда удается получить минимум длины кода состояния, достижимый при рассмотрении пар переходов. Этот прием следует использовать в том случае, когда важнее получить результат за более короткое время.

4. Минимизация переключательной активности элементов памяти

При применении описанного подхода к решению задачи противогоночного кодирования состояний асинхронного автомата для снижения интенсивности переключений элементов памяти можно использовать следующие предположения.

Каждому i-му столбцу матрицы кодирования можно поставить в соответствие множество переходов. Данными переходами связаны состояния автомата, в кодах которых переменная z_i имеет различные значения, т. е. при этих переходах i-й триггер в реальной схеме, реализующей заданный автомат, меняет свое состояние. Следовательно, для снижения интенсивности переключений элементов памяти надо выбрать такой вариант противогоночного кодирования состояний, который соответствует наименьшему множеству переходов между состояниями.

Если удается вычислить вероятности переходов, то столбцу матрицы кодирования состояний ставится в соответствие вероятность события, которое заключается в том, что происходит некоторый переход из множества переходов, связанных с данным столбцом матрицы кодирования состояний. Поскольку переходы между состояниями автомата являются несовместимыми событиями, эта вероятность равна сумме вероятностей отдельных переходов из данного множества. Для подсчета вероятностей переходов между состояниями в статье [5] используется метод Чэпмена – Колмогорова, где данные вероятности получаются в результате решения системы линейных уравнений с этими вероятностями в качестве неизвестных. Однако такой метод можно применять только тогда, когда автомат полностью определен, а его граф поведения является сильносвязным ориентированным графом. В противном случае столбцу матрицы кодирования состояний автомата можно, например, приписывать мощность связанного с ним множества переходов.

Таким образом, каждому совместимому множеству строк матрицы условий и, соответственно, вектору, имплицирующему все строки из данного множества, приписывается вес в виде числа переходов или в виде величины, пропорциональной сумме вероятностей переходов, связанных с этим вектором. Искомое решение получается в виде покрытия множества строк матрицы условий максимальными совместимыми множествами, обладающего минимальным весом. Весом покрытия является сумма весов принадлежащих ему элементов.

Вероятность перехода из состояния q_i в состояние q_j, вызываемого входным сигналом a, когда автомат находится в состоянии q_i, равна вероятности прихода входного сигнала a. Если имеется несколько входных сигналов, переводящих автомат из состояния q_i в состояние q_j, условная вероятность p'_{ij} такого перехода равна сумме вероятностей этих сигналов, поскольку поступления на вход автомата различных входных сигналов – несовместимые события. Условием перехода является то, что автомат находится в состоянии q_i. Нахождение автомата в состоянии q_i и приход сигнала, переводящего в состояние q_j, являются независимыми событиями. Поэтому абсолютная вероятность p_{ij} перехода из состояния q_i в состояние q_j в течение всего времени работы автомата равна $P_i p'_{ij}$, где P_i – вероятность того, что автомат находится в состоянии q_i.

Вероятности P_i ($i = 1, 2, \ldots, |Q|$) находятся путем решения системы уравнений Чэпмена – Колмогорова, которые имеют следующий вид:

$$\sum_{i=1}^{|Q|} P_i p'_{ij} = P_j, \quad j = 1, 2, \ldots, |Q|;$$

$$\sum_{i=1}^{|Q|} P_i = 1.$$

Вероятности p'_{ij} должны быть известны. Таким образом, решив данную систему уравнений, получим вероятности P_i. Как было сказано раньше, абсолютная вероятность $p_{ij} = P_i p'_{ij}$.

Асинхронный автомат, поведение которого описано в табл. 1, является полностью определенным, а граф его поведения является сильносвязным ориентированным графом. Следовательно, вероятности переходов между состояниями можно определять по методу Чэпмена – Колмогорова. Допустим, что вероятности входных сигналов данного автомата имеют равномерное распределение. Тогда условные вероятности p'_{ij} переходов (из состояния q_i в состояние q_j, когда автомат находится в состоянии q_i) представлены в табл. 2, где строки и столбцы соответствуют состояниям автомата и на пересечении строки q_i и столбца q_j расположена вероятность p'_{ij}. Пустые клетки означают отсутствие перехода между соответствующими состояниями (вероятность такого перехода равна нулю).

Таблица 2

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
q_1	1/2	1/4	1/4					
q_2		1/2		1/4				1/4
q_3	1/4	1/4	1/4	1/4				
q_4		1/2		1/4	1/4			
q_5					3/4	1/4		
q_6			1/4			3/4		
q_7							3/4	1/4
q_8	1/4						1/4	1/2

Для нахождения вероятностей состояний (вероятностей попадания автомата в те или иные состояния) надо решить следующую систему линейных уравнений (для упрощения вычислений используем величины, пропорциональные условным вероятностям):

$$2P_1 + P_3 + P_8 = 4P_1;$$

$$P_1 + 2P_2 + P_3 + 2P_4 = 4P_2;$$

$$P_1 + P_3 + P_6 = 4P_3;$$

$$P_2 + P_3 + P_4 = 4P_4;$$

$$P_4 + 3\,P_5 = 4\,P_5;$$

$$P_5 + 3\,P_6 = 4\,P_6;$$

$$3\,P_7 + P_8 = 4\,P_7;$$

$$P_2 + P_7 + 2\,P_8 = 4\,P_8;$$

$$P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 = 1.$$

В результате решения данной системы уравнений получаем $P_1 = 4/27$, $P_2 = P_7 = P_8 = 5/27$, $P_3 = P_4 = P_5 = P_6 = 2/27$. Абсолютные вероятности переходов, приведенные к общему знаменателю, представлены в табл. 3, где строки и столбцы соответствуют состояниям автомата и на пересечении строки q_i и столбца q_j расположена вероятность p_{ij}. Пустые клетки, так же как и в табл. 2, показывают нулевые вероятности.

Таблица 3

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
q_1	2/27	1/27	1/27					
q_2		5/54		5/108				5/108
q_3	1/54	1/54	1/54	1/54				
q_4		1/27		1/54	1/54			
q_5					3/54	1/54		
q_6			1/54			3/54		
q_7							15/108	5/108
q_8	5/108						5/108	5/54

Если рассматривать пары переходов, то максимальных совместимых множеств строк матрицы условий окажется 55, что затруднит демонстрацию предлагаемого подхода. Поэтому обратимся к менее громоздким вычислениям при рассмотрении пар *K*-множеств. В табл. 4 представлены полученные для рассматриваемого примера максимальные совместимые множества строк матрицы условий, обозначенные их номерами, и векторы, имплицирующие все строки из соответствующих множеств. В качестве весов полученных множеств представлены величины, пропорциональные суммам соответствующих вероятностей (числители этих сумм при общем знаменателе 108).

