

# 02, февраль 2016

УДК 004.932.2

## О задаче сегментации дорог на цифровых снимках Земли

*Переверзев В.А., бакалавр  
кафедра «Программное обеспечение ЭВМ и информационные технологии»,  
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана*

*Научный руководитель: Волкова Л.Л., ассистент  
кафедра «Программное обеспечение ЭВМ и информационные технологии»,  
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана  
[irudakov@bmstu.ru](mailto:irudakov@bmstu.ru)*

Сегментация изображения является одной из основных задач обработки и анализа изображения в компьютерном зрении. Сегментация – это процесс разделения изображения на сегменты. Цель сегментации заключается в изменении или упрощении представления кадра, чтобы его было легче анализировать. В частности, данный процесс используется для выделения объектов и контуров на кадре. Результатом сегментации изображения являются сегменты, покрывающие все изображение или контуры, выделенные из кадра. В сегменте все пиксели схожи по некоторому критерию, а пиксели соседних областей имеют значительную разницу по этим критериям. Критерий – это некое свойство пикселя, например, цвет, яркость или текстура.

Космические снимки Земли в высоком разрешении содержат большое количество деталей, таких, как природные ресурсы Земли и последствия человеческой деятельности, что затрудняет распознавание дорог со снимков. Вследствие этого выделение дорог с цифровых снимков Земли является нетривиальной и важной задачей, т. к. позволяет обновлять карты автоматически, без привлечения человека [1]. При решении данной задачи возникает вопрос, как можно убрать шум. Шумом на снимках являются тени, деревья, машины и другие объекты, которые могут закрывать дорогу. Разработано несколько универсальных алгоритмов и методов сегментации изображения. Для повышения эффективности работы данных методов и алгоритмов чаще всего необходимо совмещать методы со знаниями из предметной области.

## **1. Классификация методов**

Алгоритмы сегментации изображений можно разбить на группы, использующие похожие ключевые идеи. Ниже представлены несколько групп методов.

- **Методы, использующие кластеризацию данных**
  - к-средних [2];
  - сдвиг среднего [3];
  - метод Отса.
- **Эвристические методы**
  - разрастание регионов;
  - разделения и слияния регионов;
  - водораздел[4];
  - эффективный метод на графах [5];
  - метод Эйквила.
- **Энергетические методы**
  - метод уровня [6];
  - турбоПиксели;
  - алгоритм змейки [7].
- **Методы, основанные на оттенках серого и гистограммах**
  - метод Розенфельда;
  - метод Рамеша;
  - бинаризация с нижним порогом;
  - бинаризация с верхним порогом;
  - бинаризация с двойным ограничением.
- **Выделение краёв**
  - Саппу;
  - Рb-детектор.

### **1.1. Неподходящие алгоритмы**

Некоторые из написанных выше алгоритмов и методов сегментации изображения не подходят для выделения дорожного полотна со снимков.

Рассмотрим эти методы:

- метод водораздела:  
главная идея данного метода, заключается в том, что изображение в градации серого можно интерпретировать как карту высот. Представим, что льем воду на

каждый пиксель и смотрим, куда вода стекает. Образовываются бассейны, т.е. множество пикселей, поток из которых стекает к одной общей точке. Результатом работы данного метода является карта суперпикселей. Данный метод очень чувствителен к шуму – ищутся все локальные минимумы. Это, в свою очередь, приводит к чрезмерной сегментации изображения на кластеры, т.е. дорожное полотно разбивается на несколько независимых объектов, что затрудняет последующую работу с извлеченными данными. Поэтому требуется слияние кластеров.

- метод Эйквила:

в данном методе пороговой обработки применяются два окна  $g$  и  $R$ , из которых окно  $R$ , большее по размерам, служит для вычисления значения порога, в то время как окно  $g$ , меньшее, определяет область изображения, в которой будет использоваться полученный порог. Оба окна перемещаются параллельно по изображению с шагом, равным размеру меньшего окна  $g$ , и каждый раз для всех элементов окна  $R$  вычисляется оптимальный порог. Из-за наличия шумов на кадре нет возможности извлечь дорогу целым объектом, т.к. дорога будет разбита на разные сегменты.

