

Программная инженерия

Том 8
№ 5
2017
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

- Басавин Д. А., Поршнева С. В., Петросов Д. А.** Оценка максимального числа моделируемых информационных потоков с помощью параллельной программной реализации гибридной жидкостной модели интернет-трафика на основе технологии GPGPU 195
- Поляков А. В.** Метод идентификации личности по отпечаткам пальцев на основе сферического локально-чувствительного хэширования .. 207
- Светушков Н. Н.** Программный интерфейс простого графического 3D-редактора на основе метаязыка описания связей 215
- Аббасов А. Э.** Анализ экспертной информации об экологических показателях транспортных средств 222
- Стронгин М. М., Виноградов Ю. А., Здидовец А. Г., Киселёв Н. А., Попович С. С.** Использование технологий National Instruments в экспериментальных исследованиях процессов термогазодинамики ... 230

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2017

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 8

N 5

2017

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCENKO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

- Basavin D. A., Porsnev S. V., Petrosov D. A.** Estimation of Maximum Number of Simulated Information Flows for Parallel Software Implementation of a Hybrid Fluid Model of Internet Traffic using GPGPU 195
- Poliakov A. V.** Fingerprint Identification using Spherical Local-Sensitive Hashing 207
- Svetushkov N. N.** Program Interface for a Simple 3D-Graphical Editor Based on a Relationships Description Meta-Language 215
- Abbasov A. Je.** Analysis of Expert Information on Environmental Indicators Vehicles 222
- Strongin M. M., Vinogradov Yu. A., Zditovets A. G., Kiselev N. A., Popovich S. S.** Applying of National Instruments Technologies in Experimental Research of Thermal Gas Dynamics Processes 230

Information about the journal is available online at:
<http://novtex.ru/prin/eng> e-mail: prin@novtex.ru

Д. А. Басавин¹, ассистент кафедры, e-mail: basavind@gmail.com,

С. В. Поршнева², д-р техн. наук, проф., e-mail: sergey_porshnev@mail.ru,

Д. А. Петросов¹, канд. техн. наук, доц., зав. кафедрой, e-mail: scorpionss2002@mail.ru

¹ Белгородский государственный аграрный университет им. В. Я. Горина, п. Майский, Белгородская обл.

² Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, г. Екатеринбург

Оценка максимального числа моделируемых информационных потоков с помощью параллельной программной реализации гибридной жидкостной модели интернет-трафика на основе технологии GPGPU

Обсуждается гибридная жидкостная модель интернет-трафика, описывающая информационные потоки в современных компьютерных сетях. В рамках этой модели информационные потоки описываются системой дифференциальных уравнений (СДУ), которая с физической точки зрения является следствием баланса информационных потоков, входящих в соответствующий узел компьютерной сети и выходящих из него. В связи с отсутствием аналитических решений СДУ, входящих в гибридную жидкостную модель, необходима разработка программных инструментов, позволяющих получать численные решения данной СДУ. Опыт практического использования известного варианта программной реализации гибридной жидкостной модели с последовательно исполняемым кодом показал, что время расчета характеристик информационных потоков даже для относительно простых конфигураций компьютерных сетей оказывалось весьма значительным, а для компьютерных систем со сложной архитектурой — неприемлемо большим. В связи с этим была выдвинута гипотеза о возможности применения технологий параллельных вычислений для разработки программной реализации гибридной жидкостной модели на основе технологии GPGPU, которая в различных случаях обеспечивает ускорение сложных расчетов в несколько десятков раз. В связи с этим данная технология к настоящему времени является одной из передовых и общедоступных. Однако одним из ее ограничений является жесткое аппаратное ограничение объема доступной памяти GPU. В связи с этим оказывается необходимым установление связи между параметрами аппаратного обеспечения (GPU) и максимально возможным числом моделируемых информационных потоков (соответственно, размером сети и ее архитектурой). Получена оценка максимального числа моделируемых информационных потоков с помощью параллельной программной реализации гибридной жидкостной модели интернет-трафика, созданной на основе технологии GPGPU [1].

Ключевые слова: интернет-трафик, компьютерные сети, гибридная жидкостная модель интернет-трафика, технология GPGPU, GPU

Введение

В настоящее время основной объем информационных потоков в телекоммуникационных сетях составляют данные, передаваемые через сеть Интернет, — интернет-трафик. В состав глобальной компьютерной сети (КС) Интернет входят миллионы разнородных устройств, расположенных в разных странах, принадлежащих различным организациям и частным лицам. Передачу данных по сети Интер-

нет одновременно осуществляют тысячи приложений различных типов.

В связи с этим для провайдеров телекоммуникационных услуг наиболее актуальными оказываются задачи выбора конфигурации КС, характеристик сетевого оборудования (маршрутизаторов, коммутаторов и др.) и политик управления интернет-трафиком. Провайдеры уже на этапе проектирования телекоммуникационных сетей должны знать, какое число пользователей они смогут обслуживать

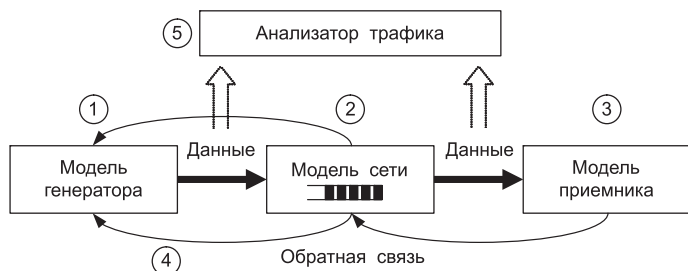


Рис. 1. Обобщенная структурная схема математической модели интернет-трафика

при условии обеспечения гарантированной скорости передачи данных для каждого пользователя.

Одним из перспективных подходов к решению подобной задачи является использование математических моделей (ММ), описывающих информационные потоки в современных КС. Такой подход, с точки зрения используемых материальных и временных затрат, представляется наиболее перспективным. Использование адекватных ММ информационных потоков в КС позволит рассчитывать для заданных архитектуры и параметров технического оборудования КС количественные оценки характеристик трафика, в первую очередь, объемов передаваемой информации и скорости ее передачи. Наличие подобных оценок позволит уже на этапе проектирования КС дать ответ на вопрос: сможет ли сетевое оборудование при выбранных технических характеристиках обеспечить заявленное качество обслуживания пользователей? Данный подход, как очевидно, дает возможность осуществить адекватный выбор сетевого оборудования проектируемой КС, исключая при этом необходимость его приобретения, монтажа, настройки и проверочного тестирования на действующих КС.

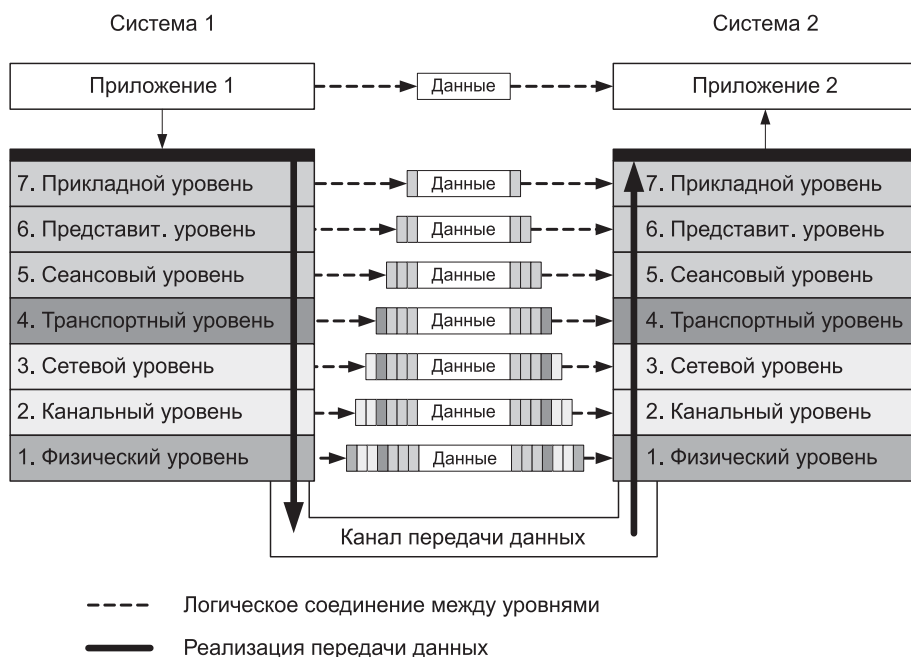


Рис. 2. OSI-модель КС, состоящая из двух информационных систем, связанных интернет-каналом [5]

Анализ работ, посвященных разработке методов построения ММ интернет-трафика, показал, что они могут быть представлены в виде единой структурной схемы, приведенной на рис. 1 [2].

На рис. 1 видно, что ММ интернет-трафика в КС объединяет ММ следующих подсистем и процессов:

- источников (генераторов) трафика;
- прохождения трафика по каналу передачи данных;
- приемников трафика.

В таких моделях интернет-трафика учитываются механизмы обратной связи, используемые для регулирования работы источников трафика и каналов передачи данных.

В рамках описанной выше обобщенной структурной схемы может быть реализовано большое число ММ. Вид конкретной ММ интернет-трафика зависит от выбранного уровня рассмотрения процесса передачи данных в КС. Согласно стандарту *OSI (Open Systems Interconnection)*, регламентирующему взаимодействие открытых систем [3], выделяют семь уровней средств взаимодействия (рис. 2).

Напомним, что под открытой системой понимается "сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами по стандартным правилам, определяющим формат, содержание и значение принимаемых и отправляемых сообщений" [4].

Здесь обмен данными происходит непосредственно между выбранным уровнем и примыкающими к нему верхним и/или нижним уровнями. Взаимодействие между уровнями регламентируется интерфейсом, определяющим структуру и форматы данных, передаваемых между уровнями стандарта *OSI*. Рассматриваемое взаимодействие является иерархическим: с верхних уровней информационных систем (ИС) передаются информационные запросы на нижние уровни, а в обратном направлении передаются результаты выполнения данных запросов. Например, транспортный уровень является связующим звеном между нижними уровнями ИС, на которых учитываются физические особенности процессов передачи данных по сети, и верхними уровнями стандарта *OSI*, на которых учитываются особенности используемого сетевого оборудования и программных средств.

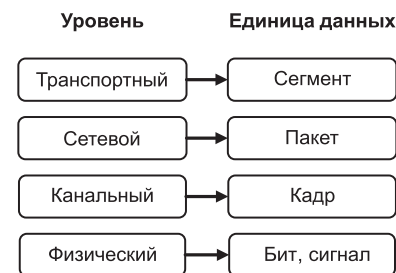


Рис. 3. Единицы измерения потока передаваемых данных на нижних уровнях стандарта *OSI*

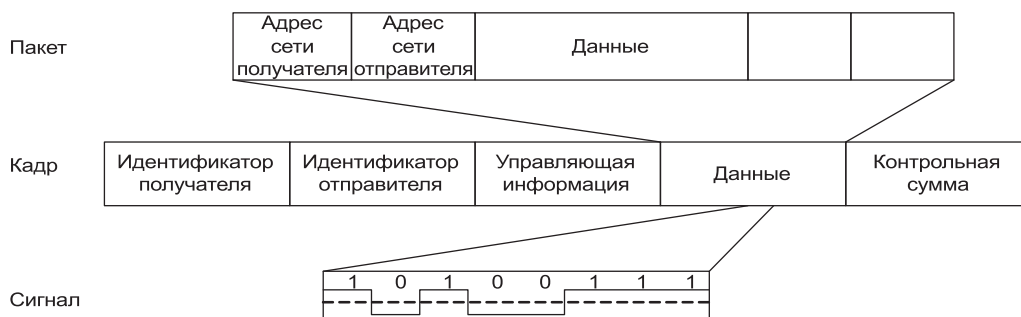


Рис. 4. Детальность физического представления о механизмах передачи данных в зависимости от уровня *OSI*

Каждому из уровней *OSI*-модели уровней соответствуют собственные информационные единицы измерения потока передаваемых данных (рис. 3).

Форма представления данных зависит от уровня, на котором в рамках *OSI*-модели рассматривается процесс передачи информации: чем более высоким оказывается уровень *OSI*-модели, тем используется более обобщенное описание механизмов передачи данных (рис. 4).

Основным протоколом передачи пакетов на транспортном уровне является протокол *TCP/IP*, на его долю приходится 85...92 % общего объема переданной информации [2]. Данная оценка получена при анализе общедоступных образцов интернет-трафика [6, 7]. Напомним, что данный протокол предусматривает возможность контроля процесса передачи данных по механизму обратной связи и повторной отправки данных. При этом скорость передачи данных изменяется динамически — чем меньше пакетов сбрасывается, тем выше скорость их передачи, и наоборот. Эта особенность протокола *TCP* позволяет адаптировать интенсивность потоков данных к текущему уровню загрузки канала передачи данных.

Обзор известных математических моделей интернет-трафика приведен в работе [2]. Их анализ показывает, что наиболее перспективными¹ оказываются ММ, в которых интернет-трафик рассматривается на уровне укрупненных групп пакетов, а не на уровне процессов распространения отдельных пакетов. К данному типу ММ относят жидкостную модель (ЖМ), предложенную Миршой (V. Mirsha) и соавторами [8], и ее последующую модификацию — гибридную жидкостную модель (ГЖМ) [2], представляющая собой результат интеграции ЖМ и *abt*-моделей источников трафика.

В соответствие с *abt*-моделью информационные потоки в сети Интернет, которыми обмениваются подключенные к ней клиенты, рассматриваются не на уровне отдельных пакетов и приложений, а более укрупненно — на уровне модулей данных приложений (МДП). При этом свойства МДП определяются свойствами некоторого множества приложений, составленного по принципу близости друг к другу. В связи с этим оказывается возможным использовать не конкретные

¹ С точки зрения учета механизмов передачи данных в современных КС.

показатели каждого из используемых в сети приложений, а некоторые усредненные по всем элементам множества данного класса приложений характеристики.

Аналогичный переход к рассмотрению укрупненных групп элементов можно наблюдать в физике. Например, методы молекулярной динамики используют для описания движения отдельных молекул газа (аналог пакетов), в то время как функции распределения используют для одновременного описания свойства всех частиц газа (аналог укрупненных групп пакетов). Такой подход позволяет:

- существенно уменьшить число рассматриваемых событий, происходящих в КС;
- учесть особенности функционирования алгоритмов ограничения скорости отдельных пользователей, применяемых в современных КС;
- учесть конечность времени распространения пакетов между узлами КС (по устоявшейся терминологии — время запаздывания);
- учесть механизмы управления размером передаваемых пакетов, предусмотренные в протоколе *TCP* [4].

Обе модели, и ЖМ, и ГЖМ представляют собой систему дифференциальных уравнений (СДУ), которая с физической точки зрения является следствием баланса потоков, входящих в соответствующий узел КС и выходящих из него. Практическое применение данных моделей осложняется отсутствием аналитических решений, используемых СДУ, что вызывает необходимость использовать соответствующие программные инструменты.

Известен вариант программной реализации ГЖМ с последовательно исполняемым кодом [9]. Однако опыт ее практического использования показал, что время расчета характеристик информационных потоков даже для относительно простых конфигураций КС оказывалось весьма значительным, а для КС со сложной архитектурой — неприемлемо большим. В связи с этим в работе [10] была высказана гипотеза о возможности существенного увеличения скорости вычислений за счет использования технологий параллельных вычислений, в частности — технологии *GPGPU*. Получены предварительные оценки, подтверждающие данную гипотезу.

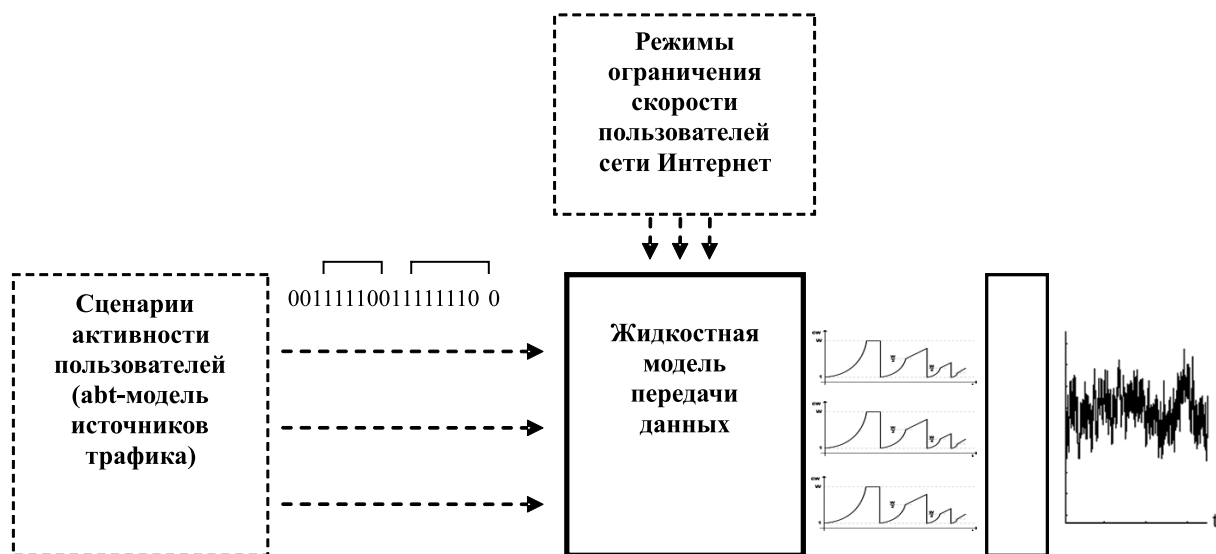


Рис. 5. Структурная схема ГЖМ

В работе [10] изложены результаты дальнейшего анализа возможности и целесообразности применения технологий *GPGPU* для создания параллельной программной реализации ГЖМ. В этой работе получены оценки максимально возможного числа моделируемых информационных потоков, которое можно разместить в памяти графического процессора без необходимости промежуточного обмена данными между графическим (*GPU*) и центральным (*CPU*) процессорами.

Основные сведения о гибридной жидкостной модели

Рассмотрим структурную схему ГЖМ, представленную на рис. 5.

Структурная модель КС, следуя работе [8], в ЖМ и ГЖМ соответственно, представляется в виде графа $G = (V, E)$, вершины которого V — хосты (узлы, маршрутизаторы), а ребра E — каналы связи между ними (рис. 6).

Для описания модели КС используются следующие определения и обозначения. Ребра графа, соответствующие каналам связи, обозначаются индексом $l \in E$. Здесь каждый канал характеризуется следующими параметрами:

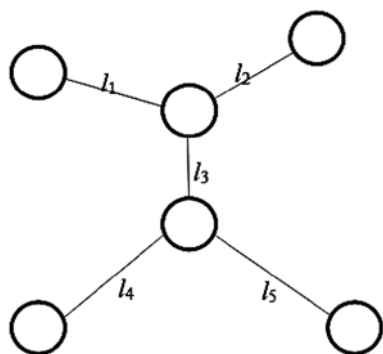


Рис. 6. Структурная модель фрагмента компьютерной сети

- пропускной способностью C_l (бит/с или пакет/с);
- длиной очереди (числом пакетов) на входе в данный канал q_l ;
- используемой политикой управления очередью на входе в канал передачи данных (маршрутизаторе), описываемой функцией $p_l(t)$ — вероятностью сброса пакетов, зависящей от уровня загрузки (перегрузки) канала l в данный момент времени t ;
- временем передачи пакетов по каналу a_l (время задержки).

В рамках рассматриваемой модели принимается, что в КС циркулируют потоки данных. Потоки данных, проходящие по одному маршруту и обладающие близкими характеристиками, можно объединить в группы или классы. Все потоки одной группы рассматриваются как единый поток, интенсивность которого равна сумме интенсивностей отдельных потоков. Для обозначения классов потоков будем использовать индекс $i = 1, \dots, N$, где N — общее число моделируемых потоков.

Пусть $F_i = \{k_{i,1}, \dots, k_{i,m_i}\}$ и $O_i = \{j_{i,1}, \dots, j_{i,m'_i}\}$ — упорядоченный перечень (список) каналов (соответственно, очередей), по которым последовательно проходит i -й поток в прямом (передача данных) и в обратном (передача подтверждений) направлениях; m_i и m'_i — число очередей, которые проходят в прямом и обратном направлениях соответственно, согласно маршрутам для потока i . Полный маршрут i -го потока данных в прямом и обратном направлениях $E_i = F_i \cup O_i$.

Тогда изменение во времени скорости передачи данных i -го потока $W_i(t)$ (размер "окна данных") и длины очереди на входе в l -й канал $q_l(t)$ описывается следующей системой дифференциальных уравнений:

$$\frac{dW_i(t)}{dt} = \frac{\theta_{(W_i(t)-M_i)}}{R_i(t)} - \frac{W_i(t)}{2} \lambda_i(t), \quad (1)$$

$$\frac{dq_i(t)}{dt} = -\theta_{q(t)}C_l + \sum_{i \in N_i} A_i^l(t), \quad (2)$$

где M_i — максимально допустимый размер "окна данных" для потока i ;

$\theta_{f(t)}$ — функция Хевисайда

$$\theta_{f(t)} = \begin{cases} 1, & \text{если } f(t) \geq 0, \\ 0, & \text{если } f(t) < 0, \end{cases}$$

$R_i(t)$ — время оборота² i -го потока, рассчитываемое в соответствии с формулой

$$R_i(t) = \sum_{l \in E_i} (a_l + q_l/C_l), \quad (3)$$

суммирование здесь проводится по всей трассе прохождения i -го потока E_i ;

$\lambda_i(t)$ — скорость потери пакетов i -го потока, вычисляемая по формуле

$$\lambda_i(t) = p(t - R_i(t)/2) A_i(t - R_i(t)/2), \quad (4)$$

C_l — пропускная способность l -го канала, который обслуживается данным маршрутизатором;

$A_i^l(t)$ — скорость передачи i -го потока по l -му каналу:

$$A_i^l(t) = \frac{W_i^l(t)}{R_i^l(t)}.$$

Таким образом, в ЖМ и ГЖМ учитывается, что подтверждение о получении данных, отправленных в момент времени $t - R_i(t)$, поступает на источник данных в момент времени t . Как следствие, в соответствии с протоколом *TCP* в течение данного временного интервала источник отправляет равное "окну данных" число пакетов. При этом учитывается, что продолжение передачи данных возможно только после подтверждения получения всех отправленных пакетов из текущего "окна данных".

Для учета существующих режимов обеспечения гарантированной скорости доступа *CAR* (*Committed Access Rate*) [11] применяются алгоритмы класса *Rate Limiting*³. Одним из таких алгоритмов является представленный в работе [2] алгоритм "корзина маркеров". Данный алгоритм имеет следующие ключевые параметры:

- *CIR* (*Committed Information Rate*) — гарантированная средняя скорость передачи информации;

- *BC* — согласованный размер всплеска (байт) — объем данных, который можно передать, если скорость превысит заданное среднее значение;

- *BE* — расширенный размер всплеска (байт) — объем передаваемых данных, при превышении которого объемом данных, переданных со скоростью, превышающей *CIR*, передаваемые пакеты начинают сбрасываться.

Идея алгоритма "корзина маркеров" состоит в том, чтобы допускать кратковременное превышение скорости передачи данных над установленным значением *CIR*. Такой подход позволяет обеспечить более плавное реагирование ограничителя информационного потока на возникающие всплески интернет-трафика. При этом предполагается, что имеется некая виртуальная "корзина" — емкость, содержащая некоторое число абстрактных маркеров. Здесь маркер представляет собой разрешение передать (пропустить) через ограничитель трафика некоторое количество информации, например, один пакет или некоторое (определенное) число байтов, которое далее для краткости назовем "квантом информации". Ограничитель трафика работает только в том случае, когда скорость поступления данных превышает установленное ограничение (*CIR*). При превышении установленного значения ограничитель позволяет передать некоторое количество данных — *BC* (согласованный размер всплеска) со скоростью, превышающей это ограничение. После передачи каждого кванта информации, соответствующего одному маркеру (в рассматриваемом примере — одного пакета), из "корзины" извлекается один маркер. Если корзина не пуста, то "квант информации" передается, несмотря на то, что при этом скорость будет превышать установленное ограничение *CIR*. Таким образом, пересылка данных в режиме превышения *CIR* сопровождается удалением соответствующего числа маркеров из "корзины". В данном алгоритме для описания зависимости вероятности сброса пакетов от размера очереди используется формула

$$p(q_i) = \begin{cases} 0, & \text{если } q_i < BC, \\ \frac{q_i - BC}{BE - BC}, & \text{если } BC \leq q_i \leq BE, \\ 1, & \text{если } q_i > BE. \end{cases} \quad (5)$$

Фактический размер всплеска рассчитывается в соответствии с формулой

$$bs_i = q_i - BC. \quad (6)$$

Помимо алгоритмов *CAR* также применяются политики управления очередью на маршрутизаторах *AQM* (*Active Queue Management*). Одним из распространенных алгоритмов *AQM* является алгоритм *RED*. Вероятность сброса пакетов при использовании алгоритма *RED* вычисляется по формуле

² Согласно стандарту протокола *TCP* $R_i(t)$ — сумма времени прохождения потока данных от момента начала их передачи источником до момента их получения приемником и времени от момента отправления приемником подтверждения о получении данного сообщения до его получения передатчиком.

³ *Rate limiting* (англ. — ограничение скорости доступа) или *Traffic shaping* (англ. — формирование трафика) — ограничение пропускной способности канала для отдельного узла сети ниже технических возможностей канала до узла. Шейпинг обычно используется как средство ограничения максимального потребления трафика со стороны узла сети.

$$p(x) = \begin{cases} 0, & 0 \leq x < \text{thres}^{\min}, \\ \frac{x - \text{thres}^{\min}}{\text{thres}^{\max} - \text{thres}^{\min}} p^{\max}, & \text{thres}^{\min} \leq x \leq \text{thres}^{\max}, \\ 1, & \text{thres}^{\max} < x, \end{cases} \quad (7)$$

где thres^{\min} , thres^{\max} , p^{\max} — настраиваемые параметры маршрутизатора (thres^{\min} — нижнее пороговое значение средней длины очереди, начиная с которого происходит сброс пакетов; p^{\max} — максимальная вероятность сброса пакетов; thres^{\max} — верхнее пороговое значение средней длины очереди: вероятность сброса пакетов при изменении длины очереди от thres^{\min} до thres^{\max} растет линейно от нуля до p^{\max}); x — средняя длина очереди на маршрутизаторе, вычисляемая в соответствии с работой [12] по формуле

$$\frac{dx}{dt} = \frac{\log_e(1-\alpha)}{\delta} x(t) - \frac{\log_e(1-\alpha)}{\delta} q(t), \quad (8)$$

где α — настраиваемый параметр маршрутизатора; δ — частота выборки, параметр, вводимый для моделирования средней длины очереди и соответствующий $1/C$.

Скорость потери пакетов вычисляется по формуле

$$\lambda'_i(t) = \frac{1 - \left[1 - \frac{\lambda_i(t)}{A_i(t - R_i(t))} \right]^{R_i(t) A_i(t - R_i(t))}}{R_i(t)}, \quad (9)$$

где t — момент времени, в который на источник поступает подтверждение о получении данных, переданных в момент времени $t-R$; $p[q_i(t - R_i/2)]$ — величина, вычисляемая в соответствии с формулой (5). Соответственно, в формуле (1) $\lambda_i(t)$, вычисляемое в соответствии с выражением (4), заменяется $\lambda'_i(t)$, вычисляемой в соответствии с формулой (9).

Для учета очереди на общем маршрутизаторе $Q(t)$ СДУ формулы (1), (2) дополняют уравнением, описывающим скорость потери пакетов на маршрутизаторе:

$$\frac{dQ}{dt} = A - D, \quad (10)$$

где A — суммарный (входной) поток данных, поступающих на маршрутизатор, рассчитываемый по формуле

$$A = \sum_{i \in N} A_i = \sum_{i \in N} \frac{W_i(t - R_i/2)}{R_i}, \quad (11)$$

а D — выходной поток данных, исходящих из маршрутизатора:

$$D = \begin{cases} A, & \text{если } Q = 0, \\ C, & \text{если } Q > 0. \end{cases}$$

Здесь второй член в ДУ (1) и уравнение (2), как очевидно, обеспечивают учет влияния друг на друга потоков, создаваемых источниками данных.

