

УДК 004.9,519.7

А. М. НЕДЗЬВЕДЬ, С. В. АБЛАМЕЙКО

## ИНТЕЛЛЕКТУАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АНАЛИЗА ИЗОБРАЖЕНИЙ, АДАПТИРУЕМОЕ К ЗАДАЧАМ МЕДИЦИНСКОЙ ДИАГНОСТИКИ

Modern software development is based on the methods of its intellectualization. Software should solve not only well-posed problems, and determine the best ways to solve a variety of problems. Therefore it is need using special software with intelligent component responsible for the behavior and evolution of the program in different situations. Technology for creating software of medical images analysis and processing based on the using of tables interpreter with an intelligent agent, which is equipped with appropriate properties of communicating tags and functions. Such software can modify by yourself or with a simple user, without the intervention of specialists. The data table for each commands shell corresponds to estimates set, which are calculated on the basis of commands performance. Comparing table interpreter with the calculated estimates allow to intelligent agent makes a decision about formation of a new script. This script consists of image processing and analysis commands. Now, such an organization of software is unique and can to solve many problems of image processing and analysis. If intelligent agent has work mistake, developers or users can change interpreter script and influence to work of software. This technology has been successfully tested for analysis of histological images, and will be adapted to other problems that related to image processing. It is developing as pilot software QTIP, which is used for the diagnosis of disease on base histological images, and will be adapted for nanoscale imaging, aerial photography and satellite images.

*Ключевые слова:* программное обеспечение, анализ изображения, интеллектуальные методы.

*Key words:* software, image analysis, intellectual methods.

В настоящее время разработано много методов и технологий получения изображений, имеющих свою специфику. Однако рост числа таких технологий вызвал дефицит методов обработки и анализа получаемых изображений. Современная технология анализа изображений представляет собой разрозненные наборы функций и команд, которые требуют одновременного участия по крайней мере двух специалистов: один освещает специфику области исследования, другой отвечает за обработку изображений. В результате процесс анализа изображений представляет собой долгую сложную процедуру. Безусловно, существуют программные продукты для специфических изображений, однако все они характеризуются ограниченным функционалом. Как правило, это связано со стремлением разработчиков защитить свое программное обеспечение от непрофессиональных действий пользователя. В то же время, если в программном продукте присутствуют сложные параметрические функции, он сразу теряет свою привлекательность.

Особенно сложная ситуация характерна для рынка медицинского программного обеспечения, где у каждого врача свои индивидуальные требования и оценки [1]. Другой стороной данной проблемы является разнообразие объектов на изображении. Использование вычислительной техники и математических методов в этой отрасли позволяет не только ускорить процесс обработки материала, но и повысить точность исследования.

В данной работе предлагается схема создания интеллектуального программного обеспечения, способного самостоятельно адаптироваться к решению частных задач диагностики и мониторинга по результатам анализа изображений. Схема основана на объединении возможностей динамических библиотек и интерпретатора с множеством функций для обработки изображений. В результате программа может быть разделена на две части: первая ориентирована на профессиональных разработчи-

ков программного обеспечения, вторая – на работу его пользователей. Интерпретатор имеет возможность привлекать дополнительные функции динамических библиотек. Это позволяет корректировать свойства программного обеспечения без стадии компиляции. Пользователи со своей стороны могут изменять графический интерфейс для улучшения комфортных условий рабочего места.

### Анализ изображений

Процесс анализа изображения состоит из выполнения набора изменяющих их команд. Как правило, его можно разделить на несколько этапов: ввод и улучшение изображений, сегментация, определение объектов (идентификация), измерение объектов и вычисление их характеристик, анализ и принятие решений [2].

Каждый шаг включает разные последовательности функций. Их применение зависит от свойств изображения или его оценок (контраста, шума или размытости), которые можно определить в большинстве случаев. Мы можем построить таблицу функций обработки изображений и оценок свойств изображений.

Искомые объекты определяются в соответствии с задачами, которые предстоит решить. Оценка особенностей изображения позволяет определить функции, необходимые для его обработки. В результате строится таблица связей функций и оценок характеристик изображения.

