ТЕЛЕКОММУНИКАЦИИ: СЕТИ И ТЕХНОЛОГИИ, АЛГЕБРАИЧЕСКОЕ КОДИРОВАНИЕ И БЕЗОПАСНОСТЬ ДАННЫХ

УДК 621.391

ДЕТЕКТИРОВАНИЕ ПРЯМЫХ ЛИНИЙ НА ОСНОВЕ ФОРМ-ФАКТОРА

О.Г. ШЕВЧУК

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь

Поступила в редакцию 10 ноября 2016

Предложен метод детектирования прямых линий на основе преобразованного форм-фактора. Проведен анализ метода по сравнению с известными алгоритмами. Осуществлен выбор параметров для разработанного метода. Показано, что метод устойчив к изменению поворота и незначительному искривлению линии.

Ключевые слова: форм-фактор, детектирование, прямая линия, нормализация.

Введение

Основными задачами при параметризации и идентификации ключевых объектов на изображениях является их детектирование. Ключевым объектом на изображении могут выступать: точка (пиксел), линия или сегмент изображения.

Несмотря на то, что большинство методов идентификации и параметризации в качестве ключевых объектов используют точки (пиксели), прямые линии обладают значительно большей устойчивостью по сравнению с ними. Для выделения прямых линий на изображениях широко используются методы, основанные на преобразовании Хафа [1], масочном поиске [2], вычислении градиента [3] и квантовании по ориентации [4]. Однако данные методы не всегда устойчивы к повороту, масштабу и изменению кривизны линии. Устранить данный недостаток можно за счет использования методов детектирования линий на основе форм-фактора [5].

Метод детектирования прямых линий на основе преобразованного форм-фактора

Предлагается метод детектирования прямых линий на основе преобразованного форм-фактора (NFFLD – New Form-Factor Line Detection). Данный метод отличается от известного метода выделения изолированных прямых линий на основе форм-фактора [5] расчетом длины контурной линии, определяемой суммой образующих ее контурных точек.

Исходными данными для метода является полутоновое изображение $I = \left\| i(y, x) \right\|_{\left(y = \overline{0,Y-1},\ x = \overline{0,X-1}\right)}$, где $i(\mathrm{y},\mathrm{x}) = \overline{0,255}$ – значение пиксела на изображении; $y = \overline{0,Y-1}, x = \overline{0,X-1}$ – координаты пиксела изображения; Y, X – размеры изображения по вертикали и горизонтали.

Алгоритм метода выделения прямых линий на изображении на основе форм-фактора основан на методе FFLD [5] и состоит из следующих шагов.

Шаг 1. Контурная фильтрация изображения. В результате контурной фильтрации на основе исходного изображения I формируется бинарная матрица $P = \left\| p(x, y) \right\|_{\left(y=\overline{0,Y-1},x=\overline{0,X-1}\right)}$ образов выделенных контуров, где $p(x, y) = \{0,1\}$ – значение пиксела в бинарной матрице образов.

Шаг 2. Сегментация контуров. Сегментация матрицы P осуществляется методом выращивания областей (Region Growing – RG) [6]. В результате формируется матрица сегментации $I_S = \left\| i_S(y, x, n) \right\|_{\left(y=\overline{0,Y-1},\, x=\overline{0,X-1}\right)}$, где $n = [1, N_s]$, N_s – число выделенных контуров.

Шаг 3. Нормализация контурных линий. Для утончения линий используется метод нормализации контурных линий по толщине на основе масочного анализа локальных ориентаций их фрагментов [7]. В результате выполнения метода происходят поиск и удаление избыточных пикселов из матриц I_S и P.

Шаг 4. Классификация выделенных контуров по числу концевых точек. Для каждого сегментированного контурного элемента $l(n) = \left\{ X(n), Y(n) \right\}_{\left(n=\overline{1,N_S}\right)}$, образованного пикселами $p(y, x) = 1$, для которых соответствующие пикселы $i_S(y, x, n) = n$, определяется число концевых точек, где $X(n) = \left\{ x(i) \right\}_{i=\overline{0,K-1}}$, $Y(n) = \left\{ y(i) \right\}_{i=\overline{0,K-1}}$ – координаты n-го контура; K – количество пикселов в n-м контуре; $n = \overline{1, N_S}$ порядковый номер линии; N_S – количество выделенных линий. Концевой точкой является единичный пиксел $p(y, x)$ бинарной матрицы P образов, который имеет только один соседний единичный пиксел $p(y + j, x + q)$, где $j = \overline{-1,1}$, $q = \overline{-1,1}$ и $j + q \neq 0$.