Таблица 4

Совместимые множества	Имплицирующие векторы	Веса множеств
{1,2,3,7,9}	(0 1 0 1 0 0 1 1)	15
{1,2,8}	(0 1 0 1 1 0 0 0)	15
{1,2,13,16}	(0 1 0 1 1 1 0 0)	15
{1,3,7,8,9}	(0 1 0 1 1 0 1 1)	15
{1,7,8,11}	(0 1 0 1 1 0 0 1)	25
{1,8,9,14}	(0 1 0 1 1 0 1 0)	25
{1,13,14}	(0 1 0 1 1 1 1 0)	25
{2,3,4,5,13,15}	(0 0 0 0 1 1 0 0)	4
{2,3,6,9,16}	(0 0 0 0 0 0 1 1)	10
{2,3,7,9,10,16}	(0 1 0 0 0 0 1 1)	16
{2,3,10,13,16}	(0 0 0 1 1 1 0 0)	13
{2,3,12,13,16}	(0 1 1 1 1 1 0 0)	15
{2,5,6}	(0 0 0 0 0 1 1 1)	14
{3,4,6,9}	(0 0 0 0 1 0 1 1)	14
{4,5,6,15}	(0 0 0 0 1 1 1 1)	14
{4,5,11,13,14,15}	(0 0 0 0 1 1 1 0)	14

Окончание табл. 4

Совместимые множества	Имплицирующие векторы	Веса множеств
{4,9,11,14}	(0 0 0 0 1 0 1 0)	14
{7,10,11}	(0 1 0 0 0 0 0 1)	30
{8,9,10,11,14}	(0 0 0 1 1 0 1 0)	23
{9,11,14,16}	(0 0 0 0 0 0 1 0)	10
{12,13,14}	(0 1 1 1 1 1 1 0)	25

В настоящей работе не рассматривается какой-то конкретный метод получения минимальных взвешенных покрытий. Эта задача известна давно и достаточно подробно исследована (см., например, [11]). Заметим только, что если не учитывать веса полученных максимальных совместимых множеств, то в качестве покрытия может быть найдена совокупность совместимых множеств {1, 8, 9, 14}, {2, 3, 12, 13, 16}, {4, 5, 6, 15}, {7, 10, 11} с весом 84. Минимальный вес, равный 67, имеет покрытие, которое состоит из множеств {1, 3, 7, 8, 9}, {2, 3, 12, 13, 16}, {4, 5, 6, 15}, {8, 9, 10, 11, 14} и согласно принятому критерию является лучшим. Соответственно имеем следующие матрицы кодирования:

$$\begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array} ; \quad \begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} z_1 & z_2 & z_3 & z_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array} .$$

Оценить преимущество первого варианта матрицы кодирования над вторым можно еще следующим образом. Кодирование состояний автомата можно представить как размещение состояний автомата в пространстве внутренних переменных $z_1, z_2, \ldots, z_k$ [5], т. е. по вершинам булева гиперкуба, представляющего это пространство. В статье [5] введен критерий качества такого размещения с точки зрения интенсивности переключений элементов памяти. Этот критерий выражается формулой $D = \sum w_{ij}(d_{ij} - 1)$, где d_{ij} – расстояние по Хэммингу между кодами состояний q_i и q_j, w_{ij} в данном случае – число переходов между состояниями q_i и q_j или величина, пропорциональная вероятности перехода между состояниями q_i и q_j ($i \neq j$). Суммирование ведется по всем парам состояний, соответствующим парам вершин в гиперкубе. Очевидно, чем меньше значение D, тем лучше результат размещения, и $D = 0$, если всем парам состояний, связанным переходами, соответствуют ребра гиперкуба. Тогда при любом переходе из состояния в состояние переключается ровно один элемент памяти.

Для первого варианта кодирования $D = 35$. Второй вариант кодирования дает $D = 18$. Сравнение по критерию D результатов решения примеров показывает целесообразность использования предлагаемого метода.

Заключение

Предлагаемый метод энергосберегающего противогоночного кодирования состояний асинхронного автомата рассчитан на использование его в автоматизированной системе логического проектирования. Сравнение результатов кодирования состояний изложенным методом и кодирования состояний без учета интенсивности переключений элементов памяти показывает, что применение данного метода дает лучший результат.

Список литературы

1. Мурога, С. Системное проектирование сверхбольших интегральных схем. В 2-х кн. Кн. 1 / С. Мурога. – М. : Мир, 1985. – 288 с.

2. Pedram, M. Power minimization in IC design: Principles and applications / M. Pedram // ACM Trans. Design Automat. Electron. Syst. – 1996. – Vol. 1. – P. 3–56.

3. Kashirova, L. State assignment of finite state machine for decrease of power dissipation / L. Kashirova, A. Keevallik, M. Meshkov // Second Intern. Conf. Computer-Aided Design of Discrete Devices. – Minsk : Institute of Engineering Cybernetics NAS of Belarus, 1997. – Vol. 1. – P. 60–67.

4. Sudnitson, A. Partition search for FSM low power synthesis / A. Sudnitson // Fourth Intern. Conf. Computer-Aided Design of Discrete Devices. – Minsk : Institute of Engineering Cybernetics NAS of Belarus, 2001. – Vol. 1. – P. 44–49.

5. Закревский, А.Д. Алгоритмы энергосберегающего кодирования состояний автомата / А.Д. Закревский // Информатика. – 2011. – № 1(29). – С. 68–78.

6. Закревский, А.Д. Алгоритмы синтеза дискретных автоматов / А.Д. Закревский. – М. : Наука, 1971. – 512 с.

7. Ангер, С. Асинхронные последовательностные схемы / С. Ангер. – М. : Наука, 1977. – 400 с.

8. Синтез асинхронных автоматов на ЭВМ / под ред. А.Д. Закревского. – Минск : Наука и техника, 1975. – 184 с.

9. Автоматизированное проектирование цифровых устройств / под ред. С.С. Бадулина. – М. : Радио и связь, 1981. – 240 с.

10. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

11. Закревский, А.Д. Оптимизация покрытий множеств / А.Д. Закревский // Логический язык для представления алгоритмов синтеза релейных устройств. – М. : Наука, 1966. – С. 136–148.

Поступила 07.05.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pott@newman.bas-net.by

Yu.V. Pottosin

LOW POWER RACE-FREE STATE ASSIGNMENT OF AN ASYNGHRONOUS AUTOMATON

The problem of a race free state assignment of an asynchronous automaton is considered. A method for the state assignment is suggested that provides the minimization of the number and the switching activity of the memory elements along with the elimination of the critical races between them.

ЗАЩИТА ИНФОРМАЦИИ

УДК 004.056:005.342 (083.74)(476)

А.И. Трубей

ОЦЕНКА РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ С ИСПОЛЬЗОВАНИЕМ СУЩЕСТВУЮЩЕЙ НОРМАТИВНО-ПРАВОВОЙ И МЕТОДИЧЕСКОЙ БАЗЫ

Проводятся обзор и анализ существующей нормативной базы в области менеджмента рисков информационной безопасности, а также теоретических основ и методов их оценки. Предлагаются практические методы оценки рисков и уязвимостей информационной безопасности.

Введение

В настоящее время все больше конфиденциальной информации хранится и обрабатывается в различных информационных системах (ИС). Практически любой ИС присущи уязвимости, обусловливающие возможность реализации угроз обрабатываемой в ней информации. Наиболее вероятными источниками возникновения угроз для защищаемых активов являются обслуживающий персонал, средства вычислительной техники, системное и прикладное программное обеспечение. В этих условиях особое внимание следует уделять анализу и оценке рисков информационной безопасности (ИБ) как необходимым составляющим комплексного подхода к обеспечению ИБ. Решение задачи оценки угроз безопасности для различных ИС является основой для применения необходимых методов и средств защиты информации. Адекватная оценка рисков ИБ позволит осуществлять прогнозирование возможного ущерба, связанного с реализацией угроз, а соответственно, и оценку необходимого размера инвестиций на построение систем защиты информации.

