

ОБРАБОТКА ИЗОБРАЖЕНИЙ, СИГНАЛОВ И РЕЧИ

УДК 004.932.2

Б.А. Залесский, П.В. Лукашевич

ВЫДЕЛЕНИЕ УГЛОВ НА ИЗОБРАЖЕНИЯХ

Рассматриваются алгоритмы автоматического выделения углов на полутоновых и цветных изображениях, основанные на использовании ориентированного градиента. В отличие от известных алгоритмов они позволяют не только надежно выделить на изображении вершины углов заданной величины, но и оценить ориентацию их сторон.

Введение

Задача выделения углов хорошо известна в области обработки изображений: многие алгоритмы распознавания и сопровождения объектов, обнаружения движения, 3D-реконструкции используют углы для описания и сравнения сцен и объектов на них. Поэтому в настоящее время известно большое число подходов к решению этой задачи.

Прежде чем сделать краткий обзор и анализ известных результатов, следует охарактеризовать само понятие «угол на изображении». Обычно в области обработки изображений под углом понимают фигуру, образованную двумя отрезками гладких кривых, выходящими из одной точки (вершины) и имеющими в ней отличающиеся по направлению касательные. Отрезки называют сторонами угла (рис. 1).

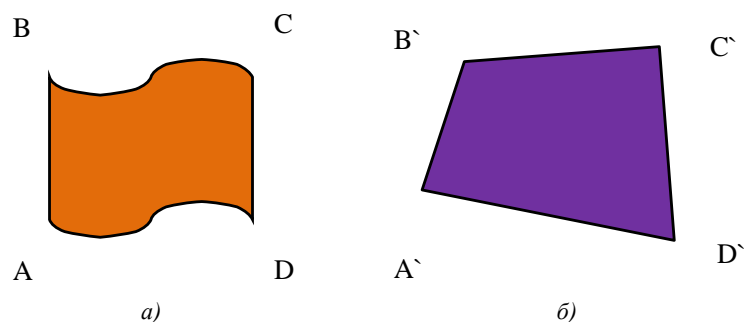


Рис. 1. Изображения фигур с углами: а) не входящими в множество Θ ; б) входящими в него

В настоящей статье речь идет лишь о выделении на изображениях углов заранее заданной величины, стороны которых являются отрезками прямых либо кривыми наперед заданной формы. Очевидно, что такое понимание геометрического объекта не является наиболее общим, однако оно более предпочтительно при решении задач обнаружения и распознавания объектов на изображениях – в данном случае важно находить углы фиксированной величины и формы.

Рассмотрим, например, актуальную задачу обнаружения зданий на аэрофотоснимках и космических изображениях. Практически все здания, за редким исключением, имеют прямолинейные контуры и прямые углы, поэтому в дальнейшем для простоты будем рассматривать углы с прямыми сторонами, обозначая их совокупность буквой Θ . Углы с криволинейными сторонами заданной формы в общем случае могут быть выделены так же, как и углы из Θ .

Под задачей обнаружения углов из множества Θ будем понимать не только нахождение координат их вершин, но и оценку ориентации их сторон, хотя часто при обработке изображений используются только координаты вершин.

Один из первых детекторов углов, основанный на анализе неоднородности изображения в окрестности каждого пиксела, предложил Моравец [1]. Харрис и Стефенс [2] разработали алгоритм выделения углов, который основан на анализе 2×2 матрицы, составленной из осредненных по окну

первых производных изображения. В 1997 г. появился детектор углов SUSAN [3]. Затем появились многоуровневые детекторы углов [4–6], использующие как первые, так и вторые производные изображений, лапласианы и гессианы изображений. Чуть позже были предложены генетические и FAST-алгоритмы обнаружения углов [7, 8]. Подробный обзор алгоритмов обнаружения углов и результаты сравнения их точности и быстродействия приведены в [9, 10].

