

N. S. Isupov, Post-graduate, Kalashnikov Izhevsk State Technical University

A. V. Kuchuganov, PhD in Engineering, Kalashnikov Izhevsk State Technical University

### Joined-up Writing Recognition with Fuzzy Logic Application

A brief survey of main recognition methods is represented in this paper. A new approach of joined-up writing recognition based on fuzzy graph presentation of vectorized text is proposed.

**Key words:** handwriting, fuzzy logic.

УДК 004.93

С. И. Зыкин, студент, Ижевский государственный технический университет имени М. Т. Калашникова

М. А. Сенилов, доктор технических наук, профессор, Ижевский государственный технический университет имени М. Т. Калашникова

## РЕАЛИЗАЦИЯ АЛГОРИТМА ВИОЛЫ – ДЖОНСА НА ОСНОВЕ ТЕХНОЛОГИИ *CUDA* ДЛЯ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ

Реализация алгоритма Виолы – Джонса для архитектуры *NVIDIA CUDA* позволяет в несколько раз ускорить его работу по сравнению с реализацией на центральном процессоре при распознавании объектов на изображениях.

**Ключевые слова:** алгоритм Виолы – Джонса, технология *CUDA*, обнаружение объектов, анализ изображений.

Задача распознавания объектов на изображениях является в настоящее время весьма актуальной. Она возникает при создании естественного человеко-машинного интерфейса, построении охранных систем и пр.

Существует множество методов обнаружения объектов на изображениях, таких, например, как методы с использованием контурных моделей, световой сегментации, нейронных сетей и др. Одним из наиболее известных алгоритмов нахождения объектов является детектор Виолы – Джонса.

Алгоритм Виолы – Джонса был представлен в 2001 г. Полом Виолой и Майклом Джонсом [1]. Этот детектор является одним из лучших по соотношению показателей «эффективность распознавания/скорость работы».

Для быстрого вычисления суммы пикселей вычисляется промежуточное представление изображения, которое называется интегральным изображением [2]. Значение элемента интегрального изображения с координатами  $x, y$  равняется сумме пикселей исходного изображения выше и левее координат  $x, y$  включительно:

$$IntImage(x, y) = \sum_{x' \leq x, y' \leq y} Image(x', y'), \quad (1)$$

где  $IntImage$  – интегральное изображение;  $Image$  – исходное изображение.

Детектор сканирует входное изображение на многих масштабах, начиная с базового значения. Процесс сканирования представляет собой процесс просмотра вырожденного дерева решений, которое называется каскадом (рис. 1) [3].

Отрицательный результат в любой точке каскада приводит к немедленному отказу от субокна. После

нескольких этапов обработки число субокон сокращается радикально, и оставшиеся окна являются решением поставленной задачи.

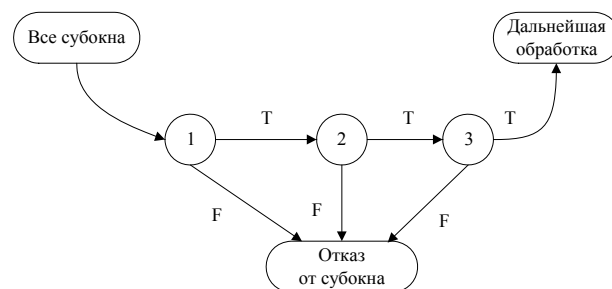


Рис. 1. Схематичное представление каскада обнаружения

Технология *CUDA* использует параллельную модель вычислений, когда каждый из *SIMD*-процессоров выполняет одну и ту же инструкцию над разными элементами данных параллельно [4].

Из-за параллельных вычислений трудно сразу вычислить интегральное изображение, так как нужно контролировать порядок вычислений. Поэтому вычисление разбито на две части.

1. Нахождение суммы по столбцам с использованием следующих формул:

$$\begin{aligned} b(x, -1) &= 0; \\ b(x, y) &= b(x, y-1) + Image(x, y), \end{aligned} \quad (2)$$

где  $b(x, y)$  – сумма значений по столбцам.