В статье основное внимание уделяется теоретическим и практическим вопросам оценки уязвимостей информации как одной из составных частей оценки риска. Приводятся оценки риска утечки конфиденциальной информации по техническим (радио-) каналам при соблюдении либо несоблюдении соответствующих мер защиты, а также оценки риска хищения конфиденциальной информации, к которой не был предусмотрен доступ по политике безопасности, для локальной сети и (для сравнения) облачной системы.

1. Нормативная база в области менеджмента рисков

Достижение требуемого уровня ИБ в любой организации должно базироваться на исследовании источников угроз информации, уязвимостей в ее защите и обусловленных их соотношением рисков. Механизм эффективного противодействия угрозам ИБ содержится в доступных международных стандартах, прежде всего в современных риск-ориентированных стандартах ISO (International Organization for Standardization) и аналогичных национальных стандартах.

Оценка рисков ИБ и их периодическая переоценка являются неотъемлемой частью создания и функционирования системы менеджмента информационной безопасности (СМИБ), разрабатываемой с целью выбора соразмерных средств управления безопасностью, которые предназначены для защиты информационных активов и придают уверенность заинтересованным сторонам. Данное положение закреплено в СТБ ISO/IEC 27001–2011 [1] (самом популярном в Беларуси стандарте в области ИБ [2]) и других нормативных документах. В частности, в соответствии с приказом Оперативно-аналитического центра при Президенте Республики Беларусь от 16.01.2015 № 3 «в целях реализации политики информационной безопасности разрабатываются локальные нормативные правовые акты организации, регламентирующие порядок … выявления угроз, которые могут привести к сбоям, нарушению функционирования ин-

формационной системы» [3]. Это является составной частью мониторинга и анализа СМИБ, а также оценки рисков.

Вследствие большого количества стандартов и подходов к анализу рисков ИБ основные понятия в этой области имеют множество определений. Наиболее подходящим для большинства практических применений является определение риска ИБ, приведенное в СТБ ISO/IEC 27005–2012 [4]. Стандарт основан на общих концепциях, изложенных в [1], и предназначен для содействия адекватному обеспечению ИБ на основе риск-ориентированного подхода. Согласно [4] риск информационной безопасности – это потенциальная возможность использования уязвимостей актива или группы активов конкретной угрозой для причинения ущерба организации. В стандарте [4] конкретизировано понятие риска ИБ (активы, угрозы, уязвимости и ущерб), рассмотрен процесс менеджмента рисков ИБ, включающий идентификацию, анализ, оценку, обработку и т. д., и предложены две критические точки принятия решения (обрабатывать ли конкретный риск и считается ли данный риск приемлемым). Из стандарта исключено детальное описание рекомендуемого подхода к оценке рисков. Организация может самостоятельно выбирать подходящую методологию.

Негативные последствия широкого круга угроз ИБ (начиная от атак хакеров и заканчивая действиями инсайдеров, применяющих свои знания и права доступа к данным для собственной выгоды) можно уменьшить, используя подход к управлению инцидентами ИБ, описанный в новом стандарте ISO/IEC 27035:2011 [5]. Интеграция системы управления инцидентами ИБ позволит уменьшить негативные последствия от реализации угроз ИБ, повысить общий уровень ИБ, качество оценки и управления рисками ИБ. Стандарт согласован с общими принципами, установленными в [1], и может применяться в любой организации независимо от ее размера.

Заслуживает также упоминания американский стандарт в области менеджмента рисков NIST 800–37:2010 [6], в котором представлен трехуровневый подход к оценке риска. Выделяют уровень ИС, уровень бизнес-процессов и уровень организации. На уровне ИС идентифицируются информационные активы, уязвимости и угрозы, а также применяемые средства защиты.

2. Теоретические основы оценки рисков информационной безопасности

Оценка риска заключается в определении его уровня (качественной либо количественной величины) и сравнении этого уровня с максимально допустимым (приемлемым) уровнем, а также с уровнем других рисков. Другими словами, оценка риска нарушения ИБ – это систематический и документированный процесс выявления, сбора, использования и анализа информации, позволяющий провести оценивание рисков нарушения ИБ, связанных с использованием информационных активов на всех стадиях их жизненного цикла.

Уровень риска определяется путем комбинирования двух величин: вероятности инцидента в области ИБ и размеров его последствий. Инцидент заключается в реализации угрозы, использующей уязвимости актива для воздействия на этот актив и нарушения его безопасности. Под безопасностью информационного актива понимаются такие свойства информации, как конфиденциальность (защита от несанкционированного ознакомления), целостность (актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения) и доступность (возможность за приемлемое время получить требуемую информационную услугу). Иногда к ИБ относят также аутентичность (возможность подтверждения подлинности и достоверности документов) и неотказуемость (невозможность отрицания совершенных действий применительно к информационным активам).

Точно определить вероятности угрозы и уязвимости либо размер ущерба на практике обычно не представляется возможным, поэтому речь может идти только о численных оценках в некотором диапазоне величин. Количественная оценка риска необходима для определения конкретной величины риска, а качественная – для интерпретации полученного результата. Количественную величину риска, связанного с осуществлением конкретной угрозы безопасности в отношении конкретного актива, можно выразить следующим образом:

$$R = P_V \cdot P_T \cdot D, \tag{1}$$

где P_V (вероятность уязвимости) – вероятность успешного использования уязвимости потенциальной угрозой;

P_T (вероятность угрозы) – вероятность того, что угроза в отношении актива будет реализовываться; успех либо неуспех реализации угрозы определяется величиной уязвимости. Анализ методик менеджмента рисков показывает, что в них используется не вероятность реализации угрозы, а примерная частота реализации угрозы за определенный промежуток времени. Для исключения путаницы в стандартах вместо термина probability используется likelihood;

D – величина потенциального ущерба, который может быть причинен при реализации угрозы.

2.1. Вероятность уязвимости

При решении практических задач защиты информации первостепенное значение имеет количественная оценка ее уязвимости. Известно, что несанкционированное получение информации возможно не только путем доступа к базам данных, но и многими другими способами, не требующими такого доступа. Основную опасность представляют преднамеренные действия злоумышленников. Воздействие случайных факторов само по себе не ведет к несанкционированному получению информации, оно лишь способствует появлению каналов утечки информации, которыми может воспользоваться злоумышленник. Потенциально возможные несанкционированные действия могут иметь место во внешней неконтролируемой зоне, зоне контролируемой территории, зоне помещений, зоне активов, зоне баз данных.

При этом для несанкционированного получения информации необходимо одновременное наступление следующих событий:

– нарушитель получил доступ в соответствующую зону;

– во время нахождения нарушителя в зоне проявляется канал утечки;

– канал утечки доступен нарушителю соответствующей категории;

– в канале утечки есть конфиденциальная информация.

По уровню возможностей нарушители могут подразделяться на следующие категории:

Первая категория – пользователи с низким уровнем возможностей, осуществляющие запуск задач (программ) из фиксированного набора, которые реализуют типовые функции по обработке информации.

Вторая категория – разработчики из числа обслуживающего персонала со средним уровнем возможностей, осуществляющие создание и запуск собственных программ с новыми функциями.

Третья категория – администраторы ИС, сотрудники подразделения технической защиты информации с высоким уровнем воздействия на базовое ПО и конфигурацию оборудования.

Четвертая категория – персонал сторонних организаций или самой организации, осуществляющий проектирование и техническое обслуживание ИС с максимальными возможностями: включение в состав ИС аппаратно-программных средств с новыми функциями.

