

Исследование зависимости качества сегментации алгоритмом Quick Shift от его параметров

1. Введение

Сегментацией изображения называется разбиение на области, объединенные по какому-либо признаку, например, по цвету или текстуре. В частности, семантической сегментацией изображения называется присвоение пикселям меток соответственно их семантическому значению. Семантическая сегментация является одной из наиболее сложных задач компьютерного зрения в связи с тем, что семантическая разметка изображения требует не только определить, какие объекты присутствуют на изображении, но и как можно точнее выделить контуров этих объектов. Одним из подходов к семантической сегментации является работа с суперпикселями, т.е. небольшими областями, объединяющими пиксели похожих цветов. Таким образом, в отличие от работы с отдельными пикселями мы имеем возможность оценивать локальную структуру изображения. Кроме того, представление изображения в виде нескольких сот суперпикселей, называемое пересегментацией, вместо десятков тысяч пикселей существенно упрощает последующую обработку [1].

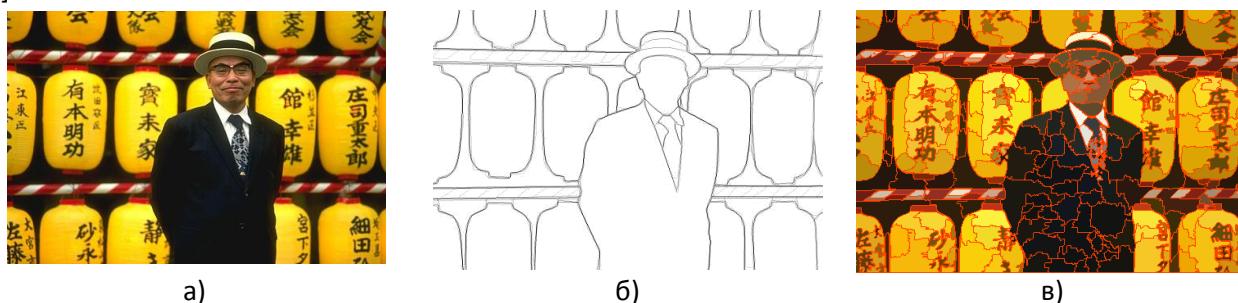


Рисунок 1. Пример изображения (а), объединения границ в эталонных сегментациях (б) и пример пересегментации алгоритмом Quick Shift (в)

Применение одной и той же конфигурации параметров алгоритма пересегментации к разным изображениям может давать совершенно разные результаты с точки зрения точности выделения границ объектов. В частности, если объекты изображения сильно отличаются по цвету друг от друга, то можно задать больший порог, определяющий разность цветов, которые можно объединять в один суперпиксель. Если же объекты похожи по цвету, то использование такого же большого порога приведет к размытию границы между объектами. Точность выделения границ может также зависеть от сложности текстур разделяемых объектов, от степени размытия границ объектов и от формы объектов [2].

Исследование проводится на примере алгоритма *Quick Shift* [3], одна из реализаций которого доступна в библиотеке *VLfeat* [4]. Этот алгоритм выполняет пересегментацию изображения как кластеризацию точек в 5-мерном пространстве: 3 компоненты цвета пикселя в пространстве *Luv* и 2 координаты пикселя на изображении. Кластеризация основывается на оценке плотности точек: там, где плотность выше, находятся центры кластеров, а точки «стекают» в центры кластеров по пути увеличения плотности. Качество сегментации оценивалось при помощи ошибки недосегментации (*under-segmentation error*):

$$U = \frac{1}{N} \left[\sum_{i=1}^M \left(\sum_{[s_j | s_j \cap g_i > 0]} |s_j| \right) - N \right] \quad (1)$$

где s_j – j -ый суперпиксель, g_i – i -ый регион эталонной сегментации, N – количество пикселей на изображении, M – количество регионов на эталонной сегментации. Именно эта метрика, как правило, используется для оценки качества пересегментации [5, 6]. Для экспериментов использовалась база изображений *Berkeley Segmentation Data Set 500 (BSDS500)* [7], состоящая из 500

изображений, к каждому из которых прилагается от 4 до 9 эталонов с выделенными границами. Эта база изображений также является общепризнанным стандартом для оценки алгоритмов подобного рода.

Задачей, которая ставится в данной работе, является изучение зависимости качества выделения границ от используемых параметров алгоритма. Была выявлена форма графика исследуемой зависимости, модель ее аппроксимации, также было исследовано, как параметры алгоритма влияют на количество суперпикселей, выявлена форма зависимости качества сегментации от количества суперпикселей. Полученные выводы могут быть использованы при подборе параметров алгоритма Quick Shift или даже для построения системы автоматической настройки этих параметров.

2. Форма зависимости

График зависимости ошибки недосегментации от значения параметров *maxdist* и *kernelsize* для одного из изображений представлен на рисунке 2а. В экспериментах значения параметра *maxdist* изменялись в промежутке от 5 до 20, а параметра *kernelsize* от 3 до 9. Выбор промежутков основывается на практической применимости пересегментаций, выполненных с параметрами из данных промежутков. Графики для большинства изображений приближенно имеют форму плоскости (2б). Для большинства графиков это достаточно хорошее приближение, о чем свидетельствует распределение ошибки аппроксимации (для оценки использовалась метрика *rmse* (*root mean square error*)), представленной на рисунке 2в. Для большинства изображений ошибка менее 0.04.

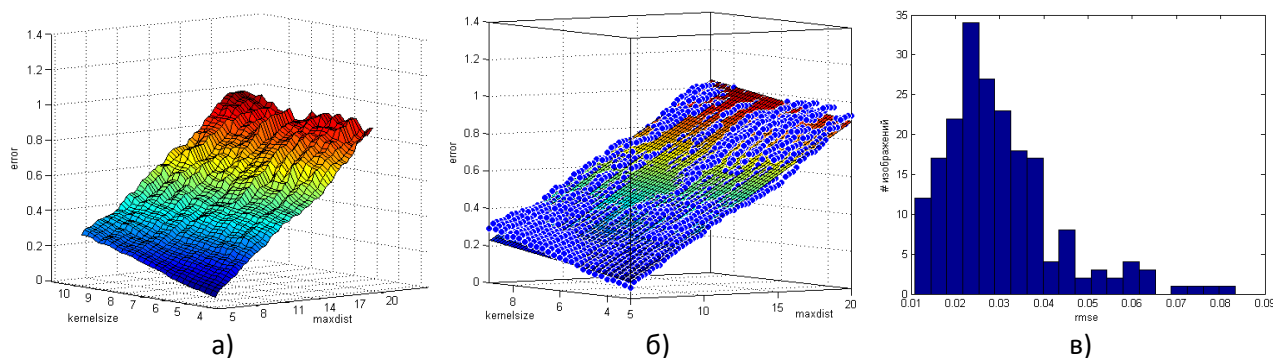


Рисунок 2. График зависимости качества сегментации от значения параметров (а), точки графика зависимости и аппроксимирующая их плоскость (б), гистограмма ошибок аппроксимации графика зависимости плоскостью (в)

Наибольшая ошибка при аппроксимации наблюдается в тех областях, где оба параметра минимальны или максимальны (рисунок 3а). Там значение ошибки соответственно резко снижается (так как происходит переход к крайне мелким суперпикселям, а значит, ошибок в определении границ практически нет) или резко возрастает (так как суперпиксели получаются настолько крупными, что такая пересегментация не имеет практического значения). Поэтому при аппроксимации допустимо игнорировать точки в этих областях. Другая ошибка аппроксимации связана с встречающейся вогнутостью графика, график с наибольшей вогнутостью представлен на рисунке 3б. Причем по параметру *maxdist* кривизна гораздо выше, чем по параметру *kernelsize*, о чем свидетельствуют данные, приведенные в таблице 1. В ней приведена средняя ошибка аппроксимации (*rmse*) в различных моделях: линейная по обоим параметрам (*poly11*), квадратичной по одному из параметров (*poly12* - *kernelsize*, *poly21* - *maxdist*), квадратичная по обоим (*poly22*). Обратите внимание, что добавление квадратичности по параметру *maxdist* (переход от *poly11* к *poly21*) гораздо больше уменьшает ошибку, чем по параметру *kernelsize* (переход от *poly11* к *poly12*).

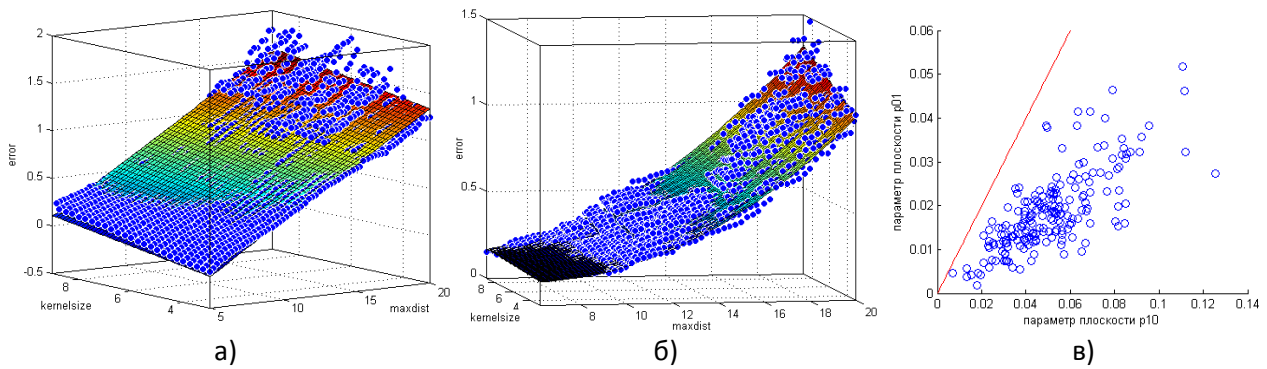


Рисунок 3. Пример зависимости, плохой аппроксимируемой плоскостью (а), существенно уменьшает ошибку введение в модель аппроксимации квадратичности по параметру *maxdist* (б). Распределение параметров аппроксимирующих плоскостей p_{10} и p_{01} по коллекции изображений (в)

Таблица 1. Средняя по коллекции изображений ошибка аппроксимации при использовании различных моделей аппроксимации

Модель аппроксимации	<i>polyX1</i>	<i>polyX2</i>
<i>poly1X</i>	0.029	0.026
<i>poly2X</i>	0.022	0.0218

Плоскость, аппроксимирующая график, представляется в виде $p_{00} + p_{10} * maxdist + p_{01} * kernelsize$. На рисунке 3в представлено распределения параметров p_{10} и p_{01} . Красным показан график $y=x$. Видно, что параметр p_{10} всегда имеет большее значение, чем p_{01} . Это говорит о большей зависимости ошибки недосегментации от параметра *maxdist*, чем от *kernelsize*. Было замечено, что параметры плоскости коррелируют с количеством сегментов на эталонах (рисунок 4). Количество сегментов характеризует «сложность изображения» для алгоритма сегментации, поэтому неслучайно при большей «сложности» быстрее растет ошибка сегментации при увеличении параметров алгоритма *maxdist* и *kernelsize*, а также выше уровень ошибки в целом (параметр плоскости p_{00}).

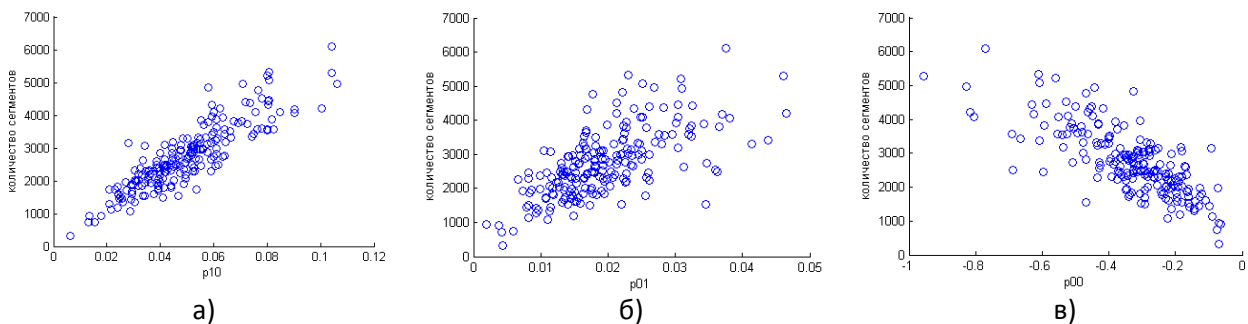


Рисунок 4. Зависимость параметров аппроксимирующих плоскостей (p_{10} (а), p_{01} (б), p_{00} (в)) от количества сегментов на эталонных сегментациях

Исследование формы зависимости подтвердило предположение о том, что параметры *maxdist* и *kernelsize* определяют, насколько точно будет искаться граница регионов изображения, причем, определяют это в разной степени. Также стоит заметить, что в контексте используемой метрики качества сегментации (ошибка недосегментации) и исследованных параметров алгоритма не имеет смысла разработка алгоритмов поиска значений параметров, при которых достигается глобальный минимум ошибки. Благодаря представленной работе подтверждено, что глобальный минимум достигается при уменьшении значения исследованных параметров, а не при какой-то особой комбинации параметров. Исключение составляют небольшие локальные минимумы (рисунок 2а), однако они не представляют большого интереса, так как выигрыш в качестве незначителен. Таким

образом, задача настройки параметров (с целью минимизации ошибки) может быть преобразована в задачу подбора таких параметров, при которых будет достигнуто заданное качество сегментации.

Обращаем внимание, что график качества сегментации похож на плоскость именно в контексте используемой метрики – ошибки недосегментации. Например, при использовании другой распространенной метрики оценки пересегментации – F1-score [7] – графики зависимости не имеют столь простую форму и однородность на разных изображениях (рисунки 5а-г). Эта ситуация еще раз подтверждает сложность выбора метрики качества сегментации: нет единой метрики, которая бы одновременно бы отслеживала все аспекты оценки сегментации, а значения разных метрик могут совершенно не коррелировать между собой.

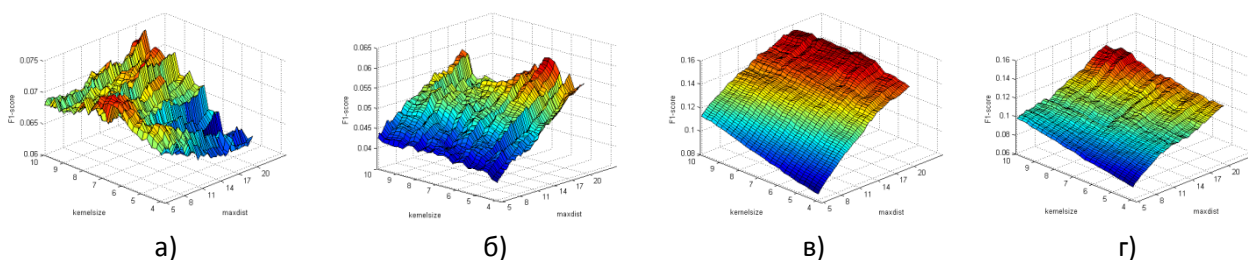


Рисунок 5. Примеры графиков зависимости ошибки пересегментации по метрике F1-score от параметров алгоритма Quick Shift (а-г)

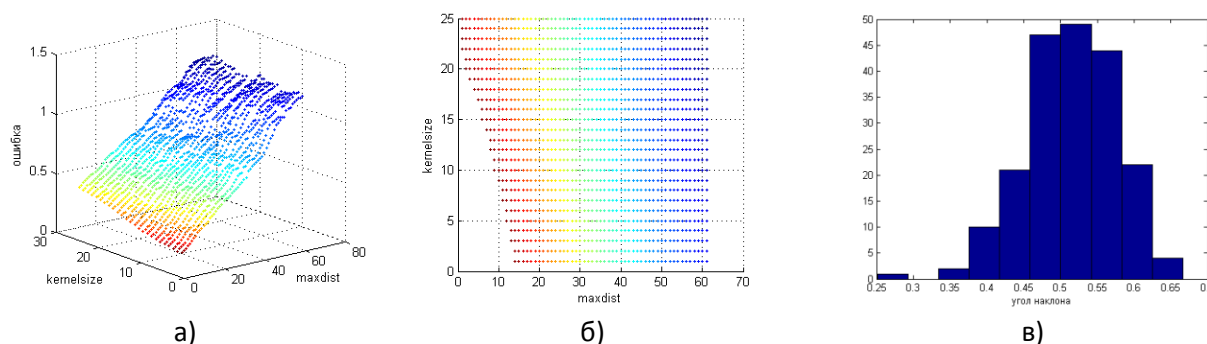


Рисунок 6. График зависимости ошибки от параметров, но здесь цвет точки показывает количество суперпикселей (а). Удалены точки, соответствующие пересегментациям с числом суперпикселей более 1000. Получаем «срез» графика (б). Гистограмма наклона «среза» в коллекции изображений (в)

3. Взаимосвязь ошибки недосегментации и количества суперпикселей

Помимо исследования зависимости ошибки недосегментации было также исследовано, как влияют параметры алгоритма на количество суперпикселей. На рисунке 6а цветом точек показан логарифм количества суперпикселей (логарифм взят для удобства отображения, так как при увеличении значений параметров число пикселей растет все быстрее и быстрее). Можно увидеть, что определенное количество суперпикселей можно получить несколькими конфигурациями параметров. Чтобы понять, как выглядит этот набор параметров, дающих примерно одинаковое количество суперпикселей, мы отсекали от графика точки, соответствующие количеству суперпикселей выше заданного порога (1000 суперпикселей) и исследовали форму и направление получившегося «среза» графика (рисунок 6б). Форма среза хорошо аппроксимируется прямой линией. Пусть прямая задается уравнением $maxdist = p1 * kernel size + p0$. Параметр прямой $p1$, как правило, лежит в пределах от 0.4 до 0.6, о чем свидетельствует гистограмма распределения этого параметра прямой (рисунок 6в). Параметр $p1$, равный, например, 0.5, говорит о том, что увеличение значение параметра $maxdist$ на 1 даст такое же изменение количества суперпикселей как увеличение параметра $kernel size$ на 2. По параметру прямой $p0$ можно судить о том, как много на данном

изображении будет суперпикселей. Например, для изображений 7а и 7б нужно гораздо больше суперпикселей (большее значение параметра прямой ρ_0), чтобы выделить детали присутствующих объектов, нежели для изображений 7в и 7г.



Рисунок 7. Изображения а) и б) требуют большего количества суперпикселей для выделения, нежели изображения в) и г)

Мы также исследовали взаимосвязь количества суперпикселей и ошибки недосегментации. График зависимости ошибки недосегментации от количества суперпикселей представлен на рисунке 8. График представляет собой набор точек, представляющих пересегментации с разными комбинациями параметров. Точки с одинаковым цветом означают пересегментации, выполненные с одинаковым значением $kernelsize$. Видно, что точки выстраиваются практически в одну линию. Для части изображений происходит «расслоение» графика, что свидетельствует о большей чувствительности этого изображения к ширине окна $kernelsize$ (при увеличении ширины окна ошибка растет быстрее, чем на других изображениях). Пример наибольшего «расслоения» представлен на рисунке 8б. Тем не менее, для большинства графиков такое «расслоение» не наблюдается или гораздо меньше, чем на рисунке 8б. Для тех изображений, где точки выстраиваются в достаточно узкий коридор, можно сказать, что качество сегментации задается только количеством суперпикселей и не зависит от конфигурации параметров, при которой это количество суперпикселей было получено. Таким образом, для большинства изображений отпадает задача подбора параметров, минимизирующих ошибку недосегментации, при заданном количестве суперпикселей, так как количество суперпикселей почти однозначно задает ошибку недосегментации.

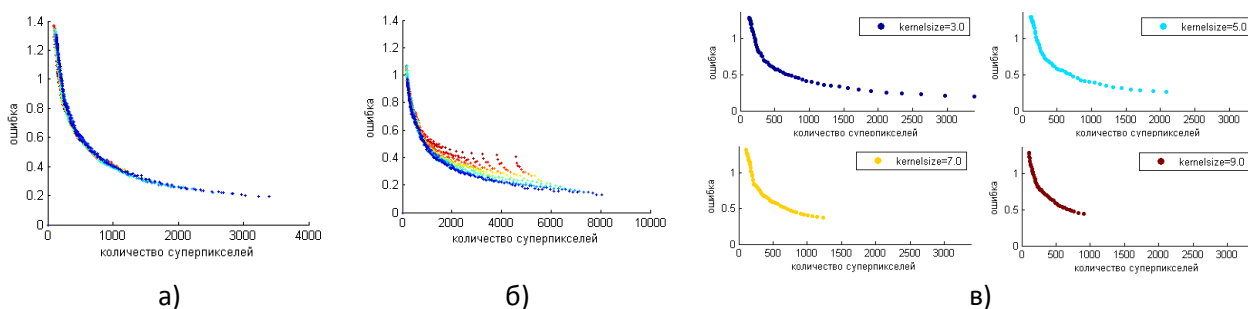


Рисунок 8. Графики зависимости ошибки недосегментации от количества суперпикселей (а-б), график зависимости, изображенный на ба, при фиксированных значениях параметра $kernelsize$ (в)

Заключение

В представленной работе была исследована зависимость ошибки недосегментации, выполненной алгоритмом Quick Shift, от параметров $maxdist$ и $kernelsize$ этого алгоритма. Были построены графики исследуемой зависимости для базы изображений BSDS. Предложена модель аппроксимации этой зависимости, оценена ошибка аппроксимации, а выполнено сравнение с более сложными моделями аппроксимации. Было выяснено, что для большинства изображений ошибка недосегментации однозначно определяется числом суперпикселей. Полученные выводы изменили представление о задаче подбора параметров. Задачей последующей работы является прогнозирование зависимости ошибки от параметров. Для этого предполагается найти набор характеристик изображения, по которым можно построить график этой зависимости и по графику

определить такие значения параметров, при которых не превышаетеcя выбранный порог ошибки и минимизируется количество суперпикселей.

Список литературы

1. Ren X., Malik J. Learning a classification model for segmentation // IEEE 9th International Conference on Computer Vision (ICCV). Nice, France. 14-17 October, 2003. Vol. 1. P. 10-17.
2. Колосовский М., Крючкова Е. Настройка параметров алгоритма сегментации изображений QuickShift // Программная инженерия. №5, 2013, сс. 11-20.
3. Fulkerson B., Soatto S. Really quick shift: Image segmentation on a GPU // Workshop on Computer Vision using GPUs (CVGPU). Crete, Greece. 10-11 September, 2010. P. 350-358.
4. Vedaldi A., Fulkerson B. VLFeat: An open and portable library of computer vision algorithms // 18th Annual ACM International Conference on Multimedia (ACM MM). Firenze, Italy. 25-29 October, 2010. P. 1469-1472.
5. Achanta R., Shaji A., Smith K. etc. SLIC Superpixels, EPFL Technical Report no. 149300, June 2010.
6. Levinshtein A., Stere A., Kutulakos K. etc. TurboPixels: Fast Superpixels Using Geometric Flows. TPAMI 2009 (vol. 31, no. 12).
7. Arbelaez P., Maire M., Fowlkes C., Malik J. Contour Detection and Hierarchical Image Segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). May, 2011. Vol. 33, No. 5. P. 898-916.