

УДК 512.543

С. И. Хашин, Ю. А. Хашина<sup>1</sup>

## Алгоритм управляемой статической сегментации

**Ключевые слова:** сегментация образов, клеточный автомат.

Строится алгоритм сегментации изображения, использующий  $R$ -сегментацию, клеточный автомат и упрощение сегментов. Оценивается качество построенного алгоритма.

**Key words:** image segmentation, cell automat.

The algorithm of image segmentation, the using the  $R$ -segmentation, the cell automat and simplification of segments is created.

### 1. Непрерывная сегментация

Сегментация изображений является одной из важнейших задач компьютерной графики. В настоящее время разработано множество различных методов сегментации [1, 3]. Однако они не являются равно пригодными для всех случаев. Поэтому постоянно появляются новые алгоритмы [2, 5, 7, 8].

Напомним вкратце определение «непрерывной сегментации» изображения [2, 5]. Пусть  $f(x, y)$  – функция яркости изображения, будем считать ее достаточно гладкой (класса не ниже  $C^2$ ) в некоторой области на плоскости. Рассмотрим квадрат длины градиента

$$S(x, y) = \|\nabla f\|^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2.$$

Точку  $(x, y)$  назовем граничной, если производная от  $S$  по направлению градиента функции  $f$  равна 0, т. е.

$$R(x, y) = \frac{1}{2}(\nabla S, \nabla f) = \left(\frac{\partial f}{\partial x}\right)^2 \cdot \frac{\partial^2 f}{\partial x^2} + 2\frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} \cdot \frac{\partial^2 f}{\partial x \partial y} + \left(\frac{\partial f}{\partial y}\right)^2 \cdot \frac{\partial^2 f}{\partial y^2} = 0.$$

В компьютерной графике чаще всего функция  $f$  задана таблично своими значениями в целочисленных точках. Простейший способ получения гладкой функции – переход к кусочно-гладкой интерполяции. Однако получаемая в этом случае линия границе оказывается чересчур сложной и фрагментированной. В работе [2] было предложено вместо кусочно-гладкой интерполяции функции  $f(x, y)$  взять билинейную интерполяцию для отдельно для частных производных функции  $f$  первого и второго порядков.

© Хашин С. И., Хашина Ю. А., 2012

<sup>1</sup>Ивановский государственный университет; E-mail: khash2@mail.ru. Работа выполнена при поддержке гранта РФФИ 11-07-00653а

Получающиеся линии границы соответствуют нашим интуитивным ожиданиям, по крайней мере в простейших случаях. Построенная таким образом линия границы на каждом единичном квадрате является некоторой алгебраической кривой 3-го порядка.

Тем не менее получающаяся линия дает нам слишком мелкие сегменты. Даже после сглаживания получающиеся сегменты оказываются слишком мелкими и имеют довольно сложную структуру. Тем не менее эта идея хорошо подходит как начальный этап более сложного и совершенного алгоритма сегментации.

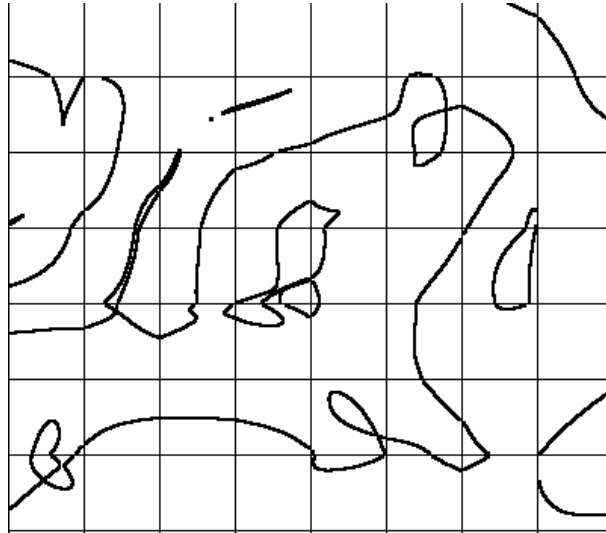


Рис. 1. Пример границы в области размера 8\*7

***R*-сегментация.** Предложенный алгоритм можно представить следующим образом.

**Шаг 1.** Для данной функции яркости  $f[x, y]$  находим ее  $r$ -функцию  $r[x, y]$ , фактически матрицу того же размера.

**Шаг 2.** Те точки, в которых  $r[x, y] \geq 0$ , относим к 1-му сегменту, остальные – ко второму.

**Шаг 3.** Если размер связной компоненты сегмента вокруг точки из сегмента  $s$  ( $s = 1, 2$ ) меньше константы `minSize`, то эту связную компоненту относим к сегменту  $3 - s$ .

