ОБРАБОТКА ИЗОБРАЖЕНИЙ

УДК 004

П.В. Лукашевич, Б.А. Залесский

МАСШТАБНО-ИНВАРИАНТНЫЙ АЛГОРИТМ ОБНАРУЖЕНИЯ ОБЛАСТЕЙ ИЗОБРАЖЕНИЙ

Описывается масштабно-инвариантный алгоритм обнаружения областей изображений по шаблону, основанный на сравнении их ориентированного градиента. Алгоритм, в частности, предназначается для нахождения в режиме реального времени областей аэро- и космических изображений, которые соответствуют кадрам видеопоследовательностей, снятых видеокамерой. Алгоритм проще и в несколько раз быстрее популярных в настоящее время алгоритмов SIFT и SURF. Он позволяет надежно находить области на изображениях (в том числе больших размеров), имеющих схожие характеристики с шаблонами.

Введение

Алгоритмы обнаружения областей на изображениях являются важным инструментом при решении широкого круга задач, включая поиск и регистрацию изображений, поиск объектов на изображениях и навигацию. Они также используются в различных областях информационных технологий, таких, как охрана окружающей среды, обнаружение изменений, обновление карт и т. д.

Задача поиска изображений понимается следующим образом: имеется изображение (например, аэро- или космическое), которое будем называть оригиналом или эталоном, требуется за реальное время найти на нем область, которая соответствует меньшему изображению, называемому шаблоном (в качестве шаблона, например, может выступать кадр видеопоследовательности, которая снята видеокамерой, установленной на борту самолета). Для кадров видеопоследовательности реальное время составляет $\approx 33 \div 40$ мс. Пространственное разрешение шаблона и угол поворота его сторон относительно эталона предполагаются неизвестными. Данная задача весьма актуальна в настоящее время. Несмотря на простоту ее формулировки, задача трудна, так как отсутствие информации о пространственном разрешении шаблона и его угле поворота относительно эталона приводит к необходимости сравнения этих изображений при всех допустимых масштабах и поворотах. В свою очередь, большое число сравнений приводит к необходимости выполнения очень большого числа операций, что делает невозможным использование известных, хорошо зарекомендовавших себя методов, без их глубокой и сложной оптимизации.

Наиболее популярные и эффективные в настоящее время алгоритмы поиска соответствия областей SIFT [1, 2] и SURF [3] решают сформулированную задачу путем сравнения так называемых особых точек, выделенных на разных масштабах изображения. В качестве особых точек могут выступать углы, места пересечений линий градиента или даже пятна определенной формы и размера [1–9]. Эти алгоритмы сначала строят масштабные пирамиды изображений [4, 5], затем выделяют особые точки на разных уровнях масштабных пирамид с помощью ориентированного градиента или гессиана, упорядочивают множества особых точек специальным образом, а затем ищут соответствие областей, устанавливая соответствие между особыми точками шаблона и оригинального изображения, используя алгоритмы быстрого поиска. Упомянутые SIFT и SURF не позволяют работать с изображениями больших размеров, в некоторых случаях не обеспечивают вычисления в режиме реального времени и используют большой объем памяти для хранения особых точек.

В статье предлагается алгоритм обнаружения областей по шаблону SICHT (Scale Invariant Compressed Histogram Transform), способный решать задачу на больших изображениях за реальное время. Алгоритм тестировался на эталонах размера 9000×9000 пикселов, хотя теоретически он может быть применен на изображениях большего размера благодаря использованной технике подвыборки, которая существенно уменьшает требуемую память и число операций при сохранении требуемой точности решения.

Очевидно, что формирование пространства признаков масштабной пирамиды оригинального изображения может быть выполнено заранее, тем не менее SIFT и SURF требуют для этого большего времени по сравнению с предложенным SICHT. Например, при использовании эталона размером 5110×5110 пикселов и шаблона размером 640×480 пикселов предварительная подготовка данных для эталона алгоритмом SURF (версия OpenCV) занимает 30,5 с, поиск соответствия – 4,8 с. SICHT требует для этого соответственно 18,5 с и 0,06 с. При этом данные, подготовленные SURF, занимают ≈ 200 МБ оперативной памяти, а SICHT – $\approx 2,8$ МБ.

В предложенном алгоритме SICHT сравнение оригинального изображения и шаблона производится на основе гистограмм ориентированного градиента (сокращенно HoG), который в общем случае вычисляется для некоторой регулярной подрешетки пикселов масштабной пирамиды оригинального изображения и шаблона.

Описание алгоритма SICHT состоит из следующих шагов. Сначала строится масштабная пирамида оригинального изображения для того, чтобы получить возможность сравнивать его с шаблоном в одинаковом пространственном разрешении. В случае если оригинальное изображение и шаблон имеют одинаковое пространственное разрешение, необходимость использования масштабной пирамиды отпадает. Затем фиксируется регулярная подрешетка пикселов оригинального изображения. Размер подрешетки выбирается так, чтобы обеспечить наиболее точное решение за реальное время. При использовании космических изображений шаг решетки выбирался от 2 до 32 пикселов. Оконные HoG вычисляются только для пикселов подрешетки оригинального изображения. Размер окна для вычисления HoG на каждом уровне пирамиды должен соответствовать размеру шаблона.