Предлагаемый в статье алгоритм обнаружения углов был специально разработан для выделения зданий. Он, в отличие от упомянутых выше алгоритмов, предназначен для выделения всех углов заранее заданного размера со сторонами заранее заданной формы, выходящих из каждой точки изображения, что позволяет получить более точные результаты при нахождении на изображениях крыш домов, чем в случае применения других известных алгоритмов. Особенностью алгоритма является также то, что для уточнения координат вершин найденных углов применялся алгоритм наращивания областей.

1. Описание алгоритма

Обозначим через $\Theta(\alpha, \rho_1, \rho_2)$, $\alpha \in [0, 2\pi]$, множества углов радианной меры α , стороны которых являются отрезками гладких кривых ρ_1, ρ_2 фиксированной формы. Предлагаемый алгоритм находит на полутоновых и цветных изображениях углы из $\Theta(\alpha, \rho_1, \rho_2)$ фиксированной радианной меры α со сторонами ρ_1, ρ_2 . Так как последовательность действий при выполнении алгоритма не зависит от формы кривых ρ_1, ρ_2 , образующих стороны, опишем его для множества углов $\Theta(\alpha, l_1, l_2)$ с прямолинейными сторонами l_1, l_2 . Для краткости воспользуемся сокращенной записью $\Theta(\alpha)$ вместо $\Theta(\alpha, l_1, l_2)$. Будем считать, что угол θ из $\Theta(\alpha)$ найден на изображении, если указаны его вершина и направления его сторон. Алгоритм основан на оценивании заполнения тонкого (толщиной от одного до нескольких пикселей) растрового шаблона сторон угла ориентированным градиентом. Он позволяет, в частности, выделять все (или почти все) углы из $\Theta(\alpha)$, выходящие из одной вершины.

Обозначим через $S = \{0, 1, K, M-1\} \times \{0, 1, K, N-1\}$, $\mathbf{j} = (j_1, j_2) \in S$ множество пикселей изображения размером $M \times N$. Значения пикселей цветного RGB-изображения $\mathbf{I}_{rgb} = (\mathbf{I}_r, \mathbf{I}_g, \mathbf{I}_b)$ представляют собой $M \times N$ матрицу с элементами-векторами $\mathbf{I}_{rgb, \mathbf{j}} = (I_{r, \mathbf{j}}, I_{g, \mathbf{j}}, I_{b, \mathbf{j}}) \in \{0, 1, K, 255\}^3$. Значения пикселей полутонового изображения \mathbf{I} формируют $M \times N$ матрицу со значениями яркостей $I_{\mathbf{j}} \in \{0, 1, K, 255\}$.

Градиент цветного изображения вычисляется либо отдельно для каждого цветового канала $\mathbf{I}_r, \mathbf{I}_g, \mathbf{I}_b$, либо только по одному из каналов, например по зеленому. Для обнаружения сторон угла используется оконный ориентированный градиент изображения.

Напомним, что ориентированный градиент изображения \mathbf{I} понимается как $M \times N$ матрица \mathbf{G} , индексированная пикселями $\mathbf{j} \in S$ изображения, элементами $G_{\mathbf{j}}$ которой являются пары чисел, первое из которых есть горизонтальная $g_{H, \mathbf{j}}$, а второе – вертикальная $g_{V, \mathbf{j}}$ компоненты градиента в данном пикселе ($G_{\mathbf{j}} = (g_{H, \mathbf{j}}, g_{V, \mathbf{j}})$). Для предложенного алгоритма использовался оконный ориентированный градиент

$$g_{H, \mathbf{j}} = \frac{\sum_{\mathbf{m} \in W_{H, \mathbf{j}}(1)} I_{\mathbf{m}} - \sum_{\mathbf{m} \in W_{H, \mathbf{j}}(-1)} I_{\mathbf{m}}}{|W_{H, \mathbf{j}}(1)| + |W_{H, \mathbf{j}}(-1)|}, \quad g_{V, \mathbf{j}} = \frac{\sum_{\mathbf{m} \in W_{V, \mathbf{j}}(1)} I_{\mathbf{m}} - \sum_{\mathbf{m} \in W_{V, \mathbf{j}}(-1)} I_{\mathbf{m}}}{|W_{V, \mathbf{j}}(1)| + |W_{V, \mathbf{j}}(-1)|},$$

вычисленный с помощью окон W (рис. 2).