**Шаг 4.** Оптимизация. Этот шаг универсален и может быть применен для произвольной начальной сегментации.

## 2. Оптимизация сегментации

В результате работы многих математически простых алгоритмов мы получаем сегментацию, далеко не полностью удовлетворяющую нашим требованиям и ожиданиям. Однако имеются некоторые стандартные методы, позволяющие существенно улучшить практически любую исходную сегментацию. В этом разделе мы их и рассмотрим. С их помощью улучшение сегментации можно описать следующими шагами:

1. Переход к связным сегментам.
2. Удаление маленьких сегментов.
3. Сглаживание с помощью клеточного автомата с параметром 5.
4. Морфологическое сглаживание, 3 повтора.
5. Упрощение сегментов.
6. Объединение сегментов до получения их требуемого количества.

**2.1. Связные сегменты.** Сегмент называется связным, если мы можем пройти от любой его точки до любой другой, переходя каждый раз к соседней точке в пределах этого же сегмента.

Очевидно, мы имеем два варианта связности: 4-связность и 8-связность, в зависимости от того, сколько соседних точек мы рассматриваем: 4 или 8. Обычно удобнее рассматривать 4-связность и, если не оговорено особо, это и будет предполагаться. На некотором этапе сегментации мы можем получить несвязные сегменты. Тем не менее практически всегда в конечном результате сегменты предполагаются связными. Поэтому в начале процесса улучшения мы должны разбить каждый сегмент на связные компоненты и в дальнейшем обеспечивать связность сегментов при всех операциях.

**2.2. Удаление маленьких сегментов.** В результате работы многих алгоритмов сегментации у нас может получиться большое количество совсем маленьких сегментов. Перед применением других методов улучшения сегментации стоит просто удалить все маленькие сегменты. Наименьший допустимый размер сегмента не должен быть меньше 40 точек и не должен превышать 1% от заданного среднего размера сегмента. Например, если мы хотим сегментировать изображение размера  $4000 * 3000$  точек на 100 сегментов, то на этом шаге должны быть удалены все сегменты размера меньше  $0.01 * 4000 * 3000 / 100 = 1200$  точек. Если размер изображения  $640 * 480$ , то минимальный размер сегмента равен  $\text{Max}(40, 0.01 * 640 * 480 / 100) = 40$  точек. С точками малых сегментов можно поступить двумя способами:

1. Собрать в один сегмент, обычно это сегмент номер 0, содержащий все несегментированные точки для последующей сегментации.
2. Каждую удаляемую точку присоединять к одному из соседних больших сегментов, а именно к тому, точек которого больше среди восьми соседей рассматриваемой точки.

**2.3. Сглаживание областей с помощью клеточного автомата.** Пусть задана сегментация  $s[*,*]$  изображения размера  $m_x * m_y$ , т. е. для каждой точки  $(x, y)$  задано натуральное число  $s[x, y]$  – номер сегмента, к которому относится точка  $(x, y)$ . Задаем  $k$  – уровень клеточного автомата из интервала  $[4..8]$ . Для каждой точки  $(x, y)$  на изображении рассмотрим восемь ее соседей. Если сама точка относится к одному сегменту, а не менее  $k$  ее соседей к другому, то и саму точку относим к этому сегменту. Обходим все точки изображения в псевдслучайном порядке и для каждой из них выполняем эту операцию. Назовем такой обход одним шагом

клеточного автомата с уровнем  $k$ . Будем выполнять такие шаги, пока за весь шаг сегментации произошла смена сегмента хотя бы в одной точке. На краях и на углах изображения требования к количеству соседей из другого сегмента надо, конечно, несколько модифицировать.

**2.4. Морфологическое сглаживание.** Пусть задана сегментация  $s[*,*]$  изображения размера  $m_x * m_y$ .

Точка называется граничной, если она принадлежит к одному сегменту, а хотя бы одна из ее четырех (восьми) соседей – к другому.

Внутренней границей сегмента называется множество граничных точек, принадлежащих сегменту. Внешней границей сегмента называется множество граничных точек, не принадлежащих сегменту, но соседних с ним.

Дилатацией сегмента назовем присоединение к нему всех его внешних граничных точек из всех других сегментов. Эрозией сегмента назовем удаление из него всех его внутренних граничных точек. Эрозия возможна в двух вариантах:

1. Присоединение удаляемых из сегмента точек к фиксированному сегменту. Обычно это 0-й сегмент, содержащий все еще не сегментированные точки.

2. Присоединение каждой из удаляемых из сегмента точек к тому сегменту, который больше представлен среди восьми соседних точек. Если таких несколько, то выбираем один из них случайным образом.

На одном шаге морфологического сглаживания выполняем последовательно эрозию и дилатацию каждого из сегментов. В целом морфологическое сглаживание – это последовательность из нескольких таких шагов.

**2.5. Упрощение сегментов.** Иногда оказывается, что некоторые сегменты, особенно большие, имеют слишком сложную структуру, типа показанной на рис. 2.



Рис. 2. Сложный сегмент

Очевидно, что этот сегмент (черный) является на самом деле объединением двух различных сегментов, на которые мы и должны его разбить.