2. Вычисление интегрального изображения по полученным ранее значениям:

$$\begin{aligned} IntImage(-1, y) &= 0; \\ IntImage(x, y) &= IntImage(x-1, y) + b(x, y). \end{aligned} \quad (3)$$

Реализация *CUDA* позволяет обрабатывать все субокна параллельно, то есть одновременно выполняется один этап каскада [5]. Ядро запускается для каждого этапа с определенным размером окна и шагом, в которых каждый поток отвечает за классификацию субокна. Если субокно классифицируется как объект на текущем этапе, в потоке записывается

в выходной массив единица, в противном случае – ноль. Получаем одномерный массив, заполнен единицами и нулями, из которого потом и создается список с координатами областей найденных лиц.

Схема работы алгоритма Виолы – Джонса, реализованного с использованием графического ускорителя, представлена на рис. 2.

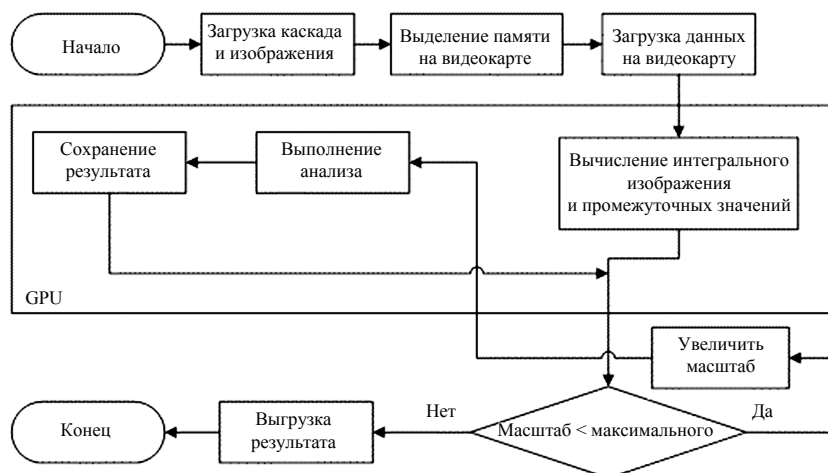


Рис. 2. Схема работы алгоритма Виолы – Джонса, реализованного с помощью *CUDA*

Описанный алгоритм был реализован в виде программы, осуществляющей анализ изображения. Для тестирования применялся каскад классификаторов, распознающий лица. После запуска программа автоматически осуществляет анализ изображения. Результат анализа наносится в виде рамок, помечающих найденные объекты, на копию изображения, как представлено на рис. 3.



Рис. 3. Пример работы программы

Тестирование проводилось с использованием видеокарты *NVIDIA GeForce 9300 GT* и процессора *Intel Core2 Quad CPU 2.40GHz*.

Время выполнения анализа изображения с расширением  $524 \times 367$  составило 0,79 с. Анализ такого

изображения с помощью ЦПУ составляет 1,20 с. Таким образом, технология *CUDA* ускорила анализ в 1,5 раза.

Как видно из результатов, алгоритм Виолы – Джонса эффективно реализуется для архитектуры *NVIDIA CUDA*. Однако следует отметить, что сам алгоритм не слишком приспособлен для реализации на графической карте, так как большое число обращений к памяти, простои потоков при работе каскада классификаторов негативно сказываются на производительности.

#### Библиографические ссылки

1. Зиновьев И. И., Овчинникова Т. Е., Шамин П. Ю. Реализация алгоритма Viola – Jones для архитектуры *NVIDIA CUDA*. – URL: [http://agora.guru.ru/hpc-h/files/011\\_Zinoviev\\_Viola\\_Jones.pdf](http://agora.guru.ru/hpc-h/files/011_Zinoviev_Viola_Jones.pdf) (дата обращения: 04.03.2012).
2. Макаров А. О., Старовойтов В. В. Быстрая обработка изображений на основе интегральных матриц изображений. – URL: [http://www.iai.dn.ua/public/JournalAI\\_2006\\_3/Razdel8/06\\_Makarov\\_Starovoytov.pdf](http://www.iai.dn.ua/public/JournalAI_2006_3/Razdel8/06_Makarov_Starovoytov.pdf) (дата обращения: 04.03.2012).
3. Viola P., Jones M. Robust Real-time Object Detection // Second International on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling. – Vancouver, Canada, July 13, 2001.
4. Борсков А. В., Харламов А. А. Основы работы с технологией *CUDA*. – М.: ДМК «Пресс», 2010.
5. Harvey J. P. GPU Acceleration of Object Classification Algorithms Using *NVIDIA CUDA* // Master's thesis, Rochester Institute of Technology, Rochester. – NY, Sept., 2009.