Каждая HoG сглаживается и выравнивается с помощью вычисления главного направления ее ориентированного градиента. После этого выровненная HoG сжимается в наперед заданное число раз и записывается на диск.

Операции, выполняемые в режиме реального времени, включают вычисление, сглаживание, выравнивание и сжатие HoG текущего шаблона и ее сравнение с записанным массивом HoG масштабной пирамиды оригинала.

Для повышения надежности и устойчивости решений задачи предложены несколько вариантов процедуры статистической кластеризации, основанной на кластерном анализе последовательности векторов – центров найденных областей соответствия. Смысл процедуры состоит в проверке для каждого текущего шаблона близости вектора-оценки его положения на оригинальном изображении к центру самого большого кластера, образованного несколькими векторами-оценками для предыдущих шаблонов.

Заметим, что предложенная статистическая процедура может быть включена в состав известных алгоритмов обнаружения областей по кадрам видеопоследовательностей. Тестирование показало применимость алгоритма для решения задач обнаружения областей.

1. Описание алгоритма

Одно из преимуществ современных алгоритмов обнаружения областей, включая SIFT и SURF, заключается в использовании особых точек, которые извлекаются из наиболее характерных частей оригинального изображения и шаблона. Во многих случаях удобнее использовать их вместо сравнения всех возможных областей (с учетом поворотов и масштабирования) оригинального изображения с шаблоном. При этом часто точность решения задачи обнаружения повышается.

Однако возникают трудности при использовании оригинальных изображений и шаблонов больших размеров или когда их пространственные разрешения отличаются более чем в четыре-пять раз. В таких случаях вычисление пространств особых точек может занять много часов, пространства могут оказаться настолько большими, что не хватает места в памяти для их сохранения, время сравнения особых точек оказывается больше допустимого.

Авторы столкнулись с упомянутыми трудностями в случае, когда в качестве оригинальных изображений использовались аэро- или космические снимки, а в качестве шаблонов – кадры видеопоследовательностей.

Для их преодоления и обеспечения возможности получать решение задачи обнаружения областей в режиме реального времени был разработан алгоритм SICHT.

Начнем описание алгоритма, обозначив через $S=\{0,\ldots,m-1\}\times\{0,\ldots,n-1\}$ множество пикселов $\mathbf{j}=(j_1, j_2)$ оригинального полутонового изображения $\mathbf{I}$ с интенсивностями $I_{\mathbf{j}}$, $\mathbf{j}\in S$. Так как цветные компоненты RGB-изображений обрабатываются SICHT независимо или даже используется только один цветовой канал, ниже будут упоминаться только полутоновые изображения.

Для того чтобы избежать ручного масштабирования изображений, используется масштабная пирамида $\Im=\{\mathbf{I}(0),\ldots,\mathbf{I}(k-1)\}$ оригинального изображения. Она образована отмасштабированными копиями оригинального изображения, которые для подходящего масштабирующего множителя $0<\rho<1$ задаются равенствами

$$\mathbf{I}(\ell)=\rho^{\ell}\mathbf{I},\quad \ell=0,\ldots,k-1, \tag{1}$$

или для $0<\tau$

$$\mathbf{I}(\ell)=\left(1+\frac{\ell\tau}{k}\right)^{-1}\mathbf{I},\quad \ell=0,\ldots,k-1, \tag{2}$$

и имеют размеры $m(\ell)=\left[\rho^{\ell}m\right]$, $n(\ell)=\left[\rho^{\ell}n\right]$ для схемы (1) и $m(\ell)=\left[\left(1+(\ell\tau)/k\right)^{-1}m\right]$, $n(\ell)=\left[\left(1+(\ell\tau)/k\right)^{-1}n\right]$ для схемы (2) (функция $[\ ]$ означает целую часть числа). Множества пикселов отмасштабированного изображения $\mathbf{I}(\ell)$ будем обозначать $S(\ell)=\{0,\ldots,m(\ell)-1\}\times\{0,\ldots,n(\ell)-1\}$. В общем случае при построении пирамиды $\Im$ могут быть использованы отрицательные значения ℓ, однако всегда можно перенумеровать изображения так, чтобы наибольшее из них имело номер $\ell=0$. Методы аккуратного построения пирамид изображений описаны в [4, 5]. Предлагаемый алгоритм так же, как и SIFT и SURF, основан на использовании оконного градиента пирамиды оригинального изображения и шаблона. Градиент изображения $\mathbf{I}(\ell)$ понимается как $m(\ell)\times n(\ell)$-матрица $\mathbf{G}(\ell)$ с элементами-векторами $\mathbf{G}(\ell)=\left\{g_{H,\mathbf{j}}(\ell), g_{V,\mathbf{j}}(\ell)\right\}_{j\in S}$, где $g_{H,\mathbf{j}}(\ell)$ и $g_{V,\mathbf{j}}(\ell)$ – горизонтальные и вертикальные компоненты оконного градиента. Окна W_H и W_V представляют собой матрицы весовых коэффициентов для подсчета градиента, конкретные размеры которых указаны в разд. 2. В рассматриваемом случае окна выбираются одинаковыми для всех изображений $\mathbf{I}(\ell)$ пирамиды (рис. 1).

