

О. Г. ШЕВЧУК, В. Ю. ЦВЕТКОВ

## ВЫДЕЛЕНИЕ ПРЯМЫХ КОНТУРНЫХ ЛИНИЙ НА АЭРОИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ЕВКЛИДОВА ФОРМ-ФАКТОРА

Белорусский государственный университет информатики и радиоэлектроники

Разработан алгоритм выделения прямых контурных линий на основе евклидова форм-фактора. Сущность алгоритма заключается в детектировании прямых контурных линий в результате сравнения с единицей значения евклидова форм-фактора, учитывающего сумму евклидовых расстояний между смежными пикселями, образующими контурную линию, при расчете ее длины. Показано, что разработанный алгоритм позволяет повысить стабильность выделения линий и быстродействие по сравнению с алгоритмом Хафа.

### Введение

Актуальной задачей обработки аэроизображений является выделение прямых линий для последующего поиска объектов. Для детектирования линий и их отрезков на аэроизображениях широко используются алгоритмы, основанные на преобразовании Хафа [1] и вычислении градиента (алгоритм LSD – Line Segment Detection) [2]. Недостатками алгоритма Хафа являются высокая вычислительная сложность и обнаружение ложных линий в местах высокой плотности пикселей, недостатком алгоритма LSD – разбиение плавной линии (линии с небольшой кривизной) на большое число изолированных фрагментов, что затрудняет параметризацию и поиск объектов. Устранить данные недостатки позволяет использование форм-фактора для поиска прямых контурных линий. Алгоритм выделения прямых линий на основе форм-фактора (FFLD – Form Factor Line Detection) [3] позволяет выделять изолированные контурные линии, не имеющие пересечений. При этом форм-фактор определяется как отношение размера контурной линии, вычисляемого по известным координатам ее конечных точек, к длине контурной линии, определяемой суммой образующих ее контурных точек:

$$f_2 = r_2 / s_2, \quad (1)$$

где  $r_2 = \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}$  – размер контурной линии;  $s_2 = \sum_{k=1}^K t(k)$  – длина контурной линии;  $(y_1, x_1), (y_2, x_2)$  – координаты конечных точек контурной линии;  $t(k) = \{1; 1,2; 1,5\}$  – коэффициент кривизны, зависящий от маски к которой принадлежит пиксел (рис. 1);  $K$  – количество пикселей в контурной линии.

Алгоритм обеспечивает уменьшение вычислительной сложности выделения прямых контурных линий за счет обработки относительно небольшого числа контурных пикселей с сохранением первоначальной формы контура. Недостатком алгоритма FFLD является пропуск некоторых линий при условии наличия у нее скопления более 3 пикселей в окрестности Мура, что приводит к потере устойчивости детектирования при изменении условий вилеорегистрации. Устранение данного недостатка возможно за счет вычисления длины контурной линии как суммы евклидовых расстояний между смежными пикселями, образующими данную линию.

Цель работы – разработка алгоритма поиска изолированных прямых контурных линий, обеспечивающего повышение устойчивости

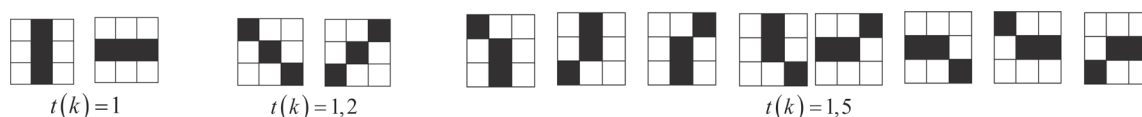


Рис. 1. Коэффициент кривизны и маски к которым он относится

детектирования за счет определения параметров линии на основе евклидовых расстояний.

### Алгоритм детектирования прямых линий на основе евклидова форм-фактора

Для повышения устойчивости детектирования прямых контурных линий предлагается определять их параметры с помощью евклидова форм-фактора, вычисляемого с помощью выражения:

$$f_{2\dot{y}} = r_2 / s_{2\dot{y}}, \quad (2)$$

где

$$s_{2\dot{y}} = \sum_{k=1}^{K-1} \sqrt{(y(k) - y(k+1))^2 + (x(k) - x(k+1))^2} -$$

евклидова длина контурной линии;  $(y(k), x(k))$  – координаты  $k$ -ого пиксела контурной линии;  $K$  – количество пикселей в контурной линии.

