САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра Системного Программирования

Петухов Дмитрий Сергеевич

Распознавание общей структуры городских сцен по единственному изображению

Курсовая работа

Научный руководитель:
к. ф.-м. н., доцент Вахитов А. Т.

Санкт-Петербург
2015

Оглавление

Введение

Понять, какой предмет изображён, и где он располагается на фотографии – одна из основных задач в области компьютерного зрения. Однако даже такие, казалось бы, простые вопросы, как "Где на фотографии расположена земля? Где небо?", то есть вопросы, касающиеся общей структуры сцены изображения, остаются сложными для распознования компьютером, несмотря на то, что человек может ответить на них с лёгкостью.

Стоит отметить, что такого рода геометрические сведения могут быть полезны при решении многих других задач компьютерного зрения. Так, например, знания о расположении земли и вертикальных объектов (например, зданий), позволили уменьшить количество ложных срабатываний в алгоритмах детекции объектов [4], также благодаря им удалось разработать алгоритм автоматического построения грубых 3D моделей по единственному изображению [3]. Далее, эти знания нашли применение в робототехнике, а именно в навигации роботов [7].

Данная работа посвящена извлечению из фотографии информации об общей структуре сцены. Множество исследований по классификации частей изображения, направлены на моделирование *семантических* классов (например, животные, деревья). Мы же заинтересованы в моделировании именно *геометрических* классов, принадлежность к которым зависит от положения объекта на сцене. Например, кусок фанеры, лежащий на земле, и такой же кусок фанеры, прислоненный к стене, будут принадлежать разным геометрическим классам, но одному и тому же семантическому.

Так же следует сказать, что в рамках нашего исследования мы будем работать только с уличными фотографиями (то есть с теми, на которых изображены здания, дороги, пешеходы, машины и так далее), снятыми в светлое время суток.

Таким образом, **целью** нашей работы является разработка устойчивого алгоритма разбиения изображений городских сцен на области "неба", "вертикальных объектов" и "земли".

В рамках нашей работы были поставлены следующие **задачи**:

- Изучить существующие решения
- Сформулировать алгоритм решения задачи
- Реализовать алгоритм и протестировать его

1. Существующие работы

Наибольший интерес для нас представлют работы Д. Хоема, как одного из основных исследователей в области геометрической сегментации изображения. В [3] автор предложил алгоритм, способный размечать изображения на области "неба", "вертикалей" и "земли" на фотографиях, сделанных вне помещения, с точностью 86%. Опишем принцип работы этого алгоритма.

В первую очередь изображение разбивается на суперпиксели (локально-однородные участки изображения, подробнее о них мы расскажем дальше) с помощью алгоритма Фелзенцвальба [1]. Затем суперпиксели, которые с определенной долей вероятности принадлежат одному и тому же геометрическому классу, объединяются в группы, число которых варьируется. В заключении, суперпикселю присваивается метка, которая максимизирует вероятность того, что суперпиксель принадлежит этому геометрическому классу и вероятность того, что он входит именно в эту группу суперпикселей, то есть не противоречит меткам одногруппников. Для приближения функций вероятности используется AdaBoost в форме логистической регрессии.

В следующей работе Хоема [4] области изображения, классифицированные как “вертикаль”, подвергаются дальнейшему разбиению на подклассы, а именно: плоские твёрдые объекты (planar solid), не плоские твёрдые объекты (non-planar solid) и пористые объекты (porous).

В [5] автор демонстрирует применимость алгоритма к фотографиям, сделанным внутри помещений (indoor images), которая достигается путём переобучения классификатора на подходящем наборе данных.

Рис. 1: Сегментация изображений алгоритмом Фелзенцвальба

2. Алгоритм

2.1. Формирование суперпикселей

Изначально изображение представляет собой обычный двумерный массив значений красного, зелёного и синего каналов (RGB). Поэтому в первую очередь, мы разбиваем исходное изображение на локально-однородные участки – **суперпиксели**. Это не только понижает размерность исходной задачи, так как нам теперь нужно назначить метку не каждому пикселю, а каждому суперпикселю, но и предоставляет больше полезной информации об изображении.

Разбивая изображение на суперпиксели, мы хотим, чтобы каждый из них представлял собой цельный объект изображения, или являлся бы частью только одного объекта, то есть из нескольких суперпикселей можно было бы составить один цельный объект. В любом случае суперпиксель считается хорошим тогда и только тогда, когда он не нарушает границ объекта.