В рамках ГЖМ в соответствии с моделью abt-источника трафика принимается, что i -й пользователь посылает запросы в (случайные) моменты времени t_k (индекс пользователя не указываем) и скачивает объем информации a_k (байт). Поэтому время завершения выполнения запроса

$$T_k = \frac{a_k}{C_k},$$

где C_k — скорость доступа данного пользователя к сети Интернет.

Следя данным работы [13], в работе [2] предложено использовать в ГЖМ трехкомпонентную модель abt-источников трафика (заметим также, что сходная классификация источников трафика рассматривалась и в других работах [14, 15]).

1. "Слоны": МДП ≥ 10 Мбайт, количество $\sim 10\%$ от полного числа активных пользователей [16], активность — в течение всего времени моделирования, скорость скачивания информации — максимально возможная. Так как время жизни потоков, отнесенных к классу "слонов", превышает время моделирования (на практике используются временные интервалы $\sim 1...10$ мин, достаточные для выявления выбросов), неравномерность загрузки сети, обусловленную этими потоками, можно учесть, разыгрывая число активных потоков данного класса. При этом считаем, что общее число потоков постоянно в течение всего времени моделирования и они работают при определенном значении выделяемой им со стороны интернет-провайдера скорости доступа.

2. "Мулы": МДП $\sim 0,3...10$ Мбайт, количество — $\sim 30\%$ от полного числа активных пользователей, активность — в соответствии со статистическими характеристиками используемых On-Off-источников⁴ [17]. Как правило, длительность пассивного периода (время обдумывания между запросами) можно описать распределением Пуассона, время скачивания — логнормальным распределением или распределением Парето.

Для описания "мулов" в процессе моделирования используют следующие параметры:

- распределение времен активации потока (вызовов по терминологии теории массового обслуживания или времени обдумывания). Согласно ряду работ (например [18]) для этой цели хорошо подходит экспоненциальное распределение с параметром $\lambda \approx 0,1...1,0$ 1/с, чему соответствует среднее время ожидания вызова $1...10$ с;

⁴ Здесь имеется в виду On/Off-модель трафика. Ее суть заключается в том, что источники (генераторы трафика) могут быть только в двух состояниях — On (включен) и Off (выключен). Источники могут генерировать трафик только когда они находятся в состоянии On. Продолжительность периодов On и Off являются статистическими характеристиками источников.

- распределение размеров скачиваемых файлов, напрямую влияющих на время активности потоков (можно использовать логнормальное распределение [19]);

- распределение времени оборота пакета (можно принять равномерным в интервале [20; 200] мс).

3. "Мыши": МДП < 0,3 Мбайт, количество — ~60 % от полного числа активных пользователей. Принимая во внимание типичную длительность активности "мышей", можно считать, что с физической точки зрения они представляют собой случайные последовательности импульсов, характеризующиеся следующими наборами параметров: A_i — амплитудами импульсов; $\tau_i = \frac{A_i}{C}$ — длительностями импуль-

сов; $\Delta t_{i-1,i}$ — длительностями интервалов между импульсами, где $i = \overline{1, N}$. Поскольку амплитуда импульсов принимает значение в диапазоне [0, 300] Кбайт, а скорость передачи данных может составлять

100 Мбит/с и более, можно утверждать, что для любого импульса

$$\tau_i = \Delta t_{i-1,i}.$$

В этой ситуации при сложении двух и более независимых случайных последовательностей вероятность одновременного совпадения двух импульсов оказывается относительно небольшой, а потому флуктуациями потока "мышей" в интернет-канале можно пренебречь.

Блок-схема последовательной реализации ГЖМ представлена на рис. 7.

На рис. 7 видно, что при использовании последовательной реализации ГЖМ выполняется следующая последовательность действий.

1. Задание дискретного интервал времени h (шаг по времени).

2. Вычисление соответствующего ряда натуральных чисел по правилу

$$n = \text{int} \left[\frac{t}{h} \right],$$

где $\text{int}[]$ — операция взятия целой части числа.

3. Задание значений параметров, определяющих сценарий действий пользователей.

4. Интегрирование СДУ (1), (2).

5. Сохранение результатов расчетов.

6. Увеличение значения счетчика интервалов на 1.

7. Если значение счетчика числа интервалов меньше заданного, выполнить п. 3—7, иначе закончить работу.

Оценка максимального числа моделируемых TCP-потоков при использовании технологии GPGPU

Анализ ГЖМ, описанной в предыдущем разделе, показывает, что на каждом шаге интегрирования СДУ ГЖМ параметры уравнений (1), (2) выбирают в зависимости от характеристики потоков, вычисленных на предыдущем шаге моделирования. В свою очередь, интегрирование ДУ, соответствующих каждому из моделируемых потоков, осуществляется независимо друг от друга. В связи с этим понятно, что, потенциально, можно разработать параллельную программную реализацию ГЖМ. Однако понятно, что ее разработка является весьма трудоемкой задачей. По этой причине для обоснования целесообразности создания программной реализации ГЖМ на основе технологии *GPGPU* получим оценку максимального числа информационных потоков, которые можно моделировать с ее помощью.

Из описания ГЖМ, приведенного в предыдущем разделе, видно, что параметры модели можно разделить на следующие группы:

- общие параметры модели (общее время моделирования, параметры распределения, моделирующие "мулов", число потоков каждого класса и т. д.);

- параметры TCP-классов (размер "окна данных", задержка прохождения сигнала, гарантированная скорость доступа и т. д.);

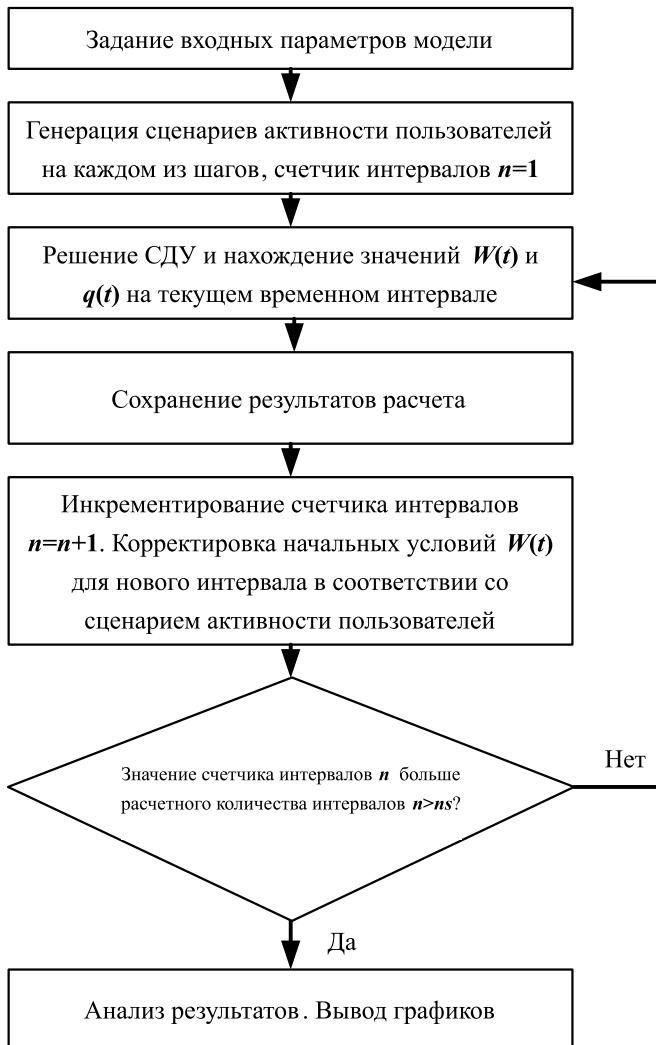


Рис. 7. Блок-схема гибридной модели информационных потоков [2]

- параметры маршрутизатора (размер очереди, пропускная способность, суммарная нагрузка и т. д.).

Также выделим следующие группы параметров ГЖМ.

- Отслеживаемые параметры — параметры, изменяющиеся на каждом шаге интегрирования СДУ и необходимые для последующих шагов интегрирования СДУ. К таким параметрам можно отнести некоторые параметры *TCP*-классов, такие как размер "окна данных", время двойного оборота пакета и др., а также некоторые параметры маршрутизаторов, такие как длина очереди, суммарная нагрузка и пр. Помимо использования этих параметров во время интегрирования СДУ, они также могут сохраняться для дальнейшего анализа, например, для оценки влияния активности *TCP*-классов на скорость потерь пакетов и др.

- Неотслеживаемые параметры — параметры, которые остаются постоянными в течение всего времени моделирования, например, общее время моделирования, число классов типа "мул" и "слон"; настройки алгоритма *RED* и т. д. Также к неотслеживаемым параметрам будем относить промежуточные, необходимые только в рамках одного шага интегрирования СДУ и не представляющие интереса для анализа, например, текущее состояние потока, счетчик смены состояния и др.

Параметры ГЖМ и типы данных, использованных для хранения этих параметров, представлены в табл. 1–3.

Принимая во внимание необходимость хранения и передачи данных параметров между *CPU* и *GPU*, целесообразно получить оценки максимально возможного числа моделируемых *TCP*-потоков, которое можно разместить в памяти *GPU*, чтобы исключить промежуточный обмен данными для расчетов, выполняемых на одном шаге алгоритма ГЖМ.

Предположим, что время моделирования составляет 10 мин, шаг интегрирования — 1 мс. Тогда для хранения "старых" значений одного параметра потребуется хранить $10 \cdot 60 \cdot 1000 = 6 \cdot 10^5$ значений, что при использовании массива типа *float* составит $\approx 2,5$ Мбайт, соответственно, 10 Мбайт на один *TCP*-поток без учета параметров маршрутизатора. Таким образом, при использовании *GPU* с размером памяти 512 Мбайт число моделируемых в течение 10 мин *TCP*-потоков не будет превосходить 40...45. При увеличении времени моделирования будет пропорционально увеличиваться число "старых" значений параметров и, соответственно, придется уменьшать число моделируемых *TCP*-потоков. Для решения отмеченной выше трудности необходимо получить оценку максимального срока хранения отслеживаемых параметров ГЖМ.

Необходимо отметить, что в ГЖМ имеются параметры, которые оказывается необходимым сохранять в течение некоторого конечного числа шагов алгоритма реализации ГЖМ, но не в течение всего времени моделирования (например, размер "окна данных" *TCP*-класса). Данная особенность ГЖМ позволяет существенно уменьшить размер используемой оперативной памяти и памяти *GPU*.

Из формул (1), (3) и (4) видно, что хранение всех значений отслеживаемых параметров ГЖМ в течение заданного интервала моделирования, фактически, не требуется, так достаточно хранить K_i значений отслеживаемого параметра:

$$K_i = \frac{\tau_i}{h},$$

где K_i — число хранимых шагов, необходимых для расчета параметров i -го *TCP*-класса; τ_i — время, определяемое как максимальное время двойного оборота для i -го *TCP*-класса; h — шаг моделирования.

Таблица 1

Неотслеживаемые параметры ГЖМ

Тип и название параметра	Физический смысл параметра
<i>(unsigned int) modeling_time</i>	Общее время моделирования
<i>(unsigned int) step_size</i>	Шаг интегрирования
<i>(unsigned int) mules_count</i>	Общее число классов типа "мул"
<i>(unsigned int) elephants_count</i>	Общее число классов типа "слон"
<i>(unsigned int) total_flows_count</i>	Общее число потоков в массиве параметров <i>TCP</i> -классов
<i>(float) t_on_min_mule</i>	Минимальное время активности <i>TCP</i> -класса типа "мул"
<i>(float) t_on_max_mule</i>	Максимальное время активности <i>TCP</i> -класса типа "мул"
<i>(float) t_off_mule</i>	Среднее время бездействия <i>TCP</i> -класса типа "мул"
<i>(unsigned int) packet_size</i>	Размер пакета в байтах
<i>(unsigned int) array_size</i>	Размер массивов для хранения "отслеживаемых" параметров <i>TCP</i> -классов и маршрутизатора
<i>(unsigned int) flows_in_row</i>	Общее число классов всех типов, участвующих при интегрировании одного шага $flows_in_row = mules_count + elephants + count$.

Параметры TCP-классов

Тип и название параметра		Физический смысл параметра
Отслеживаемые параметры	<i>(float) window</i>	Размер окна данных TCP, рассчитываемый в соответствии с формулой (1)
	<i>(float) rtt</i>	Полное время двойного оборота пакета, рассчитываемое в соответствии с формулой (3)
	<i>(float) loss_rate</i>	Скорость потерь пакетов, рассчитываемая в соответствии с формулами (4) и (9)
	<i>(float) burst</i>	Всплеск на маршрутизаторе при реализации алгоритма <i>Rate Limit</i> , рассчитываемый в соответствии с формулой (6)
Неотслеживаемые параметры	<i>(float) window_max</i>	Максимальный размер окна данных, обусловленный каналом данных
	<i>(float) propagation_delay</i>	Фиксированное время распространения сигнала по сети
	<i>(float) cir</i>	Гарантированная скорость доступа, необходимая для реализации алгоритма <i>Rate Limit</i>
	<i>(float) ebs</i>	Согласованный размер всплеска, необходимый для реализации алгоритма <i>Rate Limit</i>
	<i>(float) cbs</i>	Расширенный размер всплеска, необходимый для реализации алгоритма <i>Rate Limit</i>
	<i>(unsigned int) state_counter</i>	Счетчик состояния, необходимый для отслеживания текущего состояния потока
	<i>(unsigned char) type</i>	Класс потока ("мул" или "слон")
<i>(unsigned char) state</i>	Текущее состояние потока (активен или простаивает)	

Таблица 3

Параметры маршрутизатора

Тип и название параметра		Физический смысл параметра
Отслеживаемые параметры	<i>(float) queue</i>	Размер очереди маршрутизатора, рассчитываемый в соответствии с формулой (2)
	<i>(float) avg_queue</i>	Средний размер очереди маршрутизатора, используемый для реализации алгоритма RED, рассчитываемый в соответствии с формулой (8)
	<i>(float) drop_p</i>	p — вероятность сброса пакета на маршрутизаторах, рассчитываемая в соответствии с формулами (5) и (7)
	<i>(float) arrival_rate</i>	Суммарная нагрузка на маршрутизатор, рассчитываемая по формуле (11)
Неотслеживаемые параметры	<i>(float) max_thres</i>	Верхний порог срабатывания алгоритма RED
	<i>(float) min_thres</i>	Нижний порог срабатывания алгоритма RED
	<i>(float) max_p</i>	Максимальная вероятность сброса пакета, учитываемая при моделировании алгоритма RED
	<i>(float) bandwidth</i>	Пропускная способность маршрутизатора
	<i>(float) red_weight</i>	Коэффициент веса, учитываемый при моделировании алгоритма RED
<i>(float) buffer_size</i>	Размер входного буфера маршрутизатора	

Необходимо также учесть возможные вариации времени распространения сигнала по каналам связи для каждого TCP-класса, от которых зависит требуемое число "старых" значений для каждого класса. Максимальное значение τ_i определяется параметрами маршрутизатора и канала связи каждого класса:

$$\max(\tau_i) = a_i + B/C,$$

где a_i — время распространения сигнала по каналу данных, мс; B — размер буфера маршрутизатора,

бит; C — пропускная способность маршрутизатора, бит/мс.

В этой ситуации понятно, что для каждого TCP-класса необходимо хранить разное число "старых" данных, поэтому необходимо использовать массивы различных размерностей. Однако использование данных массивов неизбежно приведет к снижению производительности GPU. Для повышения производительности GPU предложено использовать массивы одинаковой длины. Определим максимальное время

хранения отслеживаемых параметров ГЖМ среди моделируемых TCP-классов согласно формуле

$$K = \frac{\max(a_i) + \frac{B}{C}}{h} \quad (12)$$

Для нахождения зависимости между числом моделируемых TCP-классов и размером физической памяти, занимаемой структурами данных, необходимых для моделирования данного количества TCP-классов, оценим размеры структур данных, отводимых под постоянные и неотслеживаемые параметры. Объем данных V зависит от размера типов данных используемой вычислительной платформы и числа отслеживаемых и неотслеживаемых параметров ГЖМ:

$$\begin{aligned} V_M &= V_{MS} + V_{FU}N + V_{RU} + (V_{FT}N + V_{RT})K = \\ &= V_{u\text{int}}8 + V_{float} + (V_{float}5 + V_{u\text{int}} + V_{uchar}2)N + \\ &\quad + V_{float}6 + (V_{float}4N + V_{float}4)K = \\ &= V_{u\text{int}}(8 + N) + V_{uchar}2N + V_{float}(7 + 5N + 4K(N + 1)), \end{aligned} \quad (13)$$

где V_{MS} — условная структура данных, объединяющая общие параметры модели; V_{FU} — условная структура данных, объединяющая неотслеживаемые параметры TCP-классов; V_{RU} — условная структура данных, объединяющая неотслеживаемые параметры маршрутизатора; V_{FT} — условная структура данных, объединяющая отслеживаемые параметры TCP-классов; V_{RT} — условная структура данных, объединяющая отслеживаемые параметры маршрутизатора; $V_{u\text{int}}$ — размер типа данных *unsigned int*; V_{uchar} — размер типа *unsigned char*; V_{float} — размер типа данных *float*; V_M — общий размер данных, необходимый для хранения всех параметров модели, байт; N — моделируемое число TCP-классов; K — максимальная длина массивов для хранения "старых" значений параметров модели.

Таким образом объем памяти, занимаемый структурами данных, напрямую зависит от числа пользователей и характеристик сети и ограничен сверху фактическим объемом памяти графического процессора, а именно

$$V_M < V_{AM}, \quad (14)$$

где V_{AM} — фактический объем памяти графического процессора.

Отметим также, что в формуле (14) указано строгое условие, которое обусловлено наличием дополнительных структур данных, а также тем обстоятельством, что часть памяти GPU используется подсистемой графического вывода и в служебных целях операционной системы.

Получим экспериментальную оценку максимально возможного числа TCP-классов, используемых при моделировании, для следующего случая. Имеется компьютер⁵ с GPU с объемом памяти 512 Мбайт

⁵ Будем учитывать, что процессор компьютера построен на архитектуре x86, в качестве языка программирования используется язык C, что, как было сказано ранее, влияет на размеры типов данных, используемых в модели.

($V_{AM} = 512$); шаг моделирования — 1 мс ($h = 10^{-3}$); время распространения сигнала по сети не превышает 200 мс ($a = 200$); пропускная способность маршрутизатора — 15 Мбит/с = 1,92 Кбайт/мс ($C = 1,92$); объем буфера маршрутизатора — 50 Кбайт ($B = 50$). В рассматриваемом случае максимальное число шагов моделирования K , в течение которых необходимо сохранять отслеживаемые параметры, в соответствии с формулой (12) составляет

$$K = \frac{200 + \frac{50}{1,92}}{1} \approx 226,$$

максимально допустимый объем памяти, который может быть использован для размещения данных:

$$V_M^{(\max)} \approx V_{AM} \cdot 0,9 \approx 512 \text{ Мбайт} \cdot 0,9 \approx 460 \cdot 2^{20} \text{ Байт},$$

здесь коэффициент 0,9 учитывает неизбежное использование в служебных целях самого GPU.

Объем фактической памяти V_M , используемой для моделирования N TCP-классов, в соответствии с формулой (13) равен

$$\begin{aligned} V_M &= 4 \cdot (8 + N) + 1 \cdot 2N + 4 \cdot (7 + 5N + 4K(N + 1)) = \\ &= 24 + 4N + 2N + 28 + 20N + 16K(N + 1) = \\ &= 26N + 16KN + 16K + 52 = \\ &= 3642N + 3668 \text{ Байт}. \end{aligned}$$

Приравнивая $V_M^{(\max)}$ и V_M , получаем следующее уравнение,

$$3642N + 3668 = 460 \cdot 2^{20} \text{ Байт},$$

из которого находим искомую оценку:

$$N \approx 2^{17}.$$

Для сравнения в работе [2] при моделировании были использованы только 100 информационных потоков.

Заключение

В статье описана методика расчета объемов памяти, необходимой для хранения параметров ГЖМ, и получены оценки максимально возможного числа моделируемых TCP-потоков при использовании технологии GPGPU.

Полученные оценки свидетельствуют о возможности кардинального увеличения числа одновременных моделируемых информационных потоков при использовании параллельной программной реализации в сравнении с последовательной программной реализацией ГЖМ, что, в свою очередь, подтверждает актуальность разработки.

Список литературы

1. **Басавин Д. А., Поршнева С. В.** Параллельная гибридная жидкостная модель высокоскоростных информационных потоков в магистральных Интернет-каналах // *Естественные и технические науки*. 2013. №1. С. 317–326.
2. **Гребенкин М. К., Поршнева С. В.** Гибридная жидкостная модель магистрального Интернет-канала. Saarbrücken: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. 163 с.
3. **Берлин А. Н.** Основные протоколы Интернет. М.: БИНОМ, 2008. 504 с.
4. **Олифер В. Г., Олифер Н. А.** Компьютерные сети. СПб.: Питер, 2007. 958 с.
5. **Looking** behind the automation protocols. URL: http://www.iebmedia.com/index.php?id=4582&parentid=63&themeid=255&sho_wdetail=true
6. **Архив** образцов трафика. URL: <http://ita.ee.lbl.gov/index.html>
7. **Архив** трафика MAWI Working Group. URL: <http://mawi.wide.ad.jp/mawi/>
8. **Liu Y. F., Presti L., Misra V., Towsley D., Gu Y.** Fluid Models and Solutions for Large-Scale IP Networks // *ACM SIGMETRICS Performance Evaluation Review*. 2003. Vol. 31, Is. 1. P. 91–101.
9. **Гребенкин М. К., Поршнева С. В.** Программная реализация гибридной жидкостной модели информационных потоков в высокоскоростных магистральных интернет-каналах. Свидетельство о государственной регистрации программ для ЭВМ № 2012616118. Заявка № 2012613786. Дата поступления 11 мая 2012 г. Дата регистрации в Реестре программ для ЭВМ 4 июля 2012 г.
10. **Басавин Д. А., Поршнева С. В.** О целесообразности использования графических процессоров при моделировании крупных телекоммуникационных систем // *Актуальные вопросы в научной работе и образовательной деятельности: сб. науч. тр. по материалам Международной научно-практической конференции 31 января 2013 г.: в 13 частях. Часть 2*. Тамбов: Бизнес-Наука-Общество, 2013. С. 12–15.
11. **Вересина И.** Качество обслуживания в сетях IP. М.: Вильямс, 2003. 368 с.
12. **Misra V., Gong W.-B., Towsley D.** Stochastic Differential Equation Modeling and Analysis of TCP Window Size Behavior // *Proceedings of IFIP WG 7.3 Performance*, November, 1999. Vol. 99. URL: <http://dna-pubs.cs.columbia.edu/citation/paperfile/24/Misra99-TCP-Stochastic.pdf>.
13. **Pellicer-Lostao C., Morato D., Popez-Ruiz R.** Modelling user's activity in a real-world complex network // *International Journal of Computer Mathematics*. Bristol: Taylor & Francis. 2008. Vol. 85. P. 1287–1298.
14. **Lan K., Heidemann J.** A measurement study of correlations of Internet flow characteristics // *Computer Networks*. 2006. Vol. 50, Iss. 1. P. 46–62.
15. **Ergman J., Mahanti A., Arlitt M.** Byte Me: A Case for Byte Accuracy in Traffic Classification // *Proceedings of the 3rd annual ACM workshop on Mining network data*. San Diego, 2007. P. 35–38.
16. **Гребенкин М. К., Поршнева С. В.** Статистическая абстрактная модель интернет-трафика в мультисервисных сетях // *Науч. тр. междунар. науч.-практ. конференции "СВЯЗЬ-ПРОМ 2009"*. Екатеринбург: ЗАО "Компания Реал-Медиа", 2009. Т. 2. С. 55–58.
17. **Ma A. H. I., Schormans J.** Accelerated Simulation Modelling of Power-Law Traffic via Aggregation. ICT Beijing, 2002.
18. **Tang W., Fun Y., Cherkasova L., Vahdat A.** MediSyn: A Synthetic Streaming Media Services Workload Generator // *NOSSDAV'03, Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*. ACM New York, 2003. P. 12–21.
19. **Downey A. B.** Lognormal and Pareto Distributions in the Internet // *Computer Communications*. 2005. Vol. 28, Is. 7. P. 790–801.

Estimation of Maximum Number of Simulated Information Flows for Parallel Software Implementation of a Hybrid Fluid Model of Internet Traffic using GPGPU

D. A. Basavin, basavind@gmail.com, Belgorod State Agricultural University named after V. Gorin, Mayskiy, Belgorod region, 308503, Russian Federation, **S. V. Porsnev**, sergey_porshnev@mail.ru, Ural Federal University named after the first President of Russia B. N. Yeltsin, Ekaterinburg, 620002, Russian Federation, **D. A. Petrosov**, scorpions2002@mail.ru, Belgorod State Agricultural University named after V. Gorin, Mayskiy, Belgorod region, 308503, Russian Federation

Corresponding author:

Basavin Dmitry A., Assistant, Belgorod State Agricultural University named after V. Gorin, Mayskiy, Belgorod region, 308503, Russian Federation,
E-mail: basavind@gmail.com

Received on November 22, 2016

Accepted on March 02, 2017

The article discusses the hybrid fluid model (HFM) of Internet traffic, which describes the information flows in modern computer networks. Information flows in HFM are described by a system of ordinary differential equations (ODE). From the physical point of view this system of ODEs is the result of balance of information flows, which comes into the corresponding node of the network, and goes out of it. Due to the lack of analytical solutions, it is required to develop software tools to obtain numerical solutions of this system of ODEs included in HFM. The experience of practical usage of the well-known software HFM implementation with sequentially executable code showed that the time of calculating the information flows characteristics, even for relatively simple network configurations is quite large,

and for the networks with a complex architecture it is unacceptably high. In this regard, it has been hypothesized that it is possible to apply parallel computing techniques based on GPGPU to software HFM implementation development. It should also be noted that the GPGPU technology can accelerate the complex calculations in a few dozen times in a variety of cases. Also, the technology is now one of the leading and the publicly available ones. However, one of its limitations is stiff hardware limitation of available GPU memory. It is therefore necessary to establish consistency between the hardware parameters (the GPU) and the maximum number of simulated information flows (the size of the network and its architecture, respectively). The paper proposes an estimation of the maximum number of simulated information flows via the parallel software HFM implementation based on GPGPU technology.

Keywords: Internet traffic, computer networks, parallel hybrid fluid model, modeling, simulation, GPGPU

For citation:

Basavin D. A., Porsnev S. V., Petrosov D. A. Estimation of Maximum Number of Simulated Information Flows for Parallel Software Implementation of a Hybrid Fluid Model of Internet Traffic using GPGPU, *Programmnaya Ingeneria*, 2017, vol. 8, no. 5, pp. 195–206.