Шаг 5. Выделение сегментированных контурных элементов с двумя концевыми точками, линий. Из множества $L_S = \left\{ l_S(n) \right\}_{\left(n=\overline{1,N_S}\right)}$ сегментированных контурных элементов, включающего замкнутые, пересекающиеся и разомкнутые контурные элементы, выделяется подмножество $L_2 = \left\{ l_2(n) \right\}_{\left(n=\overline{1,N_2}\right)}$ контурных линий, которые имеют две концевые точки, где N_2 – число сегментированных контурных линий с двумя концевыми точками. Как правило, пересекающиеся контурные линии имеют более двух концевых точек, поэтому на данном шаге алгоритма из множества контурных элементов выделяются в основном контурные линии, не имеющие пересечений. Исключением является пересечение замкнутого и разомкнутого контуров. Такое исключение обрабатывается на следующих шагах алгоритма.

Шаг 6. Определение длин сегментированных контурных линий. Для каждой контурной линии $l_2(n) = \left\{ X(n), Y(n) \right\}_{\left(n=\overline{1,N_2}\right)}$, имеющей две концевые точки, определяется число $s_2(n)$ образующих данную линию контурных пикселов $p(y, x)$ с помощью выражения

$$s_2(n) = \sum_{i=0}^{K-2} \sqrt{\left(x(i) - x(i+1)\right)^2 + \left(y(i) - y(i+1)\right)^2}.$$

Число $s_2(n)$ интерпретируется как длина контурной линии.

Шаг 7. Определение размеров сегментированных контурных линий. Для каждой контурной линии $l_2(n)$, имеющей две концевые точки, вычисляется расстояние $r_2(n)$ между концевыми точками (размер контурной линии) с помощью выражения

$$r_2(n) = \sqrt{\left(y_1(n) - y_2(n)\right)^2 + \left(x_1(n) - x_2(n)\right)^2},$$

где $\left(y_1(n), x_1(n)\right), \left(y_2(n), x_2(n)\right)$ – координаты концевых точек контурной линии $l_2(n)$.

Шаг 8. Вычисление форм-фактора линии. Для каждой контурной линии $l_2(n)$, имеющей две концевые точки, вычисляется значение форм-фактора $f_2(n)$ линии с помощью выражения

$$f_2(n) = r_2(n) / s_2(n).$$

Шаг 9. Анализ форм-фактора линии. Для каждой контурной линии $l_2(n)$, имеющей две концевые точки, значение ее форм-фактора $f_2(n)$ сравнивается с единицей. Если $f_2(n)$ близко

к единице, то контурная линия $l_2(n)$ является прямой. В противном случае $\left(f_2(n)<1\right)$ – линия является кривой или ломаной. Близость значения $f_2(n)$ линии к единице оценивается в результате сопоставления заданного порога T_c, определяющего кривизну линии, с модулем разности, вычисляемым с помощью выражения $\left|1-f_2(n)\right|$. Контурная линия $l_2(n)$ является прямой, если $\left|1-f_2(n)\right| \le T_c$. В результате выполнения данного шага алгоритма из подмножества L_2 выделяется подмножество $L_D = \left\{l_D(n)\right\}_{(n=\overline{1,N_D})}$ изолированных прямых контурных линий, где N_D – число изолированных прямых контурных линий.

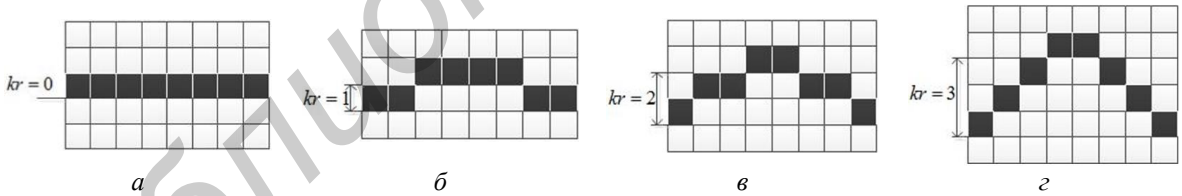
Шаг 10. Окончание алгоритма. Фиксируется информация о найденных изолированных прямых контурных линиях (координаты $\left(y_1(n), x_1(n)\right)$, $\left(y_2(n), x_2(n)\right)$ концевых точек, длина $s_2(n)$, значение форм-фактора $f_2(n)$).

В результате работы метода формируется подмножество ключевых линий $R_D = \left\{L_D, S_2, F_2\right\}_{(n=\overline{1,N_D})}$, которое в дальнейшем может быть использовано для их параметризации и идентификации.

Оценка эффективности детектора линий на основе преобразованного форм-фактора

Разработанный метод реализован на языке C++ с использованием библиотеки OpenCV 3.0. Для сравнительной оценки работы метода реализованы наиболее известные методы детектирования прямых линий на изображениях, такие как метод LSD (Line Segment Detector) [3] и метод на основе преобразования Хафа [1]. Эксперимент проведен на ЭВМ со следующими техническими характеристиками: процессор – Intel(R) Core(TM) i5-2320 CPU 3,0 ГГц; ОЗУ – 4 Гб; тип системы – 64-разрядная операционная система, процессор x64; операционная система – Windows 7.