- метод Змейка:

данный метод формулируют как задачу минимизации энергии. В качестве признаков изображения используются только градиенты. Смысл метода заключается в следующем: вокруг определенного сегмента на изображении строится контур. Для выделения данного сегмента необходимо приближать этот контур к границе данного сегмента, при этом у контура двигаются только контрольные точки. Для каждой контрольной точки выбирается новое положение из набора пикселей в окрестности. После выбора оптимальной комбинации новых положений повторяем операцию до совпадения контура и объекта. Данный метод хорош, когда надо выделить крупные объекты в изображении. В случае с выделением дорог данный метод не удобен, т.к. придется строить много маленьких независимых сегментов, пытаться их выделить и только после это объединять.

Среди большого числа разнообразных методов сегментации изображения данных можно найти такие, которые будут выдавать лучший результат при выделении дорожного полотна.

Ниже описывается методология автоматического выделения дорог, состоящая из трех частей:

- фильтрация:  
применяется медианный фильтр для удаления.
- выделение дорожного полотна:  
используется метод сдвига и эффективный алгоритм на графах.
- проверка точности выделения дорожного полотна:  
сопоставление оригинального фото и топологии выделенной дороги.

## 2. Фильтрация

### 2.1. Медианный фильтр

Медианный фильтр применяется для удаления шумов (зданий, машин и т.п.). В этом фильтре значение выходного пикселя зависит от медианных значений соседних пикселей, а не от средней выборки значений. Медианный фильтр значительно менее чувствителен по сравнению с mean фильтром для завышенных значений (пики шума). Надо отметить, что медианный фильтр убирает пиковые значения шума намного качественнее, без ухудшения резкости изображения.

## 3. Выделение дорожного полотна

### 3.1. Сдвиг среднего

Сдвиг среднего – это непараметрический метод кластеризации данных, который не требует предварительного знания количества кластеров и не ограничивает форму кластера. Главная идея алгоритма состоит в том, что центры кластеров соответствуют локальным максимумам плотности распределения данных.

#### Типы функции ядра.

Функция конечного числа точек из выборки  $x_1, \dots, x_n$ :

$$P(x) = \frac{1}{n} \sum_{i=1}^n K(x - x_i) \quad (1)$$

В свойства ядровых функций:

- нормализация:

$$\int_{R^d} K(x) dx = 1 \quad (2)$$

- симметрия:

$$\int_{R^d} xK(x)dx = 0 \quad (3)$$

- экспоненциальное убывание веса с расстоянием до точки:

$$\lim_{\|x\| \rightarrow \infty} \|x\|^d K(x) = 0 \quad (4)$$

Есть несколько разных ядровых функций:

- ядро Епанечникова:

$$K_E = \begin{cases} c(1 - \|x\|^2) & \|x\| \leq 1 \\ 0 & \text{иначе} \end{cases} \quad (5)$$

- равномерное ядро:

$$K_E = \begin{cases} c & \|x\| \leq 1 \\ 0 & \text{иначе} \end{cases} \quad (6)$$

- нормальное ядро

$$K_E = c * \exp(-\frac{1}{2} \|x\|^2) \quad (7)$$

Теперь можно перейти к описанию метода. Сдвиг среднего – это процесс нахождения максимума функции плотности с учетом дискретных данных этой функции. Это помогает определить моды этой плотности. Возьмем градиент от плотности распределения:

$$\nabla P(x) = \frac{1}{n} \sum \nabla K(x - x_i) \quad (8)$$

Рассмотрим функцию ядра:

$$K(x - x_i) = ck \left( \left\| \frac{x - x_i}{h} \right\|^2 \right), \quad (9)$$

где  $h$  – размер окна

$$g(x) = -k'(x) \quad (10)$$

$$g_i = -k' \left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \quad (11)$$

Градиент:

$$\nabla P(x) = \frac{c}{n} \sum_{i=1}^n \nabla k_i = \frac{c}{n} \left[ \sum_{i=1}^n \nabla g_i \right] \left[ \frac{\sum_{i=1}^n x_i g_i}{\sum_{i=1}^n g_i} - x \right] \quad (12)$$

Вычислим вектор сдвига среднего:

$$m(x) = \frac{\sum_{i=1}^n x_i g\left(\frac{\|x - x_i\|}{h}\right)}{\sum_{i=1}^n g\left(\frac{\|x - x_i\|}{h}\right)} - x \quad (13)$$

Результатом данного алгоритма является кластер, у которого точки приводятся к одной и той же моде распределения (рис. 1).