На каждом шаге для функций определены приоритеты их выполнения. Так, для улучшения изображения удаление импульсных шумов, следующий уровень приоритета будут иметь функции повышения контраста и коррекции границы объекта. Приоритеты определяют порядок выполнения функций и необходимость дополнительного анализа особенностей изображения.

Функции обработки изображения относятся к основным разделам компьютерного зрения. Это соответствует их использованию для изменения изображения. Каждая функция меняет свойства изображения и применяется для конкретных случаев обработки. Свойства и особенности функции должны описываться в глобальной таблице интерпретатора и сопровождаться дополнительной информацией.

Любое программное обеспечение может быть разделено на скомпилированные программы и интерпретаторы. Иногда программа представлена в виде комбинации этих вариантов. Интерпретатор позволяет создавать дополнительные простые функции на базе возможностей ядра программного обеспечения, не меняя его [3]. Мы предлагаем воспользоваться этими возможностями интерпретатора для создания программного обеспечения. В этом случае интерпретатор работает как менеджер действий и событий. Он поддерживает события и выполнение функций пользовательского интерфейса. В таком случае можно изменить дизайн программного обеспечения и без этапа компиляции. В то же время сложные функции и расчеты выполняются во внешних статически скомпилированных модулях. Это сохраняет скорость вычислений на том же уровне, что и в скомпилированном программном обеспечении [4].

Один из наиболее важных вопросов при разработке программного обеспечения – создание архитектуры необходимых для анализа и обработки изображений [5] данных, которые определяются в виде семантических и цифровых признаков.

Из них состоит таблица, являющаяся основой для команд интерпретатора (рис. 1), позволяя формировать их исполняемые последовательности. Для увеличения производительности предлагается использовать реализацию основных команд в динамических библиотеках. Интерпретатор в данной схеме служит для объединения работы команд по обработке изображений и интеллектуального агента. Интеллектуальный агент разделен на две части: первая формирует оценку изображения, вторая по этой оценке и таблице данных из интерпретатора – последовательность команд для исполнения интерпретатором.

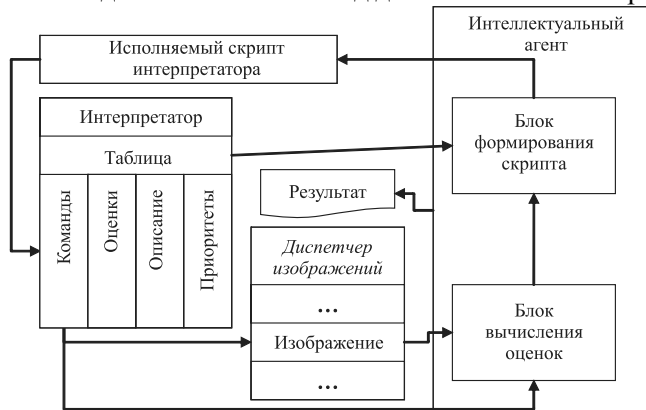


Рис. 1. Схема интеллектуального программного обеспечения для анализа и обработки изображений

В таблице данных каждой команде интерпретатора соответствует набор оценок, которые вычисляются на основе характеристик. Сравнивая таблицы интерпретатора с полученными оценками изображения, интеллектуальный агент принимает решение о формировании нового скрипта, состоящего из команд обработки и анализа изображений для последующей модификации.

В результате интеллектуальное программное обеспечение формируется на основе следующих компонентов: таблицы данных, интеллектуального агента, интерпретатора, динамических библиотек с командами.

### Интеллектуальный агент

Формирование последовательности команд для анализа изображений требует участия интеллектуальной составляющей в любой системе. В области построения экспертных систем и управления знаниями данная составляющая называется интеллектуальным агентом и отвечает за организацию изменения информации, в нашем случае – за преобразование системы значений характеристик в последовательности команд. Данный этап имеет огромное значение, так как именно он обеспечивает возможность самосовершенствования программы.

С целью определения функции обработки изображений в сценарии мы пытались использовать нейронные сети Кохонена [7], основным элементом которых является слой Кохонена, состоящий из адаптивных линейных сумматоров. Оценки глобальных условий изображения используются в качестве весов в нейронной сети. Регулировка входных весов и вектор квантования сигнала сети тесно связаны с алгоритмом кластерного анализа (например, методом динамических ядер или К-средних). Подобная организация сети позволяет построить очень гибкую систему принятия решений. Однако основное затруднение, с которым приходится сталкиваться разработчику, – процесс обучения, поскольку он очень затратный по времени и ресурсам, а неполное обучение приводит к ошибкам и дополнительным сложностям.