Непосредственное нахождение оконного градиента изображения даже на окнах малого размера занимает весьма существенное время, поэтому его вычисление следует производить,

например, с помощью интегрального изображения (summed area table) [11] или алгоритма «бегущей суммы» (running sum) [12]. При применении каждого способа время вычисления уменьшается на несколько порядков. К тому же оно в этом случае зависит от размера изображения, но не от размера окон. Использование окон подходящего размера позволяет устранить шумовую составляющую градиента.

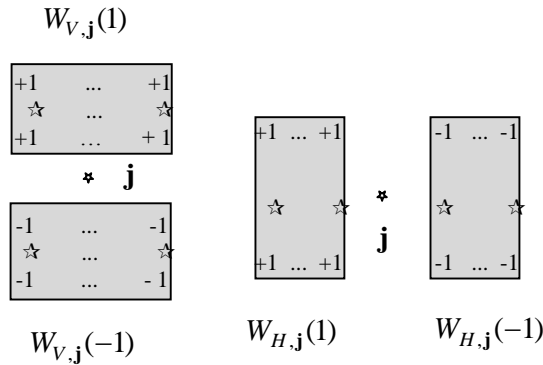


Рис. 2. Вид окон для вычисления вертикальной и горизонтальной компонент градиента

В общем случае угол наклона вектора градиента $\beta_j = \beta(g_{H,j}, g_{V,j})$ изменяется от 0 до 2π , однако при определении угла наклона сторон l_1, l_2 к оси абсцисс будем отождествлять противоположно направленные векторы G_j и $-G_j$, считая, что угол наклона градиента $-\frac{\pi}{2} \leq \beta_j < \frac{\pi}{2}$, так как в противном случае пришлось бы при вычислении значения угла учитывать соотношение яркостей внутри и вне угла. Например, при переходе от изображения к его негативу направления градиентов поменяются на противоположные, хотя углы при этом не изменятся (рис. 3).

В силу вышесказанного угол наклона градиента можно задать простой формулой

$$\beta_j = \beta(G_j) = \begin{cases} \arctan\left(\frac{g_{V,j}}{g_{H,j}}\right), & \text{если } g_{H,j} \neq 0; \\ \frac{\pi}{2}, & \text{если } g_{H,j} = 0, \end{cases} \quad (1)$$

а его длину – формулой

$$|G_j| = \sqrt{g_{H,j}^2 + g_{V,j}^2}. \quad (2)$$

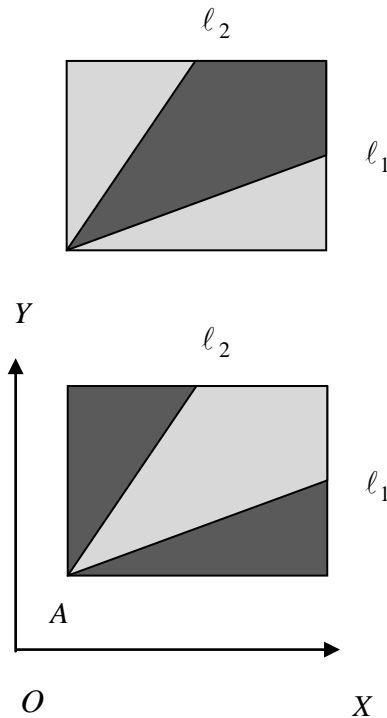


Рис. 3. Изображения, имеющие противоположенный градиент, но совпадающие углы

Сразу отметим, что, несмотря на отождествление противоположных направлений градиента, алгоритм «различает» повернутые углы равной радианной меры. Для этого использована параметризация $\theta \in \Theta(\alpha)$ минимальным углом (задающим направление) между его сторонами и осью абсцисс, равным $\gamma = \min(\angle(l_1, OX), \angle(l_2, OX))$, измеренным в точке A (рис. 3) против часовой стрелки. (Напомним, что величина всех углов из $\Theta(\alpha)$ равна α .)