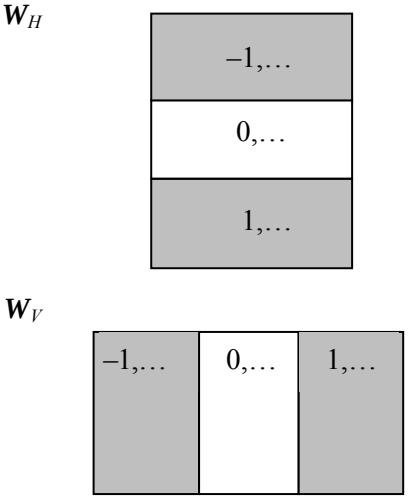

Рис. 1. Форма окон оконного градиента

Таким образом,

$$g_{H,\,\mathbf{j}}(\ell) = \frac{1}{|W_H|} \sum_{\mathbf{m}\in W_H} I_{\mathbf{m}}(\ell)\; w_{H,\mathbf{m}-\mathbf{j}}$$

и

$$g_{V,\,\mathbf{j}}(\ell) = \frac{1}{|W_V|} \sum_{\mathbf{m}\in W_V} I_{\mathbf{m}}(\ell)\; w_{V,\,\mathbf{m}-\mathbf{j}},$$

где $w_{H,\mathbf{k}-\mathbf{j}}$ и $w_{V,\mathbf{k}-\mathbf{j}}$ – элементы оконных матриц.

При вычислении HoG используются углы наклона $\beta_{\mathbf{j}}(\ell)$ векторов градиента $g_{H,\mathbf{j}}(\ell), g_{V,\mathbf{j}}(\ell)$, равные $\beta_{\mathbf{j}}(\ell) = \left[\arctan\left(\frac{g_{V,\mathbf{j}}(\ell)}{g_{H,\mathbf{j}}(\ell)}\right)\right]$.

Для обеспечения возможности работы в режиме реального времени с эталонами большого размера, которые нередко имеют высокое пространственное разрешение, в SICHT предлагается производить поиск областей только на регулярной подрешетке пикселов эталона. Использование подрешетки – «плата» за быстрые вычисления. Обычно шаг подрешетки выбирается так, чтобы достичь максимально возможной точности за реальное время. В проведенных тестах, в которых в качестве эталона были использованы космические снимки, шаг подрешетки выбирался от 2 до 32.

Вычисление градиента шаблона должно занимать как можно меньше времени, поэтому имеет смысл проводить его методом «бегущей строки» [10] или использовать интегральное изображение.

Пусть целое q – шаг подрешетки, тогда регулярная подрешетка SS имеет вид $SS=\{0, q, \ldots, i_1 q, \ldots\}\times\{0, q, \ldots, i_2 q, \ldots\}$ и состоит из пикселов основной решетки $\mathbf{j}\in \mathrm{S}$ вида $\mathbf{j}=(i_1 q, i_2 q)$. Регулярные подрешетки $SS(\ell) = SS \bigcap S(\ell)$ с одинаковым шагом q используются для всех изображений пирамиды изображений $\Im$.

Обозначим через $O = \{0, \ldots, \mu-1\}\times\{0, \ldots, \nu-1\}$ множество пикселов прямоугольного шаблона, предполагая, что $\mu < m, \nu < n$. Обозначим также через $O_{\mathbf{j}}$ множество пикселов изображения $\mathbf{I}(\ell)$, имеющее форму $\mu\times\nu$-прямоугольника с центром в пикселе $\mathbf{j}$ (напомним, что на всех изображениях пирамиды $\Im$ для подсчета HoG используются окна одинакового размера).

Гистограмма ориентированного градиента отмасштабированного эталона $\mathbf{I}(\ell)$ в пикселе $\mathbf{j}$ представляет собой одномерный вектор размерности 360

$$\mathbf{h}_{\mathbf{j}}(\ell) = \left(h_{j,0}(\ell), h_{j,1}(\ell), \ldots, h_{j,359}(\ell)\right)$$

с координатами

$$h_{j,\,i}(\ell) = \sum_{\mathbf{m}\in O_{\mathbf{j}}} \mathbf{1}_{\{\beta_{\mathbf{m}}(\ell)=i\}},$$

где индикаторная функция $\mathbf{1}_{\{\Theta\}}$ равна 1, если условие Θ выполняется, и равна 0 в противном случае. Гистограмма ориентированного градиента шаблона $\mathbf{h}$ вычисляется аналогично.

Эталон и шаблон в общем случае отличаются не только пространственным разрешением, но и поворотом относительно друг друга, поэтому перед сравнением их HoG должны быть вы-

ровнены для того, чтобы избежать сравнения всех циклических сдвигов одной HoG относительно другой.

Простейший способ выравнивания гистограмм ориентированного градиента заключается в нахождении их максимального значения и циклического сдвига вектора HoG таким образом, чтобы она начиналась со своего максимума. Однако HoG «выглядят не гладко» (рис. 2), поэтому их непосредственное выравнивание приводит к возникновению существенного числа ошибок определения главного направления градиента в окне, а это, в свою очередь, приводит к значительному числу неправильно найденных соответствий.