Хоем для получения суперпикселей использует алгоритм Фелзенцвальба [1], который основан на оптимальном разбиении графа, построенного по изображению. Алгоритм является достаточно быстрым на практике, асимптотическая сложность $O(n \log n)$, и суперпиксели, получившиеся в результате его работы, хорошо "прилипают" к краям объектов. Однако этот алгоритм не предоставляет возможности контролировать ни количество суперпикселей, ни их размер (рис. 2.1). Поэтому в нашей работе мы используем алгоритм сегментации SLIC, который не только так же качественно сегментирует изображение, но и является асимптотически быстрее, $O(n)$, а так же предоставляет возможность регулировать количество суперпикселей и их рамер (рис. 2.1).

Рис. 2: Сегментация изображений алгоритмом SLIC

2.2. Вектор особенностей

После того как суперпиксели получены, каждый из них необходимо охарактеризовать. Для этого мы вычисляем следующий вектор особенностей x_i для каждого суперпикселя i:

$$x_i = \left(\overline{R}, \overline{G}, \overline{B}, \overline{X}, \overline{Y}\right) \tag{1}$$

- $\overline{R}, \overline{G}, \overline{B}$ — средние значения красного, зелёного и синего цветов суперпикселя i;
- $\overline{X}, \overline{Y}$ — среднее местоположение суперпикселя i.

2.3. Марковское случайное поле

Рассматривая задачу с вероятностной точки зрения, нам необходимо, зная векторы особенностей суперпикселей, присвоить метки w_i суперпикселям таким образом, чтобы максимизировать апостериорную вероятность $P(w_1...w_n|x_1...x_n)$.

По теореме Байеса её можно выразить следующим образом:

$$P(w_1...w_n|x_1...x_n) = \frac{\prod_{i=1}^{n} P(x_i|w_i)P(w_1...w_n)}{P(x_1...x_n)} \tag{2}$$

Предполагая, что случайные величины w_i образуют Марковское случайное поле с кликами из двух вершин, априорную вероятость можно определить следующим образом:

$$P(w_1...w_n) = \frac{1}{Z} exp\left[-\sum_{(i,j)\in C} \psi(w_i, w_j)\right] \tag{3}$$

Здесь $\psi(w_i, w_j)$ — **функция стоимости** (cost function), возвращает как положительные, так и отрицательные значения и характеризует "цену", которую мы платим назначая ту или иную метку паре соседних суперпикселей. Z назвается статсуммой (partition function), по сути это нормирующая константа.

Таким образом, чтобы разметить изображение нам нужно решить следующую задачу оптимизации:

$$\begin{aligned}
\dot{w}_1 ... \dot{w}_n &= \underset{w_1 ... w_n}{argmax} \left[P(w_1 ... w_n | x_1 ... x_n) \right] \\
&= \underset{w_1 ... w_n}{argmax} \left[\prod_{i=1}^{n} P(x_i | w_i) P(w_1 ... w_n) \right] \\
&= \underset{w_1 ... w_n}{argmax} \left[\sum_{i=1}^{n} \log(P(x_i | w_i)) + \log(P(w_1 ... w_n)) \right] \\
&= \underset{w_1 ... w_n}{argmin} \left[\sum_{i=1}^{n} -\log(P(x_i | w_i)) + \sum_{(i,j) \in C} \psi(w_i, w_j) \right] \\
&= \underset{w_1 ... w_n}{argmin} \left[\sum_{i=1}^{n} U_i(w_i) + \sum_{(i,j) \in C} B_{ij}(w_i, w_j) \right]
\end{aligned} \tag{4}$$

В энергетической нотации задача максимизации апостериорной вероятности принимает вид минимизации энергии. По смыслу $U_i(w_i)$ задаёт то, насколько данный суперпиксель соответствует тому или иному классу, а парное слагаемое $B_{ij}(w_i, w_j)$ отражает степень взаимозависимостей классов соседних суперпикселей.