Предлагается алгоритм поиска прямых контурных линий на основе евклидова форм-фактора (DLBEFF – Detection Of Lines Base On Euclidean Form Factor). Сущность алгоритма заключается в детектировании прямой контурной линии в результате сравнения с единицей значения евклидова форм-фактора, учитывающего сумму евклидовых расстояний между смежными пикселями, образующими контурную линию, при расчете ее длины, с предварительной нормализацией по толщине выделенных контуров изображения.

Исходными данными для алгоритма является полутоновое изображение

$$I = i(y, x)_{(y=0, Y-1, x=0, X-1)},$$

где  $i(y, x) = 0, 255$  – значение пиксела на изображении;  $y = 0, Y-1$ ,  $x = 0, X-1$  – координаты пиксела изображения;  $Y, X$  – размеры изображения по вертикали и горизонтали. Алгоритм DLBEFF из следующих шагов:

1. Высокочастотная фильтрация изображения. Осуществляется с использованием алгоритма Канни [4]. В результате формируется бинарная матрица  $P = \|p(x, y)\|_{(y=0, Y-1, x=0, X-1)}$  образов выделенных контуров, где  $p(x, y) = \{0, 1\}$  – значение пиксела.

2. Сегментация контуров (разбиение на фрагменты). Сегментация матрицы  $P$  осуществляется алгоритмом выращивания областей [5]. В результате формируется матрица сегментации

$$I_S = \|i_S(y, x, n)\|_{(y=0, Y-1, x=0, X-1)},$$

где  $n = [1, N_s]$ ,  $N_s$  – число выделенных контуров.

3. Нормализация контурных линий. Для уточнения линий используется алгоритм нормализации контурных линий по толщине на основе масочного анализа локальных ориентаций их фрагментов [6].

4. Выделение сегментированных контурных элементов с двумя концевыми точками, линий. Из множества  $L_S = \{l_S(n)\}_{(n=1, N_s)}$  сегментированных контурных элементов, включающего замкнутые, пересекающиеся и разомкнутые контурные элементы, выделяется подмножество  $L_2 = \{l_2(n)\}_{(n=1, N_2)}$  контурных линий, которые имеют две концевые точки, где  $N_2$  – число сегментированных контурных линий с двумя концевыми точками.

5. Определение длин сегментированных контурных линий. Для каждой контурной линии  $l_2(n) = \{X(n), Y(n)\}_{(n=1, N_2)}$ , имеющей две концевые точки, определяется число  $s_{2\dot{y}}(n)$  образующих данную линию контурных пикселей  $p(y, x)$  с помощью выражения

$$s_{2\dot{y}}(n) = \sum_{i=0}^{K-2} \sqrt{(x(i) - x(i+1))^2 + (y(i) - y(i+1))^2},$$

6. 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

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

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

7. Вычисление форм-фактора линии. Для каждой контурной линии  $l_2(n)$ , имеющей две концевые точки, вычисляется значение форм-фактора  $f_{2\dot{y}}(n)$  линии с помощью выражения

$$f_{2\dot{y}}(n) = r_2(n) / s_{2\dot{y}}(n).$$

8. Анализ форм-фактора линии. Для каждой контурной линии  $l_2(n)$ , имеющей две концевые точки, значение ее форм-фактора  $f_{2\dot{y}}(n)$  сравнивается с единицей. Близость значения  $f_{2\dot{y}}(n)$  линии к единице оценивается в результате сопоставления заданного порога  $T_c = 0,2$  [7], определяющего кривизну линии, с модулем разности, вычисляемым с помощью выражения  $|1 - f_{2\dot{y}}(n)|$ . Контурная линия  $l_2(n)$  является прямой, если выполняется следующее условие

$$|1 - f_{2_3}(n)| \leq T_c.$$

В результате работы алгоритма формируется множество  $L_D = \{l_D(n)\}_{(n=1, N_D)}$  изолированных прямых контурных линий, где  $N_D$  – их количество.

### Экспериментальные исследования

Разработанный алгоритм реализован на языке C++ с использованием библиотеки OpenCV 3.0. 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 Windows 7. Для анализа алгоритмов DLBEFF и FFLD использовалось тестовое изображение размером 600×600 пикселей при изменении параметра  $\mu = \|i, c, s, \alpha\|$  в графическом редакторе, где  $i$  – яркость,  $c$  – контраст,  $s$  – масштаб и  $\alpha$  – поворот. Для анализа алгоритмов DLBEFF, LSD и Хафа использовались следующие изображения:

– с искусственно построенной линией, длиной 5, 11, 15, 25, 41, 65 и 101 пиксел с кривизной  $kr = 0...3$  относительно идеальной прямой линии и повернутые в графическом редакторе на угол от 0 до 180 град относительно центра изображения;

– полученные с помощью видеокамеры и относящиеся к одному из типов, выделенных в соответствии с видом яркостной гистограммы [8] при изменении параметра  $\mu$  в графическом редакторе.