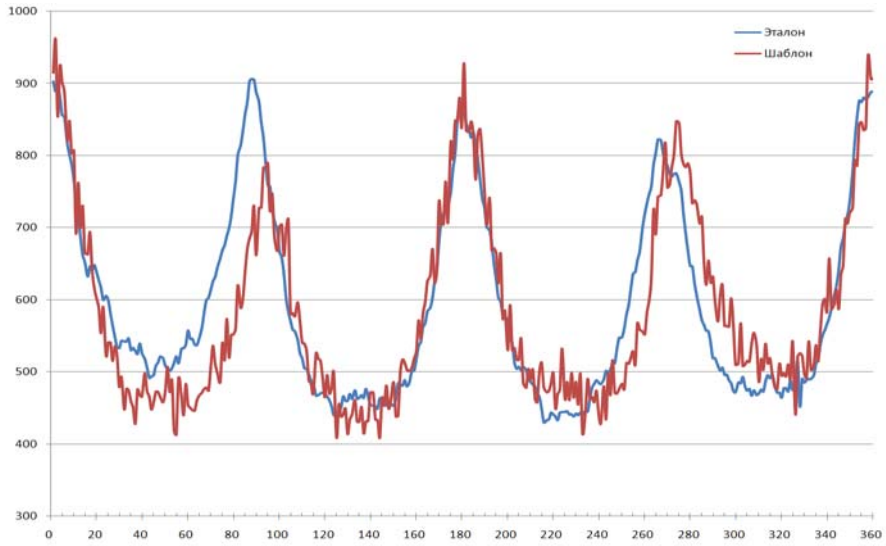

Рис. 2. Выровненные по максимуму HoG шаблона
и соответствующей ему области эталона

Уменьшение количества ошибок при выравнивании HoG достигается путем их сглаживания. Даже простейшее сглаживание с помощью скользящего среднего дает заметное уменьшение числа неправильно найденных областей эталона.

Обозначим через

$$\overline{\mathbf{h}}_{\mathbf{j}}(\ell) = \mathbf{h}_{\mathbf{j}}(\ell) \otimes \mathbf{a}, \qquad \overline{\mathbf{h}} = \mathbf{h} \otimes \mathbf{a},$$

циклические свертки HoG эталона и шаблона с усредняющим вектором

$$\mathbf{a} = \frac{1}{d}\underbrace{(1,\ldots,1)}_{d}.$$

Тогда первые координаты сглаженных выровненных HoG $\widetilde{\mathbf{h}}_{\mathbf{j}}(\ell)$ эталона и $\widetilde{\mathbf{h}}$ шаблона удовлетворяют соотношениям

$$\widetilde{h}_{j,0}(\ell) = \max_{0 \le i \le 259} \left\{ \overline{h}_{\mathbf{j},i}(\ell) \right\} \text{ и } \widetilde{h}_0 = \max_{0 \le i \le 259} \left\{ \overline{h}_i \right\}.$$

Замечание 1. Несколько видоизмененная последовательность действий: а) сначала с помощью метода скользящего среднего с окном шириной до 20÷30 градусов находятся максимумы сглаженных гистограмм oG; б) исходные несглаженные гистограммы выравниваются по максимуму осредненной скользящим средним HoG; в) выровненные несглаженные гистограммы осредняются методом скользящего среднего с маленьким окном от 3 до 10 градусов – дает несколько более точный результат.

Для нахождения на эталоне области, соответствующей шаблону, SICHT сравнивает сглаженные выровненные HoG $\widetilde{\mathbf{h}}_{\mathbf{j}}(\ell)$ масштабных изображений эталона для пикселов $\mathbf{j}$ из подрешеток $SS(\ell)$ со сглаженной выровненной HoG $\widetilde{\mathbf{h}}$ шаблона. Центр области эталона, которая считается соответствующей текущему шаблону, находится с помощью соотношения

$$\mathbf{j}^* = \underset{0\le\ell\le k,\ \mathbf{j}\in SS(\ell)}{\arg\max} \left\| \widetilde{\mathbf{h}}_{\mathbf{j}}(\ell) - \widetilde{\mathbf{h}} \right\|, \tag{3}$$

где $\|\ \|$ – евклидова норма.

Так как эталон и шаблон часто имеют разные характеристики, такие, как яркость, контраст, содержание, решение (3) может давать некоторое число ошибочных соответствий. Использование двух или трех выравниваний на основе наибольших локальных максимумов $\overline{\mathbf{h}}_{\mathbf{j}}(\ell)$ и $\overline{\mathbf{h}}$ позволяет получить более точные на несколько процентов результаты.