Вероятность уязвимости – несанкционированного получения информации нарушителем *k*-й категории по *j*-му каналу утечки информации в *l*-й зоне *i*-го структурного компонента объекта либо системы – находится по формуле [7]

$$P_{ijkl} = P^{\partial}_{ikl} \cdot P^{к}_{ijl} \cdot P^{и}_{ijkl} \cdot P^{и}_{ijl}, \qquad (2)$$

где P^{∂}_{ikl} – вероятность доступа нарушителя *k*-й категории в *l*-ю зону *i*-го компонента объекта либо системы;

$P^{к}_{ijl}$ – вероятность наличия (проявления) *j*-го канала утечки информации в *l*-й зоне *i*-го компонента объекта либо системы;

$P^{и}_{ijkl}$ – вероятность доступа нарушителя *k*-й категории к *j*-му каналу утечки информации в *l*-й зоне *i*-го компонента при условии доступа нарушителя в зону;

P_{ijl}^{u} – вероятность наличия защищаемой информации в *j*-м канале утечки информации в *l*-й зоне *i*-го компонента.

Вероятность уязвимости принимает значения в диапазоне (0, 1).

2.2. Частота реализации угрозы

Частота реализации угрозы определяется путем экспертных оценок, прогнозирования, а также на основании статистических данных. Является положительным числом, определяющим ожидаемое количество попыток реализации угрозы за определенный период времени.

2.3. Величина ущерба

Величина прямого или косвенного ущерба, причиняемого организации в результате инцидентов безопасности, связанных с раскрытием, несанкционированной модификацией, временной недоступностью или разрушением информации, определяется ценностью информационных активов. Последствия таких инцидентов могут выражаться в упущенной выгоде, потере конкурентных преимуществ, ухудшении имиджа организации, причинении вреда интересам третьей стороны, штрафах, прямых финансовых убытках или дезорганизации деятельности. При этом для каждого актива следует рассматривать наихудший сценарий развития событий.

Оценка ущерба – достаточно сложная задача, плохо поддающаяся формализации и решаемая, как правило, с использованием методов экспертных оценок. Для оценки возможного ущерба могут применяться различные критерии и качественные шкалы. Для того чтобы оценка ущерба имела экономический смысл, качественная шкала должна соотноситься с размером финансовых потерь. Размер ущерба, как правило, выражается в денежных единицах.

3. Краткий обзор методик оценки рисков и уязвимостей

Согласно [1] оценка рисков ИБ необходима для понимания требований ИБ и рисков для активов организации. Наиболее простую и понятную методику управления информационными рисками предлагает группа компаний GlobalTrust, специализирующихся на создании систем защиты информации и персональных данных, а также систем менеджмента информационной безопасности. Методика полностью соответствует требованиям международных стандартов [1–5], представляющих собой практическое руководство по управлению рисками, и базируется на популярных методах оценки рисков, таких как CRAMM, OCTAVE и RA2 [8].

Новая версия методики управления рисками GlobalTrust включает полный комплект документов, необходимых для внедрения системы управления рисками ИБ в организациях любого типа и размера. Методика оценки рисков GlobalTrust характеризуется следующими особенностями:

– риск оценивается для конкретных активов или групп активов;

– риск определяется качественно и количественно на основании трех параметров: вероятности угрозы, величины уязвимости, размера ущерба;

– формируется реестр информационных активов, определяющих для каждого информационного актива его местоположение, формат, принадлежность к классам и категориям пользователей и владельцев, приложения и бизнес-процессы, в которых он используется, а также свойства актива и требования по его доступности;

– для оценки риска используется многоуровневая качественная шкала, дополнительно риски делятся на высокие, средние, низкие; для сопоставления качественных уровней рисков и количественных значений величины риска используется процедура калибровки шкалы оценки риска;

– методика охватывает также процессы обработки риска, предусматривается формирование плана обработки рисков.

Для вычисления рисков удобно использовать период времени, равный одному году. В этом случае величина риска соответствует прогнозируемым среднегодовым потерям организации в результате инцидентов безопасности ALE (Annual Loss Expectancy). Эту величину целесообразно использовать для соотнесения расходов на безопасность с величиной риска.

Величина ALE рассчитывается по формуле

$$ALE = SLE \cdot ARO. \quad (3)$$

Здесь SLE (Single Loss Expectance) – потенциальный ущерб (в денежных единицах) для организации в результате единичного факта реализации соответствующей угрозы:

$$SLE = AV \cdot EF, \quad (4)$$

где AV (Asset Value) – стоимость актива (данных, программ, аппаратуры и т. д.);

EF (Exposure Factor) – степень уязвимости актива к угрозе;

ARO (Annualized Rate of Occurrence) – среднегодовая частота возникновения инцидентов (ожидаемая частота реализации угрозы в год). Значение ARO зависит от эффективности элементов системы защиты и вероятности того, что они не выполнят своих функций.

Таким образом,

$$ALE = SLE \cdot ARO = AV \cdot EF \cdot ARO. \quad (5)$$

Для определения EF используются оценочные количественные значения, полученные путем экспертных оценок, прогнозирования, а также на основании статистических данных из общедоступных источников, например на основании систем оценки уязвимостей, которые созданы коммерческими и некоммерческими организациями. На данный момент существует ряд организаций, занимающихся мониторингом, классификацией и накоплением данных об уязвимостях, которые предоставляют открытый доступ к своим базам уязвимостей. Перечни уязвимостей в этих системах используются как один из основных источников информации для оценки ИБ, так как описания уязвимостей содержат и предусловия, и оценки, характеризующие результат атак, эксплуатирующих эти уязвимости, а также списки конкретных программно-аппаратных средств, содержащих уязвимости.

Каждая из этих систем имеет свои преимущества, но все они отличаются по измеряемому признаку. Например, CERT/CC использует значения оценок от 0 до 180 и учитывает следующие факторы: подвержена ли интернет-инфраструктура риску и какой тип предусловий нужен для эксплуатации уязвимости. При оценке уязвимости по методике SANS кроме простоты эксплуатации учитывается и распространенность уязвимых систем. Компания Microsoft использует методику оценки уязвимостей, связанных с вредоносным программным обеспечением, которая учитывает наличие обновления и количество векторов, применяемых атакующим. В системе DREAD используются минимальное число показателей и очень простая формула для получения общего показателя (простое их усреднение).

По мнению экспертов, одной из наиболее распространенных, востребованных и проверенных на практике является система оценки общеизвестных уязвимостей (Common Vulnerability Scoring System, CVSS) [9, 10]. CVSS предназначена для оценки уязвимостей, связанных с дефектами и ошибками ПО при проектировании или кодировании. Она состоит из трех базовых, временных и контекстных метрик. Каждая метрика представляет собой число (оценку) в интервале 0–10 и вектор – краткое текстовое описание со значениями, которые применяются для вывода оценки.

Базовые метрики (Base Metrics) используются для описания основополагающих сведений об уязвимости (возможности эксплуатации уязвимости и влиянии уязвимости на систему). Их значения практически не меняются со временем и не зависят от окружения, в котором работает оцениваемый продукт. Метрики «вектор доступа (*AV*)», «сложность доступа (*AC*)» и «аутентификация (*Au*)» оценивают возможность получения доступа к уязвимости и необходимость для эксплуатации уязвимости дополнительных условий. Метрики воздействия «влияние на конфиденциальность (*C*)», «целостность (*I*)» и «доступность (*A*)» описывают разрушительность атаки в случае эксплуатации уязвимости. Это влияние определяется независимо относительно конфиденциальности, целостности и доступности.