В качестве критериев эффективности алгоритмов использованы стабильность результатов выделения прямых линий на изображении и среднее время работы алгоритмов.

Для оценки стабильности детектирования линий, их количество  $K_i(\mu)$  на  $i$ -ом изображении при изменении одного из его параметров  $\mu$  нормировалось относительно опорного изображения в соответствии с выражением:

$$S_i(\mu) = \frac{K_i(\mu)}{K_0},$$

где  $K_0$  – количество выделенных линий на исходном изображении,  $\mu = \|i, c, s, \alpha\|$ ,  $i = [-50...50\%]$ ,  $c = [-50...50\%]$ ,  $s = [50...150\%]$ ,  $\alpha = [-90...90^\circ]$  – для реальных изображений,  $\alpha = [0...180^\circ]$  – для искусственной линии.

Значение характеристик  $\Delta S(\mu) = S_{\max}(\mu) - S_{\min}(\mu)$  стабильности и среднее время обработ-

ки изображений для алгоритмов DLBEFF и FFLD приведены в табл. 1, из которой видно, что алгоритм DLBEFF стабильнее алгоритма FFLD в среднем на 8% при равном быстродействии.

Значение характеристики  $\Delta S(\mu)$  стабильности выделения линий при повороте изображения с искусственно созданной линией для разработанного алгоритма и алгоритмов Хафа и LSD приведено в табл. 2, из которой видно, что предложенный алгоритм абсолютно стабилен к изменению угла поворота и незначительному изменению кривизны контурной линии. Средняя нормированная нестабильность выделения линий для алгоритма Хафа составляет 4,4, для алгоритма LSD – 0,5. Это связано с разбиением (алгоритм LSD) и множественным выделением (алгоритм Хафа) контурных линий при незначительной кривизне, что затрудняет дальнейшую их параметризацию.

Таблица 1. Характеристика  $\Delta S(\mu)$  стабильности детектирования линий и среднее время работы для алгоритмов DLBEFF и FFLD

Алгоритм	Яркость	Контраст	Масштаб	Поворот	Время обработки, мс
DLBEFF	0,47	0,55	0,95	0,6	762
FFLD	0,51	0,6	1	0,65	770

Графики зависимости нормированного значения количества выделенных линий  $S_i$  от изменения параметра  $\mu$  при использовании разработанного алгоритма и алгоритмов Хафа и LSD для реальных изображений представлены на рис. 2, значения  $\Delta S(\mu)$  и время работы алгоритмов – в табл. 3.

Из табл. 3 видно, что алгоритм DLBEFF в среднем стабильнее алгоритма Хафа в 1,8 раз при изменении яркости, в 6,3 раза при изменении контраста, в 2 раза при изменении масштаба и в 5,9 раз при повороте изображения. При этом разработанный алгоритм уступает в стабильности алгоритму LSD в 1,1 раз при изменении яркости, в 2,9 раз при изменении контраста и 1,9 раз при повороте изображения, однако в 1,13 раз стабильнее при изменении масштаба. Разработанный алгоритм DLBEFF в 3,8 раз превосходит в быстродействии алгоритм Хафа, однако уступает алгоритму LSD в 1,3 раза (табл. 3).

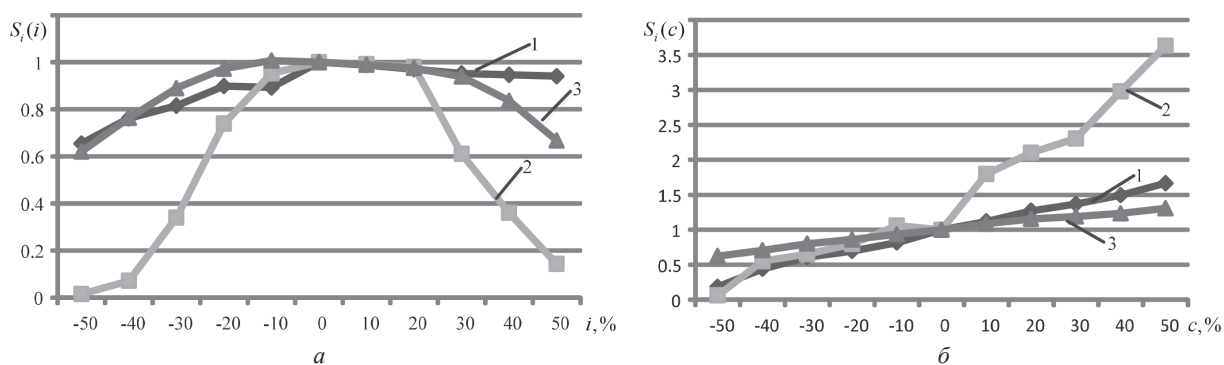


Рис. 2. Нормализованное значение  $S_i(\mu)$  количества выделенных линий на изображении в зависимости от изменяемого его параметра для:  $a$  – 1-го типа изображений при изменении яркости;  $b$  – 2-го типа изображений при изменении контраста; 1 – алгоритм DLBEFF; 2 – алгоритм Хафа; 3 – алгоритм LSD