Смысл алгоритма заключается в следующем: растровый шаблон угла $\theta \in \Theta(\alpha)$, образованный растровыми отрезками фиксированной длины, поворачивается вокруг каждого пиксела j . Для каждого поворота γ вычисляется число пикселей $n_1(\gamma)$ на l_1 и соответственно $n_2(\gamma)$ на l_2 , градиент в которых «почти» ортогонален l_1 и соответственно l_2 .

Затем проверяется гипотеза о существовании угла в данном пикселе с направлением γ . Решающие правила для такой проверки могут основываться, например, на оценке вероятностей существования и отсутствия угла и выборе решения, вероятность которого больше. При этом важно помнить, что пиксел, вообще говоря, может быть вершиной нескольких углов заданной величины α , поэтому критерии, основанные на вычислении минимальных или максимальных значений каких-либо метрик, например расстояния Махаланобиса или циркуляционных гистограмм [13], не применим. Авторы использовали для этой цели следующий простой критерий, основанный на следующих значениях $n_1(\gamma)$ и $n_2(\gamma)$:

$$n_i(\gamma) = \left\{ \mathbf{j} \in l_i : \left| \beta_j - \arctan \left(\frac{1_{y,i}}{1_{x,i}} \right) \right| \leq \tau_1, \quad |G_j| \geq \tau_2 \right\}, \quad i = 1, 2, \quad (3)$$

где $(1_{x,i}, 1_{y,i})$ – координаты отрезка l_i ; τ_1 – наперед заданный порог допустимой точности вычисления градиента, который, например, в проведенных экспериментах выбирался в интервале $\tau_1 \in \left[0, \frac{\pi}{9}\right]$ (численно $\frac{\pi}{9} \approx 0,35$); τ_2 – порог, ограничивающий снизу длину векторов градиента.

Вид формулы (3) продиктован свойствами оконного градиента \mathbf{G} . Несмотря на гладкость, значения \mathbf{G} обладают небольшой погрешностью, которая приводит к необходимости использования порога τ_1 . Кроме того, на изображениях присутствует шум, приводящий к возникновению некоторого числа шумовых векторов градиента малой длины, которые затрудняют обнаружение углов. Для отсеивания шумовых векторов используется порог τ_2 .

Замечание. Чтобы избежать оценивания величин τ_1 , τ_2 , можно учитывать в $n_i(\gamma)$ все ненулевые векторы градиента, попадающие на растровые представления отрезков l_i с весами, обратно

пропорциональными расстоянию $\left(\left| \beta_j - \arctan \left(\frac{1_{y,i}}{1_{x,i}} \right) \right| + 1 \right)^{-q}$ ($q > 0$) от истинного значения направления $\arctan \left(\frac{1_{y,i}}{1_{x,i}} \right)$ сторон l_i и прямо пропорциональными длине $|G_j|$, т. е. положить

$$n_i(\gamma) = \sum_{\mathbf{j} \in l_i} \frac{|G_j|}{\left(\left| \beta_j - \arctan \left(\frac{1_{y,i}}{1_{x,i}} \right) \right| + 1 \right)^{-q}}.$$

Для $n_i(\gamma)$, вычисленных по формуле (3), критерий имеет вид неравенства

$$\frac{n_1(\gamma)}{|l_1|} \cdot \frac{n_2(\gamma)}{|l_2|} \geq \rho \quad \text{либо} \quad \min \left(\frac{n_1(\gamma)}{|l_1|}, \frac{n_2(\gamma)}{|l_2|} \right) \geq \sqrt{\rho} \quad (4)$$

для наперед заданного положительного числа ρ . Неравенства (4) имеют понятный статистический смысл – при существовании угла они оценивают вероятность того, что его стороны имеют «правильно сориентированный» градиент. Проведенные эксперименты показали, что оптимальные значения ρ действительно близки к 0,5.