Временные метрики (Temporal Metrics) отражают характеристики уязвимостей, которые изменяются во времени. К ним относятся метрики «воздействие на возможность эксплуатации **(***E*)», «уровень устранения (*RL*)», «степень достоверности информации (*RC*)».

Группа контекстных метрик (Environmental Metrics) позволяет адаптировать оценку уязвимости к конкретной ИС. Контекстные метрики учитывают конкретное окружение, в котором работает программа. К ним относятся возможность сопутствующего ущерба (*CDP*), распределение целей (*TD*), требования к конфиденциальности (*CR*), целостности (*IR*), доступности (*AR*).

Как правило, базовые и временные метрики определяются аналитиками, разработчиками продуктов в области безопасности или разработчиками приложений, потому что они лучше осведомлены о характеристиках уязвимости, чем пользователи. Контекстные метрики определяются пользователями, поскольку они точнее могут оценить потенциальное воздействие уязвимости в рамках своей собственной среды. На первом этапе проводится расчет оценки для базовых метрик, на которую затем накладываются временные и контекстные оценки. В дальнейшем будем использовать только базовые и временные метрики для вычисления оценки по формуле

$$TemporalScore = TS = [((0,6 \cdot Impact) + (0,4 \cdot Exploitability) - 1,5) \cdot f(Impact)] \cdot E \cdot RL \cdot RC, \qquad (6)$$

где $Impact = 10,41 \cdot (1 - (1 - C) \cdot (1 - I) \cdot (1 - A))$;

$Exploitability = 20 \cdot AV \cdot AC \cdot Au$;

$f(Impact) = 0$, если $Impact = 0$; $f(Impact) = 1,176$ в противном случае.

Оценки уязвимостей по методике CVSS необходимо пронормировать для получения значений в интервале (0;1), что позволит использовать их при расчете рисков по формуле (5):

$$EF = CVSS\ Rating / 10. \qquad (7)$$

Вместо частоты реализации угрозы *ARO* будем определять вероятность эксплуатации уязвимости, которая учитывает как вероятность наличия уязвимости, так и вероятность ее использования хотя бы одной из угроз. Чем выше уровень подверженности уязвимости применению эксплойта, тем больше шансов провести успешную атаку и тем выше частота злонамеренного использования. Вычислим вероятность использования уязвимости с применением базовых метрик возможности эксплуатации и временных метрик по формуле [11]

$$P_E = AV \cdot AC \cdot Au \cdot E \cdot RL \cdot RC. \qquad (8)$$

Частоту реализации угрозы *ARO* будем определять с помощью данной вероятности в каждом конкретном случае оценки риска.

Некоторые наиболее типичные риски утечки конфиденциальной информации можно оценить с помощью методик GlobalTrust и CVSS.

При расчете вероятностей уязвимостей и угроз, а также при оценке рисков осуществляется округление результатов вычислений.

4. Оценка риска утечки информации по радиоканалу

Оценим риск утечки конфиденциальной информации, обрабатываемой в ИС, которая реализует высокопроизводительные информационно-вычислительные технологии обработки геолого-геофизических данных, по радиоканалу (посредством радиоизлучений от внедренных электронных устройств перехвата информации, модулированных информативным сигналом).

Предположим, что технические средства ИС расположены на одном объекте в пределах одной контролируемой зоны и обработка конфиденциальной информации осуществляется в рабочие дни в рабочее время в течение 2 ч. Нарушителя можно отнести к четвертой категории.

Для оценки риска будем использовать следующую качественную шкалу, в которой определенным уровням риска сопоставляются соответствующие размеры среднегодовых потерь. Предположим, что стоимость ИС совместно с активами и оказываемыми услугами составляет Ca млн руб. Поэтому максимальный риск, равный восьми, сопоставим с потерей функциональности всей ИС или ее большей части (банкротством). Минимальный уровень риска, равный нулю, соответствует отсутствию среднегодовых потерь в результате инцидентов безопасности либо минимальным потерям, не превышающим, например, $0{,}0001 \cdot Ca$ млн руб. в год (табл. 1).

Таблица 1

Откалиброванная качественная шкала оценки риска

Уровень риска	Среднегодовой ущерб в результате инцидентов безопасности (*ALE*), млн руб.
Низкие риски	
0	$0 \le ALE < 0{,}0001 \cdot Ca$
1	$0{,}0001 \cdot Ca \le ALE < 0{,}0005 \cdot Ca$
2	$0{,}0005 \cdot Ca \le ALE < 0{,}0015 \cdot Ca$
Средние риски	
3	$0{,}0015 \cdot Ca \le ALE < 0{,}003 \cdot Ca$
4	$0{,}003 \cdot Ca \le ALE < 0{,}03 \cdot Ca$
5	$0{,}03 \cdot Ca \le ALE < 0{,}1 \cdot Ca$
Высокие риски	
6	$0{,}1 \cdot Ca \le ALE < 0{,}3 \cdot Ca$
7	$0{,}3 \cdot Ca \le ALE < 0{,}5 \cdot Ca$
8	$0{,}5 \cdot Ca \le ALE$

В основу оценки ущерба может быть также положен подход, описанный в гл. 24 Уголовного кодекса Республики Беларусь. Пример калибровки шкалы с использованием данного подхода приведен в табл. 2 (при $Ca > 2000$ базовых величин).

Таблица 2

Качественная шкала оценки ущерба согласно Уголовному кодексу Республики Беларусь

Уровень ущерба (риска)	Среднегодовой ущерб в результате инцидентов безопасности (*ALE*), млн руб.
Незначительный	$0 \le ALE <$ 40 базовых величин
Значительный	40 базовых величин $\le ALE <$ 250 базовых величин
Крупный	250 базовых величин $\le ALE <$ 1000 базовых величин
Особо крупный	1000 базовых величин $\le ALE < 0{,}5 \cdot Ca$
Неприемлемый	$0{,}5 \cdot Ca \le ALE$

В качестве информационных активов, в отношении которых рассматривается угроза утечки, возьмем геологические, геофизические, геохимические и иные данные, характеризующие особенности строения и минерально-сырьевой потенциал недр, зафиксированные на материальных носителях, накопленные и обрабатываемые в ИС. Такая информация является закрытой во всех странах. Хищение данных может привести к нарушению их конфиденциальности и прав собственности. Сведения могут быть использованы физическими или юридическими лицами, которые не предусмотрены в государственных контрактах или других соглашениях (лицензиях, договорах) на производство работ по геологическому изучению недр. В результате, возможно, придется уплачивать штрафы по искам клиентов на основании заключенных с ними соглашений о конфиденциальности. Кроме того, будет нанесен ощутимый урон репутации, который выразится в сокращении числа клиентов, заказов и уменьшении коммерческой выручки.

Предположим, что согласно принятым критериям оценки финансового ущерба стоимость активов, подвергаемых угрозе утечки, $AV = 0{,}01 \cdot Ca$ млн руб. Вычислим EF и ARO.