Т а б л и ц а 2. Характеристика  $\Delta S(\mu)$  стабильности детектирования линий при повороте изображения с искусственно созданной линией

Длина линии, пиксел	5		11			15			
	0	1	0	1	2	0	1	2	3
Кривизна, пиксел	0	1	0	1	2	0	1	2	3
Алгоритм DLBEFF	0	0	0	0	0	0	0	0	0
Алгоритм Хафа	2	2	3	3	2	3	3	3	2
Алгоритм LSD	1	1	0	0	1	0	0	1	1
Длина линии, пиксел	25				41				
	0	1	2	3	0	1	2	3	
Кривизна, пиксел	0	0	0	0	0	0	0	0	
Алгоритм DLBEFF	0	0	0	0	0	0	0	0	
Алгоритм Хафа	4	4	3	3	6	6	4	4	
Алгоритм LSD	0	0	1	1	0	0	1	1	
Длина линии, пиксел	65				101				
	0	1	2	3	0	1	2	3	
Кривизна, пиксел	0	0	0	0	0	0	0	0	
Алгоритм DLBEFF	0	0	0	0	0	0	0	0	
Алгоритм Хафа	8	8	7	3	9	9	8	2	
Алгоритм LSD	0	0	1	1	0	0	1	1	

Т а б л и ц а 3. Характеристика  $\Delta S(\mu)$  стабильности детектирования линий и среднее время работы для алгоритмов DLBEFF, LSD и Хафа

Алгоритм	Яркость	Контраст	Масштаб	Поворот	Время обработки, мс
DLBEFF	0,564	1,212	1,554	0,216	587,6
LSD	0,504	0,418	1,768	0,114	442,6
Хафа	1,04	7,664	3,066	1,274	2273,2

### Заключение

Разработан алгоритм выделения прямых контурных линий на основе евклидова форм-фактора. Данный алгоритм отличается от известных детектированием прямых контурных линий в результате сравнения с единицей значения евклидова форм-фактора, учитывающего сумму евклидовых расстояний между смеж-

ными пикселями, образующими контурную линию, при расчете ее длины, с предварительной нормализацией по толщине выделенных контуров изображения. В отличие от алгоритмов Хафа и LSD предложенный алгоритм позволяет выделять контурные линии без фрагментации. Разработанный алгоритм обеспечивает повышение стабильности выделения линий: а) по сравнению с алгоритмом Хафа в 1,8 раз при изменении яркости, в 6,3 раза при изменении контраста, в 2 раза при изменении масштаба и в 5,9 раз при повороте изображения (при выигрыше в быстродействии в 3,8 раз); б) по сравнению с алгоритмом LSD в 1,13 раз при изменении масштаба (при проигрыше в быстродействии в 1,3 раза).

### ЛИТЕРАТУРА

1. Duda, R. O. Use of the Hough Transformation to Detect Lines and Curves in Pictures / R. O. Duda // Communication of the ACM. – 1972. – Vol. 15, № 1. – P. 229–246.

2. **LSD**: A Fast Line Segment Detector with a False Detection Control / R. Grompone von Gioi [et al.] // IEEE Transactions on Pattern. Analysis and Machine Intelligence. – 2010. – Vol. 32, № 4. – P. 722–732.
3. **Бородина, О. Г.** Выделение изолированных прямых линий на изображениях с использованием форм-фактора / О. Г. Бородина, В. Ю. Цветков // Изв. СПбГЭТУ «ЛЭТИ». – 2015. – № 1. – С. 41–45.
4. **Canny, J. A.** Computational Approach to Edge Detection / J. Canny // IEEE Trans. Pattern Analysis and Machine Intelligence. – 1986. – Vol. 8, no. 6. – P. 679–698.
5. **Shih, F. Y.** Automatic seeded region growing for color image segmentation / F. Y. Shih, S. Cheng // Image and Vision Computing, Newark. – 2005. – № 23. – P. 877–886.
6. **Шевчук О. Г.** Нормализация контурных линий по толщине на основе масочного анализа локальных ориентаций их фрагментов / Шевчук О. Г., Цветков В. Ю. // Информатика. – 2016 – № 51. – С. 14–24.
7. **Шевчук О. Г.** Детектирование прямых линий на основе форм-фактора / Шевчук О. Г. // Телекоммуникации: сети и технологии, алгебраическое кодирование и безопасность данных: материалы междунар. научно-технич. семинара, часть 2. Минск, апрель – декабрь 2016 г. / БГУИР. – Минск, 2016. – С. 61–67.
8. **Шевчук О. Г.** Детектирование прямых линий на изображении с использованием евклидова форм-фактора / Шевчук О. Г., Крамков Д. А. // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2017): доклады XVI Международной конференции. Минск, 16 ноября 2017 г. – Минск: ОИПИ НАН Беларуси, 2016. – с. 167–173.