Для тестирования метода детектирования прямых использованы искусственные, построенные и повернутые в графическом редакторе, линии. Для искусственно созданных линий использованы размеры 5, 11, 15, 25, 41, 65 и 101 пиксел с кривизной $kr = 0 \cdots 3$ (рис. 1) относительно идеальной прямой линии. Каждая линия поворачивалась на угол от 0 до 180 град относительно центра изображения.



а *б* *в* *г*

Рис. 1. Кривизна линии длиной восемь пикселов: *а* – $kr = 0$; *б* – $kr = 1$; *в* – $kr = 2$; *г* – $kr = 3$

В качестве критериев эффективности алгоритмов использованы количество выделенных линий для каждой из групп искусственных линий, усредненное по углу поворота, и время работы методов.

Среднее количество выделенных линий рассчитывается для множества изображений одной и той же линии при повороте на различный угол с помощью выражения

$$\overline{K} = \frac{\sum_{i=0}^{n} K_n}{n},$$

где n – количество линий, K_n – количество линий, выделенных на n-ом изображении.

Также для разработанного метода проведена оценка среднего, минимального и максимального значений форм-фактора для искусственно созданных линий различной длины и кри-

визны. Среднее значение форм-фактора рассчитывается для множества изображений одной и той же линии при повороте на различный угол с помощью выражения

$$\overline{F} = \frac{\sum_{i=0}^{n} F_n}{n},$$

где n – количество линий; F_n – значение форм-фактора n-ой линии.

Среднее $\overline{F}$, минимальное $F_{\min}$ и максимальное $F_{\max}$ значения преобразованного форм-фактора для искусственных линий различной длины и кривизны приведены в табл. 1.

Таблица 1. **Среднее, минимальное и максимальное значения преобразованного форм-фактора для искусственных линий заданной длины и кривизны**

Длина линии, пиксел	**5**		**11**			**15**			
Кривизна, пиксел	0	1	0	1	2	0	1	2	3
$\overline{F}$	0,97	0,97	0,95	0,95	0,91	0,95	0,95	0,92	0,91
$F_{\min}$	0,93	0,93	0,92	0,92	0,8	0,93	0,93	0,85	0,84
$F_{\max}$	1	1	1	1	0,97	1	1	0,97	0,95
Длина линии, пиксел	**25**				**41**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
$\overline{F}$	0,95	0,95	0,93	0,92	0,95	0,95	0,94	0,93	
$F_{\min}$	0,92	0,92	0,88	0,86	0,92	0,92	0,91	0,89	
$F_{\max}$	1	1	0,98	0,95	1	1	0,99	0,96	
Длина линии, пиксел	**65**				**101**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
$\overline{F}$	0,95	0,95	0,94	0,94	0,95	0,95	0,95	0,94	
$F_{\min}$	0,92	0,92	0,91	0,91	0,92	0,92	0,92	0,91	
$F_{\max}$	1	1	0,99	0,97	1	1	0,99	0,98	

В табл. 1 показано, что минимальное $F_{\min}$ и максимальное $F_{\max}$ значения преобразованного форм-фактора для искусственных линия различной длины и кривизны изменяется в пределах $[0{,}8;\ 0{,}93]$ и $[0{,}95;\ 1]$ соответственно. Значения $F_{\min}$ и $F_{\max}$ уменьшаются при увеличении кривизны линии kr. Усредненное значение $\overline{F}$ по углам поворота остается в пределах $[0{,}9;\ 0{,}97]$. Следовательно, при анализе преобразованного форм-фактора линию следует считать прямой, при выполнении следующего условия:

$$\begin{cases} \left|1 - f_2(n)\right| \le T_c, \\ T_c = 0,2. \end{cases}$$

Количество выделенных линий, усредненное по углам поворота, разработанного метода и методов Хафа и LSD для искусственно созданной линии заданного размера и кривизны, приведено в табл. 2.

Значение количества выделенных линий, усредненное по углам поворота, для различных методов детектирования прямых линий из табл. 2 показывает, что представленный метод устойчив к изменению угла поворота и незначительному изменению кривизны контурной линии (рис. 2, *б*), по сравнению с методами Хафа и LSD. Метод Хафа выделяет чаще всего более одной линии (рис. 2, *в*), а метод LSD при незначительном изменении кривизны линии разбивает ее (рис. 2, *г*), что усложняет дальнейшую параметризацию и идентификацию линий на изображении.