Оценки изображения и характеристик объектов образуют таблицу значений, которая постоянно присутствует в программе. Условия частных задач формируют диапазоны для этих значений, в соответствии с ними составляются наборы правил для принятия решений интеллектуальным агентом. Процесс обучения похож на процесс накопления знаний с постоянными диапазонами значений для принятия решения. При этом вариантов построения прямых решений намного больше, чем обратных.

Таким образом, для упрощения механизма работы интеллектуального агента можно применять вероятностную модель, используя таблицы данных, а в них – обратную цепочку в рассуждениях, вместо весовых коэффициентов – условную вероятность, которая определяется по теореме Байеса:

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)},$$

где  $P(A)$  – априорная вероятность гипотезы  $A$ ,  $P(A|B)$  – вероятность гипотезы  $A$  при наступлении события  $B$  (апостериорная вероятность),  $P(B|A)$  – вероятность наступления события  $B$  при истинности гипотезы  $A$ ,  $P(B)$  – полная вероятность наступления события  $B$ .

Следовательно, можно рассчитать вероятность использования команды обработки изображения на основе оценок изображения и его объектов.

### Интеллектуальное программное обеспечение анализа изображений

Для разработки структурной схемы программного интерфейса и архитектуры была выполнена оценка функциональности и совместимости имеющихся в распоряжении инструментов разработки программного обеспечения.

По материалам, размещенным Гийомом Марсо [3], который в своем исследовании использовал 72 параметра для оценки реализаций языков программирования и протестировал их 19 специальными тестами, в качестве ядра выбран интерпретатор Lua [8].

Наша система использует язык Lua в качестве основного модуля и обеспечивает взаимодействие сложных компонентов. В ядро интерпретатора дополнительно включены графический интерфейс, глобальные переменные и структуры изображения. Архитектура графического интерфейса объединяет библиотеки Highgui [9] из пакета OpenCV [9] и библиотеку Qt. Функции обработки и анализа изображений поддерживаются благодаря связи OpenCV с Lua, которые реализуются в отдельной библиотеке (рис. 2).

Структура изображения определяется в модуле графического интерфейса на основе библиотеки OpenCV, которая отвечает за визуализацию и представления изображений. Заголовки этой структуры выступают как глобальные указатели интерпретатора Lua и имеют специальный тип Userdata для пользовательских данных, который соответствует указателям адресного пространства компьютера.

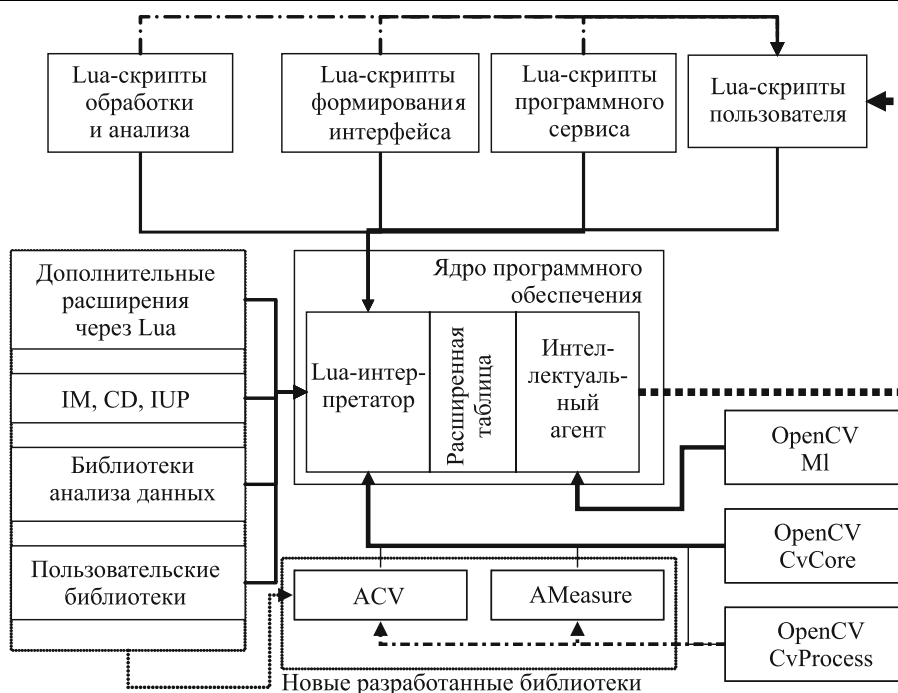


Рис. 2. Схема реализации интеллектуального программного обеспечения для анализа и обработки изображений