## REFERENCES

1. **Duda, R. O.** Use of the Hough Transformation to Detect Lines and Curves in Pictures / R. O. Duda // Communication of the ACM. – 1972. – Vol. 15, № 1. – P. 229–246.
2. **LSD**: A Fast Line Segment Detector with a False Detection Control / R. Grompone von Gioi [et al.] // IEEE Transactions on Pattern. Analysis and Machine Intelligence. – 2010. – Vol. 32, № 4. – P. 722–732.
3. **Borodina, O. G.** Isolation of isolated straight lines on images using a form factor / O. G. Borodin, V. Yu. Tsvetkov // Izv. ETU «LETI». – 2015. – No. 1. – P. 41–45.
4. **Canny, J. A.** Computational Approach to Edge Detection / J. Canny // IEEE Trans. Pattern Analysis and Machine Intelligence. – 1986. – Vol. 8, no. 6. – P. 679–698.
5. **Shih, F. Y.** Automatic seeded region growing for color image segmentation / F. Y. Shih, S. Cheng // Image and Vision Computing, Newark. – 2005. – № 23. – P. 877–886.
6. **Shevchuk O. G.** Normalization of contour lines in thickness based on mask analysis of local orientations of their fragments / Shevchuk O. G., Tsviatkou V. Yu. // Informatics. – 2016. – № 51. – P. 14–24.
7. **Shevchuk O. G.** Detection of straight lines based on the form factor / Shevchuk O. G. // Telecommunications: networks and technologies, algebraic coding and data security: materials of the international. scientific and technical. seminar, part 2. Minsk, April – December 2016 / BSUIR. – Minsk, 2016. – P. 61–67.
8. **Shevchuk O. G.** Detection of straight lines in an image using the Euclidean form factor / Shevchuk O. G., Kramkov D. A. // Development of Informatization and the State System of Scientific and Technical Information (RINTI-2017): reports of the XVI International Conference. Minsk, 16 November 2017 – Minsk: OIPI NAS of Belarus, 2016. – P. 167–173.

Поступила  
16.05.2018

После доработки  
16.08.2018

Принята к печати  
31.08.2018

*A. G. SHAUCHUK, V. YU. TSVIATKOU*

## DETECTION OF DIRECT CONTROL LINES ON AEROVIEWS BASE ON EKKLIDOV FORM FACTOR

*Belarusian State University of Informatics and Radioelectronics, o.shevchuk@bsuir.by, vtsvet@bsuir.by*

*The algorithm of detection of direct contour lines base on euclidean form factor is presented. The essence of the algorithm is the detection of direct contour lines as a result of a comparison with unity of the value of the euclidean form factor, taking into account the sum of the euclidean distances between neighbors pixels forming the contour line, when its length is calculate. The developed algorithm makes it possible to increase the stability of the allocation of lines and the speed of response in comparison with the Hough algorithm is show.*



**Шевчук Оксана Геннадьевна** *Ассистент кафедры инфокоммуникационных технологий БГУИР*

**Shevchuk Oksana** *Assistant of the Department of Infocommunication Technologies BSUIR*

*Email: o.shevchuk@bsuir.by*



**Цветков Виктор Юрьевич**, *д.т.н., доцент,*

*- заведующий кафедрой инфокоммуникационных технологий БГУИР,*

*- заведующий НИЛ 2.3 «Сетецентрические видеотехнологии и управление» НИЧ БГУИР*

**Tsviatkou Viktor**, *Doctor of Technical Sciences, Associate Professor,*

*- Head of the Department of Infocommunication Technologies BSUIR,*

*- Head of the NIL 2.3 «Network-centric video technologies and management» NICh BSUIR*

*Email: vtsvet@bsuir.by*