а *б* *в* *г*

Рис. 2. Результаты работы методов для линии длиной 41 пиксел, кривизной $kr = 2$, повернутой на угол в 60 градусов: *а* – исходное изображение; *б* – метод NFFLD; *в* – метод Хафа; *г* – метод LSD

Время работы методов для искусственно созданной заданного размера и кривизны приведено в табл. 3, оттуда следует, что предложенный метод обеспечивает выигрыш в скорости выделения прямых контурных в 6…149 раз по сравнению с методом Хафа в зависимости от длины и кривизны линии, но проигрывает методу детектирования прямых линий LSD в 1,2…2,8 раза.

Таблица 2. **Количество выделенных линий, усредненное по углам поворота для искусственно созданной линии заданного размера и кривизны**

Длина линии, пиксел	**5**		**11**			**15**			
Кривизна, пиксел	0	1	0	1	2	0	1	2	3
Метод NFFLD	1	1	1	1	1	1	1	1	1
Метод Хафа	1,2	1,2	1,6	1,6	1,7	1,6	1,6	1,5	1.4
Метод LSD	0,5	0,6	1	1	1,02	1	1	1,1	1.9
Длина линии, пиксел	**25**				**41**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
Метод NFFLD	1	1	1	1	1	1	1	1	
Метод Хафа	1,85	1,85	1,7	1,3	2,3	2,3	1,8	1,36	
Метод LSD	1	1	1,22	1,73	1	1	1,39	1,7	
Длина линии, пиксел	**65**				**101**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
Метод NFFLD	1	1	1	1	1	1	1	1	
Метод Хафа	2,27	2,27	1,97	1,08	2,5	2,5	2,01	0,74	
Метод LSD	1	1	1,47	1,96	1	1	1,49	2,08	

Таблица 3. **Время работы алгоритмов**

Длина линии, пиксел	**5**		**11**			**15**			
Кривизна, пиксел	0	1	0	1	2	0	1	2	3
Метод NFFLD, мс	0,55	0,26	0,5	0,45	0,41	0,55	0,52	0,55	0.54
Метод Хафа, мс	3,37	2,4	7,7	8,2	7,98	14	13,6	14	13,65
Метод LSD, мс	0,3	0,13	0,23	0,226	0,21	0,25	0,25	0,44	0,3
Длина линии, пиксел	**25**				**41**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
Метод NFFLD, мс	0,83	0,96	0,83	0,85	1,36	1,38	1,43	1,4	
Метод Хафа, мс	35,47	35,3	36,6	35,9	94,17	94,16	96,2	95,77	
Метод LSD, мс	0,3	0, 3	0,4	0,33	0,61	0,7	0,8	0,66	
Длина линии, пиксел	**65**				**101**				
Кривизна, пиксел	0	1	2	3	0	1	2	3	
Метод NFFLD, мс	2,23	2,2	2,3	2,25	3,8	3,76	3,75	3,8	
Метод Хафа, мс	234,3	232,5	234,9	233,2	551,6	550,9	555,9	568,3	
Метод LSD, мс	1,07	1,44	1,08	1,16	2,06	2,09	2,15	2,19	

Заключение

Разработан метод детектирования прямых линий на основе преобразованного форм-фактора. Представленный метод устойчив к изменению угла поворота и незначительному изменению кривизны контурной линии, по сравнению с методами Хафа и LSD. Также данный метод обеспечивает выигрыш в скорости до 149 раз по сравнению с методом Хафа.

С помощью оценки среднего, минимального и максимального значений форм-фактора для искусственно созданных линий различной длины и кривизны, осуществлен выбор оптимального значения порога для анализа выделенных линий: $T_c = 0,2$.

DETECTION OF STRAIGHT LINES BASED ON FORM FACTOR

O.G. SHAUCHUK

Abstract

A method of detecting straight lines based on the converted form-factor is proposed. The analysis of this method in comparison with known algorithms is conducted. Parameters for the developed method are selected. It is shown that the method is resistant to change of turn and a little curvature of the line.

Keywords: form-factor, detection, straight line, normalization of line.

Список литературы

1. *Duda R.O.* // Communication of the ACM. 1972. Vol. 15. №1. P. 229-246.
2. *Anver M.M., Stonier R.J.* // Proc. of the 2nd Intern. Conf. on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2003. Singapore. 2003. P. 344-348.
3. *Grompone von Gioi R.* // IEEE Transactions on Pattern. Analysis and Machine Intelligence. 2010. Vol. 32. № 4. P. 722-732.
4. *Chan T.S., Raymond K.K.* // Proc. of 13th Intern. Conf. on Pattern Recognition, ICPR 1996. Vienna. 1996. P. 126-130.
5. *Бородина О.Г., Цветков В.Ю.* // Изв. СПбГЭТУ «ЛЭТИ». 2015. №1. С. 41-45.
6. *Shih F.Y., Cheng S.* // Image and Vision Computing. Newark. 2005. №23. P. 877-886.
7. *Шевчук О.Г., Цветков В.Ю.* // Информатика. 2016. №51. С. 14-24