Вычисления числа пикселов $n_1(\gamma)$, $n_2(\gamma)$ на сторонах угла l_1 , l_2 можно выполнять с помощью заранее заготовленных растровых представлений отрезков или тонких полос фиксиро-

ванной длины, так как при подходящем подборе окон для подсчета ориентированного градиента матрица \mathbf{G} будет содержать широкие полосы достаточно длинных векторов (рис. 4, б).

Заметим, что поиск особых точек на изображении с помощью шаблонов не является новым [11], однако использование в данном случае ориентированного градиента позволило существенно повысить точность получаемых решений и при этом избежать появления ложных углов.

Опишем подробно один из простых вариантов алгоритма обнаружения углов θ фиксированной величины α , т. е. $\theta \in \Theta(\alpha)$:

Шаг 1. Вычисляем и запоминаем матрицу градиента \mathbf{G} с помощью интегрального изображения или метода бегущей строки.

Шаг 2. Вычисляем по формулам (1), (2) и запоминаем матрицу \mathbf{F} , индексированную пикселями $\mathbf{j} \in S$ изображения с элементами-векторами $F_{\mathbf{j}} = \langle G_{\mathbf{j}}, \beta_{\mathbf{j}} \rangle$.

Шаг 3. Вычисляем и запоминаем массив \mathcal{L} растровых отрезков фиксированной длины l , образующих с осью OX углы $0, \delta, 2\delta, \dots, \delta M$, $M = \left\lfloor \frac{2\pi}{\delta} \right\rfloor$ (δ – шаг поворота отрезков, задающий точность определения углов, выбирается исходя из условий поставленной задачи). В проведенных вычислениях использовалось $M = 360$, т. е. δ полагался равным $\frac{2\pi}{360}$.

Шаг 4. Создаем список для обнаруженных углов. Для этого просматриваем пиксели изображения \mathbf{j} . Для каждого \mathbf{j} вычисляем значения $n_1(\gamma), n_2(\gamma)$ для растровых представлений сторон $l_1, l_2 \in \mathcal{L}$, образующих угол величины α , повернутый на $\gamma = k\delta$, $k = 0 \div M$, относительно оси OX . Проверяем выполнение неравенства (4). Если оно выполняется, заносим характеристики угла \mathbf{j} и γ в список углов.

Шаг 5. Обрабатываем и модифицируем список углов для выделения в нем действительно различных углов.

Последний шаг алгоритма продиктован тем, что вследствие использования оконного градиента его векторы, характеризующие стороны какого-либо угла, образовывали сравнительно широкие полосы, поэтому нередко список углов содержит несколько характеристик одного и того же угла (рис. 4, б). Для выделения в списке (\mathbf{j}_k, γ_k) параметров действительно разных углов использовался алгоритм наращивания областей для нескольких условий соседства: в списке выделялись кластеры (связные компоненты) близких по значению векторов (\mathbf{j}_m, γ_m) и из каждого кластера выбирался только один его представитель, для которого значение статистики максимально. Такой сравнительно простой способ постобработки позволяет с большой надежностью оставить только одного представителя каждого угла $\theta \in \Theta(\alpha)$ (рис. 4, в).

2. Эксперименты

Алгоритм обнаружения углов специально разрабатывался для решения задачи поиска областей изображений с городской застройкой по шаблону (template matching), а также для поиска объектов прямоугольной формы. Для решения данной задачи важно уметь выделять на изображении углы наперед заданной величины, например только прямые. Очевидно, что выделение углов не только облегчает задачу поиска, но и повышает его точность. Применение известных алгоритмов приводило к получению значительного числа неправильно найденных углов, особенно на областях, сильно заполненных градиентом. Такие области характерны, например, для лесных зон (верхняя правая часть рис. 4, а) или гор. Все это существенно снижало значимость признака, не позволяя решить задачу выделения сооружений с достаточной степенью точности.

Предложенный алгоритм позволил с высокой вероятностью находить углы заданной величины (в частности, углы зданий), не выделяя при этом ложные (рис. 4, в, г).

Для проверки точности работы алгоритма использовались цветные RGB-аэрофотоснимки, полученные с Google Maps (<http://maps.google.com/>). Все вычисления проводились на зеленом канале изображений.