Для решения нашей задачи мы используем распространённый способ введения парного слагаемого, а именно **модель Поттса**, которая "штрафует" суперпиксели с различными метками и "поощряет" с одинаковыми:

$$B_{ij}(w_i, w_j) = \begin{cases} -\gamma, & \text{если } w_i = w_j; \\ 0, & \text{если } w_i \neq w_j. \end{cases}$$

В качестве унарного слагаемого мы задали следующую функцию:

$$U(w_i) = -\alpha P(\overline{RGB}(x_i) | w_i) - \beta P(\overline{XY}(x_i) | w_i)$$

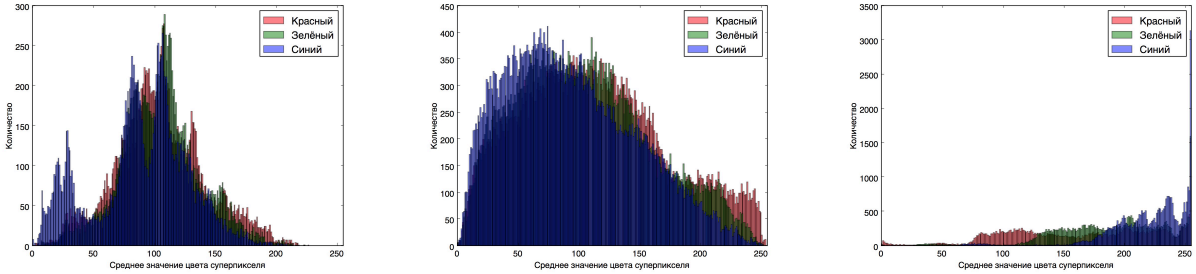
Здесь $\overline{RGB}(x_i)$ и $\overline{XY}(x_i)$ — средние значения цвета и расположения суперпикселя i соответственно. Значения параметров α, β и γ подбираются эмпирическим путём.

2.4. Минимизация энергии

Такого рода задачи можно переформулировать в терминах задач о нахождении максимального потока в графе. По теореме Форда-Фалкерсона величина этого максимального потока равна величине пропускной способности минимального разреза графа. Следовательно, мы могли бы оптимизировать энергию с помощью алгоритмов поиска минимального разреза графа, однако доказано, что при условии не бинарной сегментации (количество возможных меток больше двух) это возможно только в

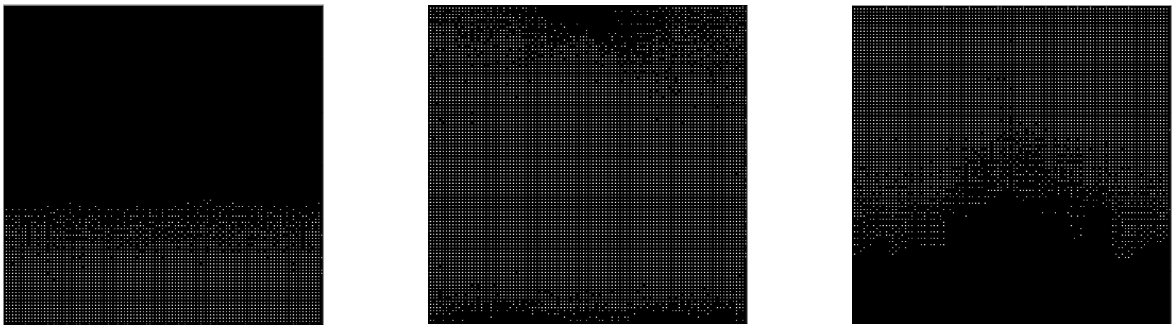
случае использования выпуклой парной функции $B_{ij}(w_i, w_j)$, иначе задача является NP-трудной [6].

Выбранная нами функция не выпукла, поэтому придётся воспользоваться приближённым методом решения задачи минимизации энергии, а именно мы используем алгоритм α**-расширения** (α-expansion) [9], который получил наибольшее распространение.



а) б) в)

Рис. 3: Распределение среднего значения цвета суперпикселей, принадлежащих классам: (а) земля (б) вертикальные объекты (в) небо



а) б) в)

Рис. 4: Распределение среднего местоположения суперпикселей, принадлежащих классам: (а) земля (б) вертикальные объекты (в) небо

3. Реализация

Для того, чтобы приблизить функции правдоподобия $P(\overline{RGB}(x_i)|w_i)$ и $P(\overline{XY}(x_i)|w_i)$ мы собрали 50 уличных фотографий, которые в сумме дают 105959 суперпикселей, разметили их и построили гистограммы распределения среднего значения цвета (рис. 3) и местоположения (рис. 4) суперпикселей в зависимости от геометрического класса.