Согласно (2) вероятность несанкционированного снятия конфиденциальной информации нарушителем четвертой категории по радиоканалу $P_{р4}$ рассчитывается по формуле

$$P_{р4} = P_4^{д} \cdot P_р^{к} \cdot P_{р4}^{н} \cdot P_р^{и}. \tag{9}$$

Вероятности $P_4^{д}$, $P_р^{к}$, $P_{р4}^{н}$, $P_р^{и}$ оценим следующим образом. Нарушитель с переносным средством технической разведки, к примеру, может посетить объект под видом сотрудника сторонней организации для проведения работ по обслуживанию инженерных коммуникаций в помещении. Перехват может осуществляться также в непосредственной близости от границы контролируемой зоны, если она недостаточно точно рассчитана. Поэтому вероятность $P_4^{д}$ равна $0,2$, если предположить, что регламентные работы проводятся еженедельно в рабочие дни. Вероятность проявления радиоканала $P_р^{к}$ равна $0,25$, т. е. составит четверть рабочего времени, в течение которого может обрабатываться информация. Вероятность доступа нарушителя к каналу при условии его доступа в зону $P_{р4}^{н}$ равна 1 (портативный приемник он имеет при себе).

Оценим вероятность наличия конфиденциальной информации в радиоканале $P_р^{и}$, т. е. вероятность выделения сигнала на фоне шума, действующего на входе приемного устройства. На практике обеспечение защиты информации от утечки по техническим каналам осуществляется с применением активной защиты – специальных широкополосных генераторов электромагнитного шума. Такие генераторы используют помехи типа «белый шум», т. е. излучают широкополосный шумовой сигнал с равномерно распределенным энергетическим спектром во всем рабочем диапазоне частот (например, гауссовский «белый шум»). Важнейшей характеристикой генератора шума является коэффициент δ – отношение сигнал-шум.

Вычислим P_o – вероятность правильного обнаружения сигнала техническими средствами разведки в пределах контролируемой зоны. При определенных допущениях вероятность P_o обнаружения сигнала с неизвестными параметрами можно рассчитать по формуле [12, 13]

$$P_o = \Phi\left[\frac{\delta \cdot \sqrt{\Delta F_{пр} T_a} - \Phi^{-1}(1 - P_{лт})}{1 + \delta}\right], \tag{10}$$

где $P_{лт}$ – вероятность ложной тревоги;

$\Delta F_{пр}$ – полоса пропускания тракта приемника, Гц;

T_a – время осреднения (анализа) процесса, с;

$T_{лт}$ – средний интервал между ложными тревогами;

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt.$$

Для обеспечения точности, достаточной для расчетов параметров перехватываемого сигнала, ошибка измерения не должна превышать 10 % от значения измеряемой величины, т. е. $P_{ош} \le 0,1 \cdot P_o$. Выбор порогового значения вероятности правильного обнаружения сигнала целесообразно осуществлять посредством минимизации вероятности полной ошибки

$$P_{ош} = P^{*}(1 - P_o) + (1 - P^{*})P_{лт}, \tag{11}$$

где P^{*} – априорная вероятность наличия сигнала на входе приемного устройства.

Подставив в формулу (11) значение вероятности полной ошибки $P_{ош} = 0,1 \cdot P_o$ для случая наибольшей неопределенности $(P^* = 0,5)$, получим $P_{оп} = 0,83$. Это означает, что при $P_o \le P_{оп}$ выделить информацию на фоне шума будет невозможно или крайне трудно.

При перехвате информации вероятности ложной тревоги составляют $10^{-6} - 10^{-3}$. Задавая пороговое значение вероятности $P_{оп}$, из формулы (10) при $\Delta F_{пр} T_a = 10, T_a \le T_{лт}$ и $P_{лт} = 10^{-3}$ получаем предельно допустимое (пороговое) значение отношения сигнал-шум на входе приемного устройства:

$$\delta_n = \frac{\Phi^{-1}(P_{оп}) + \Phi^{-1}(1 - P_{лт})}{\sqrt{\Delta F_{пр} T_a} - \Phi^{-1}(P_{оп})} = \frac{\Phi^{-1}(0,83) + 3,2}{3,16 + \Phi^{-1}(0,83)} = 1,9. \quad (12)$$

Таким образом, если $P_o \le P_{оп} = 0,83$ $(\delta \le \delta_n = 1,9)$, то перехват информации невозможен, можно только определить факт наличия или отсутствия сигнала без расчета его параметров. Поэтому вероятность наличия информации в радиоканале $P_p^и$ равна нулю.

Если $P_o > P_{оп} = 0,83$ $(\delta > \delta_n = 1,9)$, то перехват информации возможен, вероятность наличия конфиденциальной информации в радиоканале $P_p^и = P_o$.

Следовательно, вероятность несанкционированного получения конфиденциальной информации нарушителем четвертой категории по радиоканалу будет следующей:

$$P_{р4} = P_4^д \cdot P_р^к \cdot P_{р4}^н \cdot P_р^и = 0, \text{ если } \delta \le \delta_n = 1,9; \quad (13)$$

$$P_{р4} = P_4^д \cdot P_р^к \cdot P_{р4}^и \cdot P_р^и = 0,2 \cdot 0,25 \cdot 1 \cdot P_o, \text{ если } \delta > \delta_n = 1,9. \quad (14)$$

Предположим, что на объекте соблюдаются меры активной и пассивной защиты:

– границы контролируемой зоны (включающей опасные зоны R_1 и R_2) определены с учетом радиуса зоны возможного перехвата информации по результатам специальных исследований;

– осуществляется проверка импортных технических средств перед введением в эксплуатацию на отсутствие в них электронных устройств перехвата информации;

– осуществляются конструктивные доработки технических средств и помещений, где они расположены, в целях локализации возможных каналов утечки информации;

– применяются исправные адаптивные к уровню опасных сигналов генераторы шума (один генератор шума на 40 м^2, расстояние между соседними генераторами шума 20 м), которые, с одной стороны, обеспечивают гарантированную защиту информации $(\delta \le \delta_n = 1,9)$, а с другой – не засоряют эфир и не наносят вред здоровью обслуживающего персонала.

Тогда согласно (13) $EF = P_{р4} = 0$.

Из опыта можно оценить частоту реализации угрозы несанкционированного съема информации по радиоканалу (ARO): один раз в год. Следовательно, $ALE = 0,01 \cdot Ca \cdot 0 \cdot 1 = 0$ млн руб. Это минимальный уровень риска.

Если же не соблюдаются вышеперечисленные меры безопасности, некорректно рассчитаны границы контролируемой зоны, применяются неисправные или не обеспечивающие условие $\delta \le \delta_n = 1,9$ генераторы шума, то согласно (14), например, при $\delta = 2$:

$$EF = P_{р4} = 0,2 \cdot 0,25 \cdot 1 \cdot P_o = 0,2 \cdot 0,25 \cdot 1 \cdot 0,85 = 0,04.$$

Следовательно, $ALE = 0,01 \cdot Ca \cdot 0,04 \cdot 1 = 0,0004 \cdot Ca$ млн руб. Это низкий уровень риска, равный единице (см. табл. 1). Если Ca = 10 млрд руб., AV = 100 млн руб., то ALE = 4 млн руб. Согласно табл. 2 это незначительный уровень ущерба.

Аналогичным образом можно оценить риск утечки конфиденциальной информации и пороговые значения отношения сигнал-шум δ_n вследствие побочных электромагнитных излучений и наводок видеосистемы и клавиатуры компьютеров ИС.