Наиболее качественные результаты получены для небольших окон W_H, W_V размера 3×3 или 5×5 , использованных для определения ориентированного градиента \mathbf{G} (рис. 4, б).

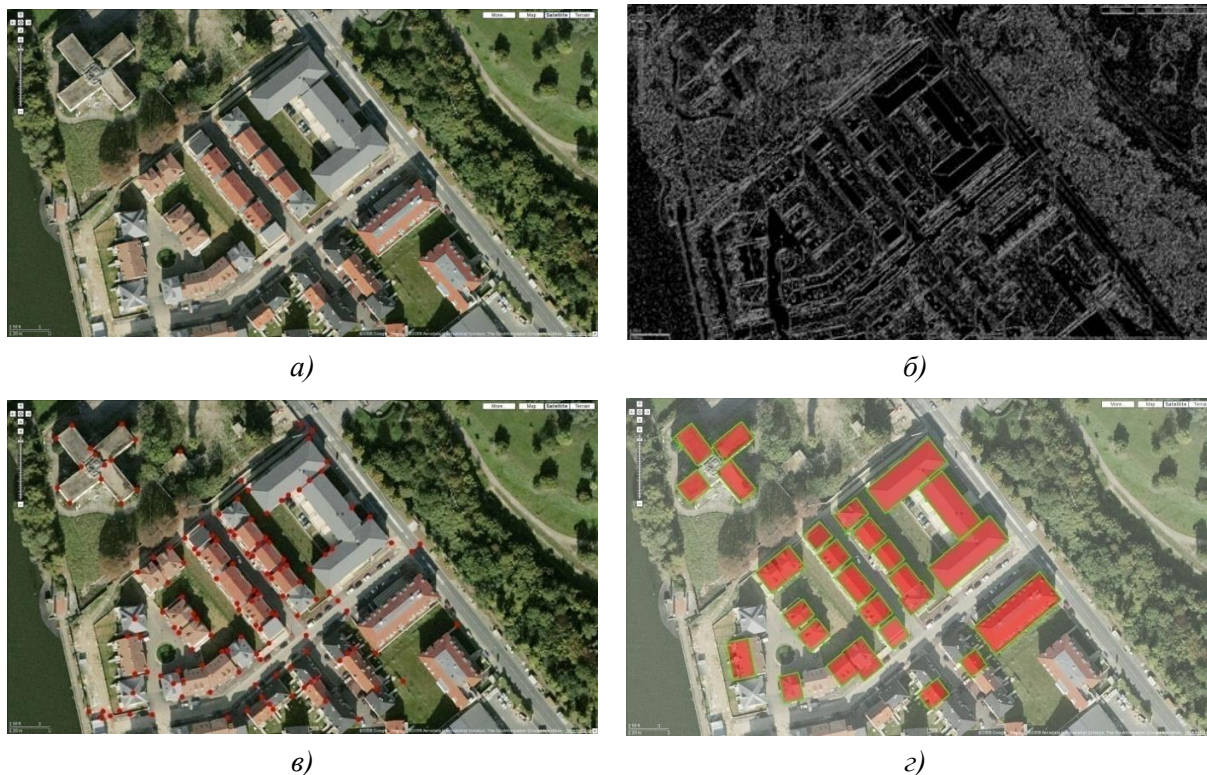


Рис. 4. Результаты работы алгоритма: а) аэрофотоснимок; б) изображение модуля ориентированного градиента \mathbf{G} ; в) результат определения прямых углов; г) результат определения положения зданий на изображении по углам алгоритмом отжига (simulated annealing)

Обнаружение углов на изображении 1540×860 пикселей на одном ядре процессора Core 2 2,13 ГГц заняло 6 мин 10 с.