DOI: 10.17587/prin.8.195-206

References

1. **Basavin D. A., Porshnev S. V.** Parallelnaia gibridnaia zhidkostnaia model vysokoskorostnykh informatsionnykh potokov v magistralnykh Internet-kanalakh (Parallel Hybrid Fluid Model of High-Speed Information Flows in Backbone Internet Channels), *Estestvennye i tekhnicheskie nauki*, 2013, vol. 1, pp. 317–326 (in Russian).
2. **Grebenkin M. K., Porshnev S. V.** *Gibridnaia zhidkostnaia model magistralnogo Internet-kanala* (Hybrid Fluid Model of Backbone Internet Channel), Saarbrücken, LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012, 163 p. (in Russian).
3. **Berlin A. N.** *Osnovnye protokoly Internet* (Main Internet Protocols), Moscow, Binom, 2008, 504 p. (in Russian).
4. **Olifer V. G., Olifer N. A.** *Kompiuternye seti* (Computer Networks), Saint-Petersburg, Piter, 2007, 958 p. (in Russian).
5. **Schwab C.** Looking behind the automation protocols. November, 2004. *Industrial Ethernet Book*, available at: <http://www.iebmedia.com/index.php?id=4582&parentid=63&themeid=255&shoedetail=true>
6. **The Internet** Traffic Archive, available at: <http://ita.ee.lbl.gov/index.html>
7. **MAWI Working** Group Traffic Archive, available at: <http://mawi.wide.ad.jp/mawi/>
8. **Liu Y., LoPresti F., Misra V., Towsley D.** Fluid Models and Solutions for Large-Scale IP Networks, *Proc. ACM Sigmetrics*, San Diego, CA, USA, 2003, pp. 91–101.
9. **Grebenkin M. K., Porshnev S. V.** Programmnaya realizatsiia gibridnoj zhidkostnoj modeli informatsionnykh potokov v vysokoskorostnykh magistralnykh internet-kanalakh (Program Implementation of Hybrid Fluid Model of Information Flows in High-Speed Backbone Internet Channels), Svidetel'stvo o gosudarstvennoj registratsii programm dlya E'VM № 2012616118, 2012. (in Russian).
10. **Basavin D. A., Porshnev S. V.** O celesoobraznosti ispolzovaniia graficheskix processorov pri modelirovanii krupnykh telekommunikatsionnykh system (On the Feasibility of Using GPUs for Modeling Large Telecommunication Systems), *Proc. Aktualnye voprosy v nauchnoj rabote i obrazovatel'noj deyatel'nosti*, Tambov, Biznes-Nauka-Obshchestvo, 2013, vol. 2, pp. 12–15 (in Russian).
11. **Vegeshna Sh.** *Kachestvo obsluzhivaniya v setyax IP* (Quality of service in IP networks), Moscow, Vil'yams, 2003, 368 p. (in Russian).
12. **Misra V., Gong W., Towsley D.** Stochastic Differential Equation Modeling and Analysis of TCP Window Size Behavior, *Proceedings of IFIP WG 7.3 Performance*, November, 1999. Vol. 99, available at: <http://dna-pubs.cs.columbia.edu/citation/paperfile/24/Misra99-TCP-Stochastic.pdf>.
13. **Pellicer-Lostao C., Morato D., Popez-Ruiz R.** Modelling user's activity in a real-world complex network, *International Journal of Computer Mathematics*, 2008, vol. 85, pp. 1287–1298.
14. **Lan K., Heidemann J.** A measurement study of correlations of Internet flow characteristics, *Computer Networks*, 2006, vol. 50, issue 1, pp. 46–62.
15. **Erman J., Mahanti A., Arlitt M.** Byte Me: A Case for Byte Accuracy in Traffic Classification, *Proc. of the 3rd annual ACM workshop on Mining network data*, San Diego, USA, 2007, pp. 35–38.
16. **Grebenkin M. K., Porshnev S. V.** Statisticheskaya abt-model internet-trafika v multiservisnykh setiakh (Statistical abt-model of Internet traffic in multiservice networks), *SVIaZ-PROM 2009*, Ekaterinburg, ZAO "Kompanija Real-Media", 2009, vol. 2, pp. 55–58 (in Russian).
17. **Ma A. H.I., Schormans J. A.** *Accelerated Simulation Modeling of Power-Law Traffic via Aggregation*, Proc ICT, Beijing, China, 2002.
18. **Tang W., Fun Y., Cherkasova L., Vahdat A.** MediSyn: A Synthetic Streaming Media Services Workload Generator, *NOSSDAV'03, Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, ACM New York, 2003, pp. 12–21.
19. **Downey A. B.** Lognormal and Pareto Distributions in the Internet, *Computer Communications*, 2005, vol. 28, is. 7, pp. 790–801.

А. В. Поляков, аспирант, e-mail: andre.levsha@gmail.com, Московский государственный университет имени М. В. Ломоносова

Метод идентификации личности по отпечаткам пальцев на основе сферического локально-чувствительного хэширования

Представлен новый алгоритм идентификации личности по отпечаткам пальцев, предназначенный для больших баз данных, в которых хранятся подобные отпечатки. Данный алгоритм создан на основе метода сферического локально-чувствительного хэширования в сочетании с моделью аппроксимации поля ориентаций отпечатка пальца обобщенными многочленами Чебышева. Анализ и экспериментальные результаты показывают, что по сравнению с аналогичным по функциональным возможностям алгоритмом программная реализация представленного в настоящей статье алгоритма демонстрирует лучшую производительность и эффективность с позиций требований по точности и скорости идентификации.

Ключевые слова: биометрическая идентификация, локально-чувствительное хэширование, обобщенные полиномы Чебышева, поле ориентаций, симплекс

Введение

Биометрическая система аутентификации личности, основанная на отпечатках пальцев, работает в двух режимах — верификации и идентификации. В режиме верификации личности по отпечаткам пальца на вход биометрической системе подается идентификатор и отпечаток пальца, при этом система решает следующую задачу: похож или нет данный отпечаток на отпечаток в базе данных биометрической системы с тем же идентификатором (сравнение один к одному). В режиме идентификации личности по отпечаткам пальца на вход биометрической системе подается лишь один отпечаток пальца. При этом система решает следующую задачу: существуют ли в базе данных этой системы отпечатки, похожие на входящий отпечаток. Если такие отпечатки существуют, от системы требуется выдать короткий список отпечатков, наиболее похожих на данный. В настоящей статье будут обсуждены вопросы, которые возникают при использовании биометрических систем в режиме идентификации.

Отпечатки пальцев являются одной из наиболее изученных биометрических характеристик, работа с ними широко распространена в гражданских и судебных приложениях. Несмотря на то что программные реализации современных алгоритмов сравнения отпечатков пальцев достаточно быстры и точны [1–3], идентификация неизвестного отпечатка в больших базах данных все еще представляет собой сложную задачу как по точности, так и по

скорости исполнения. Для уменьшения общего числа сравнений отпечатков в биометрических системах используют методы предварительного отсеивания. Эти методы можно разделить на два класса:

- явная классификация;
- непрерывная классификация (индексирование).

Методы явной классификации разбивают всю базу данных отпечатков на фиксированное число классов, и в процессе идентификации отпечаток пальца сравнивается с другими отпечатками из базы данных только из того же класса.

Основная трудность реализации этих методов заключается в том, что число классов мало и отпечатки пальцев распределены по ним неравномерно. Поэтому при поиске отпечатка в большой по объему объектов базе данных зачастую явные методы классификации не способны значительно сузить пространство поиска.

Это побудило исследователей к разработке методов поиска, которые не основываются на явной классификации. В частности, в 1998 г. в работе [4] был предложен метод локально-чувствительного хэширования, основная идея которого состоит в таком подборе хэш-функций для некоторых измерений, чтобы похожие объекты с высокой вероятностью оказались в одной корзине. Этот метод позволяет строить структуру для быстрого приближенного поиска объектов, похожих по некоторой мере на искомый шаблон.

Формальная постановка задачи поиска приближенного ближайшего соседа представлена далее в определении 1.

Методы непрерывной классификации сами могут быть классифицированы на основе перечисленных далее признаков отпечатков пальцев, которые в них используются:

- глобальные признаки, в частности, поле ориентаций папиллярных линий [5–8] — как правило, поле ориентаций преобразуется в числовой вектор и выбирается такое преобразование, чтобы похожие отпечатки отображались в близкие точки соответствующего многомерного пространства;
- локальные признаки, которые характеризуют особые точки отпечатков пальцев (минуции) — большинство подходов непрерывной классификации используют геометрические признаки, полученные из минуциальных треугольников [9–12].

В настоящей статье представлен новый алгоритм идентификации личности по отпечаткам пальцев для больших по объему таких отпечатков баз данных. Этот алгоритм создан на основе метода сферического локально-чувствительного хэширования в сочетании с моделью аппроксимации поля ориентаций отпечатка пальца обобщенными многочленами Чебышева. Анализ и экспериментальные результаты показывают, что по сравнению с аналогичным по назначению алгоритмом [13] программная реализация представленного алгоритма демонстрирует лучшую вычислительную производительность и эффективность с позиции требований по точности идентификации.

В структурном плане настоящая статья организована следующим образом.

В разд. 1 представлена формальная постановка задачи поиска приближенного ближайшего соседа, введены основные сведения и определения метода локально-чувствительного хэширования и сферического локально-чувствительного хэширования, требующиеся для дальнейшего изложения.

В разд. 2 представлена математическая модель отпечатка пальца на основе аппроксимации поля направлений обобщенными многочленами Чебышева.

В разд. 3 представлен алгоритм идентификации личности по отпечатку пальца на основе сферического локально-чувствительного хэширования, показаны оценки его сложности и приведены экспериментальные результаты.

В Заключении представлены выводы и указаны направления дальнейших исследований.

1. Предварительные сведения и определения

В этом разделе содержится формальная постановка задачи поиска отпечатка, как задача поиска приближенного ближайшего соседа в многомерном пространстве признаков. Представлено описание центральной идеи описываемого алгоритма идентификации личности по отпечаткам пальцев — метода локально-чувствительного хэширования и сферического локально-чувствительного хэширования.

1.1 Формальная постановка задачи поиска приближенного ближайшего соседа

Определение 1. Задача поиска c — приближенного ближайшего соседа (c — *approximate nearest neighbor*, c — *ANN*) формулируется следующим образом: дано множество точек P в метрическом пространстве $\mathcal{M} = (\mathbb{R}^d, \rho)$, где \mathbb{R}^d — d -мерное вещественное векторное пространство, ρ — метрика на нем; требуется построить структуру данных, которая для каждого запроса $q \in \mathbb{R}^d$ находит точку $p \in P$, удовлетворяющую следующему условию: $\forall p' \in P : \rho(p, q) \leq c\rho(p', q)$.

Определение 2. Задача поиска (R, c) — приближенного случайного соседа состоит в следующем. Пусть дано множество точек P в метрическом пространстве $\mathcal{M} = (\mathbb{R}^d, \rho)$ и параметр $R > 0$, требуется построить структуру данных, которая для каждого запроса $q \in \mathbb{R}^d$ выполняет следующее:

- если существует точка $p \in P : \rho(p, q) \leq R$, возвращает ответ "да" и точку $p' \in P : \rho(p', q) \leq cR$;
- если $\rho(p, q) > cR$ для всех $p \in P$, возвращает ответ "нет".

1.2. Локально-чувствительное хэширование и сферическое локально-чувствительное хэширование

Ранние работы по индексированию отпечатков пальцев были основаны на вложении пространства биометрических признаков в некоторое пространство поиска. Каждой точке пространства поиска соответствовал прообраз в пространстве оригинальных признаков отпечатка пальца в предположении, что если вложение сохраняет меру схожести, то близкие по этой мере векторы биометрических признаков перейдут в близкие точки многомерного пространства поиска. Далее поиск проводился сравнением входящего отпечатка пальца с теми отпечатками из базы данных, чьи соответствующие векторы близки к запрашиваемому отпечатку. Большинство из ранних подходов могут быть грубо классифицированы на основе биометрических признаков и различных преобразований.

В 1998 г. в работе [4] был представлен метод локально-чувствительного хэширования (LSH), что открыло путь для быстрого поиска по биометрическим базам данных больших объемов. Применение данного метода на больших базах данных показывает существенный выигрыш в скорости при незначительной потере точности.

Локально-чувствительное хэширование преследует, как правило, две цели: уменьшение размерности строк входящего шаблона изображения отпечатка пальца и конструирование хэш-таблиц путем кластеризации минуций. В случае бинарных векторов простые LSH-функции сохраняют расстояние Хэмминга (так как вероятность коллизии двух хэшей равна доле позиций битов, в которых совпадают две строки).

Большинство существующих и реализованных программно алгоритмов вложений множества особых точек отпечатков пальцев во множество бинарных

строк включают в себя либо прямую дискретизацию биометрических признаков, либо некоторую статистику признаков. Процесс дискретизации неизбежно влечет информационные потери и искажения. В настоящей статье показано, как метод локально-чувствительного хэширования может быть применен прямо к минимальным шаблонам без дискретизации.

Локально-чувствительное хэширование [4] — это вероятностный алгоритм решения задачи поиска приближенного ближайшего соседа. Метод LSH-индексирования основан на семействе локально-чувствительных функций H , отображающих некоторую область X в d -мерном пространстве в счетное множество S со следующим свойством: близкие точки в метрическом пространстве хэшируются одним и тем же значением с высокой вероятностью.

Введем формальные определения согласно работе [4].

Пусть $\|\cdot\|$ — евклидова норма в \mathbb{R}^d ; S^{d-1} — единичная гиперсфера с центром в начале координат; μ — равномерное распределение на S^{d-1} , $\mu(S^{d-1}) = 1$; \mathbb{B}^k — множество k -мерных булевых векторов; P_H — вероятность.

Определение 3. Семейство $H = \{h: S^{d-1} \rightarrow \mathbb{B}^k\}$ называется (r, c, p_1, p_2) — чувствительным ($r > 0, c \geq 1, p_1 > p_2$), если $\forall x, y \in S^{d-1}$:

- $P_H(h(x) = h(y)) \geq p_1$ при $\|x - y\| \leq r$ (вероятность коллизии любой хэш-функции h из семейства H не менее p_1 для достаточно близких точек на единичной гиперсфере);

- $P_H(h(x) = h(y)) \leq p_2$ при $\|x - y\| \geq cr$ (вероятность коллизии любой хэш-функции h из семейства H не более p_2 для достаточно далеких точек на единичной гиперсфере).

В работе [14] был предложен метод сферического локально-чувствительного хэширования на единичной гиперсфере. Идеей данного метода являлось вписывание в единичную гиперсферу правильного d -мерного многогранника.

Определение 4. Задача поиска (R, c) — приближенного случайного соседа на единичной гиперсфере состоит в следующем. Пусть дано множество точек P в метрическом пространстве $\mathcal{M} = (\mathbb{R}^d, \rho)$, причем $\forall p \in P, \|p\| = 1$, задан параметр $R > 0$. Требуется построить структуру данных, которая для каждого запроса $q \in \mathbb{R}^d$ выполняет следующее:

- если существует точка $p \in P: \rho(p, q) \leq R$, возвращает ответ "да" и точку $p' \in P: \rho(p', q) \leq R$;
- если $\rho(p, q) > cR$ для всех $p \in P$, возвращает ответ "нет".

2. Математическая модель отпечатка пальца, основанная на аппроксимации поля направлений обобщенными многочленами Чебышева

Содержательно в данном разделе опишем процедуру построения вектора признаков отпечатка пальца на основе поля ориентацией (направления потока папиллярных линий в каждой точке изображения

отпечатка пальца), а затем — процесс аппроксимации поля ориентаций обобщенными многочленами Чебышева (ортогональными полиномами от двух переменных).

Полиномы Чебышева являются важным семейством ортогональных полиномов в численных методах и основанных на них вычислениях.

Определение 5. Для любого $j > 0$ полином Чебышева степени j обозначается через $T_j(x)$ и определяется следующим образом:

$$T_j(x) = \cos(j \arccos(x)), \quad x \in [-1, 1].$$

Полиномы Чебышева ортогональны относительно взвешенного скалярного произведения, т. е.

$$\int_{-1}^1 \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi, & \text{если } i=j=0; \\ \frac{\pi}{2}, & \text{если } i=j \geq 1; \\ 0, & \text{если } i \neq j; \end{cases}$$

и, следовательно, удовлетворяют рекуррентному соотношению

$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x), \\ j \geq 1, T_0(x) = 1, T_1(x) = x.$$

Обобщение полиномов Чебышева на случай двух переменных рассмотрено в работе [15]. Рассмотрим полиномы Чебышева $T_{\tau-\eta}(x)$ и $T_\eta(y)$, где τ, η — неотрицательные целые числа. Тогда можно вычислить множество базисных функций для обобщенного полинома Чебышева k -го порядка следующим образом:

$$T_{\tau\eta}(x, y) = T_{\tau-\eta}(x)T_\eta(y), \quad \tau = \overline{0, k}, \eta = \overline{0, \tau}.$$

Одной из областей, в которой полиномы Чебышева играют центральную роль, является минимаксная аппроксимация функций многочленами. Структура процесса аппроксимации включает три компонента:

- класс функций (содержащий функцию, которую требуется аппроксимировать);
- форму (для аппроксимирующей функции);
- норму (ошибку аппроксимации), в рамках которой ставится задача аппроксимации.

Будем рассматривать: класс $C[a, b]$ непрерывных функций на отрезке $[a, b]$, ($a < b, a, b \in \mathbb{R}$); аппроксимацию полиномами; норму Чебышева $L_\infty: \|f\|_\infty = \max_{a \leq x \leq b} |f(x)|$.

Сформулируем задачу минимаксной аппроксимации.

Определение 6. Пусть F — нормированное линейное пространство, $f \in F, A$ — подпространство F . Аппроксимация $f_B \in A$ называется минимаксной (или наилучшей), если $\forall f^* \in A$ справедливо:

$$\|f - f_B\| \leq \|f - f^*\|.$$

При аппроксимации непрерывной функции полиномами на отрезке в одномерном случае всегда можно получить хорошую аппроксимацию, взяв в качестве аппроксимирующего многочлен доста-

точно высокой степени. Это следствие из представленной далее теоремы 1.

Теорема 1. Пусть $a < b$, $a, b \in \mathbb{R}$, $\forall f \in C[a, b]$ и $\forall \varepsilon > 0$ существует полином P_n достаточно большой степени n такой, что $\|f - f^*\|_\infty < \varepsilon$.

Доказательство теоремы 1 можно найти, например, в работе [16].

Более того, справедливы представленные далее теоремы 2—4.

Теорема 2. Для любого p , $1 \leq p < \infty$ существует и единственна минимаксная полиномиальная аппроксимация P_n для любой функции $f \in L_p[\alpha, \beta]$, где $\alpha < \beta$, $\alpha, \beta \in \mathbb{R}$.

В условиях теоремы 2 L_p — пространство интегрируемых функций, p -я степень которых интегрируема.

Доказательство теоремы 2 приведено в работе [16].

Теорема 3 (Минимаксная аппроксимация для вещественных функций двух переменных).

Пусть $f: [\alpha, \beta] \times [\gamma, \zeta] \rightarrow \mathbb{R}$ — непрерывна, где $\alpha < \beta$, $\gamma < \zeta$, $\alpha, \beta, \gamma, \zeta \in \mathbb{R}$, λ и ν — целые числа. Тогда существует полином $p_{\lambda, \nu}^{\text{лучший}}(x, y) \in \mathbb{R}[x, y]$ степени не более чем λ по x и не более чем ν относительно y , такой, что

$$\|f - p_{\lambda, \nu}^{\text{лучший}}(x, y)\|_\infty = \inf_{p \in \mathbb{R}[x, y]} \|f - p\|_\infty.$$

Доказательство теоремы 3 представлено в работе [17].

Теорема 4 (Непрерывная сходимости обобщенных полиномов Чебышева от двух переменных).

Пусть $f: [-1, 1] \times [-1, 1] \rightarrow \mathbb{R}$ — непрерывная функция ограниченной вариации, у которой существует хотя бы одна ограниченная на компакте $[-1, 1] \times [-1, 1]$ частная производная. Тогда функция f представима в виде

$$f(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{ij} T_i(y) T_j(x),$$

где $a_{ij} \in \mathbb{R}$, причем ряды сходятся к f равномерно.

Доказательство теоремы 4 приведено в работе [16].

Для аппроксимации поля ориентаций используются идеи статьи [18], однако в качестве базисных функций вместо преобразования Фурье используются обобщенные полиномы Чебышева.

Для вычисления поля ориентаций применяется градиентный метод. Его суть в том, что поле ориентаций вычисляется попиксельно и не усредняется. Обозначим удвоенное поле ориентаций в точке (x, y) через $2O(x, y)$.

Для аппроксимации двумерной дискретной функции $f(x, y)$ будем использовать линейную комбинацию s базисных функций:

$$f(x, y) = \sum_{j=0}^s a_j \varphi_j(x, y), \quad a_j \in \mathbb{R}, \quad j = \overline{0, s}.$$

Положив $\Phi_j(\mathbf{x}) = \Phi_j(x, y) = [\varphi_0(x_j, y_j), \dots, \varphi_s(x_j, y_j)]$, определим матрицу системы

$$\mathbf{V} = \begin{pmatrix} \Phi(\mathbf{x}_0) \\ \Phi(\mathbf{x}_1) \\ \dots \\ \Phi(\mathbf{x}_i) \end{pmatrix} = \begin{pmatrix} \varphi_0(x_0, y_0), \varphi_1(x_0, y_0), \dots, \varphi_s(x_0, y_0) \\ \varphi_0(x_1, y_1), \varphi_1(x_1, y_1), \dots, \varphi_s(x_1, y_1) \\ \dots \\ \varphi_0(x_i, y_i), \varphi_1(x_i, y_i), \dots, \varphi_s(x_i, y_i) \end{pmatrix},$$

где размер матрицы \mathbf{V} определяется количеством точек и базисных функций.

Далее пусть, есть вектор параметров $\mathbf{a} = [a_0, \dots, a_s]^T$, и вектор значений функции $\mathbf{f}(\mathbf{x}_j) = [f_0(\mathbf{x}_j), \dots, f_s(\mathbf{x}_j)]$, где $f(\mathbf{x}_k)$ — наблюдаемое значение функции в точке (x_k, y_k) . Далее методом наименьших квадратов моделируем f , минимизируя целевую функцию ошибок

$$F = \sum_{j=1}^i \omega_j [\Phi(\mathbf{x}_j) \mathbf{a}^T - f(\mathbf{x}_j)]^2,$$

где $\omega_1, \dots, \omega_i$ — весовые коэффициенты.

После этого решение вычисляется с помощью матрицы $(\mathbf{V}^T \mathbf{W} \mathbf{V})^{-1} \mathbf{V}^T \mathbf{W} \mathbf{f}$, где $\mathbf{W} = \text{diag}(\omega_1, \dots, \omega_i)$ — диагональная матрица весов.

Оптимизация для получения конечных параметров выполняется в два этапа. На первом этапе происходит грубая аппроксимация параметров модели методом наименьших квадратов с взвешенным псевдообратным преобразованием. На втором этапе происходит нелинейное уточнение параметров модели.

Пусть \mathbf{a} и \mathbf{b} — векторы параметров модели Чебышева для данных синусов и косинусов углов ориентации папиллярных линий, соответственно. Тогда само поле ориентаций может быть описано следующим образом:

$$O(x_j) = \frac{1}{2} \arctg \frac{\Phi(\mathbf{x}_j) \mathbf{a}^T}{\Phi(\mathbf{x}_j) \mathbf{b}^T}.$$

Для минимизации среднеквадратичной ошибки требуется минимизировать функционал

$$\min_{\mathbf{a}, \mathbf{b}} \sum_{j=1}^i \omega_j \left[\sin \left(\frac{1}{2} \arctg \frac{\Phi(\mathbf{x}_j) \mathbf{a}^T}{\Phi(\mathbf{x}_j) \mathbf{b}^T} - O(x_j) \right) \right]^2. \quad (1)$$

На первом шаге оптимизации требуется независимо вычислить данные для синусов и косинусов углов поля ориентации папиллярных линий $O(x, y)$. Вычисление происходит с помощью градиентного метода. Далее вычисляется псевдообратная матрица системы $\mathbf{a}_o = \mathbf{V}_w^+ \mathbf{W} \mathbf{f}_1$, $\mathbf{b}_o = \mathbf{V}_w^+ \mathbf{W} \mathbf{f}_2$, где

$$\mathbf{f}_1 = [\sin 2O(x_1), \dots, \sin 2O(x_i)]^T,$$

$$\mathbf{f}_2 = [\cos 2O(x_1), \dots, \cos 2O(x_i)]^T.$$

На втором шаге целевая функция (1) минимизируется нелинейными алгоритмами оптимизации, например, методом Ньютона.

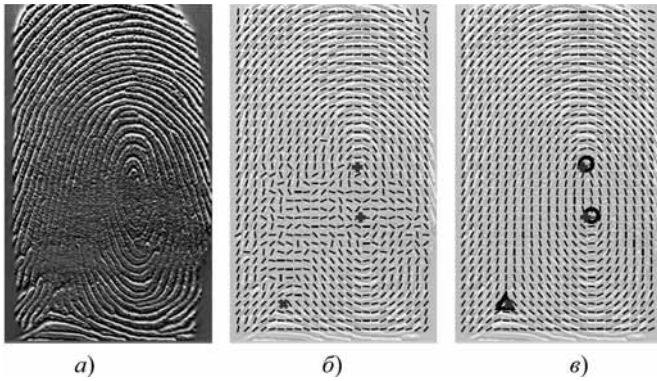


Рис. 1. Результат аппроксимации многочленами Чебышева поля направлений:

a — оригинальное изображение; *b* — поле ориентаций, полученное методом градиентов; *в* — аппроксимация многочленами Чебышева (42 параметра)

На рис. 1 представлен результат работы программной реализации алгоритма аппроксимации поля ориентаций многочленами Чебышева.

Коэффициенты модели Чебышева представляют собой вектор признаков для описания локальных особенностей в каждой минучиальной точке.

Далее для каждого отпечатка составляется матрица схожести минутий по косинусной мере с последующим двумерным шкалированием. Таким образом строится проекция биометрического шаблона на единичную гиперсферу.

3. Алгоритм идентификации личности по отпечатку пальца на основе сферического локально-чувствительного хэширования

Впишем в единичную гиперсферу правильный d -мерный многогранник. Пусть v_1, \dots, v_N — вершины многогранника, N — число вершин, \mathbf{A} — матрица вращения, тогда для точки $u \in S^{d-1}$ хэш-функция определена следующим образом:

$$h_A(u) := \arg \min_i \|\mathbf{A}v_i - u\|.$$

Путем случайного вращения многогранника получаем семейство хэш-функций $H = \{h_A\}$.

Существует всего три типа правильных многогранников в пространствах размерности $d \geq 5$ [19]: симплекс; гипероктаэдр; гиперкуб. При этом симплекс имеет $(d+1)$ вершину, гипероктаэдр имеет $2d$ вершин, гиперкуб имеет 2^d вершин.

Координаты вершин могут быть вычислены следующим образом.

- Симплекс:

$$[v_i]_j = \delta_{ij} - \frac{d+1-\sqrt{d+1}}{d(d+1)}, \quad i = 1, \dots, N;$$

$$[v_{N+1}]_j = \frac{1-\sqrt{d+1}}{d} - \frac{d+1-\sqrt{d+1}}{d(d+1)},$$

где $[v_i]_j$ — j -я координата i -й вершины; δ_{ij} — символ Кронекера.

- Гипероктаэдр: все перестановки $(\pm 1, 0, \dots, 0)$ задают нужные координаты вершин.

- Гиперкуб: координаты вершин в этом случае определяются как $\frac{1}{\sqrt{d}} (\pm 1, 0, \dots, \pm 1)$.

В случае нормализованных векторов евклидово расстояние между двумя векторами эквивалентно их косинусной мере, что позволяет эффективно вычислять хэш-функцию следующим образом:

$$h_A(u) = \arg \max_i (\mathbf{A}v_i u).$$

В работе [14] показано, что в пространстве размерности d временная сложность вычисления $h_A(u)$ составляет $\mathcal{O}(d^2)$ для любого многогранника, пространственная сложность вычисления — $\mathcal{O}(d^2 L)$ на хранение повернутых вершин многогранника, $\mathcal{O}(nL)$ — на хранение индекса всех элементов множества, где L — число хэш-таблиц, n — число элементов множества.

При этом в той же работе показано, что для каждой хэш-таблицы требуется $\mathcal{O}(d^3 + dn)$ времени на преобработку данных (d^3 — на формирование случайной матрицы вращения, dn — на хэширование всех элементов множества).

Заметим, что представленный в работе [14] алгоритм не является оптимальным. Действительно, в работе [20] представлены следующие результаты.

Теорема 5. Существует алгоритм решения задачи (R, c) -ANN с временной сложностью $\mathcal{O}(dn^\rho)$ и пространственной сложностью $\mathcal{O}(n^{1+\rho} + dn)$, где $\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2}$.