Для того, чтобы разметить фотографии, нами было написано web-приложение на языке программирования Python 2.7 с использованием библиотеки Flask, которое позволяет сегментировать изображения алгоритмом SLIC, предоставляя пользователю удобный интерфейс для дальнешей ручной разметки.

Алгоритм, описанный в главе 2, был также реализован на языке программирования Python 2.7, с использованием библиотек Numpy, Scipy, Skimage[8]. Реализация алгоритма α-расширения была взята с сайта[2].

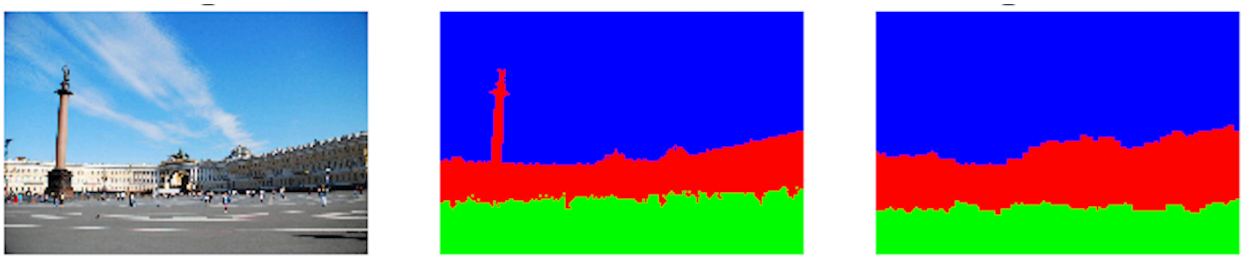


Рис. 5: Слева – исходная фотография, в центре – вручную размеченное изображение, справа – результат работы алгоритма.»

4. Тестирование

Для проведения тестирования мы обучили алгоритм на 79818 суперпикселях (40 изображений) и запустили его на 26141 суперпикселях (10 изображений). Варьируя параметры α и β унарной функции $U_i(w_i)$, было установленно, что лучшие рузельтаты достигаются при значениях 1.0 и 0.6 соответственно. Это значит, что среднее значение цвета суперпикселя оказывает большее влияние на метку суперпикселя.

При таких параметрах из общего количества суперпикселей правильно было размечено 21718, неправильно – 4423.

Точность работы алгоритма составила 82%.

Заключение

В ходе работы были изучены существующие решения в области геометрической сегментации изображения. Был сформулирован и реализован алгоритм с применением Марковского случайного поля для разметки изображений на классы ”неба”, ”вертикальных объектов” и ”земли”.

Была собрана обучающая выборка из уличных изображений, на которой алгоритм был обучен и протестирован, показав точность 82%.

В дальнейшем разработанный алгоритм может быть улучшен с помощью расширения обучающей выборки, выбора другой парной функции энергии или расширения вектора особенностей.

Список литературы

[1] Felzenszwalb P., Huttenlocher D. Efficient graph-based image segmentation. — IJCV, 2004.

[2] Group Western Computer Science. Реализация алгоритма α-расширения. — URL: `http://vision.csd.uwo.ca/code/` (дата обращения: 27.05.2015).

[3] Hoiem D., Efros A. A., Hebert M. Automatic photo pop-up. — ACM SIGGRAPH, 2005.

[4] Hoiem D., Efros A. A., Hebert M. Geometric context from a single image. — ICCV, 2005.

[5] Hoiem D., Efros A. A., Hebert M. Recovering surface layout from an image. — IJCV, 2007.

[6] M. Greig D., T. Porteous B., H. Seheult A. Exact maximum a posteriori estimation for binary images. — Journal of the Royal Statistical Society. Series B, 1989.

[7] Opportunistic use of vision to push back the path- planning horizon / B. Nabbe, D. Hoiem, A. A. Efros, M. Hebert. — Proc. IROS, 2006.

[8] Scikit-image. Collection of algorithms for image processing. — URL: `http://scikit-image.org`.

[9] Y. Boykov, O. Veksler, Zabih R. Fast approximate energy minimization via graph cuts. — IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001.