Материалы IX-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2016). - Том II. - М.: ИПУ РАН, 2016. - С.249-253. - ISBN 978-5-91450-185-0.

ЯЗЫК IMAGESCRIPT ДЛЯ КРОССПЛАТФОРМЕННОГО АНАЛИЗА, ОБРАБОТКИ И РАСПОЗНАВАНИЯ СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Выхованец В.С., Фозилов М.М.
Институт проблем управления им. В.А. Трапезникова РАН, г. Москва,
МГТУ им. Н.Э. Баумана, г. Москва
valery@vykhovanets.ru, mukim.fozilov@mail.ru

Ключевые слова: обработка сигналов и изображений, распознавание образов, язык ImageScript, кроссплатформенность.

Keywords: signal and image processing, pattern recognition, ImageScript language, cross-platform.

Введение

Задачи анализа и обработки сигналов и изображений [1, 2], задачи распознавания образов [3] являются актуальными и востребованными в различных областях науки и техники, например, в реализации человеко-машинных интерфейсов, в принятии решений роботами, в системах автоматического контроля, при управлении процессами, для обнаружения и сопровождения объектов.

Среди трудностей, возникающих при реализации таких задач, – плохая переносимость и масштабируемость программных средств анализа и обработки сигналов и изображений, недостаточная производительность используемых вычислительных устройств, и ряд других. Например, на сегодняшний день существует различные методы обнаружения объектов на изображениях [4]. Однако все эти методы требуют значительных временных и вычислительных ресурсов для их переноса с одной аппаратурной платформы на другую, при адаптации реализующих их алгоритмов в новой операционной среде.

Целью данной статьи является описание языка ImageScript, предназначенного для эффективной кроссплатформенной реализации алгоритмов анализа, обработки и распознавания сигналов и изображений. Предлагается использование интерпретируемого языка ImageScript, реализуемого на виртуальных и аппаратурных платформах со стековой организацией вычислений. Применение языка ImageScript позволит создавать аппаратурно-независимые и эффективные программы, предназначенные для анализа, обработки и распознавания сигналов и изображений.

1. Технология PostScript

Важным элементом программных средств электронной полиграфии являются RIP-программы (англ. Raster Image Processing), обеспечивающие вывод текстовых документов и графических изображений на различные устройства с высокой разрешающей способностью. Основной вклад в подготовку электронных изданий, не зависящих от аппаратурной платформы, внесла компания Adobe System Incorporated. Главным изобретением компании явился язык PostScript, на базе которого был развит универсальный формат представления документов PDF (англ. Portable Documents Format) [5].

PostScript – язык описания документов, реализует принцип WYSIWYG (англ. What You See Is What You Get). В языке используется обратная польская нотация: операнды записываются перед оператором. Интерпретатор языка записывает операнды в стек, потом выполняется оператор, снимая операнды со стека. Обратная польская нотация считается эффективной, но трудно читаемой. Однако при автоматической генерации описания документов эта трудность не существенна.

Материалы IX-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2016). - Том II. - М.: ИПУ РАН, 2016. - С.249-253. - ISBN 978-5-91450-185-0.

Язык PostScript может рассматриваться как универсальный язык программирования с мощными встроенными графическими возможностями, обеспечивающий аппаратурно-независимый формат описания документов.

Язык имеет статические или динамические средства. Статические средства обеспечивают определенный встроенный набор операторов и функций. Динамические средства используются для расширения языка. Встроенные или ранее определенные операторы языка можно в случае необходимости перекрывать, изменяя поведение программы.

Структурированное описание PostScript состоит из двух частей: пролога и сценария. Пролог – набор описаний процедур, которые могут использоваться при выполнении сценария. Главной задачей пролога является описание основного словаря, используемого при интерпретации сценария. Сценарий – создается для описания отдельных элементов документа. Он состоит из операторов, вызова функций и данных.

Существенными достоинствами технологии PostScript является машинно-независимый язык описания документов и возможность автоматической генерации описания документов с помощью сторонних программ.

2. Технология ImageScript

Технология ImageScript строится на одноименном интерпретируемом языке, который имеет много общего с языком PostScript. В отличие от PostScript технология ImageScript предусматривает встроенные графические средства, необходимые для эффективного анализа, обработки и распознавания изображений:

– предварительная обработка и сегментация сигналов и изображений;
– реализация зарекомендовавших себя способов распознавания образов;
– загрузка и сохранение сигналов и изображение в распространенных форматах.