Доказательство оптимальности данной оценки приведено в работе [21].

В работе [20] указан метод формирования псевдослучайной матрицы вращения с помощью преобразования Адамара, а именно, $x \rightarrow \mathbf{H}\mathbf{D}_3\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_1x$, где \mathbf{H} — матрица Адамара, а $\mathbf{D}_i, i = 1, 2, 3$ — случайные диагональные ± 1 -матрицы. Следовательно, за счет применения быстрого преобразования Адамара ортогональное преобразование может быть построено за время $\mathcal{O}(d)$, причем на это построение требуется $\mathcal{O}(d \log d)$ памяти.

Таким образом, если в алгоритме из работы [14] заменить блок формирования псевдослучайной матрицы, то временная сложность алгоритма снижается до $\mathcal{O}(dn)$, а пространственная сложность — до $\mathcal{O}(n^2 + dn)$.

Псевдокод модифицированного сферического LSH-алгоритма представлен на рис. 2.

Алгоритм состоит из трех этапов: преобработки, создания индекса и поиска. На этапе преобработки задается размерность пространства d и правильный многогранник (в программной реализации — симплекс), создаются псевдослучайные матрицы вращения, которые задают семейство хэш-функций

```

1 Алгоритм хэширования отпечатков
2
3 Предобработка (размерность d)
4 Пусть  $\{v_i, i = 1, \dots, d+1\}$  - вершины вписанного в гиперсферу симплекса
5 Для каждой хэш-функции  $h_{A_k}$  в хэш-таблице  $H_1$ :
6     Создать случайную матрицу вращения  $A_k$ 
7     Повернуть симплекс  $v_i \leftarrow A_k * v_i$ 
8
9 Вставка S-LSH ( $T_i = \{m_1, m_2, \dots\}$ ) # На вход даем шаблон минутий
10
11 Для каждой  $m_j$  в  $T_i$ :
12     Для каждой хэш-функции  $h_{A_k}$  в таблице  $H_1$ :
13          $b_k = h_{A_k}(m_j)$ 
14         вставить  $(i, j)$  в корзину  $b_k$  таблицы  $H_1$ 
15
16
17 Поиск S-LSH( $T_q, \{H_1\}, \text{max\_score}$ )
18  $C = []$  # счетчик коллизий
19  $s = []$  # скоринговый массив
20 Для каждой  $v_j$  из  $T_q$ :
21     Для каждой хэш-функции  $h_{A_k}$  из  $H_1$ :
22          $b_k = h_{A_k}(v_j)$ 
23         Для каждой  $(i, j)$  в корзине  $b_k$  таблицы  $H_1$ :
24              $C(i, j) = C(i, j) + 1$ 
25 Для каждого индекса  $i$  из шаблона:
26     если  $C(i, j) > 0$ :
27          $s(i) += 1$ 
28 Отсортировать  $s$  по убыванию
29 Вернуть  $\text{max\_score}$  - индексы с максимальным скорингом

```

Рис. 2. Псевдокод модифицированного S-LSH алгоритма

$H = \{h_1, \dots, h_j\}$ с последующим вычислением новых координат вершин симплекса после вращения.

На этапе создания индекса на вход алгоритму подается база отпечатков $DB = \{T_1, \dots, T_z\}$, где z — число отпечатков в базе. При этом каждый отпечаток представлен своим множеством минутий: $T_i = \{m_1, \dots, m_{i_n}\}$, где b_i — общее число минутий в отпечатке T_i .

После этого для каждого шаблона $T_i \in DB$ и для каждой минутии m_j в нем строится вектор признаков v_j с использованием аппроксимации поля ориентации обобщенными полиномами Чебышева. Для каждого вектора признаков и для каждой хэш-функции h_{A_k} , заданной матрицей вращения A_k , вычисляется хэш $b_k = h_{A_k}(v_j)$, после чего паре (i, j) ставится в соответствие корзина b_k . Таким образом на выходе данного этапа формируется множество хэш-таблиц $\{H_1, \dots, H_L\}$.

На этапе поиска на вход алгоритму подается запрос T_q , множество хэш-таблиц $\{H_1, \dots, H_L\}$ и параметр max_score — максимальное число кандидатов в итоговом коротком списке. Хэш-функции применяются к каждому вектору признаков v_q , полученному посредством аппроксимации поля ориентации обобщенными полиномами Чебышева в окрестности минутии m_q , $j = \overline{1, b_q}$, где b_q — общее число минутий в шаблоне T_q . После этого вычисляется общее число коллизий хэш-функций, и max_score шаблонов с наибольшим числом коллизий формируют итоговый список выдачи.

3.1. Экспериментальные результаты

Рабочая характеристика системы биометрической классификации, как правило, определяется как функция уровня ошибок индексирования от процента базы данных отпечатков, по которой проводился фактический поиск. Уровень ошибок, в свою очередь, определяется как процент отпечатков в тестовой выборке, которые не были найдены. Уровнем проникновения называется доля базы данных отпечатков, в которой система должна вести поиск.

Сама отмеченная выше функция зависит от параметра max_score , определяющего максимальное число рассматриваемых кандидатов на похожие отпечатки (см. рис. 2).

Целью эксперимента являлось определение уровня точности алгоритма на тестовой базе отпечатков, определение скорости работы его программной реализации и сравнение рабочих характеристик программной реализации алгоритма, описанного в этой статье, с программной реализацией алгоритма MCC-LSH, описанного в работе [13].

Эксперимент проводили на публичной базе данных отпечатков FVC2004 DB1 [22] (800 изображений отпечатков пальцев всего, по 8 изображений отпечатка каждого из 100 личностей). Один отпечаток каждого человека брали для создания индекса (всего 100 записей), по остальным отпечаткам пальцев вели поиск.

Для моделирования поля ориентаций многочленами Чебышева использовался пакет CHEBFUN [23] для Matlab и экстрактор NBIS [24], для алгоритмов

Сравнение рабочих характеристик программных реализаций алгоритмов S-LSH и MCC-LSH

Уровень проникновения, %	Уровень ошибок, %	
	S-LSH	MCC-LSH
5	12	20
10	6	14
15	5	12
20	4	10
25	3	8
30	3	6

локально-чувствительного хэширования использовались функции библиотеки FALCONN [25].

Характеристики используемой ЭВМ: Intel Core i5-2500 CPU @3.30 GHz 3.30GHz, 4 Гб ОЗУ, ОС Windows 7.

Время исполнения программной реализации S-LSH-алгоритма составило 0,53 мс на создание индекса и 3,08 мс на поиск (для сравнения, для программной реализации алгоритма, описанного в работе [13], время на создание индекса составило 0,52 мс и 3,12 мс на поиск).

Сравнение рабочих характеристик программных реализаций алгоритмов S-LSH и MCC-LSH представлено в таблице. Как видно из данных таблицы, результаты работы программной реализации алгоритма S-LSH превосходят по точности соответствующие результаты работы программной реализации алгоритма MCC-LSH. Например, на уровне проникновения в 5 % уровень ошибки для программной реализации S-LSH алгоритма составил 12 %, а для программной реализации MCC-LSH — 20 %.

Заключение

В настоящей статье представлен алгоритм биометрического поиска по отпечаткам пальцев на основе локально-чувствительного хэширования, при этом в отличие от алгоритма из этого класса, представленного в работе [13], хэши отпечатков пальцев лежат в исходном пространстве признаков, не подвергаясь дискретизации. Эксперименты показывают хорошие результаты работы алгоритма по сравнению с существующими.

Представляется, что метод сферического локально-чувствительного хэширования является перспективным инструментальным средством биометрического поиска и на данном этапе может быть использован для поиска по базам данных, содержащим несколько десятков тысяч биометрических шаблонов.

Вместе с тем предметом дальнейших исследований является определение теоретических и практических условий для качественного уменьшения уровня ошибок при уровне проникновения базы данных меньше 5 %, поскольку полученные характеристики не позволяют таким образом "безболезненно" уменьшить пространство поиска на несколько порядков. Это обстоятельство приводит к ограничению использования данного метода на больших базах данных (порядка нескольких десятков миллионов биометрических образцов и более).

Список литературы

1. **Maltoni D., Maio D., Jain A. K., Prabhakar S.** Handbook of Fingerprint Recognition, second ed. Springer, 2009.
2. **Поляков А. В.** Алгоритм сравнения отпечатков пальцев на основе структуры созвездий // Программная инженерия. 2015. № 8. С. 26—31.
3. **Поляков А. В., Ковалев И. М.** Алгоритм сравнения отпечатков пальцев на основе поиска максимального пути в графе // Интеллектуальные системы. Теория и приложения. 2016. Т. 20, № 2. С. 103—117.
4. **Indyk P., Motwani R.** Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality // Proc. 13th ACM Symp. Theory of Computing, 1998. P. 604—613.
5. **Lumini A., Maio D., Maltoni D.** Continuous versus Exclusive Classification for Fingerprint Retrieval // Pattern Recognition Letters. 1997. Vol. 18. P. 1027—1034.
6. **Cappelli R., Maio D., Maltoni D.** A Multi-Classifer Approach to Fingerprint Classification // Pattern Analysis and Applications. 2002. Vol. 5. P. 136—144.
7. **Cappelli R., Lumini A., Maio D., Maltoni D.** Fingerprint Classification by Directional Image Partitioning // IEEE Trans. Pattern Analysis and Machine Intelligence, 1999. Vol. 21, No. 5, P. 402—421.
8. **Lee S., Kim Y., Park G.** A Feature Map Consisting of Orientation and Inter-Ridge Spacing for Fingerprint Retrieval // Proc. Audio and Video-Based Biometric Person Authentication. 2005. P. 184—190.
9. **Germain R. S., Califano A., Colville S.** Fingerprint Matching Using Transformation Parameter Clustering // IEEE Computational Science and Eng., 1997. Vol. 4, No. 4. P. 42—49.
10. **Bhanu B., Tan X.** Fingerprint Indexing Based on Novel Features of Minutiae Triplets // IEEE Trans. Pattern Analysis and Machine Intelligence. 2003. Vol. 25, No. 5. P. 616—622.
11. **Liang X., Asano T., Bishnu A.** Distorted Fingerprint Indexing Using Minutia Detail and Delaunay Triangle // Proc. Third Int'l Symp. Voronoi Diagrams in Science and Eng. 2006. P. 217—223.
12. **De Boer J., Bazen A. M., Gerez S. H.** Indexing Fingerprint Databases Based on Multiple Features // Proc. Ann. Workshop Circuits, Systems and Signal Processing, 2001. P. 300—306.
13. **Cappelli R., Ferrara M., Maltoni D.** Fingerprint indexing based on minutia cylinder-code // Pattern Analysis and Machine intelligence, IEEE Transactions on, 2011. Vol. 33, No. 5. P. 1051—1057.
14. **Terasawa K., Tanaka Y.** Spherical LSH for approximate nearest neighbor search on unit hypersphere // Algorithms and Data Structures. 2007. Vol. 4619. P. 27—38.
15. **Townsend A.** Computing with functions in two dimensions, PhD Thesis, University of Oxford, 2014.
16. **Mason J. C., Handscomb D. C.** Chebyshev Polynomials, CRC Press, Florida, 2003.
17. **Rice J. R.** Tchebycheff approximation in several variables // Trans. Amer. Math. Soc. 1963. Vol. 109. P. 444—466.
18. **Wang Y., Hu J., Phillips D.** A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular point detection and fingerprint indexing // Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2007. Vol. 29, No. 4. P. 573—585.
19. **Coxeter H. S. M.** Regular Polytopes (3rd ed.). Dover., 1973. 123 p.
20. **Razenshteyn I., Smidt L.** Practical and Optimal LSH for Angular Distance // 29th Annual Conference on Neural Information Processing Systems. 2015. arXiv:1509.02897.
21. **Andoni A., Razenshteyn I.** Tight lower bounds for data-dependent locality-sensitive hashing, 2015, arXiv:1507.04299.
22. **Система** тестирования алгоритмов верификации "FVC- onGoing" лаборатории биометрических систем Болонского университета (Италия). URL: <https://biolab.csr.unibo.it/FVCOnGoing/UI/Form/Home.aspx>
23. **Chebfun-numerical** computing with functions. URL: <http://www.chebfun.org/>
24. **NIST** Biometric Image Software. URL: <http://www.nist.gov/itl/iad/ig/nigos.cfm>
25. **FAst** Lookups of Cosine and Other Nearest Neighbors, URL: <https://falconn-lib.org/>

Fingerprint Identification using Spherical Local-Sensitive Hashing

A. V. Poliakov, e-mail: andre.levsha@gmail.com, Lomonosov Moscow State University, Moscow, 119234, Russian Federation

Corresponding author:

Poliakov Andrey V., Postgraduate student, Lomonosov Moscow State University, Moscow, 119234, Russian Federation
E-mail: andre.levsha@gmail.com

Received on February 18, 2017

Accepted on March 01, 2017

Fingerprint recognition is a well-studied research area in which significant results have been achieved. Fingerprint recognition can be divided into two tasks: verification and identification. Fingerprint verification is used to verify the identity of an individual by 1:1 matching whereas identification is used to establish the identity by 1:N matching. Fingerprint identification thus becomes more challenging than verification because of high system penetration and false acceptance rate.

In the resolution of this problem, indexing algorithms have a fundamental role. In the literature, there are several proposals that make use of different features to characterize fingerprints.

Fingerprint indexing is an efficient technique that greatly improves the performance of fingerprint based person authentication systems by reducing the number of comparisons.

This article presents a novel fingerprint indexing algorithm for large databases. A spherical locality-sensitive hashing scheme has been designed relying on generalized Chebyshev polynomials approximation model.

One general approach to LSH is to "hash" items several times, in such a way that similar items are more likely to be hashed to the same bucket than dissimilar items. One can consider any pair that hashed to the same bucket for any of the hashings to be a candidate pair.

The analysis and the experimental results show a good performance in terms of accuracy and computational complexity. Indexing experiments show remarkable results using public fingerprint database FVC2004 DB1.

Keywords: biometric identification, locality-sensitive hashing, generalized Chebyshev polynomials, orientation field, simplex

For citation:

Poliakov A. V. Fingerprint Identification using Spherical Local-Sensitive Hashing, *Programmnaya Ingeneria*, 2017, vol. 8, no. 5, pp. 207–214.

DOI: 10.17587/prin.8.207-214

References

1. **Maltoni D., Maio D., Jain A. K., Prabhakar S.** *Handbook of Fingerprint Recognition*, second ed., Springer, 2009.
2. **Poliakov A. V.** Algorithm sravneniya otpechatkov paltcev na osnove struktury sozvezdii (Fingerprint matching algorithm using a constellation structure), *Programmnaya Ingeneria*, 2015, no. 8, pp. 26–31 (in Russian).
3. **Poliakov A. V., Kovalev I. M.** Algoritm sravneniya otpechatkov paltcev na osnove poiska maksimalnogo puti v grafe (Fingerprint matching algorithm using a longest path problem), *Intellectualnye sistemy. Teoriya i prilozheniya*, 2016, vol. 20, no. 2, pp. 103–117 (in Russian).
4. **Indyk P., Motwani R.** Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality, *Proc. 13th ACM Symp. Theory of Computing*, 1998, pp. 604–613.
5. **Lumini A., Maio D., Maltoni D.** Continuous versus Exclusive, Classification for Fingerprint Retrieval, *Pattern Recognition Letters*, 1997, vol. 18, pp. 1027–1034.
6. **Cappelli R., Maio D., Maltoni D.** A Multi-Classifer Approach to Fingerprint Classification, *Pattern Analysis and Applications*, 2002, vol. 5, pp. 136–144.
7. **Cappelli R., Lumini A., Maio D., Maltoni D.** Fingerprint Classification by Directional Image Partitioning, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1999, vol. 21, no. 5, pp. 402–421.
8. **Lee S., Kim Y., Park G.** A Feature Map Consisting of Orientation and Inter-Ridge Spacing for Fingerprint Retrieval, *Proc. Audio and Video-Based Biometric Person Authentication*, 2005, pp. 184–190.
9. **Germain R. S., Califano A., Colville S.** Fingerprint Matching Using Transformation Parameter Clustering, *IEEE Computational Science and Eng.*, 1997, vol. 4, no. 4, pp. 42–49.
10. **Bhanu B., Tan X.** Fingerprint Indexing Based on Novel Features of Minutiae Triplets, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003, vol. 25, no. 5, pp. 616–622.
11. **Liang X., Asano T., Bishnu A.** Distorted Fingerprint Indexing Using Minutiae Detail and Delaunay Triangle, *Proc. Third Int'l Symp. Voronoi Diagrams in Science and Eng.*, 2006, pp. 217–223.
12. **De Boer J., Bazen A. M., Gerez S. H.** Indexing Fingerprint Databases Based on Multiple Features, *Proc. Ann. Workshop Circuits, Systems and Signal Processing*, 2001, pp. 300–306.
13. **Cappelli R., Ferrara M., Maltoni D.** Fingerprint indexing based on minutia cylinder-code, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011, vol. 33, no. 5, pp. 1051–1057.
14. **Terasawa K., Tanaka Y.** Spherical LSH for approximate nearest neighbor search on unit hypersphere, *Algorithms and Data Structures*, 2007, vol. 4619, pp. 27–38.
15. **Townsend A.** Computing with functions in two dimensions, PhD Thesis, University of Oxford, 2014.
16. **Mason J. C., Handscomb D. C.** *Chebyshev Polynomials*, CRC Press, Florida, 2003.
17. **Rice J. R.** Tchebycheff approximation in several variables, *Trans. Amer. Math. Soc.*, 1963, vol. 109, pp. 444–466.
18. **Wang Y., Hu J., Phillips D.** A fingerprint orientation model based on 2d Fourier expansion (fomfe) and its application to singular point detection and fingerprint indexing, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007, vol. 29, no. 4, pp. 573–585.
19. **Coxeter H. S. M.** *Regular Polytopes* (3rd ed.), Dover, 1973, 123 p.
20. **Razenshteyn I., Smidt L.** Practical and Optimal LSH for Angular Distance, *29th Annual Conference on Neural Information Processing Systems*, 2015, arXiv:1509.02897.
21. **Andoni A., Razenshteyn I.** Tight lower bounds for data-dependent locality-sensitive hashing, 2015, arXiv:1507.04299.
22. **FVC-onGoing:** on-line evaluation of fingerprint recognition algorithms, available at: <https://biolab.csr.unibo.it/FVConGoing/UI/Form/Home.aspx>
23. **Chebfun — numerical** computing with functions, available at: <http://www.chebfun.org/>
24. **NIST Biometric Image Software**, available at: <http://www.nist.gov/itl/iad/ig/nigos.cfm>
25. **FAst** Lookups of Cosine and Other Nearest Neighbors, available at: <https://falconn-lib.org/>

Н. Н. Светушков, канд. техн. наук, доц., e-mail: svetushkov@mai.ru,
Московский авиационный институт (национальный исследовательский университет)

Программный интерфейс простого графического 3D-редактора на основе метаязыка описания связей

Представлены основные требования и подходы, которые легли в основу разработки простейшего графического редактора трехмерных объектов. Для удобства пользователя предложено интерфейс программы полностью построить на контекстном меню, которое можно в дальнейшем легко усложнять и модифицировать путем добавления всевозможных опций и специальных функций. В качестве формата сохранения данных используется разработанный ранее метаязык описания связей, одним из преимуществ которого является простое описание сложных трехмерных сцен с возможностью поддержки установленных между объектами связей. Излагаемые принципы и подходы дают возможность создать программное средство редактирования, не перегруженное сложным иерархически организованным меню и, вместе с тем, упрощают пользователю процесс его освоения, так как предоставляют интуитивно понятный интерфейс. Графический редактор при этом может быть полностью написан на языке C++ и в дальнейшем использоваться как простое средство для создания трехмерных моделей для сторонних разработчиков. Цель проекта, результаты исследований по которому рассматриваются в статье, — предоставить пользователю простые интерактивные механизмы создания трехмерных объектов на основе лексем языка описания связей. С использованием открытого для пользователя кода можно разрабатывать специализированные приложения, включая как полностью информационные приложения, так и прикладные, например, для численного моделирования сложных физических процессов.

Ключевые слова: 3D-графика, язык описания трехмерных объектов, унифицированные принципы редактирования, интуитивно понятный интерфейс, контекстное меню, диалоговые окна, программные средства

Введение

Проектирование сложных трехмерных объектов представляет собой довольно трудоемкую задачу [1], для решения которой проводятся не только специализированные конкурсы (например, "Асы компьютерного 3D-моделирования"), но и уже создано и продолжает создаваться большое число программных средств, включая такие распространенные средства, как Solid Works, 3D Max и др. Основной трудностью, которая зачастую является непреодолимым препятствием для начинающих пользователей, является перегруженный второстепенными функциями и большими числом узкоспециализированных возможностей программный интерфейс. По этой причине разработчики систем трехмерного моделирования создают специальные обучающие интерактивные программы помощи для демонстрации и объяснения, каким образом использовать ту или иную функцию. Для графического редактора Blender, например, разработано около 25 специальных обучающих демо-роликов (*tutorials*) [2]. Допол-

нительной сложностью является то обстоятельство, что существующие программные средства требуют достаточно много как дисковой, так и оперативной памяти компьютера. Кроме того, для просмотра сформированных трехмерных моделей предполагается использование соответствующих программ (*viewers* — вьюеров). Этим объясняется полное отсутствие 3D-редакторов и вьюеров для операционной системы Android.

В настоящей статье описаны базовые требования и подходы для создания простейшего графического 3D-редактора, позволяющего интуитивно понятными средствами создавать трехмерные модели, которые в дальнейшем можно использовать в любой программе, написанной на языке C++. Для этого достаточно подключить созданную автором и свободно распространяемую библиотеку, которая описана в работе [3]. Дополнительным преимуществом нового редактора будет возможность описывать связи между отдельными частями трехмерной модели, что в целом значительно упрощает разработку больших составных изделий. Преимущества

предлагаемого подхода были продемонстрированы в работе [3] при разработке метаязыка описания связей, а также прикладной информационной системы, для которой использовалась модель спутника "Спектр-РГ". Некоторые принципы, описанные в статье, были реализованы именно в этой прикладной системе, например, использование колеса мыши для вращения и перемещения трехмерной модели.

Основные функциональные возможности графического редактора

На практике, как правило, не возникает необходимости использовать все средства моделирования и модификации 3D-объектов, которые имеются в используемой для этого программной среде. Для начинающих пользователей обычно появляется необходимость быстро визуализировать какие-то простые объекты, не имеющие геометрически сложных деталей, и использовать их для решения своих задач. Для этих целей вполне достаточно иметь в распоряжении программное средство создания основных базовых трехмерных фигур, включающее механизмы по их перемещению и редактированию. Кроме этого, необходимо также, чтобы в используемом программном средстве присутствовали механизмы для сохранения модели в доступном формате, позволяющем использовать ее в качестве или демонстрационной модели, или модели для прикладной информационной системы.

Необходимо отметить, что немаловажную роль в системах трехмерного проектирования играет собственно реализация интерактивных возможностей по перемещению и вращению созданных моделей, чтобы можно было более детально рассмотреть и проанализировать разработанную трехмерную модель. С этой целью созданы даже специальные графические мыши, в которых определенные операции перемещения и вращения автоматически связываются с перемещениями управляющих частей манипулятора. Несмотря на то что перечисленные возможности действительно упрощают процесс редактирования, число управляющих кнопок манипулятора при этом может увеличиваться и, как следствие, эффективность его использования будет в значительной степени зависеть от навыка работы пользователя с такого рода устройствами. По мнению автора, в интерфейсах существующих программ графического моделирования недостаточно полно используются возможности обычной двухкнопочной мыши с колесом прокрутки. Возможности перепрограммирования манипулятора мышь, в зависимости от текущего иерархического уровня редактирования позволяют использовать колесо мыши для многих целей, в том числе и для перемещения объекта в определенном направлении, и для вращения на определенный угол относительно заданной оси.

Интуитивно понятный интерфейс

Практика показывает, что для разработки хорошего в плане предъявляемых к нему требований трехмерного редактора необходимо надежное программное обеспечение, имеющее дружелюбный и эргономичный интерфейс [4–7]. Для этого на этапе проектирования необходимо сфокусировать внимание разработчиков на целях и потребностях потенциальных пользователей создаваемого продукта.

Основной целью разработки нового трехмерного графического редактора является реализация механизмов, облегчающих для пользователя процессы создания простых трехмерных моделей и сцен, что может быть сделано в том числе с помощью упрощения интерфейса. Для этого, по мнению автора, необходимо в максимальной степени исключить из основного диалогового окна иерархические меню, которые трудны для их восприятия пользователем и сложны для описания в инструкциях, поясняющих их работу. В ходе разработки редактора, как уже отмечалось выше, предполагается построить программную оболочку, полностью основанную на контекстном меню, опции которого меняются в зависимости от выполняемых пользователем действий. Несмотря на то что такой подход предложен давно и частично используется в операционной системе Windows, в программах графического моделирования полностью реализовать его пока не удастся. Изложенные далее требования к разработке интерфейса графического редактора, по мнению автора, позволят реализовать основные положения предлагаемого подхода.

Создание элементарных фигур

Первым шагом на пути упрощения интерфейса пользователя графического редактора является разработка основного диалогового окна, которое будет содержать только такие опции, которые обеспечивают внешние функции программной среды. Такие опции позволяют реализовать процессы сохранения файлов, печать, конвертацию и т. д. Как отмечалось ранее, для того чтобы упростить процесс создания и позиционирования геометрических фигур, которые относятся к стандартным для графического редактора (их число можно наращивать), необходимо максимально использовать возможности контекстного меню. Другими словами, функция создания должна быть исключена из основного меню, а ее инициация может происходить при нажатии правой кнопки мыши. Для того чтобы по возможности предоставить пользователю максимальное число способов создания фигур, контекстное меню должно содержать информацию не только о типе фигуры, но и о том, каким образом она будет создаваться. Наиболее очевидный с точки зрения пользователя способ создания объекта и его позиционирования — это использование размеров объекта, заданных по умолчанию, и помещение объекта в той точке экрана, на которую указывает курсор мыши. Контекстное меню

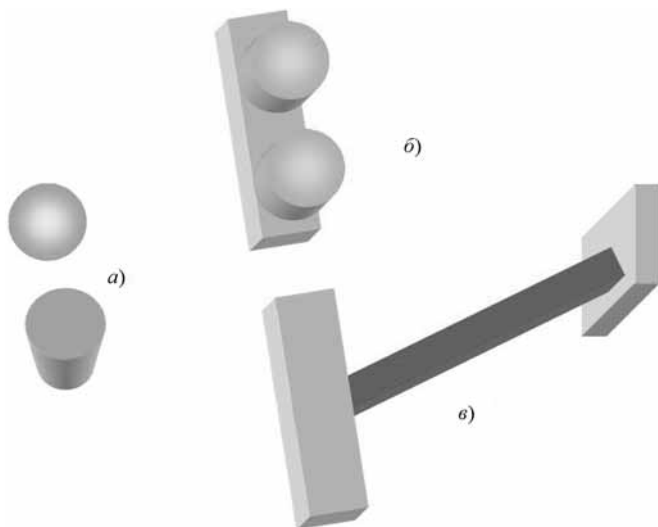


Рис. 1. Примеры создания базовых трехмерных фигур:
 а — отдельно расположенные объекты в виде цилиндра и шара;
 б — два цилиндра, прикрепленных к одной грани заданного ранее параллелепипеда; в — параллелепипед, построенный как связующий элемент для граней двух других параллелепипедов

содержит и другие способы создания и позиционирования, которые реализованы в языке описания связей. Согласно общему подходу, представленному в работе [3], фигура может создаваться или как отдельно расположенный объект, или как объект, прикрепленный к выбранной грани, или как объект, определяемый как связующий элемент для двух различных граней (рис. 1).