5. Оценка риска хищения информации пользователем локальной сети, реализующей обработку геолого-геофизических данных

В качестве объекта оценки будем рассматривать приведенную выше ИС, реализующую высокопроизводительные информационно-вычислительные технологии обработки геолого-геофизических данных. Оценим риск хищения аналогичной конфиденциальной информации, к которой не был предусмотрен доступ по политике безопасности, злоумышленником из числа пользователей ИС с использованием привилегий. Особенно опасны атаки, позволяющие нарушителю получить системные права, так как в этом случае он может иметь практически полный доступ ко всем активам ИС. Это позволяет нарушителю выполнять любые несанкционированные действия по компрометации ИС, а также хищению конфиденциальных сведений.

Вероятность уязвимости оценим по методике CVSS. Уязвимости будем оценивать по наиболее широко используемым привилегиям. При вычислении оценки уязвимости, которая имеет несколько способов эксплуатации (векторов атаки), будем выбирать тот метод, который оказывает наибольшее воздействие на систему. В метрике *Au* требования аутентификации учитываются с того момента, как осуществлен доступ в систему. Для локально эксплуатируемых уязвимостей значения *S* или *M* будем присваивать, если необходима дополнительная аутентификация помимо той, которая требуется при регистрации в системе. Уязвимости, предоставляющие доступ с пользовательским уровнем привилегий, оцениваются как частичная потеря конфиденциальности, целостности и доступности.

Проанализируем базовые метрики и определим их значения: *AV* – локальный (*L* = 0,395); *AC* – высокая (*H* = 0,35) (потому что уязвимость не может произвольно эксплуатироваться злоумышленником); *Au* – многократная (*M* = 0,45) (необходима дополнительная аутентификация, помимо той которая требуется при регистрации в сети); *C, I* и *A* – частичные (*P* = 0,275). В результате получим базовый вектор $AV:L/AC:H/Au:M/C:P/I:P/A:P$.

Далее проанализируем временные метрики и определим их значения: *E* – функционально использованное (*F* = 0,95); *RL* – временное решение (*TF* = 0,90); *RC* – не подтверждена (*UC* = 0,90). В результате получим временной вектор $E:F/RL:TF/RC:UC$.

Воспользовавшись формулой (6), а также при необходимости калькулятором CVSS Calculator [14], получим *TS* = 2,6. Согласно шкале, разработанной FortiGuard Center [15] и приведенной в табл. 3, уровень опасности данной уязвимости – низкий.

Таблица 3

Шкала опасности уязвимостей FortiGuard

Уровень опасности	Оценка по CVSS 2.0
Критический	9 – 10
Высокий	7 – 8,9
Средний	4 – 6,9
Низкий	0,1 – 3,9
Информационный	0

В соответствии с (7) получим $EF = \frac{2,6}{10} = 0,26.$

По формуле (8) определим вероятность использования уязвимости:

$$P_E = L \cdot H \cdot M \cdot F \cdot TF \cdot UC = 0,395 \cdot 0,35 \cdot 0,45 \cdot 0,95 \cdot 0,9 \cdot 0,9 = 0,05.$$

В предположении, что пользователь еженедельно имел возможность осуществлять несанкционированный доступ с использованием привилегий, получим $ARO = 52 \cdot 0,05 = 2,6.$

Следовательно, $ALE = 0,01 \cdot Ca \cdot 0,26 \cdot 2,6 = 0,007 \cdot Ca$ млн руб. Это средний уровень риска, равный четырем (см. табл. 1). Если Ca = 10 млрд руб., AV = 100 млн руб., то ALE = 70 млн руб. Согласно табл. 2 это крупный уровень ущерба.

6. Оценка риска хищения геолого-геофизических данных, обрабатываемых в публичной облачной системе

Оценим риск хищения вышеописанной конфиденциальной информации, к которой не был предусмотрен доступ по политике безопасности, злоумышленником с использованием привилегий при ее обработке в публичной облачной системе.

Следует учитывать, что провайдеры сервисов SaaS и PaaS несут ответственность за управление конфигурацией своих платформ, а клиенты IaaS обеспечивают управление доступом к операционной системе, бизнес-приложениям и ПО уровня Middleware, безопасность конфигурации развертываемых на виртуальных серверах приложений и ПО, антивирусную защиту.

Вероятность уязвимости оценим по методике CVSS. Уязвимости, при которых обеспечивается доступ на корневом уровне, будем оценивать как полную потерю конфиденциальности, целостности и доступности. Например, нарушение целостности, которое позволяет злоумышленнику изменять файл паролей операционной системы, оценим как полное нарушение конфиденциальности, целостности и доступности. С учетом этого получим следующие результаты.

Базовые метрики принимают значения: AV – сетевой (N = 1,0); AC – низкая (L = 0,71); Au – однократная (S = 0,56); C – полное (C = 0,66); I – полное (C = 0,66); A – полное (C = 0,66). Получим базовый вектор $AV:N / AC:L / Au:S / C:C / I:C / A:C$.

Временные метрики принимают значения: E – высокое (H = 1,0); RL – официальное исправление (OF = 0,87); RC – подтверждена (C = 1,0). Получим временной вектор $\mathrm{E:H/\ RL:OF/\ RC:C}$. Следовательно, по формуле (6) получим TS = 7,8, что согласно табл. 3 соответствует высокому уровню опасности. После нормирования получим EF = 0,78.

По формуле (8) определим вероятность использования уязвимости:

$$P_E = N \cdot L \cdot S \cdot H \cdot OF \cdot C = 1,0 \cdot 0,71 \cdot 0,56 \cdot 1,0 \cdot 0,87 \cdot 1,0 = 0,35.$$

В предположении, что пользователь еженедельно имел возможность осуществлять несанкционированный доступ с использованием привилегий, получим $ARO = 52 \cdot 0,35 = 18,2$. Следовательно, $ALE = 0,01 \cdot Ca \cdot 0,78 \cdot 18,2 = 0,14 \cdot Ca$ млн руб. Это высокий уровень риска, равный шести (см. табл. 1). Если Ca = 10 млрд руб., AV = 100 млн руб., то ALE = 1400 млн руб. Согласно табл. 2 это особо крупный уровень ущерба. В случае если имеются более подробные сведения об ИС и облачной системе, то полученные временные оценки можно улучшить путем использования контекстной формулы, в которой метрики среды объединяются с временной оценкой. В конечном итоге учет показателей временной метрики позволит уменьшить вероятность успешного применения уязвимости.

Справедливость расчетов по оценке вероятностей использования уязвимостей для локальной ИС и облачной системы подтверждается статистическими данными. В [16] отмечается, что по вектору эксплуатации уязвимости распределились следующим образом: 77 % уязвимостей эксплуатировались удаленно, 15 % уязвимостей – по локальной сети, 8 % уязвимостей требовали физического доступа. Полученная оценка возможного риска для облачных вычислений показывает, что использование публичного облака неприемлемо в случае обработки конфиденциальной информации, для этой цели может использоваться, например, гибридное облако или следует применять криптографическую защиту конфиденциальной информации.

Заключение

В статье предложен метод оценки рисков утечки конфиденциальной информации на основании методик GlobalTrust и CVSS. Проведены оценки рисков утечки конфиденциальной информации по радиоканалу, а также хищения конфиденциальных сведений, к которым не был предусмот-

рен доступ по политике безопасности, для различных типов развертывания информационных систем. Уровень риска утечки конфиденциальной информации по радиоканалу при конкретной структуре ИС и стоимости активов оценен как низкий, а уровень риска хищения конфиденциальной информации пользователем локальной сети – как средний. Для сравнения, уровень риска хищения конфиденциальной информации, обрабатываемой в публичном облаке, оценен как высокий. Другими словами, при одних и тех же условиях риск хищения конфиденциальной информации в публичном облаке примерно в 20 раз выше, чем в локальной сети.

