

УДК 004.023

## РАСПАРАЛЛЕЛИВАНИЕ ЭВОЛЮЦИОННЫХ МЕТОДОВ СЕГМЕНТАЦИИ ТРЕХМЕРНЫХ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

*Дрындик Р.В., Привалов М.В.*

*Донецкий национальный технический университет,  
кафедра автоматизированных систем управления  
rman-dryndik@rambler.ru*

*Проанализирован эволюционный алгоритм сегментации трехмерных изображений. Рассмотрена адаптация алгоритма кластеризации K-средних под эволюционную парадигму. Показаны особенности реализации эволюционного алгоритма кластеризации. Рассмотрены некоторые технические характеристики GPGPU.*

**Общая постановка проблемы.** Задача анализа изображений, видео и других данных является очень актуальной в настоящее время, особенно в медицине. Зачастую, постановка правильного диагноза болезни основывается на информации, полученной посредством магнитно-резонансного исследования, позитронно-эмиссионного исследования, рентгена и прочих. Под сегментацией изображения понимают разбиение изображения на непохожие области по некоторому признаку. Все многообразие алгоритмов сегментации трехмерных изображений можно разделить на 3 основные группы [1].

- 1) структурные алгоритмы;
- 2) стохастические алгоритмы;
- 3) комбинированные алгоритмы.

Данные алгоритмы относятся к алгоритмам поиска оптимальных решений. Следует отметить, что хотя данные методы в той или иной мере решают поставленную перед собой задачу, они не лишены недостатков. К основным недостаткам можно отнести:

- 1) высокую вычислительную сложность при большом объеме обрабатываемых данных;
- 2) проигрыш в использовании ресурсов памяти;
- 3) чувствительность некоторых алгоритмов к шумам;
- 4) необходимость в задании дополнительной информации;
- 5) необходимость в настройке некоторых параметров;
- 6) плохие результаты работы с изображениями, на которых яркостные характеристики области различаются незначительно.

**Решение задачи и результаты исследований.** Под сегментом подразумевается некоторая область, такая, что расстояние между ее элементами минимально, а расстояние между двумя соседними областями – максимально. Такой факт дает возможность построить генетический алгоритм, в основе которого может использоваться известный метод кластеризации. В рамках данной статьи рассмотрена адаптация алгоритма K-средних для применения в генетическом алгоритме сегментации.

Требуется разбить множество, состоящее из  $n$  элементов, на  $K$  групп так, чтобы расстояние между элементами внутри группы было минимальным, а расстояние между группами – максимальным. Каждый элемент рассматриваемого множества представляет собой вектор длины  $d$ .

Пусть дано множество  $A = \{x_1, x_2, \dots, x_{n-i+1}, \dots\}$  – множество векторов длины  $d$  и пусть элемент  $x_{ij}$  – это  $j$ -е свойство  $i$ -го элемента множества  $A$ . Введем следующую величину:

$$\mathbf{W} = \begin{pmatrix} w_{11} & \dots & w_{1K} \\ \vdots & \ddots & \vdots \\ w_{N1} & \dots & w_{NK} \end{pmatrix} = \|w_{ik}\| = \begin{cases} 1, & x_i \in G_k \\ 0, & x_i \notin G_k \end{cases} \quad (1)$$

Матрица  $\mathbf{W}$  обладает следующим важным свойством:

$$w_{ij} \in \{0, 1\} \text{ и } \sum_{1 \leq j \leq K} w_{ij} = 1 \quad (2)$$

Обозначим центры кластеров следующим образом  $c_k = (c_{k_1}, c_{k_2}, c_{k_3}, \dots, c_{k_d})$ , тогда

$$c_{k_j} = \frac{\sum_{1 \leq i \leq n} w_{ik} x_{ij}}{\sum_{1 \leq i \leq n} w_{ik}} \quad (3)$$

Тогда внутрикластерное отклонение может быть определено по формуле:

$$S^k(W) = \sum_{1 \leq i \leq n} w_{ik} \sum_{1 \leq j \leq d} (x_{ij} - c_{kj})^2 \quad (4)$$

Требуется найти такую матрицу  $W^* = \|w_{ik}^*\|$ , что  $S(W^*) = \min_W \{S(W)\}$ .