Объединение этих операций в единое действие пользователя может выглядеть следующим образом:

- выбор некоторой области (пустого места), куда необходимо поместить фигуру с помощью указателя мыши;
- нажатие правой клавиши манипулятора, вызывающее открытие меню, в котором перечислены все возможные фигуры и способы их создания; после нажатия левой клавиши мыши выбранная фигура с размерами, заданными по умолчанию, или будет помещена в указанную область, или будет создана в виде соединяющего или прикрепленного объекта (при этом потребуются указать либо соединяемые грани, либо просто требуемую грань).

Операции перемещения и вращения объекта

Следующий шаг на пути разработки упрощенного интерфейса — это разработка последовательности действий пользователя для точного позиционирования трехмерной фигуры относительно других объектов. Эта операция, с точки зрения требований интуитивно понятного интерфейса, может состоять из двух этапов, каждый из которых является полностью независимым. На первом этапе происходит "грубое" перемещение в выбранную область, а на втором

этапе — "тонкая" настройка положения модели, осуществляемая путем ее перемещения или вращения относительно базовых осей.

Первая часть операции позиционирования может быть реализована с помощью выделения грани (в том числе и на координатной поверхности) и собственно объекта, над которым она выполняется. Для пользователя данная процедура может выглядеть следующим образом.

- Для того чтобы выделить трехмерный объект, пользователь "кликает" левой клавишей мыши на нужную фигуру, которая при этом изменяет цвет, демонстрируя таким образом, что следующие действия будут связаны именно с ней.

- Удерживая левую клавишу, пользователь перемещает курсор указателя мыши по рабочему полю и выбирает нужную грань, которая, при наведении на нее курсора также изменяет цвет. При отпускании левой клавиши происходит заключительное перемещение выбранного объекта в центр выбранной грани. Данные действия аналогичны тому, что выполняет пользователь при перемещении файлов в операционной системе Windows (*drag and move*). Отметим, что окончательное перемещение выбранной фигуры будет осуществляться таким образом, чтобы нормали грани и фигуры стали сонаправлены.

С учетом изложенного выше заметим, что в результате для "грубого" перемещения объекта в заданную область пространства пользователь должен всего лишь один раз нажать и отпустить левую клавишу мыши.

Интуитивная ясность этой операции связана с тем обстоятельством, что прослеживается аналогия с осознанными действиями человека, который хочет, например, поставить чашку на стол. В результате проведенного человеком действия чашка, как правило, будет перемещена и поставлена на поверхность стола таким образом, чтобы нормаль к поверхности стола совпала с нормалью ко дну чашки. А процесс точного позиционирования фигуры вполне аналогичен передвижению чашки на столе в нужное место.

Операция перемещения или вращения объекта в пространстве относительно базовых осей (локальной системы координат) инициируется нажатием и отпусканием левой клавиши мыши и также выполняется в два этапа.

- На первом этапе пользователю необходимо выбрать одно из трех базовых направлений, вдоль которого будет проведена операция перемещения (базисные оси высвечиваются сразу после отпускания левой клавиши мыши).

- На втором этапе осуществляется перемещение по выбранному направлению на определенное расстояние. Отметим, что точное значение такого перемещения может быть заранее не определено, и пользователю необходимо предоставить возможность плавного изменения объекта с отображением результатов перемещения. Именно для этого и предлагается использовать колесо мыши.

Для выполнения операции вращения, так же как и при перемещении объекта, используется локальная

система координат, состоящая из базовых осей, которые в соответствии с общим подходом, изложенным в работе [3], определены для каждой элементарной фигуры. Пользователь с помощью курсора выбирает одно из трех направлений, а затем, используя колесо манипулятора мыши, вращает объект как по часовой стрелке, так и против нее относительно соответствующей оси.

Таким образом, действия пользователя по перемещению и вращению модели или ее части можно представить в виде последовательности, изображенной на рис. 2 (см. вторую сторону обложки):

- пользователь выбирает объект и щелкает по нему левой клавишей мыши, при этом изменяется цвет объекта и на экран выводятся координатные оси локальной системы координат;
- пользователь щелкает по выбранному объекту правой клавишей мыши, и появляется контекстное меню, из которого он может выбрать нужное действие — перемещение по осям, вращение относительно осей, присоединение к поверхности и др. (по умолчанию подразумевается присоединение объекта к какой-либо грани) или операцию изменения (редактирования) параметров всего объекта;
- если пользователем выбрана опция вращения, то при наведении курсора на какую-либо из осей появляется стрелка для вращения и одновременно активируется колесо мыши для вращения объекта относительно выбранной оси;
- если используется опция перемещения, то при наведении курсора на соответствующую ось на экране появляются дополнительные стрелочки, указывающие возможность перемещения, и пользователь также может использовать колесо мыши для плавного перемещение объекта вдоль этой оси;
- для того чтобы выйти из режима перемещения и вращения, можно использовать двойное нажатие левой клавиши мыши по выбранному объекту (такое действие представляется вполне логичным, так как одинарное нажатие задает фокус на данном объекте) или правой клавиши, если указатель находится на пустом поле.

Предлагаемая интерактивная процедура изменения положения объекта в пространстве имеет ряд очевидных преимуществ:

- интуитивно понятные для пользователя действия;
- точное позиционирование объекта с помощью колеса мыши;
- перегружаемые функции управления объектом в зависимости от контекстного меню, что избавляет пользователя от необходимости запоминания значений дополнительных кнопок и переключателей;
- минимум нажатий кнопок мыши при выполнении всей операции в целом.

Операции редактирования

В операциях редактирования пользователю предоставляется возможность видоизменять трехмерную модель различными способами, используя при этом принятые и описанные ранее подходы к ее

компоновке на основе понятия грани. Отметим, что традиционное понятие редактирования объекта подразумевает изменение каких-либо его характеристик. С этой целью, как отмечалось выше, представляется удобным использовать колесо мыши.

В силу того что в языке описания трехмерная модель представляет собой набор связанных граней (каждая из которых имеет собственный базис), построенных в соответствии с заданным алгоритмом, операции редактирования можно разделить на две категории:

- операции, при которых изменяются общие параметры, описывающие весь объект целиком, такие как высота, ширина, радиусы оснований и др.;
- операции, при которых изменяются параметры отдельных граней, такие как линейные размеры и пространственное положение.

Для редактирования общих параметров предлагается использовать контекстное диалоговое окно,

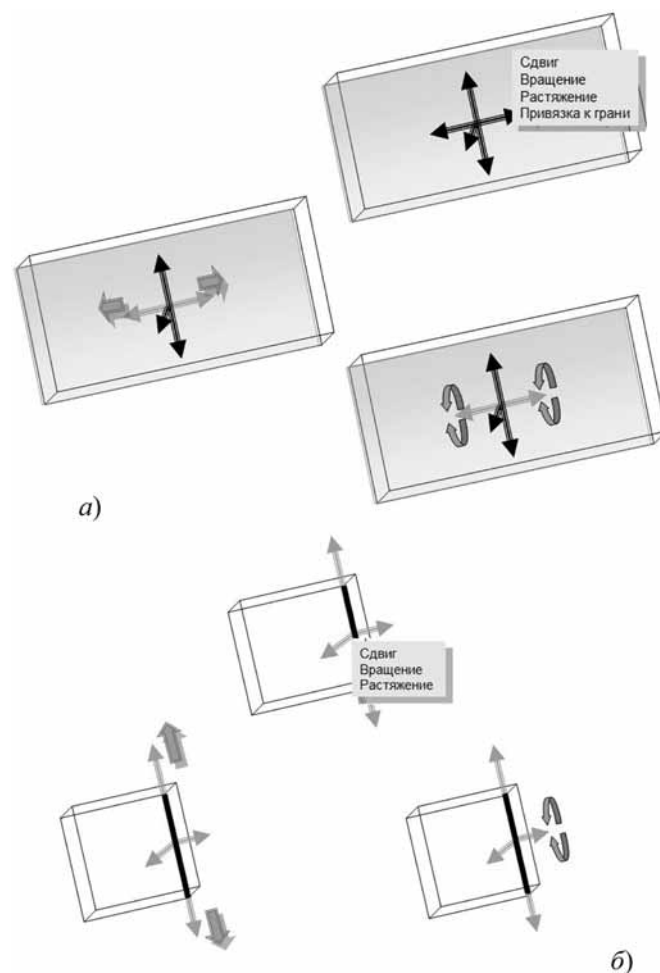


Рис. 3. Редактирование объекта на уровне граней и ребер, осуществляемое в соответствии с порядком, определенным для перемещения всей фигуры:

a — действия пользователя при модификации выбранной грани; *b* — действия пользователя при модификация выбранного ребра

которое должно открываться при нажатии правой клавиши мыши на уже выделенном объекте.

Для того чтобы перейти в режим редактирования граней и ребер (рис. 3), пользователь должен еще раз навести курсор на объект и нажать левую клавишу манипулятора мыши. При этом весь объект станет прозрачным и будет показан лишь его каркас.

Редактирование граней и ребер выполняется однотипным образом, аналогичным ориентации всего объекта в пространстве — т. е. выбор направления и перемещение или вращение грани или ребра относительно оси.

После перехода в режим редактирования ребер или граней пользователь работает только с выделенным объектом, а идентификация нужной грани или ребра происходит путем изменения ее цвета во время перемещения курсора. Повторное нажатие левой клавиши приведет к переходу в режим собственно редактирования выбранного ребра или грани. При этом, точно так же как и при перемещении всей фигуры, на экране отображаются связанные с объектом базисные оси. Пользователь также может с помощью правой клавиши мыши активировать контекстные меню и диалоги, открывающие детализированные опции по модификации выбранной грани или ребра.

Контекстное меню и диалоги

Подход к редактированию с использованием контекстного меню, как уже отмечалось ранее, позволяет организовать всю структуру управления на интуитивно понятном уровне и избавиться от многочисленных структурированных иерархических меню, которые так распространены в среде Microsoft Windows. Хотя, и это следует отметить, и в этой среде встречаются управляющие модули, организованные в виде контекстного меню. Подход к редактированию на основе контекстного меню в разрабатываемом интерфейсе реализуется в следующем виде (см. рис. 2, в, вторая сторона обложки): при нажатии правой клавиши мыши на каком-либо объекте, в зависимости от выбранного ранее действия, отображается список опций, доступных пользователю именно в этом режиме.

Такой подход предлагается использовать и для инициирования диалоговых окон. Например, в случае, когда пользователь навел курсор на ось для перемещения или вращения объекта, нажатие правой клавиши мыши может приводить к открытию диалогового окна для указания точного значения смещения (рис. 4), на которое нужно сдвинуть или повернуть модель. Эта опция удобна в случае, если перемещение необходимо выполнить вдоль определенного вектора, не являющегося базовым вектором, координаты которого можно задать в локальной системе координат (рис. 4), или в абсолютных единицах измерения, используемых для всей сцены.

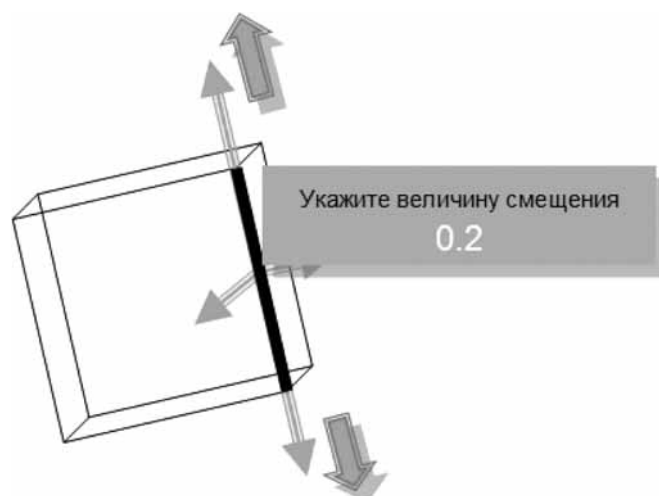


Рис. 4. Пример контекстного диалогового меню (смещение указано в относительных единицах измерения)

Для того чтобы пользователь имел информацию о том, есть ли связанное с конкретным объектом контекстное меню или нет, предлагается использовать следующий прием: при наведении указателя мыши на объекты на экране форма курсора меняется в зависимости от того, предоставляется или нет данная возможность.

Контейнеры и невидимые объекты

В программе, реализующей интерфейс пользователя для редактирования объекта, предусмотрена возможность сделать любой такой трехмерный объект невидимым. Такая возможность представляется удобной с точки зрения создания сложных составных моделей на основе требований к языку описания связей. Невидимый объект обладает всеми свойствами обычного объекта, за исключением того, что он не заносится в видеобуфер. Как следствие, его можно использовать для установления связей между ним и видимыми объектами. Другими словами, если в сцене присутствует невидимый объект, связанный с определенным набором видимых объектов, то при его модификации соответствующим образом меняется положение и видимых объектов. Например, если пользователь изменяет высоту невидимого объекта цилиндр (рис. 5), к верхней и нижней частям которого прикреплены, например, несколько шаров, то такое действие приводит к изменению положения всех прикрепленных шаров. Использование такого подхода приводит к значительному сокращению проводимых пользователем действий, по сравнению со случаем, когда пришлось бы каж-

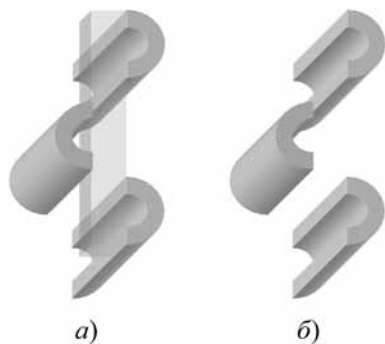


Рис. 5. Пример использования невидимого объекта для точного позиционирования трех отдельных объектов:

а — присоединение фигур к поверхности параллелепипеда;
б — определение параллелепипеда как невидимого объекта

дый из шаров перемещать на одно и то же расстояние в одном и том же направлении. Таким образом, невидимые объекты могут эффективно использоваться для структурирования и параметризации сложной составной конструкции трехмерной модели.

Другим немаловажным и полезным на практике свойством невидимых объектов является использование их в качестве контейнеров. Понятие "контейнер" вводится для того чтобы уменьшить машинное время обработки перемещения курсора мыши. Такой подход значительно обогащает (увеличивает) интерактивные возможности программной оболочки. Этот вывод обусловлен тем обстоятельством, что для проверки факта попадания курсора мыши в нужную грань необходимо перебрать все имеющиеся в сцене грани. При большом числе мелких объектов (≈ 50) такой перебор приводит к значительному замедлению работы программы. По этой причине требование к языку описания связей предусматривает для сложных объектов или группы объектов создание контейнера (нового графического объекта), который содержит минимальное число граней и предназначен для того, чтобы включить в себя эту группу. Требование к использованию контейнера для анализа события попадания курсора в любой объект из этой группы позволяет проверять не все грани объектов, а только грани самого контейнера. Такой прием несколько ухудшает точность указания курсора, однако значительно повышается производительность программы при интерактивных действиях пользователя. Учитывая, что контейнер не отображается на экране монитора, этот прием никак не сказывается на представлении всей разрабатываемой трехмерной модели.

Расширение функциональных возможностей редактора

Существующие системы трехмерного моделирования, меню в которых организовано в основном в виде иерархической структуры, фактически не

представляют возможности внесения изменений в интерфейс программы без его глубокой переработки. Поэтому неслучайно новые версии одной и той же программы такого назначения внешне и в плане их функциональных возможностей выглядят различным образом.

Предлагаемая архитектура, полностью основанная на контекстном меню, позволяет простыми средствами наращивать возможности по созданию и редактированию трехмерных моделей, не изменяя общих подходов и требований к созданию интерфейса с точки зрения его функциональных возможностей. Другими словами, в программу можно постепенно добавлять новые дополнительные возможности, и пользователь будет их осваивать, уже имея представление о полном наборе исходных базовых операций (в соответствии с концептуальными положениями, изложенными в работе [7]). Описанные способы организации программного интерфейса в принципе позволяют уйти от необходимости создания демо-роликов, предназначенных для обучения пользователя работе с текущей версией программы редактирования и с ее последующими версиями.

Заключение

Изложенные подходы и требования к разработке программного интерфейса для графического редактора на основе языка описания связей целесообразно использовать при разработке простого интуитивно понятного программного средства для проектирования 3D-объектов. Унифицированные возможности редактирования (как объекта в целом, так и его частей), построенные на контекстном меню, позволяют неподготовленному пользователю легко освоить общие приемы создания трехмерных моделей.

В силу того что в основе создания объектов лежит язык описания связей, формируемый трехмерный объект не представляется в виде простого набора точек, а создается после выполнения определенного набора команд. Именно эти команды и являются тем форматом, который может быть сохранен на жестком диске в виде отдельного файла и впоследствии считан для загрузки модели. Для того чтобы созданную трехмерную модель можно было просматривать в других редакторах, необходимо перевести ее в соответствующий формат. Учитывая, что после выполнения команд метаязыка каждый трехмерный объект получает соответствующие трехмерные координаты, по ним можно осуществить конвертацию модели в любой другой графический формат, основанный на точечном представлении. Очевидно, что при такой конвертации теряются связи между объектами, однако появляется возможность использовать другие средства редактирования и отображения.

Список литературы

1. Хайфец А. В., Логиновский А. Н., Буторина И. В., Васильева В. Н. Инженерная 3D-компьютерная графика. Уч. пособие, М.: Юрайт, 2011. 464 с.
2. Прахов А. Самоучитель Blender 2.6, СПб.: БХВ-Петербург, 2013, 384 с.
3. Светушков Н. Н. Метаязык описания связей для прикладных информационных систем// Программная инженерия. 2017. Т. 8, № 3. С. 136—139.

4. Купер А., Рейманн Р. М., Кронин Д., Носсел К. Интерфейс. Основы проектирования взаимодействия. 4-е изд., СПб.: Питер, 2016. 720 с.
5. Тидвелл Д. Разработка пользовательских интерфейсов, СПб.: Питер, 2011, 480 с.
6. Raskin J. The Human Interface. New Directions for Designing Interactive Systems, Addison-Wesley Professional, 2000, 256 p.
7. Будиу Р., Нильсен Я. Mobile Usability. Как создавать идеально удобные приложения для мобильных устройств. М.: Эксмо, 2013. 254 с.

Program Interface for a Simple 3D-Graphical Editor Based on a Relationships Description Meta-Language

N. N. Svetushkov, svetushkov@mai.ru, Moscow Aviation Institute (National Research University), Moscow, 125993, Russian Federation

Corresponding author:

Svetushkov Nikolaj N., Associate Professor, Moscow Aviation Institute (National Research University), Moscow, 125993, Russian Federation
E-mail: svetushkov@mai.ru

Received on February 09, 2017

Accepted on March 03, 2017

The basic requirements and approaches are presented as basis for the development of a simple graphics editor of three-dimensional objects. For the convenience it was decided to build the user interface fully on the context menu, which can easily be further complicated and modified by adding all sorts of options and special functions. As the data storage format a previously developed meta-language for describing relationships is used, one advantage of which it is a simple description of the complex three-dimensional scenes with the ability to support the established relationships between objects. The suggested the principles and approaches make it possible to create a software tool for editing, not overloaded with complex hierarchically organized menu and, at the same time, simplify the user process of learning to use it, as they provide an intuitive interface. Graphic editor in this case can be written entirely in C ++, and then be used as a simple tool for creating three-dimensional models for third-party developers. The purpose of the project, studies for which are discussed in the article, is providing the user simple interactive mechanisms for creating three-dimensional objects on the basis of tokens description language lexeme. Using the public API, one can develop custom applications, including both fully informational and applicable, for example, for the numerical simulation of complex physical processes.

Keywords: 3D graphics, a language for describing three-dimensional objects, unified principles of editing, intuitive interface, context menus, dialog boxes, software.

For citation:

Svetushkov N. N. Program Interface for a Simple 3D-Graphical Editor Based on a Relationships Description Meta-Language, *Programmnyaya Inzheneriya*, 2017, vol. 8, no. 5, pp. 215—221.

DOI:10.17587/prin.8.215-221

References

1. Hajfec A. V., Loginovskij A. N., Butorina I. V., Vasil'eva V. N. *Inzhenernaya 3D-komp'yuternaya grafika. Uchebnoe posobie* (Engineering 3D-computer graphics. Tutorial), Moscow, Jurajt, 2011, 464 p. (in Russian).
2. Prahov A. *Samouchitel' Blender 2.6* (Blender 2.6 Tutorial), Saint-Petersburg, BHV-Peterburg, 2013, 384 p. (in Russian).
3. Svetushkov N. N. Metajazyk opisanija slozhnyh 3D-ob#ektov dlja prikladnyh informacionnyh sistem (Meta-Language for 3D-objects Description in Applied Information Systems), *Programmnyaya Inzheneriya*, 2017, vol. 8, no. 3, pp. 136—139 (in Russian).

4. Kuper A., Rejmann R. M., Kronin D., Nossel K. *Interfejs. Osnovy proektirovanija vzaimodejstvija* (Interface. Fundamentals of interaction.), 4-e izd., Saint-Petersburg, Piter, 2016, 720 p. (in Russian).
5. Tidvell D. *Razrabotka pol'zovatel'skih interfejsov* (The development of user interfaces), Saint-Petersburg, Piter, 2011, 480 p. (in Russian).
6. Raskin J. *The Human Interface. New Directions for Designing Interactive Systems*, Addison-Wesley Professional, 2000, 256 p.
7. Budiu R., Nil'sen Ja. *Mobile Usability. Kak sozdat' ideal'no udobnye prilozhenija dlja mobil'nyh ustrojstv* (Mobile Usability. How to create a perfectly comfortable applications for mobile devices), Moscow, Exmo, 2013, 254 p. (in Russian).

А. Э. Аббасов, аспирант, e-mail: alexn1smo@mail.ru, Государственное бюджетное образовательное учреждение высшего образования Московской области "Технологический университет", г. Королёв

Анализ экспертной информации об экологических показателях транспортных средств

Представлены результаты анализа проблемных вопросов, которые возникают при создании и сопровождении информационных систем измерения и регулирования содержания вредных веществ от загрязняющих атмосферу транспортных средств. На основании теоретического анализа предметной области для автоматизированной информационной системы, в основе которой лежит база данных об экологических показателях, предложена структура интернет-ресурса для хранения и обработки сведений по экологическому контролю автотранспортных средств. Для разработки интернет-ресурса предложены инструментальные средства базы данных MySQL и системы управления контентом MODx Revolution, технологии Apache и phpMyAdmin.

Ключевые слова: автоматизированный контроль, унификация данных, математическая модель, интегральный экологический показатель, производительность интернет-ресурса

Введение

Большая часть вредных веществ, выбрасываемых в атмосферу, приходится на автомобильный транспорт (60 %) [1]. Решение задачи создания и сопровождения систем автоматизированного контроля и обработки экспертной информации об экологических показателях транспортных средств во многом осложняется отсутствием в структуре центра организации дорожного движения (ЦОДД) автоматизированной системы экспертной оценки (АСЭО), осуществляющей обработку сведений о результатах экологического осмотра автотранспортных средств. Для обеспечения работы АСЭО необходимо создание соответствующего алгоритмического, программного и информационно-технического обеспечения [2]. Трудности в решении этих задач связаны также и с отличиями стандартов на экологические показатели, которые представлены в различных нормативно-законодательных актах Российской Федерации, а также с отсутствием широкого доступа к результатам оценки этих показателей.

На территории РФ экологический стандарт Евро-5 введен 1 января 2016 г. В настоящее время в Европе, США и Японии вводится стандарт Евро-6. Сроки перехода России на стандарт Евро-6 неизвестны. В табл. 1 [3, 4] представлены экологические стандарты для легковых автомобилей в сравнении с Евро-6 (в единицах г/км).

В нормативных документах РФ проверку выхлопов делят на два типа — для бензиновых и дизельных двигателей. Для автомобилей с бензиновыми двига-

телями положения этих документов определяют содержание оксида углерода и углеводородов в составе веществ, загрязняющих атмосферу, в соответствии с требованиями ГОСТ Р 52033—2003 [5]. Для автомобилей с дизельными двигателями проводят измерения уровня задымления (дымности) в соответствии с требованиями ГОСТ 21393—75 [6]. Однако практика показывает, что этих проверок недостаточно для того, чтобы обеспечить соответствие требованиям стандарта Евро-5.

Экологический класс авто (Евро-2, Евро-3, Евро-4, Евро-5) определяется органами по сертификации. В настоящее время начата работа по созданию стационарных и передвижных пунктов по контролю и регулировке содержания загрязняющих веществ в отработавших газах транспортных средств. Однако только подобных рейдовых проверок недостаточно. Необходим перманентный контроль работы таких передвижных пунктов. При проведении государственного технического осмотра транспортного средства, например, проверка прохождения процедуры измерения и регулировки содержания загрязняющих веществ в специализированных пунктах может быть подтверждена с помощью различных удостоверяющих ее документов. Таким документом может быть, например, талон, подтверждающий факт проведения проверки, как это было сделано, к примеру, в республике Башкортостан. Сотрудники экологической ГИБДД могут выборочно остановить любое автотранспортное средство и проверить его на содержание выхлопа. Однако в эпоху активного развития информационных технологий [7] представляется

Требования к нормам по выбросу различных экологических стандартов, г/км

Экологический стандарт	Оксид углерода (II) (CO)	Углеводород	Летучие органические вещества	Оксид азота (NO _x)	HC + NO ₂	Взвешенные частицы (PM)
Для дизельного двигателя						
Евро-1	2,72 (3,16)	—	—	—	0,97 (1,13)	0,14 (0,18)
Евро-2	1,0	—	—	—	0,7	0,08
Евро-3	0,64	—	—	0,50	0,56	0,05
Евро-4	0,50	—	—	0,25	0,30	0,025
Евро-5	0,50	—	—	0,180	0,23	0,005
Евро-6	0,50	—	—	0,080	0,17	0,005
Для бензинового двигателя						
Евро-1	2,72 (3,16)	—	—	—	0,97 (1,13)	—
Евро-2	2,2	—	—	—	0,5	—
Евро-3	2,3	0,20	—	0,15	—	—
Евро-4	1,0	0,10	—	0,08	—	—
Евро-5	1,0	0,10	0,068	0,06	—	0,005
Евро-6	1,0	0,10	0,068	0,06	—	0,005
Прочерк означает отсутствие проверок выбросов данного вредного вещества						

целесообразной разработкой автоматизированной системы экспертной оценки (АСЭО) количества вредных веществ автотранспортных средств. Такая система должна быть у центра ЦОДД и включать интернет-ресурс, на котором следует отражать результаты экологического контроля автотранспортных средств и соответствующих им систем снижения токсичности. Эти результаты определяют соответствие автотранспортных средств классу экологичности с учетом нормативов (положений), представленных в законодательстве РФ [5, 6].

Методы исследования

Для автоматизации экспертной оценки экологических показателей автотранспортных средств был проведен анализ предметной области, характеризующей влияние на экологию автотранспортных средств, для последующего создания программного обеспечения многопользовательской информационной системы. Проанализированы и сгруппированы отдельные факты, характеризующие контроль экологических показателей автотранспортных средств, и соответствующие им системы снижения токсичности. Выявлен обобщенный интегральный показатель, характеризующий наиболее важные составляющие вредных веществ, выбрасываемых автотранспортными средствами в атмосферу.

Использованы математические методы для обработки экспериментальных данных, полученных по результатам обзора литературы для установления количественных зависимостей между техническими характеристиками автотранспортных средств и

экологическими показателями. Для создания экспертно-статистической модели построения интегральных экологических показателей применен метод главных компонентов из классической теории факторного анализа.

Результаты исследования

Цель АСЭО — обработка сведений о результатах экологического осмотра автотранспортных средств и предоставление широкого доступа к оценке экологических показателей. В задачи разрабатываемой АСЭО входит информационная поддержка процедур проверки в соответствии с требованиями государственного технического осмотра транспортного средства. Процедуры проверки включают процедуры измерения загрязняющих веществ.

На рис. 1 приведена структурная схема ЦОДД, состоящая из автоматизированных систем управления дорожным движением (АСУДД) и систем управления различного назначения. Схема ЦОДД дополнена АСЭО для обработки сведений о результатах экологического осмотра автотранспортных средств [1].