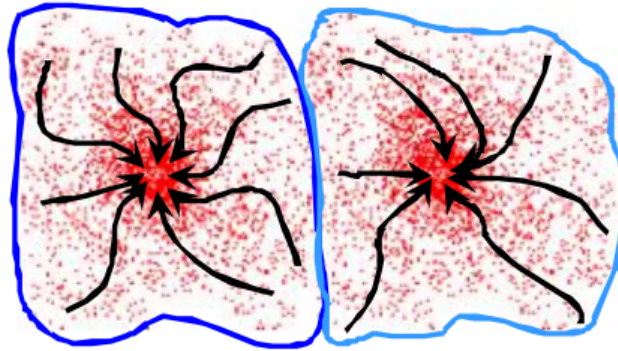
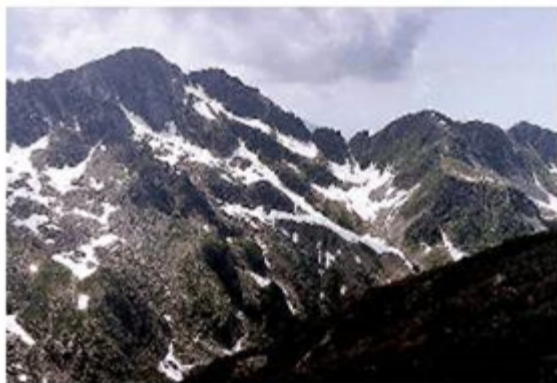


Рис. 1. Мода распределения

На рис. 2 представлена работа данного метода.



оригинальное изображение



сегментированное изображение

Рис. 2. Результат работы метода Сдвиг среднего

Сложность метода

$$O(dN^2T), \quad (14)$$

где  $d$  – размерность вектора,  $N$  – число точек во множестве,  $T$  – число итераций алгоритма.

### 3.2. Эффективный алгоритм на графах

Пусть каждый пиксель изображения – вершина графа. А вес ребра, соединяющего соседние вершины, определяется формулой:

$$w(v_i, v_j) = |I(p_i) - I(p_j)|, \quad (16)$$

где  $I(p_i)$  – интенсивность (яркость) пикселя  $p_i$ , так же и для  $I(p_j)$ . С помощью алгоритма Краскала строится минимальное остовное дерево (МОД/MST) для взвешенно связного неориентированного графа. В ходе работы алгоритма получим несколько разрозненных сегментов с минимальным суммарным весом ребер внутри, т.е. с минимальными «перепадами интенсивностей» между соседними пикселями. Поэтому соседние пиксели внутри одного сегмента будут схожи по цвету, но только до некоторого значения максимального ребра, т.е. перепада интенсивности.

Берется минимальное на текущем шаге ребро. Для двух пикселей, являющихся смежными вершинами этого ребра, определяется из одного ли они (уже построенного) сегмента:

- если из одного, то следует продолжить выполнение алгоритма;
- если нет, то необходимо определить, являются ли сегменты частью одного и того же объекта на изображении и их следуют объединить, или интенсивность сегментов существенно различается.

Для определения различий между построенными и строящимися сегментами ассоциируем величину – максимальный перепад интенсивностей внутри сегмента, т.е. определяется самое длинное ребро MST внутри сегмента:

$$Int(C) = \max_{e \in MST(C)} w(e) \quad (17)$$

Для этого достаточно сохранять длину ребра, добавляемого при объединении составляющих «подсегментов». На момент объединения длина добавляемого ребра больше, чем в уже построенных MST каждого «подсегмента», т.к. ребра обрабатываются в возрастающем порядке. На основе этого можно вывести правило для сегментов  $C_1, C_2$ :

$$D(C_1, C_2) = \begin{cases} true, & Diff(C_1, C_2) > MInt(C_1, C_2) \\ false, & \text{слить сегменты} \end{cases} \quad (18)$$

|                                                                                    |                                                                                        |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| $D(C_1, C_2)$                                                                      | Признак: следует ли разграничить сегменты                                              |
| $Diff(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} (w(v_i, v_j))$ | Текущее ребро минимальной длины, соединяющее два разрозненных сегмента                 |
| $MInt(C_1, C_2) = \min(Int(C_1), Int(C_2))$                                        | Меньший (из больших) перепад интенсивностей внутри одного из рассматриваемых сегментов |

Для объединения сегментов необходимо, чтобы перепад интенсивностей на их границе был меньше максимального перепада внутри каждого из объединяемых сегментов (рис. 3).