**Определение 1.** Связный сегмент  $U$  называется простым, если он остается связным после любого количества эрозий (т. е. удаления из него граничных клеток).

С практической точки зрения это определение является слишком сильным и для применения при улучшении сегментации его стоит ослабить.

**Определение 2.** Связный сегмент  $U$  называется полупростым, если он остается связным после некоторого количества эрозий, до тех пор пока количество точек в нем не уменьшится в два раза. Полупростые сегменты мы будем оставлять без изменений, а неполупростые – разбивать на меньшие части.

**Алгоритм упрощения сегмента.**

1. Выполняем эрозию сегмента до тех пор, пока количество точек в нем не уменьшится в два раза или он перестанет быть связным.
2. Если сегмент остался связным, мы его оставляем без изменений.
3. Если сегмент в результате очередной эрозии распался на несколько компонент, то мы присваиваем каждой из этих компонент новый номер сегмента и выполняем для каждого из этих новых сегментов последовательные дилатации: присоединение граничных точек из исходного сегмента.
4. Каждый из построенных таким образом новых сегментов опять подвергаем упрощению по тому же алгоритму.

**2.6. Объединение сегментов, топология.** После проведения всех указанных операций количество сегментов может оказаться больше, чем нам требуется. Поэтому некоторые сегменты надо объединить. Для выбора объединяемых сегментов введем понятие топологии сегментации. Они реализованы в классе `segm2`. Основная идея – предварительный расчет количества граничных точек для каждой пары сегментов (`matr top`) и сумма разностей цветов по граничным парам у сегментов (`matr ctop`). Вот важнейшие данные и методы:

```

matr top; // топология, количество граничных пар у сегментов
matr neib; // упорядочение сегментов по размеру общей границы
matr ctop; // сумма разностей цветов по граничным парам
void Topology ( matr &top);
void Topology ( matr &top, matr3 &a);
void Topology ( matr &top, matr &a);
//дополнительные данные:
int nSgm; // количество сегментов, их номера 0..nSgm-1
int aSz; // размер массивов (обычно равно nSgm)
int *area; // размеры сегментов
double *midX, *midY, *offs; //центр и ср.кв.расст.до центра
double *midR, *midG, *midB; //средний цвет по сегменту
// работа с доп.данными
void sRelease(); // удалить доп.данные
void getInfo(); // данные о сегментах
void getInfo(matr &a);
void getInfo(matr3 &a);

void getInfoT(); // данные о сегментах с топологией
void getInfoT(matr &a);
void getInfoT(matr3 &a);

```

```

// извлечение
int Area( int iSegm );           // площадь сегмента
double MidX( int iSegm );
double MidY( int iSegm );
double Offs( int iSegm );

double MidR( int iSegm );
double MidG( int iSegm );
double MidB( int iSegm );
// цветовое расстояние между сегментами
double colDist( int iSegm1, int iSegm2 );
// длина общей границы между сегментами
int bordSize( int iSegm1, int iSegm2 );
int Neib( int iSegm, int k );
// соседние сегменты по убыванию размера общей границы
double colBDist( int iSegm1, int iSegm2 );
// цветовое расстояние между сегментами вдоль границы
void saveInfo( char *fname);

```

Предложенный набор методов позволяет эффективно реализовать алгоритм улучшения начального разбиения.

## Список литературы

1. Барينوва О., Вержневцев А. Методы сегментации изображений: автоматическая сегментация // Компьютерная графика и мультимедиа. Вып. 4(4)/2006. URL: <http://cgm.computergraphics.ru/content/view/147>.
2. Кручинин А. Е., Хашин С. И. Сегментация изображения путем выделения непрерывных границ // Вестник Иван. гос. ун-та. 2007. Вып. 3. С. 80–83.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М. : Техносфера, 2006. 1072 с.
4. Хашин С. И. Оценка качества сегментации изображения // Вестн. Иван. гос. ун-та. 2010. Вып. 2. С. 112–118.
5. Хашин С. И., Хашина Ю. А. Свойства  $r$ -сегментации изображения // Математика и ее приложения : журн. Иван. мат. о-ва. Иваново : Иван. гос. ун-т, 2007. Вып. 1(4). С. 93–98.
6. Хашин С. И., Хашина Ю. А. Реализация алгоритма непрерывной сегментации изображений // Математика и ее приложения : журн. Иван. мат. о-ва. Иваново : Иван. гос. ун-т, 2011. Вып. 1(8). С. 151–156.
7. Чочиа П. А. Пирамидальный алгоритм сегментации изображений // Информационные процессы. 2010. Т. 10, № 1. С. 23–35. URL: <http://www.jip.ru/2010/23-35-2010.pdf>.
8. Farag A. A., El-Baz A. S., Gimel'farb G. Precise segmentation of multimodal images // IEEE Trans. Image Process. 2006. Vol. 15, № 4. P. 952–968.

Поступила в редакцию 26.11.2012.