Для информирования владельцев транспортных средств и заинтересованных организаций на интернет-ресурсе по результатам проведенных процедур проверки предоставляются рекомендации для проведения организационно-технических мероприятий по улучшению экологических показателей и снижению расхода топлива (по улучшению рециркуляции отработавших газов, их очистке; диагностике агрегатов, от которых зависит состав веществ, загрязняющих атмосферу). Также приводятся сведения об освобождении

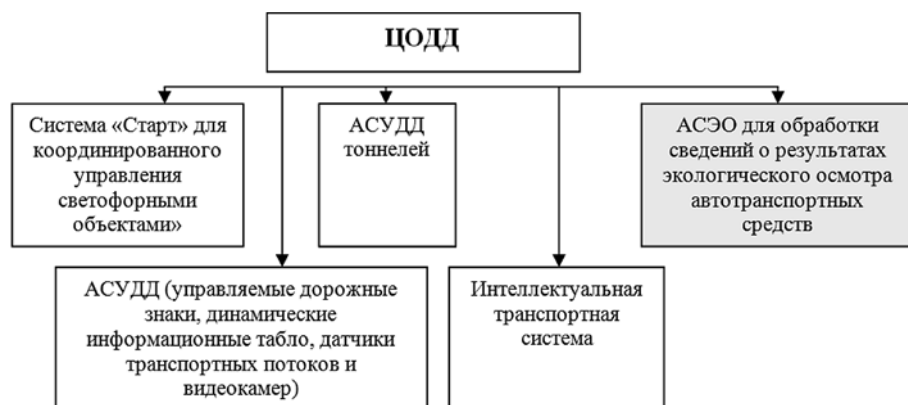


Рис. 1. Структура ЦОДД с встраиваемой АСЭО

системы снижения токсичности выбросов будет характеризоваться k -мерным вектором признаков $\tilde{x} = (\tilde{x}^{(1)}, \dots, \tilde{x}^{(k)})$. Измерив значение вектора \tilde{x} в N точках, получим матрицу исходных данных:

$$\tilde{B} = \begin{pmatrix} \tilde{x}_1^{(1)} & \dots & \tilde{x}_1^{(k)} \\ \dots & \dots & \dots \\ \tilde{x}_N^{(1)} & \dots & \tilde{x}_N^{(k)} \end{pmatrix}. \quad (1)$$

Представленную в матрице (1) многомерную информацию трудно использовать для анализа уровня экологичности транспортных средств. Например, нельзя провести сравнение систем

снижения токсичности автотранспортных средств, рассчитать время для освобождения от налога за автотранспортное средство в случае значений составных компонентов интегрального показателя качества, не превышающих значений норм по выбросу, указанных в стандартах.

Для вычисления интегрального показателя z , который зависит от k признаков, характеризующих состав вредных веществ, $z = (\tilde{x}^{(1)}, \dots, \tilde{x}^{(k)})$, необходимо проанализировать максимальный вклад каждого вредного вещества в интегральный показатель. Для разработки математической модели интегрального показателя применен метод главных компонентов.

Для унификации исходных данных используется преобразование: если $\tilde{x}^{(i)}$ — количественный монотонный показатель, характеризующий количество выбросов определенного вида, тогда

$$x^{(i)} = \frac{\tilde{x}^{(i)} - \tilde{x}_{\min}^{(i)}}{\tilde{x}_{\max}^{(i)} - \tilde{x}_{\min}^{(i)}}, i = 1, \dots, k,$$

где $\tilde{x}_{\min}^{(i)}$ — минимальное значение количественного показателя; $\tilde{x}_{\max}^{(i)}$ — максимальное значение количественного показателя.

Если $\tilde{x}^{(i)}$ — качественный показатель, для которого существует некоторое оптимальное значение $\tilde{x}_{\text{opt}}^{(i)}$, такое, что $\tilde{x}_{\min}^{(i)} < \tilde{x}_{\text{opt}}^{(i)} < \tilde{x}_{\max}^{(i)}$, то

$$x^{(i)} = 1 - \frac{|\tilde{x}^{(i)} - \tilde{x}_{\text{opt}}^{(i)}|}{\max\{|\tilde{x}_{\text{opt}}^{(i)} - \tilde{x}_{\min}^{(i)}|, |\tilde{x}_{\text{opt}}^{(i)} - \tilde{x}_{\max}^{(i)}|\}}, i = 1, \dots, k.$$

Целесообразно введение интегрального экологического показателя. На интернет-ресурсе будут представлены места, способы и результаты проверки соответствия показателей требованиям стандартов с помощью различных процедур проверки.

Состав веществ, загрязняющих атмосферу автотранспортными средствами, представлен на рис. 2 (см. вторую сторону обложки) [8]. Существуют достаточно сложные процедуры, позволяющие выбрать метод оценки выбросов дорожного транспорта, описание которых приведено в работе [9]. Обычным пользователям интернет-ресурса, на котором отображаются результаты экологического контроля автотранспортных средств, сложно будет выбрать метод оценки. Поэтому предлагается пользоваться упрощенными расчетными зависимостями для определения интегрального показателя.

Чем больше количество вредных веществ, тем больше значение интегрального показателя и хуже уровень экологичности. Проанализировано в общем случае k показателей, тогда каждая экспертная оценка по результатам диагностики состояния

Все полученные признаки будут обладать монотонностью относительно уровня токсичности и дымности вредных веществ: $x^{(i)} = 0$ при минимальном объеме вредных веществ и $x^{(i)} = 1$ при максимальном объеме вредных веществ для всех их видов. Матрицу преобразованных данных обозначим через B . Найдем вектор главных компонент Y с помощью преобразования

$$y = Ax,$$

где A является матрицей размера $k \times k$, строками которой являются нормированные собственные век-

торы $c_j = (c_j^{(1)}, \dots, c_j^{(k)})$, соответствующие неотрицательным собственным значениям $\lambda^{(i)} = D[y^{(i)}]$, $i = 1, \dots, k$, выборочной матрицы ковариаций $S = (s_{im})_{i,m=1, \dots, k}$, где D — дисперсия значений $y^{(i)}$; $s_{im} = \frac{1}{N} \sum_{j=1}^k (x_j^{(i)} - \bar{x}^{(i)})(x_j^{(m)} - \bar{x}^{(m)})$, при этом $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$. При этом $y^{(i)}$ являются независимыми векторами.

Известно [10], что сумма первых нескольких компонент составляет основную долю суммарной дисперсии. Следовательно, если дисперсия первой компоненты $D[y^{(1)}]$ составляет более 50 % от суммарной дисперсии, то она может рассматриваться как интегральный показатель экологического состояния

$$y^{(1)} = \sum_{i=1}^k c_1^{(i)} (x^{(i)} - \bar{x}^{(i)}).$$

В том случае, когда $D[y^{(1)}] < 50 \%$, а $D[y^{(1)}] + D[y^{(2)}] \geq 60 \%$, для оценки интегрального экологического показателя автотранспортного средства можно использовать сумму двух первых компонент $y^{(1)}$ и $y^{(2)}$, причем

$$y^{(2)} = \sum_{i=1}^k c_2^{(i)} (x^{(i)} - \bar{x}^{(i)}).$$

При введении весовых коэффициентов α_i , которые присваиваются каждой компоненте при интегральном оценивании, интегральные экологические показатели можно вычислить с помощью следующих соотношений:

$$y^{(1)} = \sum_{i=1}^k \alpha_i c_1^{(i)} (x^{(i)} - \bar{x}^{(i)}) \text{ в случае одного показателя,}$$

$$y^{(2)} = \sum_{i=1}^k \alpha_i c_2^{(i)} (x^{(i)} - \bar{x}^{(i)}) \text{ в случае двух показателей.}$$

Для уменьшения числа показателей, участвующих в вычислении обобщенного интегрального показателя, с потерей наименьшего количества информации, предлагается учитывать наиболее токсичные компоненты вредных веществ, которыми являются: оксид углерода (CO), оксиды азота (NO_x), углевод-

ороды (C_nH_m), а в случае применения этилированного бензина — свинец.

Для разработки интернет-ресурса, который входит в состав АСЭО, предлагается использовать БД MySQL, веб-сервер Apache с установленной CMS (CMS — Content Management System, система управления контентом) MODx Revolution.

Для выбора инструментальных средств разработки интернет-ресурса был сделан пилотный проект данной разработки в широко распространенных средах Wix, MODx, WordPress и проведена оценка эффективности применения этих сред с помощью диаграммы Ганта. На рис. 3 показаны временные, стоимостные и ресурсные затраты на разработку интернет-ресурса с помощью предложенных инструментальных средств. В табл. 2 представлены результаты сравнительного анализа затрат, масштабируемости, веса страниц, скорости загрузки. Также проводилась SEO-оценка интернет-ресурса и возможность адаптивного дизайна. На создание интернет-ресурса в средах Wix и WordPress уходит меньше времени за счет использования готовых эффектов, но по масштабности, весу страницы, скорости загрузки и SEO-оценке лидирующие позиции занимает разработка в среде MODx.

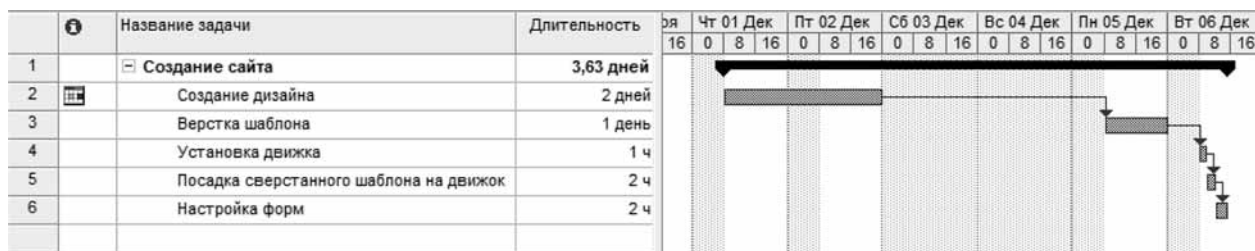
При работе с базами данных необходимо обеспечить максимальную производительность и удобство интерфейса [11]. Внесение новых данных в БД с использованием MODx Revolution более удобно, так как редактирование БД с помощью стандартных программ, таких как PhpMyAdmin, требует больших временных затрат на загрузку страниц интернет-ресурса и имеет плохой человеко-компьютерный интерфейс. При использовании MODx Revolution можно быстро создать удобную форму редактирования БД, которая будет работать по технологии AJAX и значительно сократит временные затраты, так как при этой технологии осуществляется обмен данными браузера с веб-сервером в фоновом режиме, позволяющем при обновлении данных интернет-страницы не перезагружать ее полностью.

В программе Erwin разработана структура БД MySQL, в которой хранится информация о разных автомобилях со справочной информацией и нормами выхлопов для них. В структуре БД предусмотрены пользователи — владельцы автотранспортных средств и представители заинтересованных

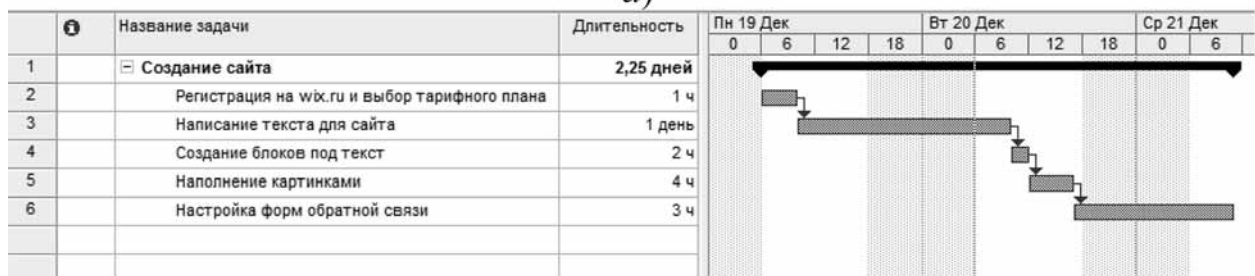
Таблица 2

Результаты сравнительного анализа инструментальных средств для разработки интернет-ресурса

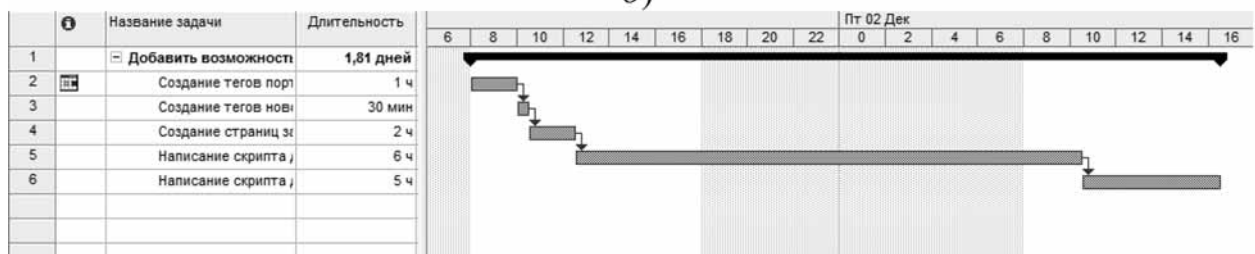
Средство	Время разработки, дни	Масштабируемость	Стоимость владения, руб.	Скорость загрузки, мс	Вес страницы, Мбайт	SEO-оценка сайта, %	Адаптивный дизайн
MODx	3,63	5	346,50	511	2,4	68	Да
Wix	2,25	2	2988	1840	3,7	56	Нет
WordPress	1,81	4	1530	920	3,1	59	Да



а)



б)



в)

Рис. 3. Затраты на разработку интернет-ресурса с помощью:
а — MODx; б — Wix; в — WordPress

организаций. Использование системы управления контентом MODx Revolution требует установки определенного количества системных таблиц, например, modx_session, в которой хранятся сведения о сессии каждого пользователя, в том числе не авторизованного на сайте.

Список таблиц (всего 75 таблиц) получен в программе phpMyAdmin, веб-интерфейс которой представлен на рис. 4.

Интернет-ресурс (сайт) при открытии пользователем будет направлять его на главную страницу. На главной странице размещаются: описание этого

modx_access_actionidom	modx_actions	modx_element_property_sets	modx_migx_elements	modx_site_tmplvars
modx_access_actions	modx_actions_fields	modx_extension_packages	modx_migx_formtabs	modx_site_tmplvar_access
modx_access_category	modx_active_users	modx_fc_profiles	modx_migx_formtab_fields	modx_site_tmplvar_contentvalues
modx_access_context	modx_categories	modx_fc_profiles_usergroups	modx_namespaces	modx_site_tmplvar_templates
modx_access_elements	modx_categories_closure	modx_fc_sets	modx_property_set	modx_system_eventnames
modx_access_media_source	modx_class_map	modx_lexicon_entries	modx_register_messages	modx_system_settings
modx_access_menus	modx_content_type	modx_manager_log	modx_register_queues	modx_transport_packages
modx_access_permissions	modx_context	modx_media_sources	modx_register_topics	modx_transport_providers
modx_access_policies	modx_context_resource	modx_media_sources_contexts	modx_session	modx_users
modx_access_policy_templates	modx_context_setting	modx_media_sources_elements	modx_site_content	modx_user_attributes
modx_access_policy_template_groups	modx_dashboard	modx_membergroup_names	modx_site_htmlsnippets	modx_user_group_roles
modx_access_resources	modx_dashboard_widget	modx_member_groups	modx_site_plugins	modx_user_group_settings
modx_access_resource_groups	modx_dashboard_widget_placement	modx_menus	modx_site_plugin_events	modx_user_messages
modx_access_templatevars	modx_documentgroup_names	modx_migx_configs	modx_site_snippets	modx_user_settings
modx_actionidom	modx_document_groups	modx_migx_config_elements	modx_site_templates	modx_workspaces

Рис. 4. Список системных таблиц движка MODx

ресурса; ссылки на каталог автомобилей с нормами выхлопов, которые регламентируются заводом-изготовителем по каждой модели, а также на личный кабинет, в котором записана модель (модели) автомобилей, находящихся в собственности пользователя, и информация по ним. При открытии страницы личного кабинета будет предоставляться возможность войти на сайт, используя логин и пароль. В случае если пользователь не зарегистрирован на сайте, он сможет пройти по ссылке на страницу регистрации. Если пользователь не помнит данные для входа, для него предусмотрена ссылка на восстановление регистрационных данных.

При входе в личный кабинет со своим логином и паролем открывается список автомобилей, находящихся в собственности. При выборе автомобиля открывается страница, на которой указаны нормы выхлопа, регламентированные заводом-изготовителем для этой модели, справочная информация для данного автомобиля, список пользовательских свойств для данного автомобиля и список тестов, пройденных автомобилем.

При выборе процедуры проверки открывается новая страница, на которой также указаны нормы выхлопов, регламентированные заводом-изготовителем, и результаты процедуры проверки транспортного средства для возможности их сравнения с нормами, регламентированными изготовителем. После прохождения теста информация отображается в личном кабинете владельца.

Интерфейс пользователя системы управления контентом MODx показан на рис. 5, а, общий вид домашней страницы интернет-ресурса — на рис. 5, б.

Также в личном кабинете должна быть реализована возможность добавить, удалить или изменить существующий автомобиль.

Нормирование выбросов вредных веществ и налоговая политика

В зависимости от значения рассчитанного интегрального показателя для владельца автотранспортного средства и определения соответствия нормам выбросов вредных веществ, можно рассчитать время освобождения владельца от уплаты налога на автотранспортное средство в том случае, если этот показатель соответствует требованиям стандартов по экологической безопасности [8]. В случае несоответствия нормам выбросов освобождения от части налога не будет. Однако в этом случае необходимо четко выполнять мероприятия по несению автовладельцем административной ответственности по ст. 8.22 и ст. 8.23 КоАП РФ: "Выпуск в эксплуатацию и эксплуатация механических транспортных средств с превышением нормативов содержания загрязняющих веществ в выбросах либо нормативов уровня шума".

Для расчета времени освобождения от налога, в зависимости от интегрального показателя качества, необходимо проанализировать особенности транспортного налога в России, который зависит от объема двигателя автомобиля и количества вредных веществ в соответствии с официальными данными автопроизводителей для каждой модели автотранспортного средства. Налоговые льготы на малотоннажные автомобили в России пока в разработке, повышающий коэффициент для расчета налога зависит от стоимости транспортного средства и срока его эксплуатации. Налог на выхлоп и объем двигателя не взимается, но существует налог на роскошь. Дорогие автомобили, владельцы которых платят налог за роскошь (этот налог существует только в России), как правило, имеют лучшие экологические показатели, и это надо учитывать при налогообложении.



а)



б)

Рис. 5. Создание домашней страницы сайта:

а — интерфейс пользователя MODx; б — структура домашней страницы пользователя сайта

Оптимизация работы системы снижения токсичности в недорогих автомобилях с более низкими экологическими показателями (например, регулирование угла поворота клапана рециркуляции [12]) также позволит освободить владельца автотранспорта от части налога.

Заключение

Обоснована целесообразность разработки автоматизированной системы экспертной оценки, входящей в ЦОДД и включающей интернет-ресурс для отображения результатов экологического контроля автотранспортных средств и соответствующих им систем снижения токсичности. Определены аналитические зависимости для вычисления интегрального обобщенного показателя уровня экологичности автотранспортных средств. Обоснован выбор инструментальных средств для интернет-ресурса АСЭО.

Список литературы

1. Керножитская А. Ф., Германова Т. В. К вопросу загрязнения атмосферного воздуха выхлопными газами автотранспорта // Сб. тр. конференции "Проблемы функционирования систем транспорта", Тюмень, 14–16 ноября 2012 г. Тюмень: Изд-во Тюменского индустриального университета, 2012. С. 184–191.
2. Аббасов А. Э., Зиновьев В. Н., Васильев Н. А., Посеренин С. П. Организация хранения, обработки и передачи информации в автоматизированной информационной системе экологического контроля // Информационные технологии. Радиоэлектроника. Телекоммуникации (ITRT-2016): сб. статей VI международной заочной научно-технической конферен-

ции. Ч. 1 / Поволжский гос. ун-т сервиса. Тольятти: Изд-во ПВГУС, 2016. С. 13–18.

3. Электронный ресурс "За рулем". URL: <http://www.zr.ru>.
4. Shen G. T., Dunbar R. B. Environmental controls on uranium in reef corals // *Geochimica et Cosmochimica Acta*. 1995. Vol. 59. No. 10. P. 2009–2024.
5. ГОСТ Р 52033–2003. Автомобили с бензиновыми двигателями. Выбросы загрязняющих веществ с отработавшими газами. Нормы и методы контроля при оценке технического состояния (с Изменением № 1). М.: ИПК Издательство стандартов. 9 с.
6. ГОСТ 21393–75. Межгосударственный стандарт. Автомобили с дизелями. Дымность отработавших газов. Нормы и методы измерений. Требования безопасности. М.: ИПК Издательство стандартов. 9 с.
7. Артюшенко В. М. Информационное обеспечение деятельности предприятий автосервиса // *Промышленный сервис*. 2009. № 4 (33). С. 3–10.
8. Программа самообучения 230. Выброс вредных веществ с отработавшими газами автомобильных двигателей. ООО "ФОЛЬКСВАГЕН Груп Рус" 2000. 27 с.
9. Нтциахристос Л., Самарас З. Руководство по инвентаризации выбросов ЕМЕП/ЕАОС 2009, редакция: июнь 2010 г. 129 с.
10. Звягинцева А. В. О вероятностном анализе данных наблюдений о состоянии природно-антропогенных систем в многомерных пространствах // *Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика*. 2016. Т. 37. № 2. С. 93–100.
11. Аббасов Э. М., Акимкина Э. Э. Достижение максимальной производительности при работе с крупными хранилищами данных // *Информационные технологии. Радиоэлектроника. Телекоммуникации (ITRT-2016): сб. статей VI международной заочной научно-технической конференции. Ч. 1 / Поволжский гос. ун-т сервиса. Тольятти: Изд-во ПВГУС, С. 7–12.*
12. Аббасов А. Э. Конвертирование трехмерных компьютерных геометрических моделей для оптимизации параметров моделируемых устройств // *Компьютерные исследования и моделирование*. 2015. Т. 7, № 1. С. 81–91.

Analysis of Expert Information on Environmental Indicators Vehicles

A. Je. Abbasov, alexn1smo@mail.ru, State Educational Institution of Higher Education of the Moscow region "University of Technology", Korolev, Moscow region, 141070, Russian Federation

Corresponding author:

Abbasov A. Je., Graduate Student, State Educational Institution of Higher Education of the Moscow region "University of Technology", 141070, Korolev, Moscow region, Russian Federation
E-mail: alexn1smo@mail.ru

*Received on October 11, 2016
Accepted on March 06, 2017*

The results of the analysis of the problematic issues that arise in the creation and support of information systems for measuring and regulation of hazardous substances coming from polluting vehicles are given in the article. Based on the theoretical analysis of the subject area for an automated information system, which is based on a database of environmental indicators, a structure of an Internet resource for storing and processing information on the environmental control of vehicles is proposed. For the development of the Internet resource tools MySQL database and content management system MODx Revolution, Apache and phpMyAdmin technologies are offered.

To automate the expert assessment of the environmental performance of vehicle we have analyzed domain characterizing the environmental impact of vehicles, for subsequent creation of software multi-user information system. We have analyzed and grouped individual facts characterizing the control of the environmental performance of

vehicles and the corresponding decrease in toxicity of the system. It revealed a generalized integral measure of the most important components of harmful substances emitted by motor vehicles in the atmosphere.

We used mathematical methods for processing experimental data obtained by the results of literature review to establish quantitative relationships between the technical characteristics of vehicles and environmental indicators. To create a statistical model of integrated environmental indicators, the method of the main components of the classical theory of factor analysis was applied.

We have defined the analytical dependences for calculating the integral of the generalized indicator of the level of environmental performance of vehicles. The choice of tools for Internet resource information system is proved.

Keywords: automated control, standardization of data, mathematical model, integrated environmental indicators, the performance of a web resource.

For citation:

Abbasov A. Je. Analysis of Expert Information on Environmental Indicators Vehicles, *Programmnyaya Ingeneria*, 2017, vol. 8, no. 5, pp. 222–229.

DOI: 10.17587/prin.8.222-229

References

1. **Kernozhitskaja A. F., Germanova T. V.** K voprosu zagraznenija atmosfernogo vozduha vyhlopnymi gazami avtotransporta (On the issue of air pollution by exhaust fumes of vehicles), *Sb. trudov konferencii "Problemy funkcionirovanija sistem Transporta"*, Tjumen', 14–16 November, 2012, Tjumen', Izd-vo Tjumenskogo industrial'nogo universiteta, 2012, pp. 184–191 (in Russian).

2. **Abbasov A. Je., Zinov'ev V. N., Vasil'ev N. A., Poserenin S. P.** Organizacija hranenija, obrabotki i peredachi informacii v avtomatizirovannoj informacionnoj sisteme jekologicheskogo kontrolja (Organization of storage, processing and transmission of data in the automated information system of environmental control), *Informacionnye tehnologii. Radioelektronika. Telekommunikacii (ITRT-2016): sb. statej VI mezhdunarodnoj zaochnoj nauchno-tehnicheskij konferencii. Povolzhskij gos. un-t servisa*, 24–25 March 2016, Tol'jatti, Izd-vo PVGUS, 2016, vol. 1, pp. 13–18 (in Russian).

3. **Elektronnyy** resurs "Za rulem", available at: <http://www.zr.ru>

4. **Shen G. T.** Environmental controls on uranium in reef corals, *Geochimica et Cosmochimica Acta*, 1995, vol. 59, no. 10, pp. 2009–2024.

5. **GOST R 52033–2003.** Avtomobili s benzinovymi dvigateľlyami. Vybrosov zagryznyayushchikh veshchestv s otrabotavshimi gazami. Normy i metody kontrolya pri otsenke tekhnicheskogo sostoyaniya (s Izmeneniyem no. 1) (GOST R 52033–2003. Cars with petrol engines. Emissions of pollutants from the exhaust gases. Norms and methods of control when assessing the technical condition (with Amendment N 1)), Moscow, IPK Izdatel'stvo standartov, 2003, 9 p. (in Russian).

6. **GOST 21393–75.** Mezhgosudarstvennyy standart. Avtomobili s dizelyami. Dymnost' otrabotavshikh gazov. Normy i metody izmereniy. Trebovaniya bezopasnosti (GOST 21393–75. Interstate standards. Cars with diesel engines. Exhaust gas opacity. Norms

and methods of measurement. Safety requirements), Moscow, IPK Izdatel'stvo standartov, 1975, 9 p. (in Russian).

7. **Artjushenko V. M.** Informacionnoe obespechenie dejatel'nosti predpriyatij avtoservisa (Information support of enterprises Car), *Promyshlennyj servis*, 2009, no. 4 (33), pp. 3–10 (in Russian).

8. **Programma** samoobucheniya 230. Vybrosov vrednykh veshchestv s otrabotavshimi gazami avtomobil'nykh dvigateley (Self-study program 230. The release of harmful substances in exhaust gases of automobile engines), OOO "FOL'KSVAGEN Grup Rus", 2000, 27 p. (in Russian).

9. **Nitsiakhristos L., Samaras Z.** Rukovodstvo po inventarizatsii vybrosov YEMEP/YEAOS (Inventory of EMEP/EEA emission Guide), 2009, ed.: June' 2010, 129 p. (in Russian).

10. **Zvjaginceva A. V.** O verojatnostnom analize dannyh nabljudenij o sostojanii prirodno-antropogennykh sistem v mnogomernykh prostranstvakh (Probabilistic analysis of the observational data on the state of natural and human systems in multidimensional spaces), *Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Serija: Jekonomika. Informatika*, 2016, vol. 37, no. 2, pp. 93–100 (in Russian).

11. **Abbasov Je. M., Akimkina Je. Je.** Dostizhenie maksimal'noj proizvoditel'nosti pri rabote s krupnymi hranilishhami dannyh (Achieving peak performance when working with large data stores), *Informacionnye tehnologii. Radioelektronika. Telekommunikacii (ITRT-2016): sb. statej VI mezhdunarodnoj zaochnoj nauchno-tehnicheskij konferencii. Povolzhskij gos. un-t servisa*, 24–25 March 2016, Tol'jatti, Izd-vo PVGUS, 2016, vol. 1, pp. 7–12 (in Russian).