Модуль генерации скрипта работает на основе механизма интеллектуального агента и определяет новые возможности программы в виде набора простых функций библиотек. Он состоит из двух частей: оценки-анализа изображений и непосредственно генерации скриптов. В соответствии с такими оценками изображения формируются необходимые последовательности функций для их обработки. После сопоставления оценок с расширенной глобальной таблицей интерпретатора определяется сценарий предварительной обработки изображения. Пользователи могут анализировать эти сценарии и вносить в них изменения.

**Самостоятельная адаптация программного обеспечения для анализа медицинских изображений**

Требования к программному обеспечению, предназначенному для работы медицинского персонала, как правило, очень высокие. Для организации рабочего места каждого врача требуются уникальные интерфейсы. Технология интеллектуального программного обеспечения позволяет автоматически настраивать интерфейс в зависимости от решаемой задачи. Как известно, существует несколько типов медицинских изображений, поэтому интерфейсы программ для гистологического анализа и компьютерной томографии отличаются кардинально. Однако интеллектуальный агент в ядре программного обеспечения способен распознать тип изображения.

Предварительный анализ изображений позволяет определить функции обработки. В результате задача может быть решена только для конкретного случая, в частности для анализа изображений гистологии и компьютерной томографии (рис. 3).

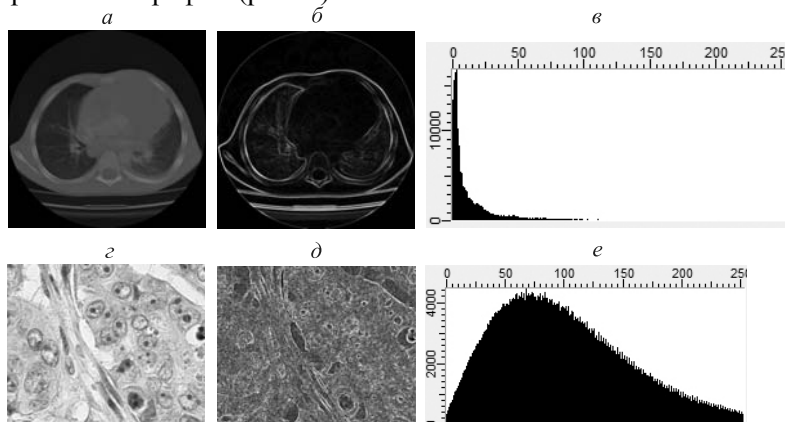


Рис. 3. Примеры результатов этапа классификации изображений:

*а* – томографическое изображение, *б* – градиент томографического изображения, *в* – гистограмма градиента томографического изображения, *г* – гистологическое изображение, *д* – градиент гистологического изображения, *е* – гистограмма градиента гистологического изображения

Первый шаг анализа заключается в определении глобальных характеристик изображения и типа изображения. Одноканальному полутоновому изображению часто соответствуют радиологические методы исследования в медицине, если глубина пикселя яркости, превышающая восемь бит, сопоставима с томографическими изображениями. Цветное изображение состоит из трех каналов и более и определяется как гистологическое. Остается неясным класс в восьмибитных полутоновых изображениях, к которым можно отнести частично обработанные изображения гистологии и радиологической медицины. Разница между полутоновыми радиологическими и гистологическими изображениями чаще всего заключается в способе формирования объектов изображения. Цвет в гистологическом изображении обусловлен окрашиванием посредством аминокислот. В результате объект на изображении представляет собой небольшое пятно, которое образует дополнительные локальные границы. Границам соответствуют локальные экстремумы яркости изображения. Таким образом, фильтр выделения границ (например, фильтр Собеля) выделяет эти границы на изображении (см. рис. 3). Таким образом, томографическому изображению свойствен резкий перепад градиента изображения, отраженный на гистограмме. Распределение яркости на гистологическом изображении похоже на распределение Гаусса.