Примерами встроенных команд, используемыми при анализе, обработке и распознавании изображений, являются [6]:

– двумерное прямое дискретное косинусное преобразование icos;
– обратное двумерное дискретное косинусное преобразование idicos;
– установка размера окна обработки isize;
– масштабирование изображений iscale;
– построение гистограммы ihistogram;
– определение порога изображения ithreshold;
– свертка изображений icon;
– двумерная линейная фильтрация ifilter;
– ранговая фильтрация iorder;
– медианная фильтрация imed;
– сегментация методом разделения iqtdecomp;
– выделение границ iedge;
– алгоритм маркерного водораздела iwatershed;
– двумерное преобразование Фурье ifft;
– обратное двумерное преобразование Фурье iifft;
– изменение размеров изображений irsize;
– поворот изображения irotate;
– конвертация изображений в формате RGB в формат HCV ihcv;
– конвертация изображений в формате HCV в формат RGB irgb;
– конвертирование из RGB в YCbCr iycbcr.

Для цифровой обработки сигналов предусматривается использование следующих встроенных команд [7]:

– преобразование вектора в формате с плавающей запятой в формат Q31 sfltoq;
– преобразование вектора в формате Q31 в формат с плавающей запятой sqtofl;

– обратные значения элементов вектора srecip;
– квадратный корень вектора ssqr;
– возведение вектора в натуральную степень sexpn;
– возведение вектора в квадрат spower;
– минимальный порядок элементов вектора sbexp;
– вычисление синусов элементов вектора ssine;
– натуральный логарифм элементов вектора slogn;
– вычисление псевдослучайного вектора srand и srandinit;
– вычисление бит-реверсивных индексов sbrev;
– корреляционная функция scorr;
– автокорреляционная функция sacorr;
– свертка двух векторов sconvol;
– нерекурсивная фильтрация sfir;
– комплексная нерекурсивная фильтрация scfir;
– интерполяционная нерекурсивная фильтрация sfirinterp;
– прямая решетчатая нерекурсивная фильтрация sfirlat;
– симметричная нерекурсивная фильтрация sfirs;
– децимационная нерекурсивная фильтрация sfirdec;
– адаптивная среднеквадратическая фильтрация sdlms;
– преобразование Гильберта shilb;
– рекурсивная фильтрация siir;
– каскадная рекурсивная фильтрация с 4 коэффициентами siircas4;
– каскадная рекурсивная фильтрация с 5 коэффициентами siircas5;
– инверсная решетчата рекурсивная фильтрация siirlat;
– прямое действительное преобразование Фурье srfft;
– обратное действительное преобразование Фурье srifft;
– прямое комплексное быстрое преобразование Фурье scfft;
– обратное комплексное быстрое преобразование Фурье scifft.

3. Язык ImageScript

Как и его аналог язык ImageScript содержит стандартное множество встроенных типов данных, таких как числа, строки, массивы, а также основные элементы управления, такие как условия, циклы, блоки команд. Данные представляются виде объектов. Простые объекты хранятся в стеке, а сложные – в виртуальной памяти, указатели на которые хранятся в стеке. Интерпретатор ImageScript управляет следующими стеками:

– стек операндов, который содержит объекты или их указатели, являющиеся операндами операторов и результатами выполнения операторов;

– стек исполнения, который содержит указатели на точки возврата из процедур, находящиеся в стадии выполнения;

– стек контекста, который хранит настроенные определенным образом объекты, задающие текущее состояние программы.

Синтаксис языка ImageScript задается следующей формальной грамматикой, где правила вывода пронумерованы, определяемые нетерминальные символы грамматики записаны в левой части правил вывода до символа →, терминальные символы заключены в апострофы, альтернативы в правой части правил вывода перечислены через разделитель |, а необязательные вхождения символов заключены в квадратные скобки.

```
1   Program   → Operator [Program]
2   Operator  → Procedure | Command | Literal | Comment
3   Procedure → '{' Program '}'
```