Процесс сравнения сглаженных выровненных HoG $\widetilde{\mathbf{h}}_{\mathbf{j}}(\ell)$ эталона и $\widetilde{\mathbf{h}}$ текущего шаблона занимает большую часть времени, доступного алгоритму для обработки одного кадра, равного $\approx 0{,}03 \div 0{,}04$ с. Известно несколько способов его ускорения. Один из них заключается в использовании параллельных вычислений, второй – в применении SIMD-инструкций, третий – в сжатии векторов HoG, позволяющем уменьшить число операций. Практические вычисления показали, что среди нескольких подходов к сжатию векторов наиболее простой алгоритм квантизации позволяет уменьшить время вычисления во много раз, сохраняя высокую точность результатов обнаружения. Смысл использованного алгоритма состоит в следующем. Для некоторого целого числа δ, являющегося множителем 360, вычисляются сжатые векторы HoG $\widetilde{\mathbf{h}}^c_{\mathbf{j}}(\ell)$ и $\widetilde{\mathbf{h}}^c$ размерности $d = 360/\delta$ по простым формулам

$$\widetilde{h}^c_{\mathbf{j},i}(\ell) = \sum_{\gamma=i\delta}^{(i+1)\delta-1} \widetilde{h}_{\mathbf{j},\gamma}(\ell)$$

и

$$\tilde{h}_i = \sum_{\gamma=i\delta}^{(i+1)\delta-1} \tilde{h}_\gamma, \quad i = 0, 1, ..., d\,.$$

Поиск соответствия с помощью сжатых HoG производится на основе соотношения

$$\mathbf{j}^\circ = \underset{0\le\ell\le k,\ \mathbf{j}\in SS(\ell)}{\arg\max} \left\| \widetilde{\mathbf{h}}^c_{\mathbf{j}}(\ell) - \widetilde{\mathbf{h}}^c \right\|. \tag{4}$$

Соотношение (4) может вычисляться не только для выровненных по максимуму HoG, но также для нескольких их выравниваний, например, по первому, второму и третьему максимумам.

Зависимость решений от текущего времени t будем обозначать $\mathbf{j}^*(t), \mathbf{j}^\circ(t)$. Для тестовых видеопоследовательностей точность оценок (3), (4), которая понимается как доля кадров-шаблонов, корректно обнаруженных на эталоне, достигает 85 %. Тем не менее она была повышена с помощью реализованной в алгоритме процедуры статистической кластеризации оценок, выполняемой для последовательности решений. Для описания предложенной процедуры для некоторого целого Δ зафиксируем последовательность оценок центров найденных на эталоне областей соответствия шаблонам

$$\mathbf{j}^*(t), \mathbf{j}^*(t-1), ..., \mathbf{j}^*(t-\Delta)\,. \tag{5}$$

Последовательность решений $\mathbf{j}^{\circ}(t)$ может быть обработана по аналогии. Из формулировки задачи вытекает, что последовательные решения задачи должны располагаться близко друг к другу. Данное свойство позволяет уточнить текущее решение $\mathbf{j}^{*}(t)$ на основе предыдущих. Были разработаны несколько процедур кластеризации решений $\mathbf{j}^{*}(t)$, которые показали хорошие результаты.

Опишем одну из кластерных процедур коррекции текущего решения на основе последовательности (5). Параметрами процедуры является радиус кластеров R, который выбирается в зависимости от характеристик камеры и скорости ее движения; целое число $L\,(\Delta/2 \le L \le \Delta)$, задающее минимальный размер кластера, созданного из последовательности (5) для проверки гипотезы о корректности текущей оценки $\mathbf{j}^{*}(t)$, и N_{out} – число допустимых последовательных выбросов среди $\mathbf{j}^{\circ}(t)$ для остановки текущего и начала нового цикла процедуры кластеризации.

Предложенная процедура стартует в момент времени $t = \Delta$ с процесса обучения. Одно из решений

$$\mathbf{j}^{*}(\Delta), \mathbf{j}^{*}(\Delta-1), ..., \mathbf{j}^{*}(0), \tag{6}$$

например $\mathbf{j}^{*}(0)$, выбирается в качестве центра $\mathbf{c}_1 = \mathbf{j}^{*}(0)$ первого кластера $C_1 = \left\{\mathbf{j}^{*}(0)\right\}$. Затем производится проверка близости остальных векторов $\mathbf{j}^{*}(\tau), \tau = 1, ..., \Delta$, к центру $\mathbf{c}_1$ первого кластера C_1. Все $\mathbf{j}^{*}(\tau)$, удовлетворяющие условию

$$\left\|\mathbf{c}_1 - \mathbf{j}^{*}(\tau)\right\| \le R, \tag{7}$$

добавляются в кластер C_1. Первый вектор $\mathbf{j}^{*}(\tau)$, не удовлетворяющий (7), фиксируется как центр $\mathbf{c}_2$ второго кластера $C_2 = \left\{\mathbf{j}^{*}(\tau)\right\}$. Оставшиеся векторы последовательности (6) проверяются условием

$$\left\|\mathbf{c}_i - \mathbf{j}^{*}(\tau)\right\| \le R, i = 1, 2. \tag{8}$$

Если условие (8) выполняется для некоторого i, вектор $\mathbf{j}^{*}(\tau)$ добавляется в кластер C_i, иначе он рассматривается как центр третьего кластера C_3. Процедура обучения повторяется по аналогии для оставшихся векторов последовательности (6). После ее окончания образуется некоторое число кластеров $C_1, C_2, ..., C_p$. Если для некоторого i' число векторов в кластере $C_{i'}$ превосходит L, т. е.

$$card(C_{i'}) > L \tag{9}$$

(заметим, что только один кластер может удовлетворять данному условию), процесс обучения останавливается, иначе он продолжается для большей последовательности

$$\mathbf{j}^{*}(\Delta+1), \mathbf{j}^{*}(\Delta), \mathbf{j}^{*}(\Delta-1), ..., \mathbf{j}^{*}(1)$$

и т. д., пока не выполнится условие (9).