12. **Abbasov A. Je.** Konvertirovaniye trekhmernykh komp'yuternykh geometricheskikh modeley dlya optimizatsii parametrov modeliruyemykh ustroystv (Conversion of three-dimensional computer geometric models to optimize the parameters of simulated devices), *Komp'yuternyye issledovaniya i modelirovaniye*, 2015, vol. 7, no. 1, pp. 81–91 (in Russian).

М. М. Стронгин, ст. науч. сотр., e-mail: strongin@imec.msu.ru,
Ю. А. Виноградов, канд. техн. наук, доц., e-mail: vinograd@imec.msu.ru,
А. Г. Здидовец, канд. техн. наук, доц., e-mail: zditovets@mail.ru,
Н. А. Киселёв, аспирант, e-mail: kiselev.nick.a@gmail.com,
С. С. Попович, канд. техн. наук, науч. сотр., e-mail: pss1@mail.ru,
Научно-исследовательский институт механики Московского государственного университета им. М. В. Ломоносова

Использование технологий National Instruments в экспериментальных исследованиях процессов термогазодинамики

Представлен опыт авторов, полученный при использовании технологий National Instruments в экспериментальных исследованиях в термогазодинамике. На примерах оснащения экспериментальных стендов для исследования различных аспектов теплообмена контрольно-измерительным оборудованием и программным обеспечением для работы с этим оборудованием продемонстрирована оперативность создания аппаратно-программных средств самими экспериментаторами без привлечения профессиональных программистов.

Ключевые слова: National Instruments, LabVIEW, термогазодинамика, теплообмен, аэродинамическая установка, виртуальный прибор

Введение

Поиск путей интенсификации теплообмена привлекал, привлекает и будет привлекать пристальное внимание инженеров, научных сотрудников и всех, кто тем или иным образом связан с теплоэнергетикой. От эффективности передачи энергии в теплообменном оборудовании во многом зависят значения технико-экономических показателей всей энергоустановки в целом. Следовательно, в настоящее время использование эффективных способов интенсификации процессов теплообмена является одним из главных и наиболее доступных путей совершенствования, повышения эффективности и экономичности энергетического оборудования. Такой вывод подтверждается постоянно возрастающим числом публикаций, посвященных данному вопросу. В связи с возрастающим интересом к проблеме интенсификации теплообмена в Институте механики Московского государственного университета им. М. В. Ломоносова (МГУ) совместно с Московским Государственным техническим университетом им. Н. Э. Баумана создана совместная межвузовская научно-учебная лаборатория "Термогазодинамика". Основными приоритетными направлениями экспериментальных работ этой лаборатории были определены следующие задачи: исследование теплоги-

дравлической эффективности (соотношение между теплообменом и сопротивлением) "вихреобразующих" поверхностей; влияние вдува охлаждающего газа через пористые поверхности на эффективность тепловой защиты; исследование газодинамической температурной стратификации (безмашинного энергоделения) потоков газа — труба Леонтьева.

Задачи экспериментальной термогазодинамики

Наиболее приемлемым методом интенсификации теплообмена в теплообменном оборудовании является применение различных турбулизаторов потока. Существуют различные типы интенсификаторов теплообмена, реализующих данный подход. Однако использование большинства из них (ребер, штырьков, закрученных лент и пр.) обычно ведет к существенному увеличению гидравлического сопротивления. Тем интереснее результаты, полученные на теплообменных поверхностях с вихревыми интенсификаторами ("лунками"), так как в данном случае росту теплообмена соответствует практически равноценный рост гидравлического сопротивления. Благодаря этим качествам в настоящее время к таким поверхностям проявляется повышенный интерес [1]. Существует большое число работ, посвящен-

ных экспериментальному и численному исследованию вихреобразующих поверхностей. Однако до сих пор остаются открытыми вопросы об оптимальной (с точки зрения теплогидравлической эффективности) форме, компоновке и взаимном влиянии лунок в каналах. Несмотря на большое число работ по вихревой интенсификации, остаются вопросы о способах определения искомым величин и достоверности опубликованных экспериментальных данных (в том числе теплогидравлических характеристик наиболее эффективных "облуненных" поверхностей). Следует отметить, что в современных исследованиях (в связи с развитием экспериментального оборудования, уточнением методов проведения эксперимента и численных расчетов) вихреобразующие поверхности показывают менее "выдающиеся" результаты. Несмотря на это, лунки до сих пор остаются перспективными интенсификаторами теплообмена. Все эти факторы свидетельствуют о необходимости создания новых методов определения теплогидравлических характеристик, проведения детальных исследований вихреобразующих поверхностей с применением современного высокоточного оборудования.

Развитие авиационной и космической техники, а также промышленной энергетики обуславливает интерес к методам тепловой защиты поверхности аппаратов от воздействия высокотемпературных и высокоскоростных газовых потоков.

Теоретическое решение этой проблемы связано с большими трудностями и поэтому представляет интерес экспериментальное исследование некоторых аспектов этой проблемы. Одним из наиболее перспективных способов защиты поверхности является так называемое "пористое" охлаждение [2]. В этом случае стенки аппарата частично или полностью изготавливаются из специального пористого материала, способного пропускать через себя охладитель (жидкость или газ), что приводит к снижению теплового потока через поверхность аппарата. Другим способом защиты является применение тепловых завес [3]. Охладитель подается через непрерывную щель, через пористую стенку или через устройство, состоящее из более коротких щелей или ряда дискретных отверстий перед элементом конструкции, подлежащим защите, и создает на поверхности пленку, предохраняющую элементы конструкции от воздействия высокотемпературного газового потока.

Интерес к газодинамическому безмашинному методу энергоразделения был вызван исследованиями энергоустановок космического назначения для осуществления пилотируемой экспедиции на Марс. В наиболее эффективной компоновке с использованием энергодвигательного комплекса на основе ядерных и газотурбинной установок, работающих по замкнутому циклу Брайтона, в качестве рабочего тела рассматривается смесь легких газов с тяжелыми и низкими значениями критерия Прандтля. В этом случае перепад температур между температурой торможения и температурой адиабатной стенки, определяющей количество переданной теплоты между

дозвуковым и сверхзвуковым потоками газа, разделенными теплопроводной стенкой, может достигать существенных значений, что позволяет предложить принципиально новые эффективные термодинамические циклы замкнутых газотурбинных установок и газотранспортных систем [4].

Пути решения экспериментальных задач

С учетом изложенного выше, решение перечисленных и подобных им задач при физическом моделировании связано с изучением тепловых полей, а это, в свою очередь, диктует необходимость многоточечных измерений, которые в целях снижения энергозатрат и повышения надежности исследования приходится проводить на автоматизированных измерительных комплексах.

Поставленные выше задачи по исследованию процессов теплообмена при различных граничных условиях выполнялись на сверхзвуковых аэродинамических установках с различной рабочей частью и различными расходами рабочего газа, а также на малой дозвуковой аэродинамической установке с возможностью управлять скоростью потока в диапазоне 0...130 м/с. Все экспериментальные стенды укомплектованы современным измерительным оборудованием, инфракрасными тепловизионными приборами для наблюдения температурных полей на исследуемых поверхностях, оптическим оборудованием для визуализации исследуемых процессов, а также оснащены современными сенсорными устройствами — датчиками давления, термопарами, тензочувствительными аэродинамическими весами.

На рис. 1 показан экспериментальный стенд на базе сверхзвуковой аэродинамической трубы исследования эффективности тепловой защиты при вдуве газа через пористые поверхности.

Наиболее распространенными исследованиями в экспериментальной аэродинамике являются рабо-

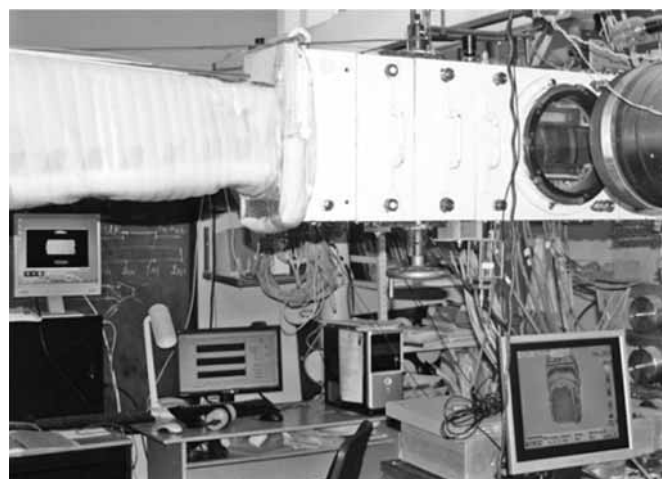


Рис. 1. Экспериментальный стенд исследования эффективности тепловой защиты при вдуве газа через пористые поверхности ("пористое" охлаждение)

ты по изучению взаимодействия модели с набегающим газовым потоком на аэродинамических установках с различными параметрами, характеризующими такое взаимодействие. При этом анализируют:

- поля давления в различных областях модели или в следе за моделью;
- силомоментные характеристики процесса взаимодействия потока с моделью;
- поля температуры на поверхности модели или поля температуры в следе за моделью при изучении термодинамического взаимодействия потока с моделью.

Сложная организация самого аэродинамического эксперимента обусловлена технологической сложностью как базовых средств его проведения (аэродинамические установки), так и используемого в этих целях измерительного оборудования.

Выбор средств решения

В Институте механики МГУ накоплен определенный опыт по разработке автоматизированных систем экспериментальных исследований в области аэродинамики [5, 6]. Первыми были системы на базе компьютеров СМ и РС/АТ/ХТ, модульной аппаратуры в стандарте КАМАК и набора нормализующих устройств. Создание таких систем предполагало присутствие в штате исследователей программистов, да и сама процедура создания программы с помощью традиционных языков программирования достаточно затратная по времени, которое требуется на компиляцию кода и последующую отладку программ (и так, иногда, по несколько раз). Кроме того, есть некоторые трудности в представлении необходимой оперативной экспресс-информации в процессе эксперимента.

При проектировании аппаратно-программного комплекса для решения предполагаемых задач термогазодинамики было решено остановиться на технологиях, предлагаемых фирмой National Instruments. Причиной такого решения была широкая хорошо проработанная номенклатура коммутационной и измерительной аппаратуры с очень удобным программным обеспечением в виде среды графического программирования LabVIEW [7]. На разных этапах экспериментов в качестве аппаратных средств использовался целый набор устройств с шиной USB для соединения с компьютером, в который входили многоканальные модули: NI USB 6210, NI USB 6218, NI USB 6225, NI USB 6255, NI USB 9213, многоканальный аналогово-цифровой преобразователь NI PCI-6071 и контакторный блок NI SCB-100, а также комплект на базе шасси SCXI с контакторными модулями SCXI-1303, мультиплексор-усилитель термопарный SCXI-1102, мультиплексор-усилитель SCXI-1102B и многоканальный аналого-цифровой преобразователь PCI 6220. Общий объем измерительных каналов в этом комплекте — 256, что удовлетворяет потребности при исследовании больших полей измеряемых параметров. Использование устройств с шиной USB

и ноутбуков позволяет строить легкие и мобильные комплексы для работы даже в полевых условиях.

Для экспериментаторов LabVIEW представляет возможности гибкости и универсальности традиционных программных языков, например, таких как С. В сущности, язык графического программирования, предоставленный LabVIEW, состоит из тех же конструкций, что и традиционные языки программирования, включая переменные, типы данных, структуры циклов и последовательностей, а также средства обработки ошибок. К тому же, LabVIEW снабжен развитой интегрированной контекстной справочной системой.

Быстро создавать наглядные приложения автоматизации измерений и испытаний позволяет графическая среда разработки приложений LabVIEW со специализированными библиотеками функций. Графическое программирование и использование принципа потока данных открывает интуитивно понятный экспериментаторам подход к созданию автоматизированных измерительных систем. Создание приложений сводится к извлечению из палитры и вставке в проект любой из множества встроенных функций. Следует отметить, и это подтверждают эксперименты, что разработанная таким образом программа обладает скоростью исполнения, которая сопоставима со скоростью программ, разработанной на языке Си. Кроме того, LabVIEW позволяет проверить правильность работы программы с помощью таких интегрированных графических средств отладки, как, например, исполнение графических узлов по шагам или транзитом.

Процесс программирования виртуальных приборов превращается из процесса, который подразумевает процедуру последующей отладки программы, в процесс построения логической схемы взаимодействия набора инструментальных средств, который необходим для решения поставленной задачи, состоящей из сбора данных, их анализа и эффективного представления полученных результатов. Все перечисленное выше позволяет обходиться без профессионалов-программистов в коллективе экспериментаторов. Программы, созданные в среде LabVIEW, называют виртуальными приборами, потому что их внешний вид и выполняемые ими функции схожи с внешним видом и функциями реальных физических приборов.

Особенности инструментальных средств для решения отдельных задач

Средствам автоматизации исследований в аэродинамике, и в частности, в термогазодинамике, присущи некоторые общие черты (структуры, этапы технологического цикла) вне зависимости от физической сущности обслуживаемого эксперимента.

Учитывая, что работа проводится в расчете на проведение эксперимента на аэродинамических установках, в темпе эксперимента происходит накопление наблюдений с последующей обработкой

накопленной информации. В условиях термодинамического эксперимента, когда время выхода на заданный режим в связи с большими теплоемкостями элементов стенда значительно, появляется возможность проводить первичную обработку данных в темпе эксперимента, что позволяет сократить время эксперимента в целом.

Структура программных средств экспериментальных исследований в предметной области, связанной с условиями проведения эксперимента и потребностями адекватных методов обработки информации и представления результатов экспериментальных исследований, может быть представлена и следующем виде:

- 1) планирование эксперимента в целом и его отдельных этапов;
- 2) тестирование измерительно-регистрирующего тракта;
- 3) градуировка измерительно-регистрирующего тракта;
- 4) контроль режимных параметров экспериментальной установки;
- 5) съем и накопление наблюдений, управление приводами исполнительных механизмов экспериментального стенда;
- 6) обработка результатов отдельных этапов эксперимента;
- 7) обработка результатов эксперимента в целом и их представление.

Отметим некоторые особенности использования инструментальных средств LabVIEW на перечисленных этапах. На этапе планирования эксперимента определяется порядок работы с измерительными каналами измерительно-регистрирующего тракта (порядок опроса каналов и число циклов опросов). Затем определяется программа положения модели или ее элементов на каждом этапе эксперимента, а также программа управления перемещением зондов с сенсорными элементами с помощью координатных механизмов, и вводятся необходимые коэффициенты, константы и различные паспортные данные.

Тестирование измерительно-регистрирующего тракта может осуществляться на нескольких этапах проведения эксперимента. Для каждого этапа характерен свой способ. К элементам тестирования следует отнести программные блоки, позволяющие проверить отработку исполнительными механизмами и устройствами различных команд и их последовательности согласно алгоритму основной программы. Программы тестирования разрабатываются в среде LabVIEW в виде виртуальных приборов. Для тестирования можно использовать приложение LabVIEW Measurements & Automation Explorer.

На этапе градуировки измерительно-регистрирующего тракта необходимо проверить метрологические характеристики тракта, подавая на его вход эталонный сигнал и анализируя прохождения этого сигнала по всему тракту. Градуировку удобно проводить с помощью эталонных калибраторов, если

есть возможность управления калибратором по шине с компьютером.

Обычно собственно эксперименту предшествует градуировка измерительно-регистрирующего тракта. Процессы градуировки однокомпонентных и многокомпонентных датчиков несколько отличаются друг от друга. Различия сохраняются также при обработке градуировочных данных и при построении градуировочной зависимости.

Процесс градуировки заключается в серии эталонных воздействий на чувствительный элемент одного датчика или группы однотипных датчиков по заранее оформленному плану и регистрации соответствующих оцифрованных показаний измерительной аппаратуры. Данные формируются в виде таблицы откликов регистрирующей аппаратуры U и значений физического воздействия на чувствительный элемент датчика P и записываются в файл. Программа обработки данных градуировки начинает свою работу со считывания файла с градуировочным материалом. Одновременно можно обрабатывать несколько градуировок (до десяти), проведенных в разное время. Для рассматриваемого датчика формируется последовательность нагружений P_i и последовательность откликов регистрирующей аппаратуры U_i по всему объему введенного градуировочного материала. По полученному набору опорных точек (U_i, P_i) методом наименьших квадратов строятся аппроксимации, включающие более десятка разновидностей функциональных зависимостей, в том числе полиномы (до пятого порядка), логарифмическую и степенную функции, а также некоторые комбинации перечисленных элементарных функций. Затем с использованием критерия Фишера выбирается зависимость с наименьшей остаточной дисперсией, которая и принимается за искомую градуировочную функцию:

$$P = f(U).$$

За погрешность определения градуировочной зависимости принимается величина

$$\Delta P = \frac{St(X, n) \sqrt{\sum_{i=1}^n (P_i - f(U_i))^2}}{n},$$

где $St(X, n)$ — квантили распределения Стьюдента для выборки объемом n с доверительной вероятностью X ; P_i — серия эталонных нагружений; U_i — серия откликов регистрирующей аппаратуры; $f(U_i)$ — серия восстановленных значений нагружений.

Градуировка признается удовлетворительной, если относительная погрешность градуировки, определенная по формуле

$$\overline{\Delta P}_t = \frac{\Delta P}{P_{\min} + 0,5(P_{\max} - P_{\min})} \cdot 100\%,$$

где ΔP — погрешность градуировочной зависимости; P_{\min} — минимальное значение диапазона датчика;

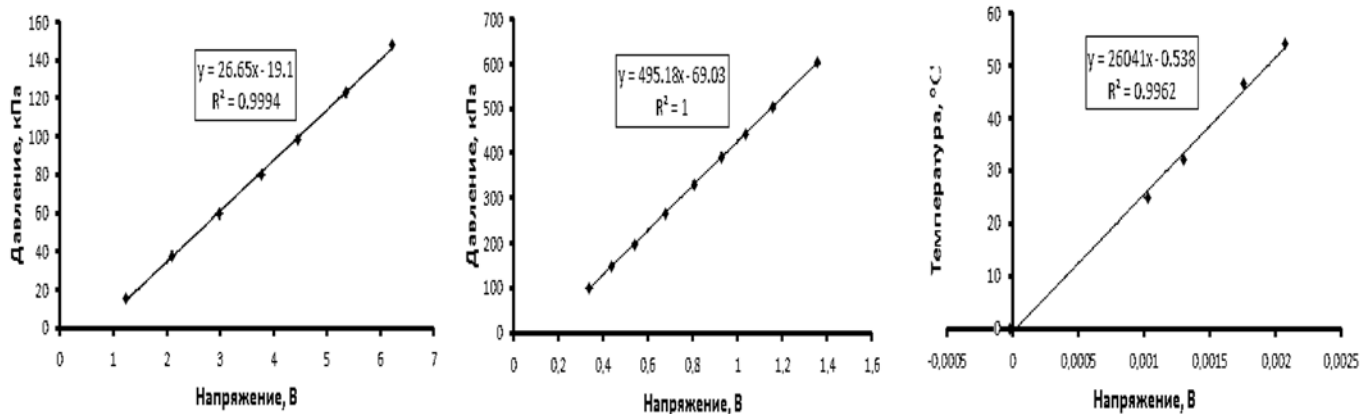


Рис. 2. Градуировочные характеристики датчиков:

a — датчик давления ИКД27Да-1500; *б* — датчик давления Honeywell ML300PS2PC; *в* — хромель-копелевых термопар ($y = f(x)$ — уравнение аппроксимации градуировочной зависимости; R^2 — значение достоверности аппроксимации)

P_{\max} — максимальное значение диапазона датчика, не превосходит 3...5 %.

Опыт работы с современными датчиками показывает, что, как правило, они аппроксимируются линейными функциями. Поэтому возможно упростить процедуру градуировки и свести программы накопления наблюдений с программой обработки градуировочной зависимости в один блок. Для примера на рис. 2 представлены характеристики применяемых в экспериментах датчиков.

При градуировке датчиков давления использовались калибраторы давления типа РС-6 и DPM, связь с компьютером осуществлялась по шине COM-2 под управлением программы в среде LabVIEW. Все программы градуировки выполнены в виде виртуальных приборов.

Результаты градуировки всех используемых в экспериментальных работах датчиков формируются в сводную таблицу в виде некоторой базы данных, в которую заносятся все данные датчиков. Из этой таблицы потом легко формируются файлы с данными набора датчиков для конкретного эксперимента.

Во всех экспериментальных исследованиях требуется определять большое число параметров (поля давлений и температур), что приводит к необходимости накапливать информацию с различного рода сенсорных устройств. Для этих целей экспериментальный стенд оборудован аппаратурой, способной работать в многоточечном режиме и отвечать высоким требованиям по надежности и точности регистрируемых данных. На рис. 3 показаны компоненты системы сбора данных, в состав которой входят

датчики и измерительные преобразователи, нормирующие устройства (устройства кондиционирования сигналов), соединительные кабели между различными устройствами и аксессуарами к ним, программное обеспечение и персональный компьютер.

Общие принципы построения виртуальных приборов для экспериментальных работ по обсуждаемой тематике сводятся к программированию следующих процедур:

1) инициирование оборудования, которое состоит из выбора типа и расположения коммутационно-измерительного модуля; порядка опроса измерительных каналов; режима опроса измерительных каналов; ввода идентификаторов имен файлов, в которые записывается накопленная информация; ввода имен файлов, из которых считывается необходимая для проведения эксперимента информация (градуировочные коэффициенты, необходимые константы); оформления режима работы необходимых исполнительных устройств;

2) формирование формы протокола для записи накопленной информации;



Рис. 3. Общая схема формирования контрольно-измерительной системы

- 3) организация мониторов для наблюдения за выходом стенда на заданный режим;
- 4) управление записью накопленной информации;
- 5) выключение измерительной аппаратуры по окончании этапа эксперимента.

Примеры решения некоторых задач

Рассмотрим примеры схем измерительно-регистрающих трактов экспериментальных стендов.

Экспериментальный стенд исследования теплообмена (соотношение между теплообменом и сопротивлением) на "вихреобразующих" поверхностях. Комплект аппаратных средств (рис. 4) стенда исследования теплообмена на "вихреобразующих" поверхностях состоит из набора датчиков давления, термопар, аэродинамических тевовесов, многофункционального блока NI PCI-6071 (мультиплексор, аналого-цифровой преобразователь, цифро-аналоговый преобразователь), контакторного блока NI SCB-100, термопарного мультиплексора-усилителя NI USB-9213, координатных устройств перемещения зондов и персонального компьютера (ПК). При градуировке датчиков давления использовался калибратор РС-6.

Экспериментальный стенд исследования газодинамической температурной стратификации (безмашинное температурное разделение) потоков. Комплект аппаратных средств (рис. 5) экспериментального стенда исследования газодинамической температурной стратификации состоит из набора датчиков давления, термопар, датчика относительной влажности газа, модуля NI USB-6255 (мультиплексор,

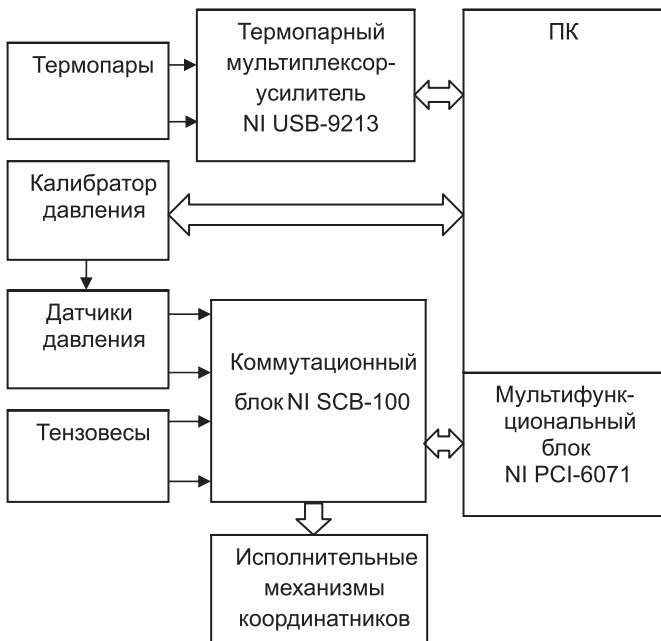


Рис. 4. Пример схемы измерительно-регистрающей системы на базе модуля с шиной USB и встроенного аналого-цифрового преобразователя

аналого-цифровой преобразователь, цифро-аналоговый преобразователь), термопарного мультиплексора-усилителя NI USB-9213, координатных устройств перемещения зондов и персонального компьютера. При градуировке датчиков давления использовался калибратор РС-6.

Экспериментальный стенд для исследования эффективности тепловой защиты при вдуве охлаждающего газа через пористые поверхности ("пористое" охлаждение) и влияния падающей ударной волны на адиабатную температуру стенки и коэффициент восстановления температуры. Комплект аппаратных средств (рис. 6) экспериментального стенда исследования эффективности тепловой защиты при вдуве охлаждающего газа через пористые поверхности и влияния падающей ударной волны на адиабатную температуру стенки и коэффициент восстановления температуры состоит из набора датчиков давления, термопар, модульной платформы SCXI, состоящей из четырех контакторных блоков SCXI-1303, двух мультиплексоров-усилителей SCXI-1102B для работы с датчиками давления, двух мультиплексоров-усилителей SCXI-1102 для работы с термопарами, многофункционального блока NI PCI-6220, координатных устройств перемещения зондов и персонального компьютера. При градуировке датчиков давления использовался калибратор РС-6.

Примеры виртуальных приборов экспериментальных стендов

На рис. 7 показан общий вид виртуального прибора управления градуировкой датчиков в среде LabVIEW. На экране отображается представление виртуального прибора с выводами результатов наблюдения. Левая часть экрана — инициализация оборудования, правая часть экрана — монитор, отображающий выход параметров каждого градуируемого датчика на заданный режим и результаты первичной обработки данных (среднее значение, отклонение в абсолютных и относительных величинах).

На рис. 8 в логической последовательности работы слева направо расположены: в левой части кода функции ввода параметров и инициализация оборудования, далее — часть кода, которая осуществляет выход параметров измерительных каналов на заданный режим и показывает этот процесс на мониторе в режиме реального времени.

На рис. 9 показан только фрагмент виртуального прибора, отображающий процесс наблюдений контроля в процессе эксперимента таких параметров, как полное давление, температура торможения, число Маха набегающего потока, распределение по длине модели статического давления, числа Маха и температуры стенки. На экран выведены мониторы выхода параметров на установившийся режим и элементы управления записью информации в файл.

На рис. 10 показана часть кода, которая выполняет чтение информации с учетом градуировоч-

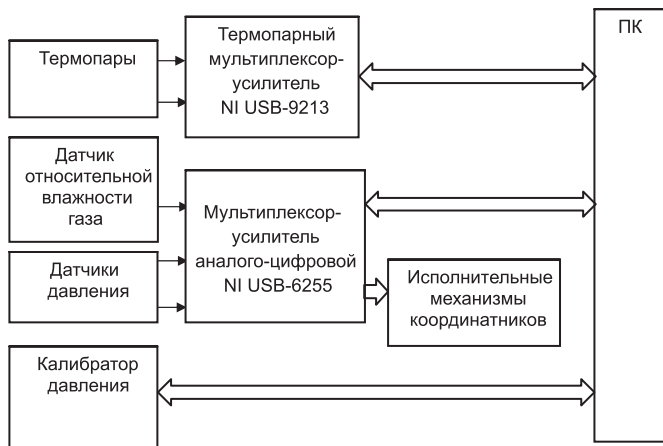


Рис. 5. Пример схемы измерительно-регистрающей системы на базе модулей с шиной USB

ных коэффициентов каналов давления и температур. После чего параметры выводятся на монитор для контроля выхода на установившийся режим.

На рис. 11 (см. третью сторону обложки) показано окно монитора вывода на заданные режимные параметры потока и управление записью информации. Информация выводится в виде осциллограмм для отслеживания динамики процесса и в цифровом виде, как результат осреднения за определенный интервал времени.