Материалы IX-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2016). - Том II. - М.: ИПУ РАН, 2016. - С.249-253. - ISBN 978-5-91450-185-0.

```
4  Command    → Name | Stack | Convert | Bitwise | Shift | Logic | Arithmetic |
                | Relation | Math | Streams | Strings | Array | Vector | Matrix
                | Dictionary | Image | Signal | Control | Context | File
5  Stack      → 'drop' | 'dup' | 'swap' | 'roll' | 'get' | 'put' |
                | 'clear' | 'depth'
6  Convert    → 'cvb' | 'cvi' | 'cvfi' | 'cvfl' | 'cvsm' | 'cvst' | 'cva'
7  Bitwise    → 'btst' | 'bclr' | 'bset' | 'bnot' | 'bcnt'
8  Shift      → 'asl' | 'asr' | 'lsl' | 'lsr' | 'rol' | 'ror'
9  Logic      → 'not' | 'and' | 'or' | 'xor'
10 Arithmetic→ 'neg' | 'add' | 'sub' | 'mul' | 'div' | 'mod' | 'abs' |
                | 'round' | 'floor' | 'ceil'
11 Relation   → 'eq' | 'ne' | 'ge' | 'gt' | 'le' | 'lt'
12 Math       → 'sqrt' | 'cos' | 'sin' | 'atan' | 'exp' | 'log' |
                | 'pow' | 'rand'
13 Streams    → 'stream' | 'size' | 'extract' | 'insert' | 'find'
14 Strings    → 'string' | 'length' | 'substring' | 'replace' | 'search' |
                | 'lower' | 'upper'
15 Array      → 'array' | 'asize' | 'aload' | 'astore' | 'aindex' | 'afill'|
                | 'aresize' | 'subarray' | 'aremove'
16 Vector     → 'vector' | 'vsize' | 'vload' | 'vstore' | 'vindex' | 'vfill'|
                | 'vresize' | 'subvector' | 'vremove'
17 Matrix     → 'matrix' | 'msize' | 'mload' | 'mstore' | 'mindex' | 'mfill'|
                | 'mresize' | 'submatrix' | 'mremove' | mtranslate | 'mident' |
                | 'mscale' | 'mrotate' | 'mtransform' | 'minvert'
18 Dictionary→ 'dict' | 'def' | 'undef' | 'count' | 'exists' | 'purge'
19 Image      → 'icos' | 'idicos' | … | 'iycbcr'
20 Signal     → 'sfltoq' | 'sqtofl' | … | 'scifft'
21 Control    → 'proc' | 'exec' | 'if' | 'elseif' | 'for' | 'repeat' |
                | 'loop' | 'exit' | 'bind' | 'nop'
22 Context    → 'save' | 'restore' | 'init' | 'current' | 'set' | 'let'
23 File       → 'file' | 'aopen' | 'close' | 'status' | 'seek' | 'position' |
                | 'read' | 'write' | 'flush' | 'run'| 'rename' | 'delete'
24 Literal    → 'null' | Boolean | Integer | Fixed | Float | Stream | String
25 Boolean    → 'true' | 'false'
26 Integer    → [Sign] Number
27 Number     → Digit [Number]
28 Fixed      → Integer ['.' Number]
29 Float      → Fixed ['e' Integer]
30 Stream     → '[' Binary ']'
31 Binary     → Hex [Binary]
32 String     → '(' [Symbols] ')'
33 Comment    → '<' Symbols '>'
34 Symbols    → Symbol [Symbols]
35 Symbol     → Sign | Digit | Letter | Others
36 Name       → Symbol [Name]
37 Digit      → '0' | '1' | ... | '9'
38 Hex        → Digit | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'
39 Sign       → '+' | '-'
40 Letter     → '_' | 'a' | 'b' |... | 'z' | 'A' | 'B' |... | 'Z' |
                | 'а' | 'б' |... | 'я' | 'А' | 'Б' |... | 'Я'
```

Материалы IX-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2016). - Том II. - М.: ИПУ РАН, 2016. - С.249-253. - ISBN 978-5-91450-185-0.

```
41  Others     → '`' | '!' | '@' | … | '/'
```

Программы на языке ImageScript состоит из операторов (правило 1), в качестве которых выступают процедуры, литералы, команды и комментарии (правило 2).

Процедуры являются последовательность операторов, заключенных в фигурные скобки (правила 2, 3). Процедуры сохраняются в словарях и могут быть вызваны по имени (правило 4), которое им присваивается при сохранении (правило 18). Именем процедуры может быть любая последовательность символов (правило 36), что позволяет переопределять как ранее сохраненные процедуры, так и литералы, встроенные команды. Если имя процедуры не задано, то ее тело рассматривается как блок операторов, подлежащих условному или циклическому выполнению (правило 21).