После успешного окончания процесса обучения кластерного алгоритма устойчивое решение в момент времени Δ, которое обозначается $\mathbf{j}^{*}_{rob}(\Delta)$, вычисляется по формуле

$$\mathbf{j}^{*}_{rob}(\Delta) = \frac{1}{card(C_{i'})} \sum_{\mathbf{j} \in C_{i'}} \mathbf{j}.$$

В следующий момент времени $\Delta+1$ начинается рабочая фаза алгоритма кластеризации. Счетчик числа ошибочно найденных соответствий n_{out} полагается равным 0, т. е. $n_{out} := 0$. Проверяется условие

$$\left\| \mathbf{j}^*_{rob}(\Delta) - \mathbf{j}^*(\Delta+1) \right\| \le R. \tag{10}$$

Если оно выполняется, полагаем

$$\mathbf{j}^*_{rob}(\Delta+1) = \rho \mathbf{j}^*_{rob}(\Delta) + (1-\rho)\mathbf{j}^*(\Delta+1) \tag{11}$$

для некоторого наперед заданного числа $0 < \rho < 1$, которое выбирается в соответствии с качеством эталона и шаблона.

Если условие (10) не выполнилось, вектор $\mathbf{j}^*_{rob}(\Delta+1)$ с большой вероятностью был вычислен неточно. Счетчик числа выбросов увеличивается на 1: $n_{out} := n_{out} + 1$, и предыдущая устойчивая оценка берется в качестве текущей: $\mathbf{j}^*_{rob}(\Delta+1) = \mathbf{j}^*_{rob}(\Delta)$. Рабочая фаза алгоритма продолжается, пока выполняется условие $n_{out} \le N_{out}$. Условие (10) проверяется для $\mathbf{j}^*(t-1)$ и $\mathbf{j}^*(t)$, и если оно выполняется, текущее устойчивое решение $\mathbf{j}^*_{rob}(t)$ полагается равным

$$\mathbf{j}^*_{rob}(t) = \rho \mathbf{j}^*_{rob}(t-1) + (1-\rho)\mathbf{j}^*(t),$$

иначе счетчик числа ошибок увеличивается на 1, т. е. $n_{out} := n_{out} + 1$, и

$$\mathbf{j}^*_{rob}(t) = \mathbf{j}^*_{rob}(t-1).$$

Как только счетчик числа ошибок n_{out} становится больше, чем N_{out}, рабочая фаза алгоритма прекращается и снова начинается процесс обучения.

Замечание 2. Оценка $\mathbf{j}^*_{rob}(t)$ является смещенной – она немного отстает от координат реального соответствия, однако при этом обладает существенно большей устойчивостью по сравнению с $\mathbf{j}^*(t)$ и точностью, достаточной для решения практических задач.

Очевидно, что в данном случае могут быть использованы другие статистические алгоритмы, в том числе и более простые, однако приведенный подход позволяет не только повысить точность результатов, но и оценить долю ошибочно найденных соответствий.

Приведем пошаговое представление SICHT:

Шаг 1. Вычислить ориентированный градиент (oG) $g_{H,\mathbf{j}}(\ell), g_{V,\mathbf{j}}(\ell)$ изображения-эталона и на его основе HoG $\mathbf{h}_{\mathbf{j}}(\ell)$ пирамиды эталона.

Шаг 2. Сгладить HoG $\mathbf{h}_{\mathbf{j}}(\ell)$ изображения-эталона (получить $\overline{\mathbf{h}}_{\mathbf{j}}(\ell)$). Сглаженные HoG $\overline{\mathbf{h}}_{\mathbf{j}}(\ell)$ выровнять по максимальному значению, т. е. вычислить $\widetilde{\mathbf{h}}_{\mathbf{j}}(\ell)$.

Шаг 3. Вычислить сжатые выровненные HoG $\widetilde{\mathbf{h}}^c_{\mathbf{j}}(\ell)$ изображения-эталона.

Шаг 4. Повторить шаги 1–3 для шаблона, т. е. вычислить oG шаблона, затем его HoG $\mathbf{h}$, сгладить его (найти $\overline{\mathbf{h}}$), выровнять по максимуму (найти $\widetilde{\mathbf{h}}$) и сжать, получив в итоге $\widetilde{\mathbf{h}}^c$ шаблона.

Шаг 5. Найти соответствие $\mathbf{j}^*$ по формуле (3) или $\mathbf{j}^\circ$ на основе (4).

Шаг 6. Скорректировать решение с помощью кластеризационной процедуры; иными словами, найти $\mathbf{j}^*_{rob}$.

Шаги 1–3 алгоритма являются предварительными – они могут быть выполнены заранее, поэтому не требуют режима реального времени. Шаги 4–6 выполняются для каждого текущего шаблона, в силу чего они должны выполняться как можно быстро.