На основе характеристик асимметрии и эксцентриситета гистограммы определяется тип изображения и его принадлежность гистологическому или томографическому классу. Интеллектуальный агент генерирует отдельный пользовательский интерфейс в виде скрипта интерпретатора. В результате формируется уникальный интерфейс для гистологического (рис. 4 а) или томографического изображения (рис. 4 б).

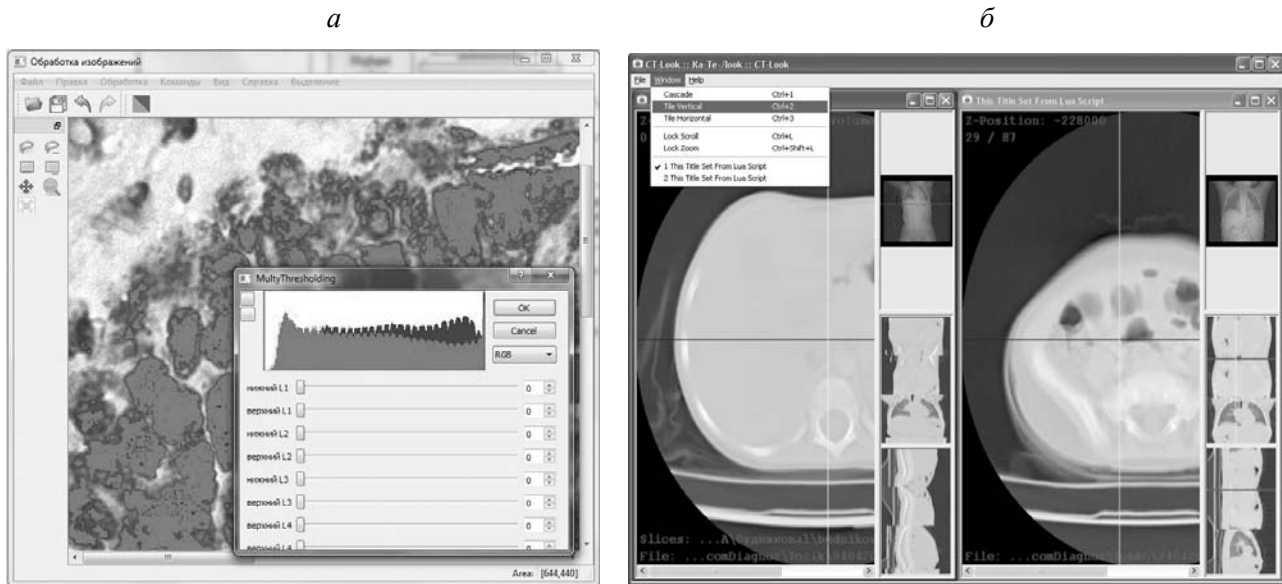


Рис. 4. Пользовательский интерфейс для обработки изображения:  
а – для гистологических задач, б – для задач компьютерной томографии

Тестирование работы интеллектуального программного обеспечения проводилось на 248 медицинских изображениях разного типа (таблица).

**Результаты проверки программного обеспечения на медицинских изображениях**

Тип изображений	Вероятность	
	определения типа изображений, %	успешной обработки изображений, %
Гистологические с высокой плотностью клеток	98	83
Гистологические со средней плотностью клеток	99	90
Гистологические с низкой плотностью клеток	87	90
Компьютерной томографии	100	93
Метода магнитоядерного резонанса	100	95

Тестирование выявило хорошую подготовку скриптов пользовательского интерфейса. Однако во время их генерации остается вероятность ошибки, поэтому разработчикам следует предусматривать возможность ее коррекции.

Использование встроенных таблиц интерпретатора совместно с интеллектуальным агентом, снабженным соответствующим механизмом связей дескрипторов и функций в этой схеме программного обеспечения, позволяет программному обеспечению модифицироваться самостоятельно или при помощи простого пользователя без вмешательства специалистов.