Множество команд интерпретатора разделяются на следующие подмножества (правило 4):

– команды вызова определенных ранее процедур по их именам (правило 4);
– стековые команды Stack (правило 5);
– команды преобразования типов Convert (правило 6);
– поразрядные логические команды Bitwise (правило 7);
– команды сдвигов Shift (правило 8);
– логические команды Logic (правило 9);
– арифметические команды Arithmetic (правило 10);
– команды отношений Relation (правило 11);
– команды математических функций Math (правило 12);
– команды работы с битовыми потоками Streams (правило 13);
– строковые команды Strings (правило 14);
– команды работы с массивами Array разнотипных элементов (правило 15);
– команды работы с векторами Vector однотипных элементов (правило 16);
– команды работы с матрицами Matrix (правило 17);
– команды работы со словарями Dictionary (правило 18);
– команды анализа и обработки изображений Image (правило 19);
– команды управления потоком команд Control (правило 21);
– команды управления контекстом Context (правило 22);
– команды файлового доступа File (правило 23).

Комментарии служат для документирования программ, состоят из последовательности символов, заключенных в угловые скобки (правила 33, 34).

В качестве литералов язык допускает (правило 24):

– неинициализированное значение null;
– Boolean – логические константы false и true (правило 25);
– Integer – целые числа (правила 26, 27);
– Fixed – числа с фиксированной запятой (правило 28);
– Float – числа с плавающей запятой (правило 29);
– Stream – битовые потоки (правила 30, 31);
– String – строки (правило 32).

Используемые подмножества символов определены в конце описания грамматики (правила 35-41).

Заключение

В настоящей статье приведено описание технологии и языка ImageScript для обработки, анализа и распознавания сигналов и изображений, и его сравнение с языком противоположного назначения – языком генерации изображений PostScript. Основными достоинствами языка ImageScript является кроссплатформенность и эффективность.

Материалы IX-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2016). - Том II. - М.: ИПУ РАН, 2016. - С.249-253. - ISBN 978-5-91450-185-0.

Для эффективной интерпретации программ конструкции языка имеют предельно простую синтаксическую структуру: программа на ImageScript – это последовательность команд-слов, разделенных пробелами. Такие программы не нуждаются в компиляции и выполняются интерпретатором последовательно в порядке чтения текста программы.

Встроенные в язык такие объекты как битовый поток, массив, вектор, матрица и словарь, а также многочисленные специализированные команды, позволяют компактно и высокоуровневыми средствами описывать задачи по анализу и обработке сигналов и изображений, распознаванию образов.

Язык ImageScript может также эффективно использоваться для анализа, обработки и распознавания сигналов, для чего в его состав введен тип данных с фиксированной запятой, распространенных при реализации алгоритмов цифровой обработки сигналов.

Интерпретируемая природа языка, помимо простоты синтаксиса, имеет ещё ряд преимуществ. Программа на языке ImageScript, обрабатывается последовательно и не имеет ограничений на размер, что очень важно для анализа и распознавания сложных образов.

На языке ImageScript можно создавать и включать в текст программы стандартные библиотеки, разработанные сторонними производителями, в том числе и использующие различные аппаратурные ускорители.

В качестве аппаратурных ускорителей могут применяться специализированные устройства, предназначенные для эффективной обработки сигналов, изображений, видеопотоков, также устройства для параллельных и конвейерных вычислений. Встраивание таких устройств в программу может осуществляться динамически. В случае отсутствия ускорителей или выхода их строя программа полностью сохраняет свою функциональность благодаря механизму переопределения команд языка.

ImageScript – адаптивный язык, позволяющий переопределять существующие команды или определять новые процедуры, не отличающиеся по синтаксису от стандартных команд. Механизм переопределения существующих процедур и стандартных команд позволяет исправлять ошибки, выявленные в загружаемых библиотеках и аппаратурных ускорителях.

Таким образом ImageScript – это не фиксированный язык, а скорее интерпретационная среда, подстраиваемая под конкретную задачу. ImageScript программы – это обычные текстовые файлы, которые можно обрабатывать стандартными текстовыми редакторами и относительно просто автоматически создавать с помощью других программ.

Литература

1. *Оппенгейм А., Шафер Р.* Цифровая обработка сигналов. 2-е издание. – М.: Техносфера, 2006. – 858 с.
2. *Яне Б.* Цифровая обработка изображений. – М.: Техносфера, 2007. – 584с.
3. *Фомин Я.А.* Распознавание образов: теория и применения. – 2-е изд. – М.: ФАЗИС, 2012. – 429 с.
4. *Гонсалес Р, Вудс Р.* Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
5. PostScript Language Reference. Adobe Systems Incorporated. – 3-rd edition. – Addison-Wesley Publishing. 1999. – 912 p.
6. TMS320C55x. Image/Video Processing Library. – Texas Instruments, 2004.
7. TMS320C55x. DSP Library. – Texas Instruments, 2013.