2. Эксперименты

Для экспериментов в качестве эталонов были использованы космические изображения и аэрофотоснимки (рис. 3). Из-за отсутствия реальной видеопоследовательности, соответствующей данному космическому изображению, была применена видеопоследовательность, полученная камерой, которая движется над бумажной копией этого изображения. Были также использованы видеопоследовательности, снятые виртуальной видеокамерой, которая движется над космическим изображением по заданному криволинейному маршруту переменной высоты.

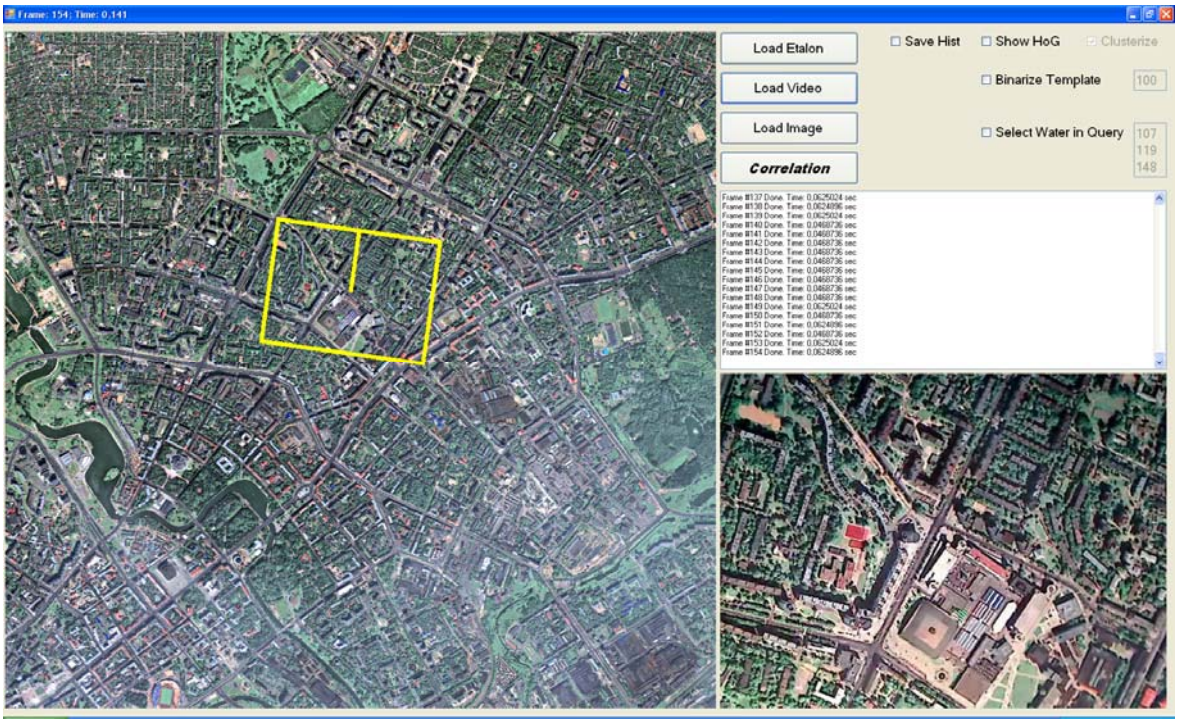

Рис. 3. Снимок экрана экспериментальной программы: оригинальное космическое изображение-эталон расположено в левой части экрана, кадр видеопоследовательности, использованной в качестве шаблона, – в правой части; найденное соответствие выделено желтым прямоугольником

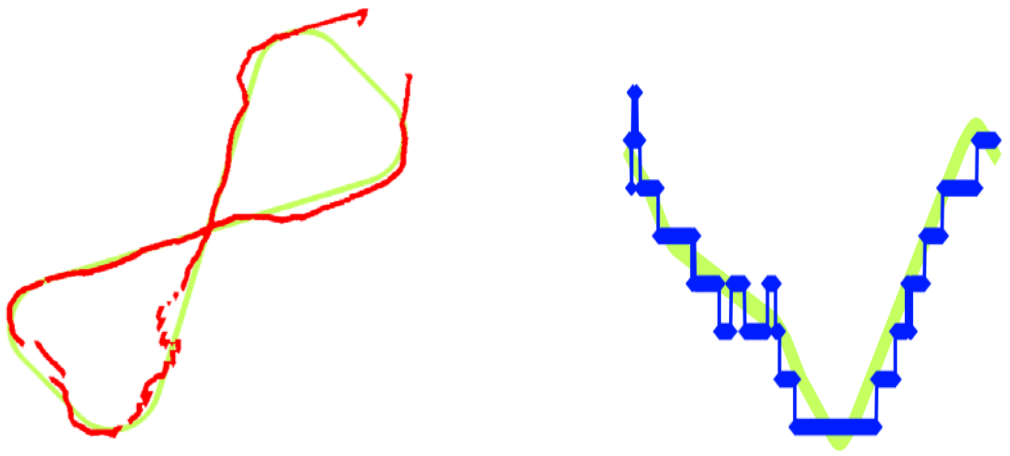

Рис. 4. Результаты тестирования алгоритма

Размеры космических изображений менялись от $10^3 \times 10^3$ до $10^4 \times 10^4$ пикселов. Разрешение камеры – 640×480 пикселов. Соотношение пространственных разрешений эталонов и шаблонов менялось от 1 до 5.