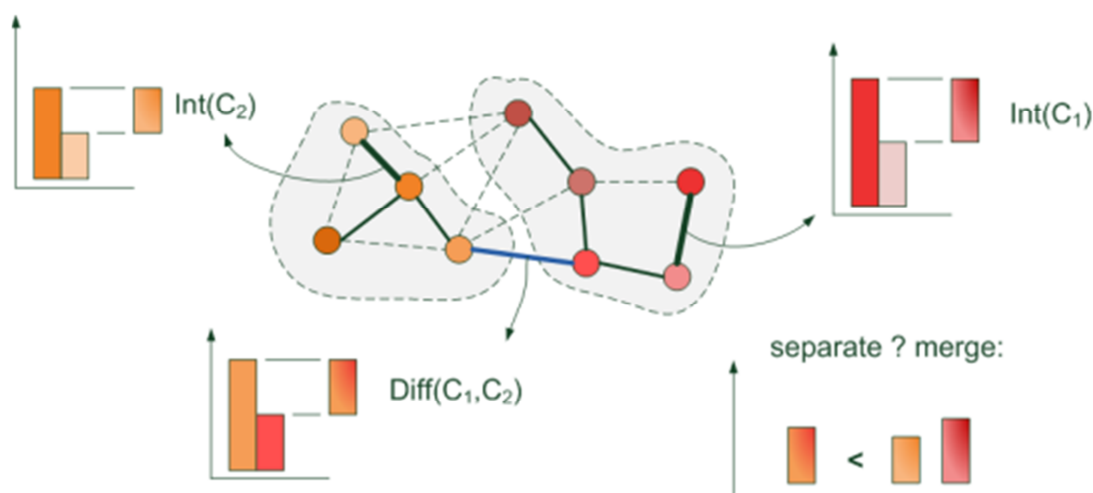


Рис. 3. Перепады интенсивности пикселей

Различают два способа построения графа по изображению (рис. 4):

- **пиксель с четырьмя связями:** каждый пиксель соединен с соседним пикселем сверху, снизу, слева, справа. В полученном графе количество ребер минимально.
- **пиксель с восемью связями:** к предыдущему варианту добавляются соединения соседних пикселей по диагонали. В полученном графе ребер больше.

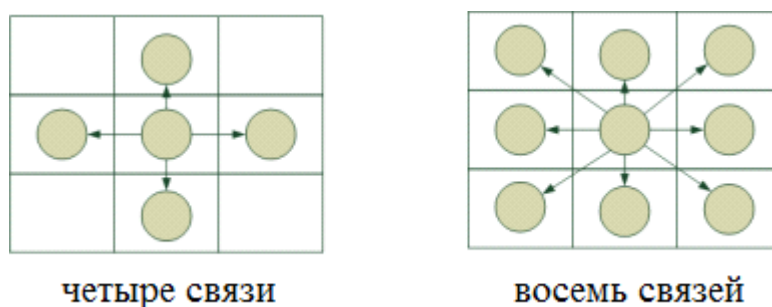


Рис. 4. Виды связей пикселей в графе

Более качественная сегментация изображения будет при использовании восьми связей в пикселе, но работать данный метод будет дольше, в то время, как алгоритм с четырьмя связями выдает приемлемый результат и работает значительно быстрее.

В качестве расстояния между пикселями используется Евклидово расстояние, зависящее как от положения пикселей  $(x, y)$ , так и от их цвета  $(r, g, b)$ :

$$dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad (19)$$

При использовании данной формулы пиксели будут считаться соседними, если они расположены рядом, либо имеют схожий оттенок, хотя физически могут находиться на некотором удалении. Для построения графа каждый пиксель соединяют с 10-20 или 30 «ближайшими». Это дает более качественную сегментацию, но требует больше вычислительных ресурсов.



Для того, чтобы пиксели сливались в большей степени, чем уже построенные сегменты, в решающем правиле добавляют величину, зависящую от размера построенного сегмента

$$T(C) = \frac{k}{|C|}, \quad (20)$$

где  $|C|$  - мощность рассматриваемого сегмента, т.е. количество пикселей в нем, а  $k$  – это параметр сегментации, который задается вручную. Тогда формула с учетом данного замечания будет выглядеть так:

$$M(C_1, C_2) = \min(Int(C_1) + T(C_1), Int(C_2) + T(C_2)) \quad (21)$$

Результат работы алгоритма (рис. 5).