Значения параметра τ_1 выбирались в интервале 20° , а параметра τ_2 — $\approx 40^\circ$. Полученные результаты позволяют сделать вывод о применимости алгоритма для поиска изображений домов на аэрофотоснимках. После несложного обучения алгоритм выделял более 95 % углов заданной величины. Точность оценки координат вершин углов составила примерно два-три пиксела. При этом на использованных для тестов изображениях он выделил углы крыш, что позволило обнаружить до 95 % зданий, не совершив ошибки второго рода — выделения ложных углов, а вследствие этого — и ложных зданий.

Заключение

Предложенный алгоритм обнаружения углов заданной формы на основе ориентированного градиента был использован как один из критериев для задачи выделения зданий и других объектов на космических и аэрофотоизображениях. Тщательное тестирование алгоритма показало его высокую надежность и пригодность для практического применения в задачах такого рода.

Для повышения количества найденных углов (в том числе и для обнаружения слабоконтрастных или зашумленных углов) в дальнейшем планируется усовершенствовать алгоритм, включив в него анализ содержания части изображения, находящегося внутри угла.

Список литературы

1. Moravec, H. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover / H. Moravec // Technical Report CMU-RI-TR-80-03. – Carnegie Mellon Univ. [Electronic resource] – 1980. – Mode of access : http://www.ri.cmu.edu/pub_files/pub4/moravec_hans_1980_1/moravec_hans_1980_1.pdf. – Date of access : 29.07.2011.
2. Harris, C. Combined Corner and Edge Detector / C. Harris, M. Stephens // Proc. of the 4th Alvey Vision Conf. – Manchester, UK, 1988. – P. 147–151.
3. Smith, S.M. SUSAN – a new approach to low level image processing / S.M. Smith, J.M. Brady // International Journal of Computer Vision. – 1997. – Vol. 23 (1). – P.45–78.
4. Baumberg, A. Reliable feature matching across widely separated views / A. Baumberg // Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. – Hilton Head Island, SC, USA, 2000. – P. 1774–1781.
5. Lindeberg, T. Feature detection with automatic scale selection / T. Lindeberg // International Journal of Computer Vision. – 1998. – Vol. 30 (2). – P. 77–116.
6. Mikolajczyk, K. Scale and affine invariant interest point detectors / K. Mikolajczyk, C. Schmid // International Journal of Computer Vision. – 2004. – Vol. 60 (1). – P. 63–86.
7. Trujillo, L. Synthesis of interest point detectors through genetic programming / L. Trujillo, G. Olague // Proc. of the 8th annual conference on Genetic and evolutionary computation GECCO '06. – Seattle, Washington, USA, 2006. – P. 887–894.
8. Rosten, E. Faster and Better: A Machine Learning Approach to Corner Detection / E. Rosten, R. Porter, T. Drummond // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – Vol. 32, № 1. – P. 105–119.
9. Rohr, K. Landmark-based image analysis: using geometric and intensity models / K. Rohr. – N.Y. : Springer, 2001. – 303 p.
10. Corner detection [Electronic resource]. – Mode of access : http://en.wikipedia.org/wiki/Corner_detection. – Date of access : 30.06.2011.
11. Viola, P. Robust Real-time Object Detection / P. Viola, M. Jones // International Journal of Computer Vision. – 2002. – Vol. 57 (2). – P. 137–154.
12. McDonnel, M. Box-filtering techniques / M. McDonnel // Computer Graphics and Image Processing. – 1981. – Vol. 17 (3). – P. 65–70.
13. Marimon, D. Combination of video-based camera trackers using a dynamically adapted particle filter / D. Marimon, T. Ebrahimi // Proc. 2nd Intern. Conf. on Computer Vision Theory and Applications (VISAPP07). – Barcelona, Spain, 2007. – P. 363–370.

Поступила: 30.08.11

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: {zalesky, pavel.lukashevich}@newman.bas-net.by*

B.A. Zalesky, P.V.Lukashevich

CORNER DETECTION ON THE IMAGE

The algorithms of automatic corner detection on grayscale and color images are presented. They are based on the use of the oriented gradient, and, unlike other known algorithms, not only identify reliably vertices of detected corners on the image, but also estimate orientation of their edges.

Application of the algorithms to extract buildings and other objects from satellite and aerial images showed their high reliability and suitability for practical use.