Эксперименты с видеопоследовательностями, снятыми обычной и виртуальной камерами, дали удовлетворительные результаты. Более чем 90 % кадров каждой видеопоследовательности были оценены в устойчивом рабочем режиме кластерного алгоритма с точностью, превышающей 100 пикселов (рис. 3 и 4).

В левой части рис. 4 зеленым цветом показана траектория движения камеры, а красным – оценка траектории, сделанная алгоритмом; справа изображена реальная высота камеры в разные моменты времени и ее оценки, полученные алгоритмом.

Следует отметить, что точность алгоритма существенно зависит от освещенности – она высока при дневном освещении и уменьшается при уменьшении освещенности шаблона.

Предложенный SICHT сравнивался с версией популярного в настоящее время алгоритма SURF, опубликованной в OpenCV-библиотеке. Тестирование показало, что SICHT имеет в несколько раз большее быстродействие, позволяющее решать задачу обнаружения в режиме реального времени, при этом он обладает примерно одинаковой с SURF точностью при работе с видеопоследовательностью и, кроме того, требует меньшего объема памяти на диске.

Заключение

В статье предложен масштабно-инвариантный алгоритм SICHT обнаружения областей изображений по изображению-шаблону, основанный на сравнении их ориентированного градиента. В отличие от популярных в настоящее время алгоритмов SIFT и SURF он не сравнивает локально инвариантные особенности изображений, а использует сжатые гистограммы ориентированного градиента. Для обеспечения работы алгоритма в режиме реального времени в нем использована техника работы на подрешетках пикселов. Даже при использовании разреженных подрешеток, например с шагом 16 или 32 пиксела, алгоритм дает достаточно точные оценки за реальное время.

К недостаткам SICHT следует отнести его чувствительность к изменению свойств изображений: контрастности, близости содержания.

В будущем для повышения точности планируется усовершенствовать процесс сравнения гистограмм ориентированного градиента, включить в состав алгоритма процедуры верификации результатов, например корреляционные.

Список литературы

1. Lowe, D. Object recognition from local scale-invariant features / D. Lowe // Proc. of the Intern. Conf. on Computer Vision. – Washington, DC, USA, 1999. – Vol. 2. – P. 1150–1157.

2. Scovanner, P. A 3-dimensional sift descriptor and its application to action recognition / P. Scovanner, S. Ali, M. Shah // Proc. of the 15th Intern. Conf. on Multimedia. – N.Y., USA, 2007. – P. 357–360.

3. SURF: Speeded Up Robust Features / H. Bay [et al.] // Computer Vision and Image Understanding (CVIU). – 2008. – Vol. 110, № 3. – P. 346–359.

4. Lindeberg, T. Scale-space theory: A basic tool for analysing structures at different scales / T. Lindeberg // Journal of Applied Statistics (Supplement on Advances in Applied Statistics: Statistics and Images). – 1994. – Vol. 21 (2). – P. 224–270.

5. Lindeberg, T. Scale-space / T. Lindeberg // Encyclopedia of Computer Science and Engineering. – 2008. – Vol. IV. – P. 2495–2504.

6. Robust wide baseline stereo from maximally stable extremum regions / J. Matas [et al.] // Proc. Brit. Machine Vision Conf. – Cardiff, UK, 2002. – P. 384–393.

7. Lindeberg, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention / T. Lindeberg // Int. J. Comput. Vis. – 1993. – Vol. 11. – P. 283–318.

8. Lindeberg, T. Feature detection with automatic scale selection / T. Lindeberg // Int. J. Comput. Vis. – 1998. – Vol. 30. – P. 77–116,.

9. A KFCM and SIFT Based Matching Approach to Similarity Retrieval of Images / P. Hao [et al.] // Proc. of the Fifth Intern. Conf. on Image and Graphics (ICIG12009). – Xi1an, Shanxi, China, 2009. – P. 372–377.

10. Porikli, F. Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces / F. Porikli // Proc. of the 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (Cvpr'05). – Washington, 2005. – Vol. 1. – P. 829–836.

Поступила 01.07.11

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: Pavel.Lukashevich@newman.bas-net.by,
zalesky@newman.bas-net.by

P.V. Lukashevich, B.A. Zalessky

SCALE INVARIANT ALGORITHM FOR MATCHING IMAGE REGIONS

A scale invariant algorithm is presented for matching regions of aero and satellite images by means of comparison of local histograms of oriented gradients (HoG). In particular, it solves the task of finding in real time regions of aero and satellite images that correspond to frames of videosequences produced by a video camera. Computing and analysis of HoG are simpler than finding and extraction of appropriate image keypoints forming feature spaces of SIFT and SURF, therefore, the presented algorithm has simpler flowchart and it is several time faster than the above mentioned algorithms. It also provides high reliable matching of image sequences with the properties similar to properties of the original image.