Данная технология была успешно применена для анализа гистологических изображений. В дальнейшем она будет адаптироваться к другим задачам, связанным с обработкой изображений. Развитие данной технологии будет происходить по сценарию: от построения концепции до формирования интерфейса пользователя и алгоритма решения задачи.

В настоящее время представленная технология реализована в экспериментальном программном комплексе QTIP, который использовался в качестве дополнительных средств для диагностики заболевания по гистологическим изображениям. В дальнейшем комплекс будет адаптироваться для наноскопических изображений и изображений спутниковой или аэросъемки земной поверхности. Для достижения этой цели необходимо сформировать новые дескриптивные признаки, которые будут связаны с функциями обработки изображений. Кроме того, общая концепция построения такого программного обеспечения, безусловно, проявит себя в ближайшей перспективе при построении сложных программных комплексов.

1. Гулиева И. Ф., Рюмина Е. В., Гулиев Я. И. // Врач и информ. технологии. 2009. № 3. С. 4.
2. Абламейко С. В., Недзьведь А. М. Обработка оптических изображений клеточных структур в медицине. Минск, 2005.
3. Marceau G. The speed, size and dependability of programming languages // Blog Square root of x divided by zero, Saturday, May 30, 2009. Mode of access: <http://gmarceau.qc.ca/blog/2009/05/speed-size-and-dependability-of.html>.
4. Reitmaier G., Schmalstieg D. // Proc. of VRST'01, November 15–17, 2001. Banff, Alberta. P. 47.
5. Цыбульник В. А. // Рос. журн. программистов RSDN. 2010. № 2. С. 39.
6. Kaski S. // Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series. Espoo, 1997. № 82.
7. Васильев И. А. // Искусств. интеллект. 2003. № 3. С. 48.
8. Ierusalimsky R. Programming in Lua 2<sup>nd</sup> ed. Lua. org. 2006.
9. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, 2008.

Поступила в редакцию 27.08.12.

*Александр Михайлович Недзьведь* – кандидат технических наук, ведущий научный сотрудник Объединенного института проблем информатики НАН Беларуси.

*Сергей Владимирович Абламейко* – академик НАН Беларуси, доктор технических наук, профессор, ректор БГУ.

УДК 681.3.06+347.78.031

Г. А. ЛОМАКИН, А. С. ГУСЕЙНОВА

## ОБЩИЕ ПОДХОДЫ К РАЗРАБОТКЕ ИНТЕГРИРОВАННОЙ СРЕДЫ ДЛЯ ПОСТРОЕНИЯ ГРАФИЧЕСКИХ ПРИЛОЖЕНИЙ

The paper describes the main approaches for creating interactive web environment for building 3D-applications. The modern development of graphics systems and technologies is urgent transferring of imaging to the next level, which greatly simplifies the development of 3D-objects. Scientific novelty of the approach is to create an integrated system for the implementation of graphics applications that abstracts the end user from the intermediate levels graphics creation and provides the ability to directly construct graphic scenes through a graphical interface on modern platforms such as Windows 7 and Android. The main concept of the proposed solution involves the creation of a set of classes and utilities that implement a common approach to graphics rendering and hide by the direct use of various mathematical models and graphics algorithms. This will allow a wider range of stakeholders to create the desired 3D-objects and scenes, spending minimal time to develop the necessary solutions.

*Ключевые слова:* 3D-приложение, графическое ядро, фреймворк, контент, Android, графический интерфейс пользователя.

*Key words:* 3D-application, graphic kernel, framework, content, Android, graphic interface of the user.

Компьютерная графика достаточно широко используется для решения задач визуализации как объектов реального мира, так и несуществующих изображений различных объектов. Значительный научный и практический интерес представляют также задачи создания виртуальных реальностей, которые отождествляют существующие прототипы, например, визуализации различных памятников архитектуры и археологии. Отметим также, что при современном развитии графических систем и технологий актуален подход, связанный с переводом визуализации 3D-объектов на новый уровень, позволяющий упростить разработку 3D-графики. Таким образом, реализация обобщенной среды (графического ядра) для визуализации различных объектов, которая предоставит пользователю возможность быстрого создания 3D-объектов, используя современные интернет- и мобильные технологии, является востребованным направлением в развитии компьютерной графики.