**Реализация генетического алгоритма.** Так как пространство поиска представляет собой множество всех существующих двоичных матриц  $W$ , удовлетворяющих условию (2), то можно представить хромосому в виде строки длины  $n$ , при этом каждый аллель может принимать значения из множества  $\{1, 2, \dots, K\}$ . Инициализация популяции может быть осуществлена одним из классических способов.

Селекция предполагает выбор случайной хромосомы из популяции с помощью метода рулетки, либо какого-либо другого метода. Решение в текущей популяции может задавать некоторый вес, который влияет на выживание в следующем поколении. Это означает, что каждой особи задается некоторое число, либо вычисляется значение специальной фитнес-функции. Существует много способов вычисления этой фитнес-функции [3]. Авторы [2] предлагают следующий вариант решения. Пусть  $f(s_w) = -S(W)$ ,  $g(s_w) = f(s_w) - (\bar{f} - c \cdot \sigma)$ , где  $\bar{f}$ ,  $\sigma$  – среднее значение и стандартное отклонение  $f(s_w)$  соответственно,  $c \in [1..3]$  – некоторая константа. Тогда значение фитнес-функции  $F(s_w)$  определяется следующим образом:

$$F(s_w) = \begin{cases} g(s_w), & g(s_w) \geq 0 \\ 0, & g(s_w) < 0 \end{cases} \quad (5)$$

Оператор мутации меняет значение аллеля в зависимости от расстояния от данной точки до центра кластера. Зададим его следующим образом: аллель заменяется значениями, которые выбираются случайно, согласно следующему распределению вероятностей:

$$p_j = \Pr\{s_W(i) = j\} = \frac{c_m d_{\max} - d_j}{\sum_{1 \leq i \leq K} (c_m d_{\max} - d_i)}, \quad (6)$$

где  $c_m$  – некоторая константа,  $d_{\max} = \max_j \{d_j\}$ .

Алгоритм с представленным выше оператором селекции может сходиться достаточно долго. Кроме того, вероятность мутации должна быть достаточно мала, так как большие значения ведут к колебаниям решений. Чтобы исключить такие ситуации вводится новый оператор, называемый K-means operator (КМО) [2]. Выполнение данного оператора состоит из следующих двух шагов:

- 1) вычисляются центры кластеров по формуле (3);
- 2) каждая точка данных переназначается кластеру с ближайшим центром, тем самым формируя новую матрицу  $\tilde{W}$ .

Имея технику построения начальной популяции и генетические операторы, можно построить генетический алгоритм, блок-схема которого представлена на рис. 1.