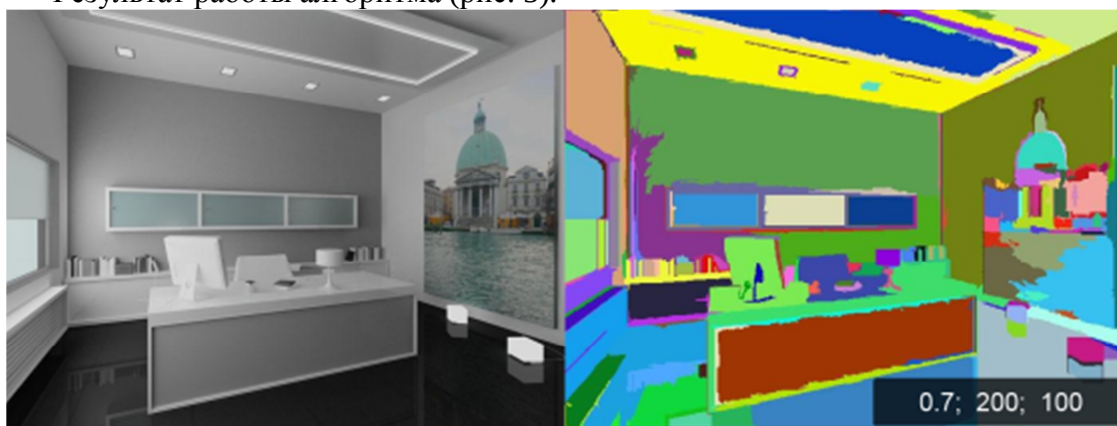


Рис. 5. Результат работы алгоритма – эффективная сегментация на графах

Сложность данного метода

$$O(e \log e) \quad (22)$$

где  $e$  – количество ребер в графе. Для изображения  $N \times M$  пикселей будет  $e = 4 \cdot N \cdot M$ .

### 3.3. Проверка точности выделения дорожного полотна

Для того, чтобы показать точность работы, выделенное дорожное полотно преобразуют в бинарный формат. Это бинарное изображение накладывается на оригинальный кадр, что позволяет увидеть дорожную топологию дороги, не обращая внимания на шум в изображении. В наложенном изображении тонкие линии показывают дорожную топологию.

## 4. Заключение

В данной статье приведена классификация методов сегментации изображений. Разработано огромное количество методов и алгоритмов сегментации изображения, каждый из них имеет свои плюсы и минусы. Для достижения наилучшего результата необходимо применять совокупность методов и сведений из предметной области.

Показана неприменимость некоторых методов, таких, как водораздел, метод Эйквила и метод Змейка.

Описывается метод автоматического выделения дорожного полотна. Был проведен анализ сложности используемых методов.

### **Список литературы**

- [1]. Jing Li, Tao Yang, Jingyi Yu, Zhaoyang Lu, Ping Lu, Xia Jia and Wenjie Chen. Fast Aerial Video Stitching. Int J Adv Robot Syst, 2014, vol. 11, 167. DOI: 10.5772/59029
- [2]. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест Алгоритмы: построение и анализ. М.: Вильямс, 2013. 1324 с.
- [3]. Chunming Li, Chenyang Xu, Martin D. Fox Level Set Evolution Without Re-initialization: A New Variational Formulation. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, DOI: 1063-6919/05.
- [4]. Allan Hanbury, Beatriz Marcotegui Waterfall Segmentation of Complex Scenes. Institute of Computer-Aided Automation, Vienna University of Technology, Favoritenstraße 9/1832, A-1040 Vienna, Austria. 2005.
- [5]. Pedro F. Felzenszwalb and Daniel P. Huttenlocher Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, 2004, vol. 59, no. 2.
- [6]. Tapas Kanungo, David M. Mount, Nathan S. Netanyahu An Efficient k-Means Clustering Algorithm: Analysis and Implementation IEEE Trans. on pattern analysis and machine intelligence. 2002. Vol. 24, no. 7. P. 881-892.
- [7]. Comaniciu D., Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 2002. Vol. 24, no. 5. P. 603–619.
- [8]. Anthony Yezzi, Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum A Geometric Snake Model for Segmentation of Medical Imagery IEEE Transactions on medical imaging. 1997. Vol. 16, no. 2. P. 199-209.