Полученные результаты позволяют оценить реальные возможности перехвата информации средствами технической разведки и обосновать целесообразность использования тех или иных средств защиты от утечки информации по техническим каналам. Кроме того, можно сделать вывод о необходимости применения дополнительных средств защиты конфиденциальной информации, например гомоморфного шифрования [17], при ее обработке в публичном облаке.

Список литературы

1. Информационные технологии. Методы обеспечения безопасности. Системы менеджмента информационной безопасности. Требования : СТБ ISO/IEC 27001–2011. – Минск : Госстандарт, 2011. – 28 с.

2. Домнич, К. Беларусь в международном исследовании компании EY по информационной безопасности за 2013 год / К. Домнич, А. Ворошилов // Банкаўскі веснік. – 2014. – № 2. – С. 64–67.

3. Приказ Оперативно-аналитического центра при Президенте Республики Беларусь от 16.01.2015 № 3 «О внесении дополнений и изменений в приказ Оперативно-аналитического центра при Президенте Республики Беларусь от 30.08.2013 № 62» [Электронный ресурс]. – 2015. – Режим доступа : http://www.oac.gov.by/files/files/pravo/prikazi_oac/Prikaz_OAC_3.htm. – Дата доступа : 06.04.2015.

4. Информационные технологии. Методы обеспечения безопасности. Менеджмент рисков информационной безопасности : СТБ ISO/IEC 27005–2012. – Минск : Госстандарт, 2012. – 61 с.

5. Information technology – Security techniques – Information security incident management : ISO/IEC 27035:2011. – Geneva : ISO/IEC, 2011. – 78 p.

6. Guide for Applying the Risk Management Framework to Federal Information Systems. A Security Life Cycle Approach : NIST 800–37:2010. – Gaithersburg : NIST, 2010. – 93 p.

7. Малюк, А.А. Теория защиты информации / А.А. Малюк. – М. : Горячая линия – Телеком, 2013. – 184 с.

8. Астахов, А.М. Искусство управления информационными рисками / А.М. Астахов. – М. : ДМК Пресс, 2010. – 312 с.

9. Common Vulnerability Scoring System (CVSS-SIG) [Electronic resource]. – 2015. – Mode of access : http://www.first.org/cvss. – Date of access : 06.04.2015.

10. Система оценки общеизвестных уязвимостей. Рекомендация. МСЭ-Т X.1521 (04/2011) [Электронный ресурс]. – 2015. – Режим доступа : https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.1521-201104-I!!PDF-R&type=items. – Дата доступа : 06.04.2015.

11. Малюк, А.А. Один из подходов к оценке рисков информационной безопасности в облачных средах / А.А. Малюк, А.В. Царегородцев, Е.В. Макаренко // Безопасность информационных технологий. – 2014. – № 4. – С. 68–74.

12. Хорев, А.А. Оценка возможностей средств радиоразведки по перехвату информации / А.А. Хорев // Специальная техника. – 2009. – № 2. – С. 54–63.

13. Трубей, А.И. Статистические и нейросетевые методы оценки эффективности защиты информации, обрабатываемой средствами вычислительной техники / А.И. Трубей, В.А. Дмитриев, В.В. Анищенко // Электроника инфо. – 2013. – № 6. – С. 24–26.

14. NVD Common Vulnerability Scoring System Support v.2 [Electronic resource]. – 2015. – Mode of access : http://www.nvd.nist.gov/cvss.cfm?calculator&version=2. – Date of access : 06.04.2015.

15. Vulnerability Severity Level [Electronic resource]. – 2015. – Mode of access : http://www.fortiguard.com/static/intrusionprevention.html. – Date of access : 06.04.2015.

16. Статистика уязвимостей в 2011 году [Электронный ресурс]. – 2015. – Режим доступа : http://www.securitylab.ru/analytics/422328.php. – Дата доступа : 06.04.2015.

17. Трубей, А.И. Гомоморфное шифрование: безопасность облачных вычислений и другие приложения (обзор) / А.И Трубей // Информатика. – 2015. – № 1. – С. 90–101.

Поступила 17.04.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: trubeia@newman.bas-net.by

A.I. Trubei

INFORMATION SECURITY RISK ASSESSMENT USING EXISTING LEGAL AND METHODOLOGICAL BASE

The article provides a survey of the existing regulatory framework for information security risk management. Practical methods for information security risk and vulnerability assessment are proposed.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Статьи принимаются в редакцию через электронную систему подачи по адресу **http://jinfo.bas-net.by** в формате файлов текстовых редакторов Microsoft Word 97 и Word 2000 для Windows. Основной текст статьи набирается с переносами шрифтом Times New Roman 11 пт, интервал между строками – одинарный, абзацный отступ 1 см, поля по 2,5 см со всех сторон.

2. Статья должна иметь индекс УДК (универсальная десятичная классификация).

3. Название статьи, фамилии всех авторов и аннотация должны быть переведены на английский язык. Для каждого из авторов приводится развернутое название учреждения с полным почтовым адресом, а также номер телефона и электронный адрес (e-mail) для связи с редакцией.

4. Формулы, иллюстрации, таблицы, встречающиеся в статье, должны быть пронумерованы в соответствии с порядком цитирования в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Необходимо избегать повторения одних и тех же данных в таблицах, графиках и тексте статьи.

Рисунки должны быть выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, набираются шрифтом гарнитуры основного текста, размер символов 9 пт. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала.

5. Набор формул выполняется в формульных редакторах Microsoft Equation или Math Type и должен быть единообразным по применению шрифтов и знаков по всей статье.

Прямо (_) набираются: греческие и русские буквы; математические символы (sin, lg, ∞); символы химических элементов (C, Cl, $CHCl_3$); цифры (римские и арабские); векторы; индексы (верхние и нижние), являющиеся сокращениями слов.

Курсивом (~) набираются: латинские буквы – переменные, символы физических величин (в том числе и в индексе).

6. Сокращения в тексте статьи (за исключением единиц измерения) могут быть использованы только после упоминания полного термина. Единицы измерения физических величин следует приводить в Международной системе СИ.

7. Литература приводится автором общим списком в конце статьи. Ссылки на литературу в тексте идут по порядку и обозначаются цифрой в квадратных скобках. Ссылаться на неопубликованные работы не допускается. С примерами оформления библиографического описания в списке литературы можно ознакомиться в приложении 2 к *Инструкции по оформлению диссертации, автореферата и публикаций по теме диссертации* на сайте Высшей аттестационной комиссии Республики Беларусь **http://vak.org.by**.

8. Поступившие в редакцию статьи направляются на рецензирование специалистам. Основным критерием целесообразности публикации является новизна и информативность статьи. Если по рекомендациям рецензента статья возвращается автору на доработку, а переработанная рукопись вновь рассматривается редколлегией, датой поступления считается день получения редакцией ее окончательного варианта. Статьи не по профилю журнала возвращаются авторам после заключения редколлегии.

9. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все вопросы.

10. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

11. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее, или статей, принятых к публикации другими изданиями.

12. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

Журнал «Информатика» включен Высшей аттестационной комиссией Республики Беларусь в список научных изданий для опубликования результатов диссертационных исследований.

Индексы

00827

для индивидуальных подписчиков

008272

для предприятий и организаций

ИНФОРМАТИКА 2 (46) апрель–июнь 2015