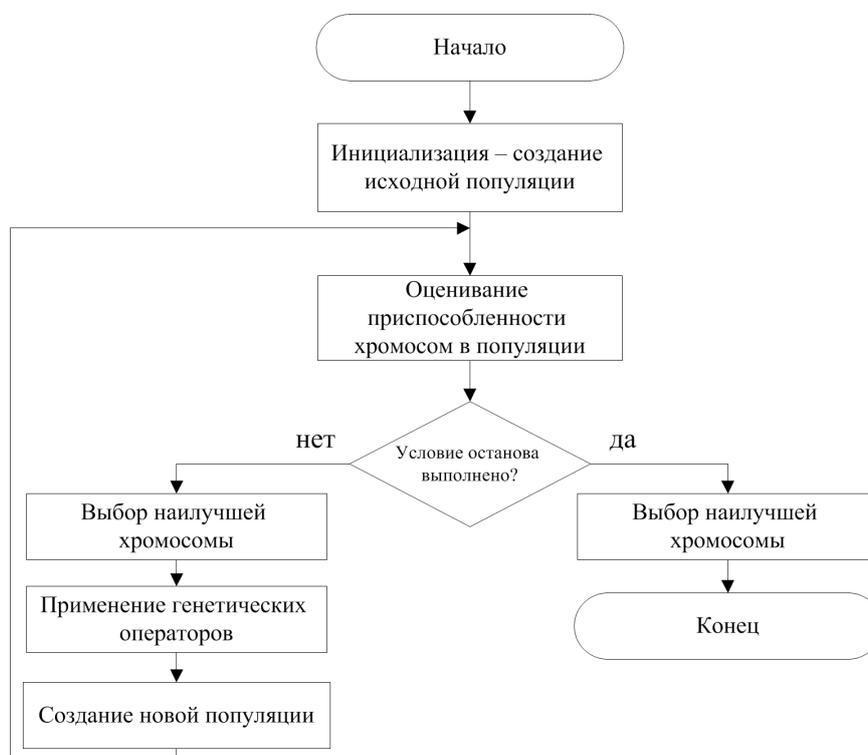


Рисунок 1. Общая схема работы генетического алгоритма

**Исследования возможности распараллеливания алгоритмов сегментации.** Сегментация часто является первым этапом предварительной обработки изображения перед началом работы другого алгоритма. Данная задача является вычислительно трудной, так как алгоритмы работают с отдельными пикселями, а количество операций достаточно велико. Ситуация усугубляется если сегментацию требуется выполнять в реальном времени. Для преодоления таких проблем необходимо выполнять сегментацию параллельно с использованием нескольких вычислительных узлов. В настоящее время активно ведутся исследования по распараллеливанию алгоритмов сегментации.

Одни из ранних исследований проводили Alireza Khotanzad и Abdel Majud Bouarfafa. Их работа [5] посвящена сегментации изображений с использованием порогового

алгоритма кластеризации. В результате их работы было получено восьмикратное ускорение работы алгоритма при использовании 11 процессоров.

В исследованиях, проведенных Shahram Rahimi, M. Zargham и A. Thakre [6], рассматривается возможность распараллеливания алгоритма C-Mean, использующего нечеткое разбиение пикселей на кластеры. Было достигнуто девятикратное ускорение при эффективности близкой к единице. В ходе исследований была выявлена прямопропорциональная зависимость между количеством процессоров и приростом скорости сегментации изображения.

Исследования Nawal Coptu, Sanjay Ranka, Georey Fox, Ravi V. Shankar [7] также показывают хорошие результаты при распараллеливании алгоритма роста регионов. Исследования проводились с использованием вычислительной системы, построенной с архитектурой MIMD. В результате также замечен значительный рост скорости обработки при использовании большого числа вычислительных узлов.

В работе Jan Wassenberg, Wolfgang Middelmann и Peter Sanders [8] описали алгоритм сегментации на графах с использованием вычислительной системы с архитектурой SIMP. Время работы разработанного алгоритма линейно зависит от размерности входных данных.

В ходе работы проводятся исследования о возможности эффективной реализации алгоритмов сегментации на графических процессорах с использованием технологий NVIDIA CUDA и OpenCL.

Развитие современных графических ускорителей привело к появлению графических ускорителей с программируемым конвейером. Если раньше программист графических приложений мог использовать только фиксированную функциональность GPU, то теперь две стадии обработки информации в конвейере GPU стали программируемыми. Более точно, программируемой стала обработка вершин, что позволило создавать, в частности, более разнообразные модели освещения, а так же обработка фрагментов. Столь коренное изменение архитектуры GPU привело к возможности использования GPU не только для целей компьютерной графики, но и для задач, которые ранее решались на CPU, таким образом, появилась технология вычислений общего назначения на графическом ускорителе или *general purpose computation on graphic processor unit*.

Современные GPU обладают высокой скоростью доступа к своей собственной оперативной памяти, которая обычно именуется текстурной памятью и имеют высокую вычислительную мощность. Например: у Intel 3.0 GHz Pentium 4 пиковая вычислительная мощность оценивается как 12 GFLOPs, а у видеокарты ATI Radeon X1800XT 120 GFLOPs, для этого же оборудования пиковая скорость обмена данными между процессором и соответствующей памятью составляет соответственно 5.96 GB/s у CPU против 42 GB/s у GPU.