На рис. 12, а (см. третью сторону обложки) представлена часть кода, которая выполняет считывание информации с учетом градуировочных коэффициентов каналов давления и температур. После чего параметры выводятся на монитор для контроля выхода

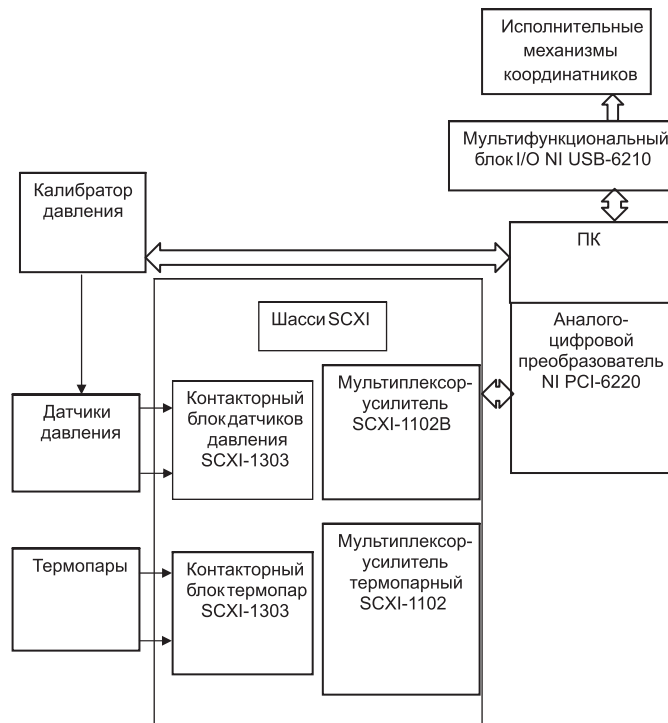


Рис. 6. Пример схемы измерительно-регистрающей системы на базе платформы SCXI

на установившийся режим. На рис. 12, б (см. третью сторону обложки) показана часть кода управления исполнительными устройствами координатников и считывания накопленных наблюдений.

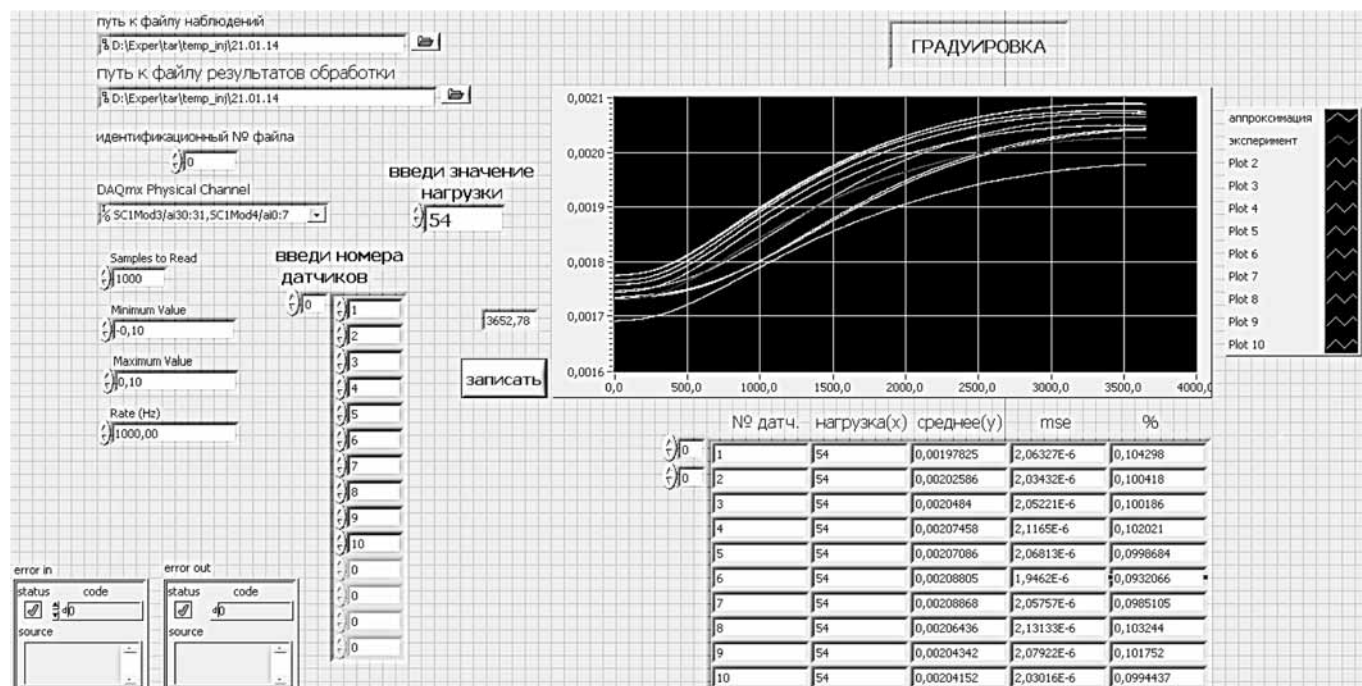


Рис. 7. Общий вид окна управления виртуального прибора градуировкой датчиков в среде LabVIEW

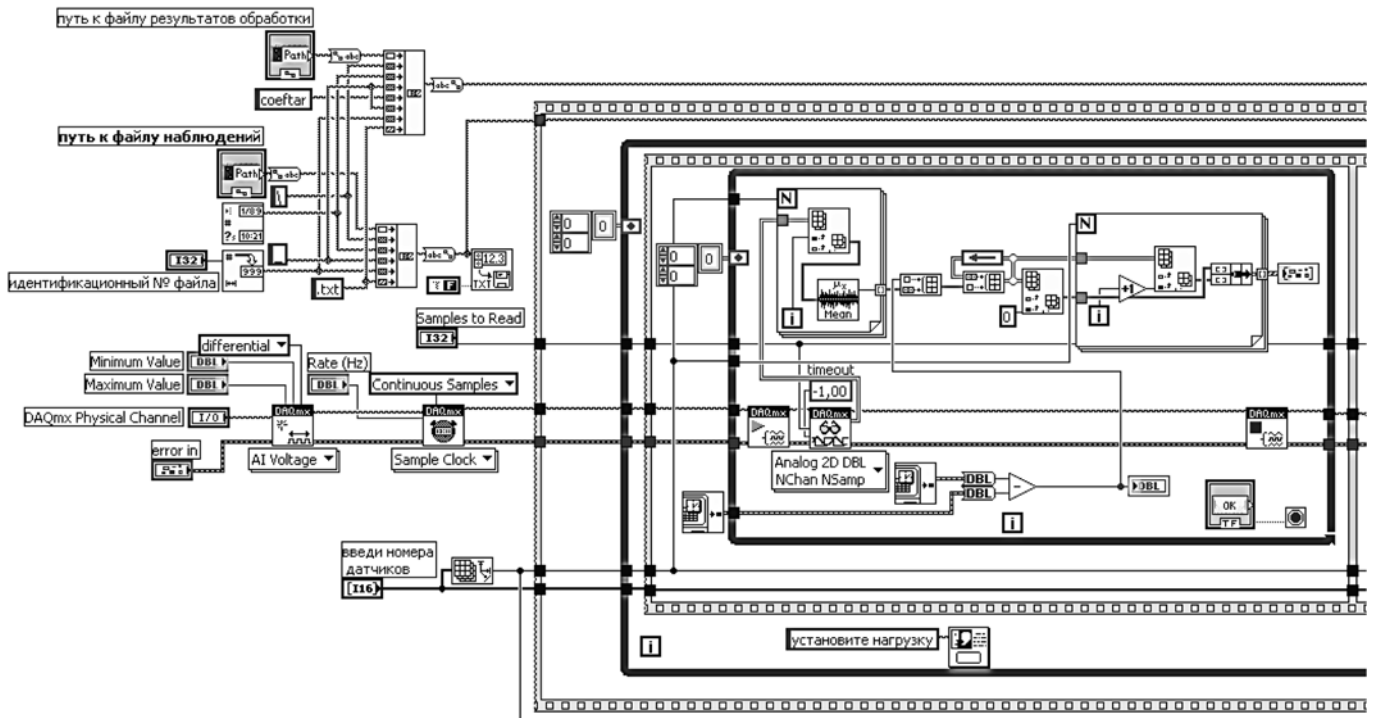


Рис. 8. Фрагмент кода программы градуировки датчиков в среде LabVIEW

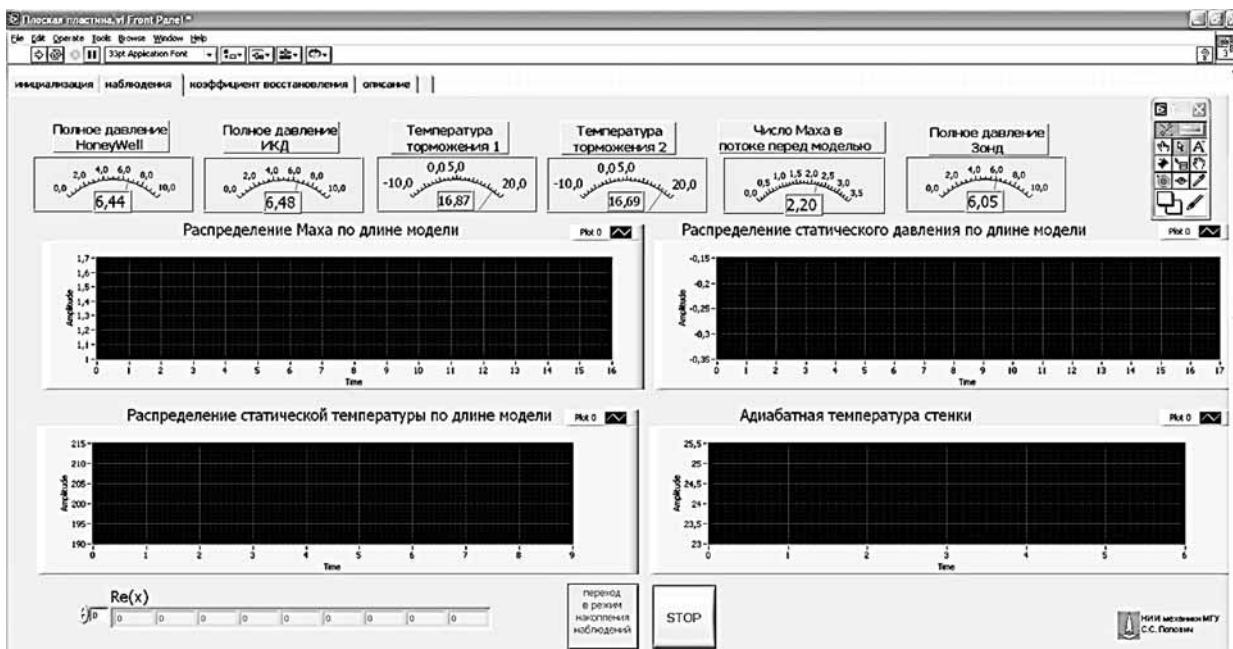


Рис. 9. Фрагмент виртуального прибора исследования эффективности тепловой защиты при вдуве охлаждающего газа через пористые поверхности ("пористое" охлаждение) в среде LabVIEW

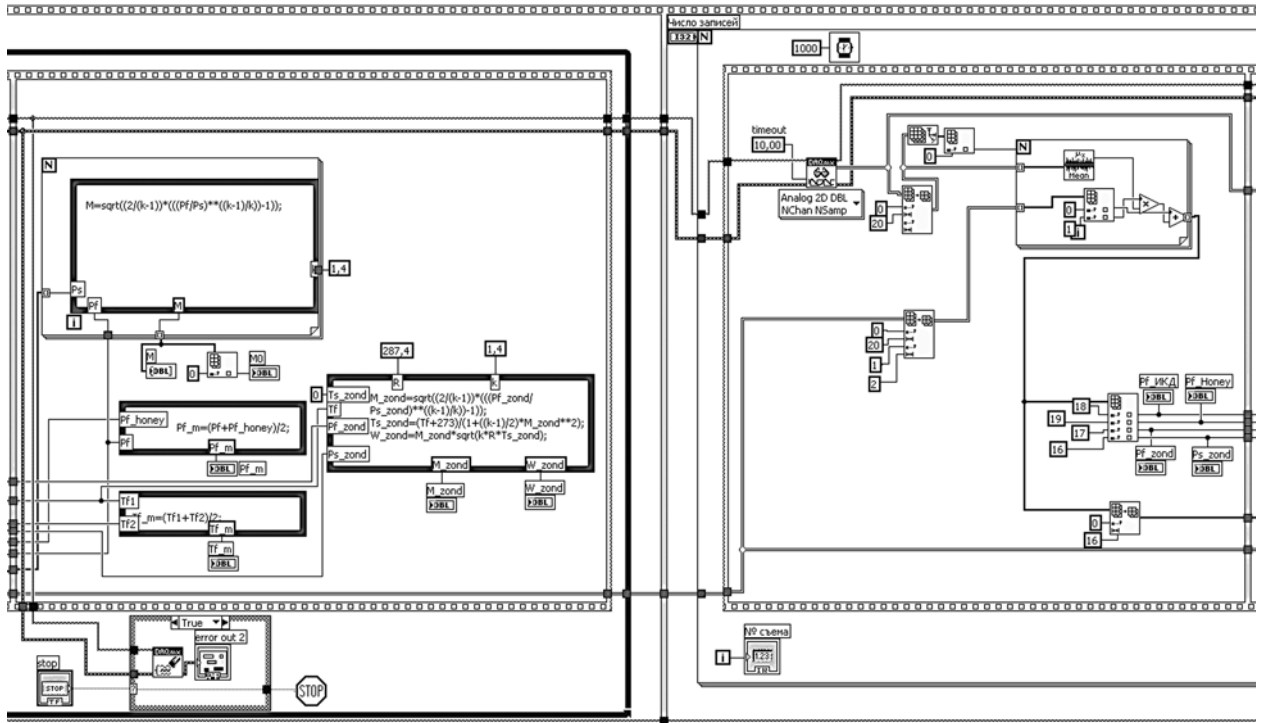


Рис. 10. Фрагмент кода программы исследования эффективности тепловой защиты при вдуве охлаждающего газа через пористые поверхности ("пористое" охлаждение) в среде LabVIEW

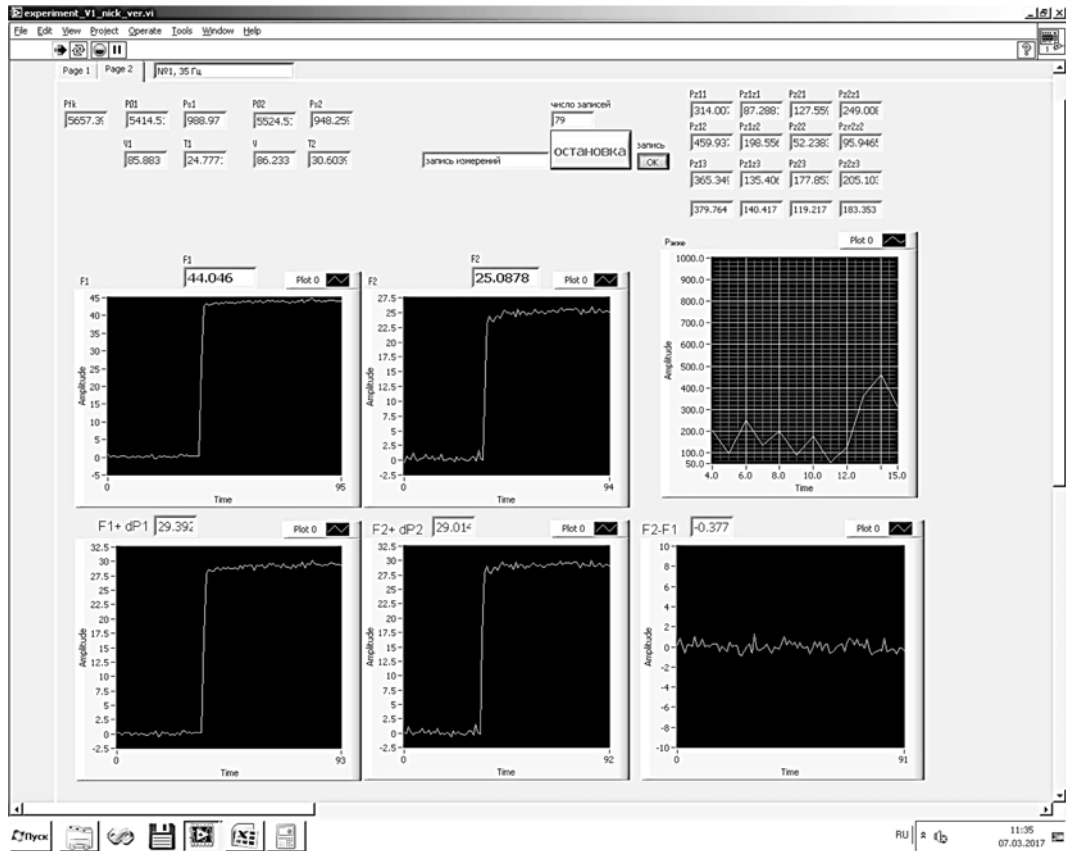


Рис. 13. Фрагмент виртуального прибора экспериментального исследования теплообмена на "вихреобразующих" поверхностях в среде LabVIEW

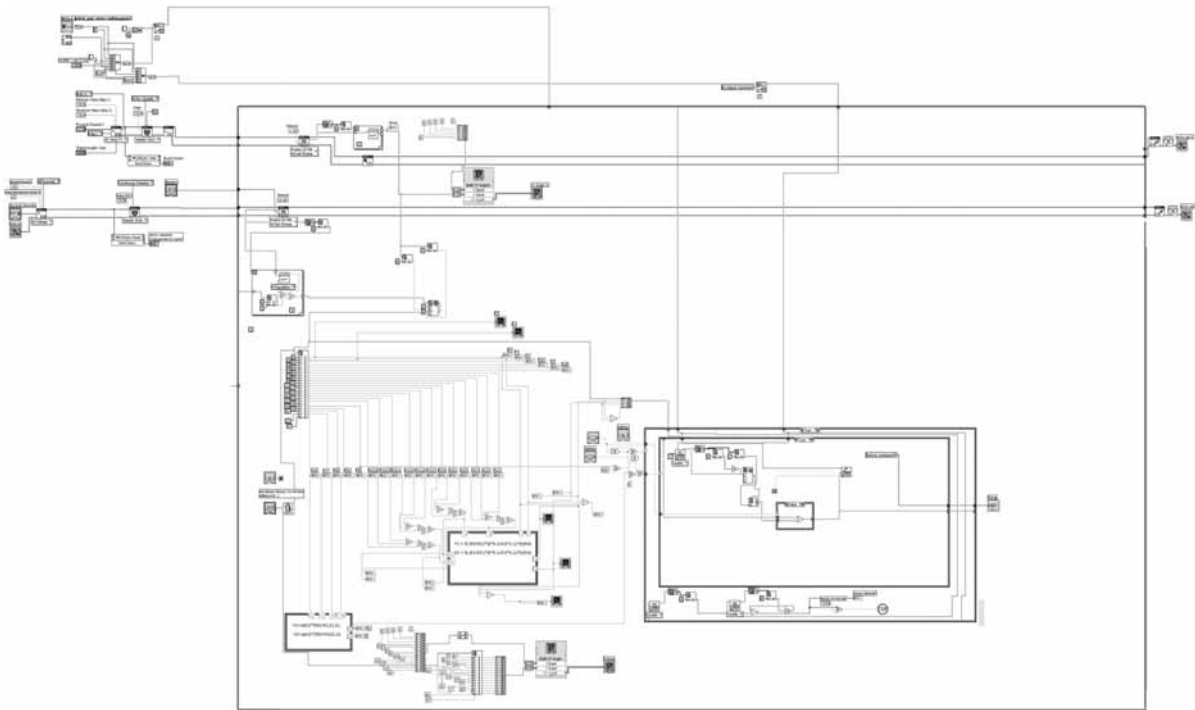


Рис. 14. Фрагмент кода программы экспериментального исследования теплообмена на "вихреобразующих" поверхностях в среде LabVIEW

На рис. 13 показан фрагмент виртуального прибора экспериментального стенда исследования теплообмена (соотношение между теплообменом и сопротивлением) на "вихреобразующих" поверхностях. На окно монитора выведены осциллограммы параметров выхода на заданные режимные параметры потока для отслеживания динамики процесса, цифровые индикаторы как результат осреднения за определенный интервал времени и элементы управления записью информации.

На рис. 14 показана часть кода виртуального прибора монитора вывода стенда на заданные режимные параметры потока и температурного режима модели.

Заключение

Опыт экспериментальной работы в области термогазодинамики в лаборатории с использованием систем регистрации данных с помощью оборудования фирмы National Instruments, автоматизированных программ опроса и обработки показаний сенсорных устройств в среде LabVIEW и первичной математической обработки накопленной информации демонстрирует высокую эффективность применяемых технологий, связанную с хорошей проработкой измерительного оборудования (многофункциональность, многоканальность, быстрдействие и высокое разрешение). Программное обеспечение аппаратно-измерительных комплексов экспериментальных стендов в среде LabVIEW может быть разработано без помощи профессиональных программистов самими экспериментаторами-исследователями

на основе интуитивно понятного подхода к созданию автоматизированных измерительных систем. В целом, аппаратно-измерительный комплекс получается очень мобильным и легко перестраиваемым даже в процессе эксперимента. На экспериментальных стендах проводятся лабораторные работы специального факультета МГУ имени М. В. Ломоносова и факультета энергомашиностроения МГТУ им. Н. Э. Баумана.

Работы выполнялись при финансовой поддержке РФФИ (грант 15-08-08428).

Список литературы

1. Leontiev A. I., Kiselev N. A., Zditovets A. G., Burtcev S. A., Strongin M. M., Vinogradov Yu. A. Experimental investigation of heat transfer and drag on surfaces with spherical dimples // Exp. Temp. Fluid Sci. 2016. Vol. 79. P. 74–84.
2. Кикнадзе С. С., Леонтьев А. И. Теплообмен и трение в турбулентном пограничном слое. М.: Энергия, 1985. 320 с.
3. Волчков Э. П. Пристенные газовые завесы. Новосибирск: Наука, 1983. 240 с.
4. Здиовец А. Г., Виноградов Ю. А., Стронгин М. М., Титов А. А., Киселев Н. А. Безмашинное энергоразделение газовых потоков. М.: Курс, 2016. 110 с.
5. Стронгин М. М. Создание и развитие автоматизированных систем физического моделирования в механике. Пакет программ для проведения аэродинамического эксперимента. НИИ механики МГУ, Отчет № 4304, 1993. 36 с.
6. Виноградов Ю. А., Стронгин М. М. Пакет прикладных программ для проведения аэродинамического эксперимента (дополнение для решения задач эффективности "пористого" охлаждения). НИИ механики МГУ, Отчет № 4404, 1995. 68 с.
7. Тревис Д. LabVIEW для всех. Пер. с англ. Клушин Н. А. М.: ПриборКомплект, 2005. 537 с.

Applying of National Instruments Technologies in Experimental Research of Thermal Gas Dynamics Processes

M. M. Strongin, strongiv@imec.msu.ru, **Yu. A. Vinogradov**, vinograd@imec.msu.ru,
A. G. Zditovets, zditovets@mail.ru, **N. A. Kiselev**, kiselev.nick.a@gmail.ru,
S. S. Popovich, pss1@mail.ru, Lomonosov Moscow State University, Institute of mechanics, Moscow,
119899, Russian Federation

Corresponding autor:

Strongin Mark M., Senior Researcher, Moscow Lomonosov State University, Institute of Mechanics, Moscow,
119899, Russian Federation,
E-mail: strongin@imec.msu.ru

Received on March 10, 2017

Accepted on March 27, 2017

Research of heat and mass transfer augmentation methods has attracted, attracts and will attract great attention of engineers, research workers and all those who are by some way connected with heat power engineering. At present, application of efficient methods of heat transfer processes augmentation is one of the main and most accessible ways to improve the efficiency and economy of power equipment. The paper presents the authors' experience gained with the use of National Instruments technologies during experimental studies of thermal gas dynamics. Examples of equipping an experimental test stand for studying various aspects of heat and mass transfer by measuring equipment and special software demonstrate the efficiency of creating hardware and software packages solely by experimenters without professional programmers involvement. Successful experimental experience in the laboratory of thermal gas dynamics is described including National Instruments data recording systems, automated programs for sampling and processing of sensor device readings in LabVIEW development environment and primary mathematical processing of the accumulated data. These examples demonstrate high efficiency of the applied technologies in connection with high quality engineering of the measuring equipment (multifunctionality, multichannel, high response speed and high resolution). The software of experimental hardware measuring units in the LabVIEW environment can be easily developed by researchers thanks to intuitive approach to the creation of automated measuring systems. In general, a hardware measuring unit is rather mobile and easily reconfigurable even during the experiment. Special students' practical works of the Mechanics and Mathematics faculty of Lomonosov Moscow State University and the Power Engineering faculty of Bauman Moscow State Technical University are conducted on the basis of the Thermal Gas Dynamics Laboratory experimental test stand.

Keywords: National Instruments, LabVIEW, thermal gas dynamics, heat mass transfer, aerodynamic wind tunnel, virtual device

Acknowledgements: This work was supported by the Russian Foundation for Basic Research, project nos. 15-08-08428.

For citation:

Strongin M. M., Vinogradov Yu. A., Zditovets A. G., Kiselev N. A., Popovich S. S. Applying of National Instruments Technologies in Experimental Research of Thermal Gas Dynamics Processes, *Programmnyaya Ingeneriya*, 2017, vol. 8, no. 5, pp. 230–240.

DOI: 10.17587/prin.8.230-240

References

1. **Leontiev A. I., Kiselev N. A., Zditovets A. G., Burtcev S. A., Strongin M. M., Vinogradov Yu. A.** Experimental investigation of heat transfer and drag on surfaces with spherical dimples, *Exp. Temp. Fluid Sci.*, 2016, vol. 79, pp. 74–84.
2. **Kutateldze S. S., Leontiev A. I.** *Teploobmen i trenie v turbulentnom pogranchnom sloe* (Heat exchange and friction in turbulent boundary layer), Moscow, Energiya, 1985, 320 p. (in Russian).
3. **Volchkov E. P.** *Pristennyye gazovyye zavesy* (Near-wall gas screens) Novosibirsk Nayka, 1983, 240 p. (in Russian).
4. **Zditovets A. G., Vinogradov Yu. A., Strongin M. M., Titov A. A., Kiselev N. A.** *Bez mashinnoe energorazdelenie gazovykh potokov* (Machineless energy separation of gas flows), Moscow, Kyrs, 2016, 110 p. (in Russian).
5. **Strongin M. M.** *Sozdanie i razvitiye avtomatizirovannykh sistem fizicheskogo modelirovaniya v mehanike. Paket programm dlja provedeniya aerodinamicheskogo eksperimenta* (Creation and development of automatized systems of physical modelling in mechanics. Programs pack for running aerodynamic experiment), Moscow, NII mehaniki MGU, Otchet no. 4304, 1993, 36 p. (in Russian).
6. **Vinogradov Yu. A., Strongin M. M.** *Paket prikladnykh programm dlja provedeniya aerodinamicheskogo eksperimenta (dopolnenie dlja resheniya zadach effektivnosti "poristogo ohlajdeniya")* (Applied programs pack for carrying out aerodynamic experiment (supplement for solution the problem of "porous" cooling efficiency)), Moscow, NII mehaniki MGU, Otchet no. 4404, 1995, 68 p. (in Russian).
7. **Travis J.** *LabVIEW for Everyone*, Saddle River NJ, Printice Hall, 2002, 589 p.

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4
Технический редактор *Е. М. Патрушева*. Корректор *И. Е. Назарова*

Сдано в набор 09.03.2017 г. Подписано в печать 00.04.2017 г. Формат 60×88 1/8. Заказ П1517
Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru