

Детектирование жирных линий и древовидных структур на изображениях

А.Е. Левашов¹, Д.В. Юрин²

Лаборатория математических методов обработки изображений³,
Факультет вычислительной математики и кибернетики

Московский государственный университет им. М. В. Ломоносова, Россия
¹alexeylevashov89@gmail.com, ²yurin@cs.msu.su, ³<http://imaging.cs.msu.su>

Аннотация

Предлагается новый многомасштабный детектор жирных линий с улучшенным методом немаксимального подавления в пространстве переменных разрешений. Используя результаты работы детектора как предварительную разметку, методом адаптивного выбора порога и заливки производится дополнение линий в точках ветвления, установление верного соединения разорванных линий и уточнение границ жирных линий. Алгоритм тестируется на синтетических, ландшафтных и медицинских изображениях.

Ключевые слова: Ridge detection, feature detection, scale space.

1. ВВЕДЕНИЕ

Существует несколько основных алгоритмов для нахождения жирных линий. По-видимому, самым первым из них был подход [2], он не использует многомасштабного анализа и, как следствие, детектирует линии только небольшой толщины, что является очень серьезным ограничением. Подход [5] основан на многомасштабном анализе, однако он представляет собой скорее хорошее теоретическое обоснование для многомасштабного подхода, чем практически эффективный алгоритм. За истекшие 15 лет было немного случаев его использования в научных исследованиях. Детектор [3], основанный на обобщении процедуры немаксимального подавления на случай трехмерного пространства переменных разрешений (x, y, t) представляется более практичным для поиска жирных линий на изображениях. Однако одним из важных недостатков детектирования жирных линий при таком подходе является пропуск точек ветвления таких линий и потеря самих линий в окрестности точек ветвления. Это нарушает связность линий и существенно усложняет анализ древовидных структур на изображениях. Примером таких задач могут быть выделение веток и проводов на любительских фотографиях с целью их ретуширования, выделение дорожной и речной сети на аэрокосмических изображениях с целью актуализации картографической информации, выделение кровеносных сосудов на медицинских изображениях и обработка трехмерных изображений компьютерной томографии с целью диагностики заболеваний. Во всех этих задачах точки ветвления играют важную роль, однако, в окрестности точек ветвления жирных линий детектор часто ошибочно не детектирует такие линии по следующим причинам. Во-первых, для процедуры немаксимального подавления возникает неопределенность выбора направления, т.к. оба собственных значения Гессiana оказываются близки. Во-вторых, в окрестности точки ветвления две (или более) жирные линии близко сходятся, поэтому оператор Лапласа может дать максимальный отклик на существенно большем масштабе (в два и более

раза, что приводит к разрыву линий). В-третьих, отклик может снижаться из-за нарушения математической модели согласованной фильтрации - линия одного цвета в пределах центрального лепестка второй производной функции Гаусса, и фон другого цвета в боковых лепестках. Целью данной статьи является устранение описанных недостатков в подходе [3], что приведет к существенному улучшению качества и упрощению дальнейшего анализа.

2. ДЕТЕКТИРОВАНИЕ ЖИРНЫХ ЛИНИЙ

Рассмотрим алгоритм [3], пусть $I(x, y)$ – яркость исходного дискретного изображения в оттенках серого. Определим

$$L(x, y, t) = G(x, y, t) * I(x, y),$$

$$G(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}, \quad t = \sigma^2, \quad (1)$$

где $*$ – есть операция свертки, а $G(x, y, t)$ – функция Гаусса. Под $L_x, L_y, L_{xx}, L_{xy}, L_{yy}, \dots$ будем понимать свертки изображения $I(x, y)$ с соответствующей производной функции Гаусса. Непосредственным дифференцированием легко показать [6], что многомасштабное представление изображения $L(x, y, t)$ удовлетворяет уравнению

$$L_t = \frac{1}{2} (L_{xx} + L_{yy}), \quad (2)$$

где производные взяты на масштабе t .

3. ДЕТЕКТИРОВАНИЕ ЖИРНЫХ ЛИНИЙ С ВЫБОРОМ ОПТИМАЛЬНОГО МАСШТАБА

Рассмотрим алгоритм [3] для детектирования жирных линий с пропуском точек ветвлений. Пусть толщина линии фиксирована и равна 2σ , Тогда фильтром для поиска жирных линий может служить оператор Лапласа или наибольшее собственное значение матрицы Гессе \mathbf{H}_{22} в точке (x, y)

$$\mathbf{H}_{22} = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix}, \quad (3)$$

где производные взяты на масштабе $t = \sigma^2$. В каждой точке (x, y) собственные вектора $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2)$ будут показывать

главные направления кривизны: собственный вектор V_1 , соответствующий наибольшему по модулю собственному числу λ_1 , будет направлен перпендикулярно касательной к линии в этой точке. Исходя из этого, получается алгоритм детектирования центральных точек жирных линий [3] как нахождение локальных максимумов и локальных минимумов величины λ_1 по направлению V_1 . Т.е. двукратным выполнением процедуры немаксимального подавления по окну 3×3 пикселя. Если масштаб детектирования t не вполне соответствует ширине линии, то очевидно, что Лапласиан и λ_1 будут меньше, чем при точном совпадении, поэтому анализируя отклик фильтра на соседних масштабах можно определить, детектирована ли линия на своем истинном масштабе. В пространстве переменных разрешений матрица Гессе имеет вид:

$$\mathbf{H}_{33} = \begin{pmatrix} L_{xx} & L_{xy} & L_{xt} \\ L_{xy} & L_{yy} & L_{yt} \\ L_{xt} & L_{yt} & L_{tt} \end{pmatrix}, \text{ где с учётом (2)}$$

$$L_{xt} = \frac{1}{2}(L_{xxx} + L_{xyy}), \quad L_{yt} = \frac{1}{2}(L_{xxy} + L_{yyy}),$$

$$L_{tt} = \frac{1}{4}(L_{xxxx} + 2L_{xxyy} + L_{yyyy})$$

В [3] предлагается выполнять модифицированную процедуру немаксимального подавления (NMS) сразу по двум направлениям в трехмерном пространстве переменных разрешений [6] (x, y, t) и окну $3 \times 3 \times 3$. В качестве направлений, по которым будет делаться NMS, выбирались проекции собственных вектора \bar{V}_1 и \bar{V}_2 , соответствующих двум максимальным собственным числам (λ_1, λ_2) матрицы \mathbf{H}_{33} на плоскость, задаваемую собственным вектором \bar{V}_1 матрицы \mathbf{H}_{22} и осью t . Дискретный шаг по масштабам $\sigma_0, \dots, \sigma_i, \dots, \sigma_n$ в [3] выбирался как и в [7]: $\sigma_0 = 1, \sigma_i = \sigma_{i-1} \sqrt[4]{2}$. Для описанной трехмерной процедуры немаксимального подавления достаточно держать в памяти 3 слоя, соответствующих трем последовательным масштабам, шаг за шагом накапливая 3-х мерные локальные максимумы.

Описанный выше метод [3] будет также детектировать еще и особенности типа пятно (blob), которые не являются строго круглыми, и точки контурных линий (edge).

В настоящей работе для отсекация пятен используется векторная обработка линий. После сборки цепочки связанных пикселей (локальных максимумов), они объединяются в граф. Цепочки, не связанные ни с какими другими, в которых число пикселей не велико, отбрасываются. Минимальное количество точек в цепочке определяется пороговой константой, зависящей от ширины линии, что позволяет установить явно минимальную вытянутость пятна, которое считается линией.

Для подавления точек контурных линий в [3] при выполнении немаксимального подавления отбрасываются точки не

прошедшие дополнительный тест на изменение знака скалярного произведения градиентов в соседних точках:

$$(\bar{\mathbf{g}}(\bar{\mathbf{x}} + \bar{\mathbf{v}}_1), \bar{\mathbf{g}}(\bar{\mathbf{x}} - \bar{\mathbf{v}}_1)) < 0. \quad (5)$$

Наши эксперименты показали, что условие (5) может приводить к ошибочному отбрасыванию осевых точек жирной линии в случае разного цвета фона по её разные стороны и в точках сильной извилистости жирной линии. В настоящей работе предлагается модифицированное условие

используется специальная процедура подавления граничных точек. Детектированные на масштабе σ_i точки с координатами $\bar{\mathbf{X}} = (x, y)$, в которых не происходит смена знака градиента вдоль направления V_1 , остаются, если для них выполняется следующее условие:

$$(\bar{\mathbf{g}}(\bar{\mathbf{x}} + \sigma_i \bar{\mathbf{v}}_1), \bar{\mathbf{g}}(\bar{\mathbf{x}} - \sigma_i \bar{\mathbf{v}}_1)) < T_g \quad (6)$$

где σ_i соответствует текущему масштабу, пороговая величина удовлетворяет условию

$$-1 < T_g < 0, \quad (7)$$

а $\bar{\mathbf{g}} = (L_x, L_y)^T$ – градиент яркости изображения.

4. АДАПТИВНАЯ ЗАЛИВКА

Пусть после предыдущего шага мы получили набор детектированных линий, которые представлены в виде множества точек $S = \{(\bar{\mathbf{x}}_j, \sigma_j, l_j)\}$, где $\bar{\mathbf{x}}_j$ – координаты центральных точек жирных линий, σ_j – масштаб, на котором была обнаружена точка жирной линии в координатах $\bar{\mathbf{x}}_j$, l_j – величина отклика признака в этой точке (отклик от свертки с лапласианом, взятым на масштабе $\bar{\mathbf{x}}_j$, нормированный на единицу). По этим точкам построим 3 изображения M, I_{\min}, I_{\max} . Будем считать, что $M(\bar{\mathbf{x}}) = 1$ если пиксель принадлежит жирным линиям, и $M(\bar{\mathbf{x}}) = 0$ иначе. I_{\min}, I_{\max} – дополнительные изображения, показывающие допустимые границы изменения интенсивности у жирных линий, по которым будет происходить заливка. Пусть изначально во всех точках $M(\bar{\mathbf{x}}) = 0$, далее для всех $\bar{\mathbf{x}}_j \in S$ положим

$$\begin{aligned} M(\bar{\mathbf{x}}_j) &= 1, \\ I_{\max}(\bar{\mathbf{x}}_j) &= I(\bar{\mathbf{x}}_j) + 0.5|l_j|, \\ I_{\min}(\bar{\mathbf{x}}_j) &= I(\bar{\mathbf{x}}_j) - 0.5|l_j|. \end{aligned} \quad (8)$$

Здесь учтено, что как показано в [3], величина Лапласиана на масштабе, соответствующем ширине линии, равна разнице средних яркостей фона и жирной линии (а в случае цветного

изображения по-канальные значения Лапласианов формируют вектор вариации цвета).

Далее будем проводить заливку изображения M . Предлагаемый алгоритм заливки представляет собой алгоритм flood fill [1] с дополнительными условиями на закраску пикселей.

Рассмотри два соседних пикселя \bar{y}_k и \bar{y}_j - такие, что

$$M(\bar{y}_k) = 1, M(\bar{y}_j) = 0. \text{ Если выполняется условие}$$

$$I_{\min}(\bar{y}_k) < I(\bar{y}_j) < I_{\max}(\bar{y}_k), \quad (9)$$

то будем закрашивать пиксель \bar{y}_j следующим образом:

$$M(\bar{y}_j) = 1,$$

$$I_{\min}(\bar{y}_j) = I_{\min}(\bar{y}_k), \dots$$

$$I_{\max}(\bar{y}_j) = I_{\max}(\bar{y}_k)$$
(10)

а алгоритм адаптивной заливки примет вид:

Алгоритм AdaptiveFloodFill(\bar{x}, \bar{d})

1. **if** $M(\bar{x}) = 1$ **return**
2. **if** $NOT(I_{\min}(\bar{x}) < I(\bar{x} + \bar{d}) < I_{\max}(\bar{x}))$ **return**
3. $M(\bar{x} + \bar{d}) = 1$
 $I_{\min}(\bar{x} + \bar{d}) = I_{\min}(\bar{x})$
 $I_{\max}(\bar{x} + \bar{d}) = I_{\max}(\bar{x})$
4. **AdaptiveFloodFill**($\bar{x} + \bar{d}, (0 \ 1)$)
5. **AdaptiveFloodFill**($\bar{x} + \bar{d}, (1 \ 0)$)
6. **AdaptiveFloodFill**($\bar{x} + \bar{d}, (0 \ -1)$)
7. **AdaptiveFloodFill**($\bar{x} + \bar{d}, (-1 \ 0)$)

В самом начале алгоритма для всех соседей $\bar{x}_i + \bar{d}$ точек для всех $\bar{x}_i : M(\bar{x}_i) \neq 0$ вызываем алгоритм **AdaptiveFloodFill**(\bar{x}_i, \bar{d}), который отличается от классического [1] строками 2 и 3. В предлагаемом алгоритме дополнительно необходимо обрабатывать ситуацию, когда $\bar{x} + \bar{d}$ оказывается за границами изображения. Это схема самой простой, не оптимальной, рекурсивной реализации алгоритма заливки. Более быстрый алгоритм можно найти в [1], который также может быть адаптирован для решения поставленной задачи путем введения условия закраски пикселя (9), (10).

5. СКЕЛЕТИЗАЦИЯ ЖИРНЫХ ЛИНИЙ

Бинарное изображение M после предыдущего шага будет сегментировать изображение на пиксели, которые принадлежат жирным линиям, и на фоновые пиксели. В том случае,

когда в точках ветвления жирные линии мало меняют свой цвет по сравнению с фоном, все стыки и точки ветвления будут включены в состав жирных линий, даже если они не были найдены многомасштабным детектором. Далее можно воспользоваться либо непрерывной скелетизацией бинарного изображения [11], и в качестве результата брать скелет как центры жирных линий. Либо воспользоваться пиксельным алгоритмом утончения [9]. В нашем подходе используется последний вариант, т.к. при таком подходе можно оставлять уже найденные многомасштабным детектором линий и добавлять только стыки между ними как результат утончения.

Алгоритм выделения скелета через утончение [9] заключается в последовательном удалении точек, не являющихся осевыми в линиях. Каждая итерация алгоритма [9] состоит из двух проходов по всем пикселям. При каждом проходе точки, которые не являются скелетными, помечаются, но удаляются только после окончания прохода. Вокруг каждой точки \bar{x} изображения M берется окно 3×3 пикселя P_1, P_2, \dots, P_9 в последовательности, изображенной на Рис. 1, и вычисляются характеристики $B(\bar{x}) = \sum_{i=2}^9 P_i$ и $A(\bar{x}) =$ количеству паттернов 01 в последовательности P_2, \dots, P_9, P_2 .

P_9 ($i-1, j-1$)	P_2 ($i-1, j$)	P_3 ($i-1, j+1$)
P_8 ($i, j-1$)	P_1 (i, j)	P_4 ($i, j+1$)
P_7 ($i+1, j-1$)	P_6 ($i+1, j$)	P_5 ($i+1, j+1$)

Рисунок 1: Расположение пикселей в окне 3×3

На первом проходе пиксель \bar{x} помечается для удаления из изображения M , если выполняется все следующие условия:

- a) $2 \leq B(\bar{x}) \leq 6$
- b) $A(\bar{x}) = 1$
- c) $M(P_2) \cdot M(P_4) \cdot M(P_6) = 1$
- d) $M(P_4) \cdot M(P_6) \cdot M(P_8) = 1$

На втором проходе пункты c) и d) заменяются на следующие:

- c') $M(P_2) \cdot M(P_4) \cdot M(P_8) = 1$
- d') $M(P_2) \cdot M(P_6) \cdot M(P_8) = 1$

Все остальное остается без изменений. Итерации делаются до тех пор, пока в двух проходах не будет удалено ни одного пикселя.

Если не удалять пиксели, которые были найдены многомасштабным алгоритмом, тогда все изначально найденные линии останутся, и к ним добавятся только стыки между ними.

6. ДЕТЕКТИРОВАНИЕ ТОЧЕК ВЕТВЛЕНИЯ

После того, как потерянные точки ветвления восстановлены, рассмотрим алгоритм их детектирования. Такой алгоритм был описан в [10] как часть этапа векторизации.

Как и в предыдущем пункте будем делать итеративный проход по всем пикселям, и рассматривать окрестность 3×3 каждой точки. Будем считать, что точка \bar{X} является узлом, если для нее выполняется следующее условие:

$$B(P_1) \geq 5 \vee A(P_1) > 2.$$

Точки помечаются как узел и удаляются вместе с ее окрестностью 3×3 из M сразу при сканировании и запоминаются в другом изображении. Процесс повторяется пока удастся найти хотя бы один узел.

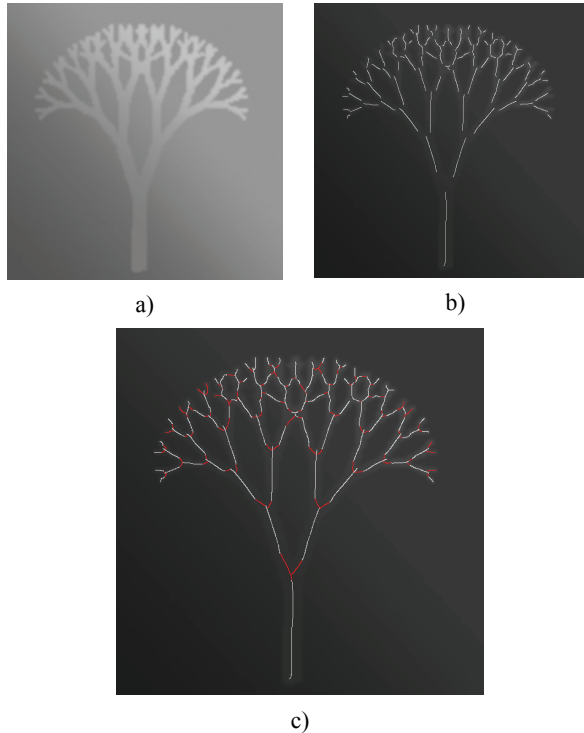


Рисунок 2: а) Исходное изображение, б) детектирование жирных линий, с) результат сшивки жирных линий

При сборке пикселей в списки связанных точек [4, 8] определяется их соседство с найденными узлами. Данные векторизуются и хранятся как граф.

7. ЗАКЛЮЧЕНИЕ

Предложен алгоритм, позволяющий детектировать жирные линии без разрывов в точках ветвлений и выполнять их векторизацию. Разработан алгоритм адаптивной заливки, который показал хорошие результаты сшивки линий на синтетических и медицинских изображениях, результат работы представлен на Рис. 2. Из недостатков стоит отметить, что адаптивная заливка может не находить точных границ жирных линий. Точные границы планируется находить через минимальный разрез на графе после того, как адаптивная заливка сшила линии одной структуры, что упрощает задачу разметки до двухполюсной задачи с помеченными точками, относящимися к приемнику и источнику максимального потока.

8. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ 13-07-00584.

9. ЛИТЕРАТУРА

- [1] S.V. Burtsev, Y.P. Kuzmin. "An efficient flood-filling algorithm" // Computers & graphics, Elsevier, -V. 17, -No 5, -P. 549–561, 1993
- [2] R. Haralick, "Ridges and Valleys on Digital Images" // Computer Vision, Graphics, and Image Processing, -V. 22, -No. 10, 1983
- [3] N. A. Khanina, E. V. Semeikina, D. V. Yurin. "Scale-space color blob and ridge detection" // Pattern Recognition and Image Analysis, -No. 1, -P. 221–227, 2012
- [4] M. Klaiber, L. Rockstroh, Z. Wang, Y. Baroud, S. Simon (2012). "A Memory-Efficient Parallel Single Pass Architecture for Connected Component Labeling of Streamed Images" // Field-Programmable Technology (FPT), International Conference, -P. 159–165, 2012
- [5] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection" // International Journal of Computer Vision, -V. 30, -No. 2, -P. 117–154, 1998
- [6] T. Lindeberg "Scale-Space Theory in Computer Vision" // Kluwer Academic Publishers/Springer, Dordrecht, Netherlands, 1994.
- [7] D.G. Lowe. "Distinctive image features from scale-invariant keypoints" // Int. Journal of Computer Vision, -V. 60, -No. 2, -P. 91–110, 2004
- [8] B. Michael, Dillencourt and Hannan Samet and Markku Tamminen (1992). "A general approach to connected-component labeling for arbitrary image representations" // Journal of the ACM, -V. 39, -N. 2, -P. 253–280, 1992
- [9] T.Y. Zhang, C.Y. Suen "A fast parallel algorithm for thinning digital patterns" // Communications of the ACM, -V. 27, -No. 3, -P. 236–239, 1984.
- [10] А.Е. Левашов, Д.В. Юрин "Система быстрого обнаружения параметрических кривых на серых и цветных изображениях с контролем достоверности" // 21-я международная конференция по компьютерной графике и зрению GraphiCon'2011. Москва, Россия, -с. 212–215, 2011
- [11] Л.М. Местецкий "Непрерывный скелет бинарного растрового изображения" //Труды международной конференции Графикон-1998, Москва, -С. 71-78.

Об авторах



Левашов Алексей Евгеньевич – аспирант кафедры математической физики, факультета вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова.

alexeylevashov89@gmail.com



Юрин Дмитрий Владимирович, к.ф.-м.н., с.н.с. лаборатории математических методов обработки изображений, факультета вычислительной математики и кибернетики Московского государственного университета им. М.В.Ломоносова. yurin_d@inbox.ru