Но GPU не только быстры и мощны, но и что важно, их мощности растут очень быстрыми темпами: годовой рост производительности CPU - 1.4 раза, а для GPU - 1.7 раз для фрагментного процессора и 2.3 раза для вершинного.

Если учесть при этом относительно низкую стоимость видеокарт по сравнению с равномоощными CPU, то станет понятна эффективность использования видеокарт для вычислений общего назначения.

Современные языки шейдеров, разработанные с использованием синтаксиса

языка C позволяют программистам относительно быстро начать писать эффективные программы для графического ускорителя. При этом язык шейдеров является достаточно гибким и удобным средством для разработки в том числе и сложных вычислительных подпрограмм, к тому же многие математические функции реализованы на GPU аппаратно, что добавляет быстродействия вычислительным программам.

Причина вычислительной мощности GPU в специализированной под графические приложения высоко параллельной архитектуре. Но это вызывает определенные неудобства при использовании GPU для нетрадиционных задач - сохраняются многие термины и некоторый синтаксис, который может показаться программистам, не связанным с компьютерной графикой, несколько искусственным. Ввиду специфичности архитектуры GPU (потокковая модель, а не фон Неймана, как на CPU), программирование GPU оказывается сложным для многих разработчиков.

Существенные ограничения накладываются такие аспекты, как наличие аппаратной поддержки только одинарной точности для чисел с плавающей точкой. На последних моделях и на профессиональных видеокартах аппаратно поддерживаются и целые числа, но пока распространенность такого оборудования не велика.

### **Вывод**

На основании вышеизложенного можно сделать вывод, что применение эволюционных алгоритмов может помочь устранить некоторые недостатки известных методов при обработке трехмерных изображений. Показано, что одним из подходов может быть применение метода кластеризации в совокупности с генетическим алгоритмом.

Анализ предлагаемых решений показал, что эволюционные алгоритмы основаны на ряде независимых подзадач, что позволяет значительно повысить скорость их работы, выполнив распараллеливание с применением вычислений на CPU, а также GPGPU. Таким образом, направлением дальнейшей работы является разработка и реализация параллельных вычислительных схем для эволюционных вариантов алгоритмов сегментации, а также их проверка на задачах кластеризации и классификации КТ и МРТ изображений.

### **Список источников**

- [1] 3D Segmentation Techniques for Medical Volumes Sarang Lakare Center for Visual Computing Department of Computer Science State University of New York at Stony Brook Research Proficiency Exam Advisor: Prof. Arie Kaufman
- [2] Genetic K-Means Algorithm K. Krishna and M. Narasimha Murty
- [3] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley, 1989.
- [4] C. von der Malsburg. The correlation theory of brain function. Technical report, Max-Planck-Institute Biophysical Chemistry, 1981.
- [5] Alireza Khotanzad, Abdelmajid Bouarfa. Image segmentation by a parallel, non-parametric histogram based clustering algorithm. Image Processing and Analysis

---

Laboratory, Electrical Engineering Department, Southern Methodist University, Dallas, Texas 75275, U.S.A.

- [6] Shahram Rahimi, M. Zargham, A. Thakre. A Parallel Fuzzy C-Mean algorithm for Image Segmentation. Режим доступа [http://opensiuc.lib.siu.edu/cs\\_pubs/26/](http://opensiuc.lib.siu.edu/cs_pubs/26/) свободный. – Загл. с экрана. – Яз. англ.
- [7] Nawal Coptu , Sanjay Ranka, Geoffrey Fox , Ravi V. Shankar. A Data Parallel Algorithm for Solving the Region Growing Problem on the Connection Machine. Режим доступа <http://www.new-prac.org/users/fox/pdftotal/sccs-0596.pdf> свободный. – Загл. с экрана. – Яз. англ.
- [8] Wassenberg Jan. Middelmann Wolfgang, Sanders Peter. An Efficient Parallel Algorithm for Graph-Based Image Segmentation. Электронный ресурс. Режим доступа <http://algo2.iti.kit.edu/wassenberg/wassenberg09parallelSegmentation.pdf> свободный. – Загл. с экрана. – Яз. англ.