*The implementation of Viola-Jones algorithm for the NVIDIA CUDA architecture will allow to accelerate its performance a few times as compared to implementation of objects detection on images at the CPU.*

**Key words:** Viola-Jones, CUDA, object detection, image analysis.

УДК 519.853.3

**А. Г. Исавнин**, доктор физико-математических наук, профессор, Камская государственная инженерно-экономическая академия, Набережные Челны

**М. Р. Хамидуллин**, соискатель, Камская государственная инженерно-экономическая академия, Набережные Челны

## ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОБ ОПТИМАЛЬНОМ УПРАВЛЕНИИ ЗАПАСАМИ АЛГОРИТМАМИ МЕТОДА ШТРАФОВ

*Для задачи об оптимальном управлении запасами реализован программный комплекс на основе метода штрафных функций с неполной минимизацией вспомогательных функций. Приложение разработано в среде программирования Borland Delphi 7.0. Для удобства ввода исходной функции и ограничений использован синтаксический анализатор математических формул. Показано, что подобные алгоритмы могут найти свое практическое применение, например, в задаче об оптимальном управлении запасами.*

**Ключевые слова:** вспомогательная функция, выпуклое программирование, оптимальное управление, штрафная функция.

**З**адача отыскания минимума некоторой функции  $f(x)$  – одна из основных проблем теории оптимизации, которая эквивалентна задаче отыскания максимума той же функции, взятой с противоположным знаком.

Метод штрафных функций может использоваться для исключения части или всех ограничивающих уравнений. Он сводит задачу на условный экстремум к решению задач на безусловный экстремум, что часто приводит к упрощению вычислений.

На данный момент актуальной проблемой является поиск эффективных решений нелинейных оптимизационных задач, позволяющих находить приближенное решение вспомогательных задач вида

$$f(x) \rightarrow \min, \quad (1)$$

с ограничениями

$$\theta_i(x) \leq 0, \quad i \in I \quad (2)$$

с заданной по  $f(x)$  точностью  $\varepsilon > 0$ . Известные алгоритмы для решения задач (1), (2) являются достаточно трудоемкими, так как в них необходимо решать вспомогательные задачи точными методами. В общем случае это означает бесконечный процесс минимизации вспомогательных функций. Поэтому при решении задач математического программирования удобно пользоваться эвристическими критериями остановки, не гарантирующими выполнения неравенства

$$f(x_k) - f^* \leq \varepsilon, \quad (3)$$

где  $f^* = \min\{f(x), x \in D(0)\}$ , (4)

даже при включении итерационной точки  $x_k \in D(0)$ , где  $\varepsilon$  – заданная точность нахождения  $f^*$ , а  $\{x_k\}$  – последовательность точек приближения.

В разработанном нами программном комплексе был реализован метод штрафных функций с неполной минимизацией вспомогательных функций. Приложение для решения задач выпуклого программирования с неполной минимизацией вспомогательных функций имеет синтаксический анализатор математических формул. При этом задачи выпуклого программирования решаются двумя различными алгоритмами метода штрафов, реализованными в среде программирования Borland Delphi 7.0. Нами предложены практически реализуемые правила задания управляющих параметров, при использовании которых выполнение условий остановки в алгоритмах выполняется не более чем за требуемое число этапов минимизации вспомогательных функций. Помимо этого программный продукт можно использовать для расчета суммарных затрат на использованные ресурсы, на хранение запасов [1, с. 23], на замену оборудования и т. д.

Рассмотрим некоторые алгоритмы в программном комплексе, допускающие приближенное решение вспомогательных задач. Определенный выбор параметра  $p$ , в зависимости от заданной точности решения вспомогательных задач, обеспечивает требуемую точность решения задачи (4) [2, с. 46].

### Алгоритм 1

Задается требуемая точность решения  $\varepsilon > 0$ ,  $x_0 \in R_n$ , натуральное число  $N$ , число  $\delta \in (0, \varepsilon)$ . Выбирается  $0 < p \leq \min(\frac{\beta\gamma\bar{\alpha}}{(V(x^*) - \gamma^2\bar{\alpha})L}, p', \bar{p})$ , возрастающая функция  $\varphi(t)$  такая, что  $\varphi(1) \geq 0$ ,

$$\varphi(N) = \frac{Lp}{\beta s(1-s)^{s-1} \bar{\alpha}^p}. \text{ Полагается } k = 1.$$

1. Вычисляется  $C_k = \varphi(k)$ .