

Исследование алгоритмов обнаружения и локализации объекта на изображении в условиях структурных искажений*

Д. С. ПЕРЕВАЛОВ

Институт математики и механики УрО РАН, Екатеринбург, Россия
e-mail: perevalovds@mail.ru

Предложен новый метод построения упрощенных моделей для исследования устойчивости качества алгоритмов обнаружения и локализации объектов к различным искажениям входных изображений.

Ключевые слова: обработка изображений, качество работы алгоритма, локализация объекта.

Введение

При разработке систем технического зрения часто возникает задача обнаружения и локализации объекта на изображении. Она заключается в проверке наличия на входном изображении некоторого объекта и вычислении его геометрических характеристик типа положения и угловой ориентации [1–3]. При разработке такого алгоритма важнее всего оценка качества его работы [4, 5]. Такая оценка обычно формулируется в терминах вероятностей ошибок алгоритма.

Как правило, оценки этих вероятностей получают эмпирически, с помощью тестирования алгоритма по некоторой базе примеров [4]. Такой способ универсален и позволяет сравнивать показатели работы несколько алгоритмов. Кроме того, накладывая шум на изображения, можно вычислять изменение качества алгоритма в зависимости от уровня зашумления. Но при использовании этого подхода трудно оценить воздействие структурных изменений типа увеличения/уменьшения числа посторонних объектов в сцене. Общим методом, позволяющим испытать влияние произвольных параметров на алгоритм, является моделирование входных данных с помощью метода Монте-Карло [5]. К сожалению, пока нет общей теории, которая бы позволила быстро строить точные модели для произвольных алгоритмов анализа изображений.

В работе предлагается метод построения упрощенных моделей для исследования устойчивости алгоритмов обнаружения и локализации объектов к различным типам искажений, в том числе структурных. Он применим к достаточно широкому классу алгоритмов, работу которых можно представить в виде двух этапов [6, 7]:

- 1) поиск признаков на изображении;
- 2) анализ их геометрического расположения.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 06-01-00229).

© Институт вычислительных технологий Сибирского отделения Российской академии наук, 2009.

Предлагается построить вероятностную модель появления признаков на изображении и, используя эту модель, с помощью метода Монте-Карло вычислить оценки вероятностей ошибок для разных значений параметров модели. На основе полученных оценок можно сделать качественное заключение об устойчивости алгоритма на интересующей области значений параметров. Для количественного описания устойчивости алгоритма предлагается характеристика, не зависящая от масштаба измерения интересующих параметров.

Предложенный подход проиллюстрирован на примере прикладной задачи распознавания изображения автосцепки [8].

1. Предварительные сведения и обозначения

Качество алгоритма обнаружения и локализации естественно описывать тремя вероятностями [4]:

- 1) P_{ta} — вероятность правильного обнаружения (true acceptance) объекта на изображении при его фактическом присутствии в сцене;
- 2) P_{II} — вероятность ошибки второго рода или ложной тревоги, когда алгоритм обнаруживает объект при его фактическом отсутствии в сцене;
- 3) P_{loc} — вероятность правильного обнаружения и правильной локализации объекта при наличии объекта в сцене.

Для определения этих величин необходимо знание распределения вероятностей появления всех возможных входных изображений, что в практических задачах недостижимо. Поэтому вместо вероятностей P_{ta} , P_{II} , P_{loc} будем использовать частоты F_{ta} , F_{II} , F_{loc} соответствующих событий по некоторому числу испытаний.

Далее будем считать, что зафиксирована некоторая задача T обнаружения и локализации объекта на изображении, которая заключается в проверке наличия на входном изображении определенного объекта и вычислении его геометрических характеристик. При этом предполагается, что на изображении может быть не более одного интересующего объекта. Последнее предположение — не слишком жесткое ограничение, так как к задаче в такой постановке сводится общая задача обнаружения нескольких объектов, принадлежащих разным классам.

Будем говорить, что некоторый алгоритм A решает задачу T , если его входом является изображение \mathbf{I} , а выходом — конечномерный вектор геометрических характеристик $\mathbf{V} = (x, y, \varphi, \dots)$, который описывает расположение объекта на изображении. Состав компонент вектора \mathbf{V} определяется требованиями конкретной задачи. Например, в него могут входить центр объекта (x, y) , угловая ориентация φ и другие характеристики. Если объект на изображении не обнаружен, алгоритм выдает $\mathbf{V} = 0$.

Нас будут интересовать лишь алгоритмы, представимые в виде композиции двух алгоритмов A_1 и A_2 :

$$A = A_2 \circ A_1.$$

Алгоритм A_1 осуществляет поиск на \mathbf{I} признаков нескольких классов. Результатом его работы является множество

$$\mathbf{L} = \{\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_n\},$$

где $\mathbf{L}_i = (s_i, x_i, y_i, \varphi_i, \dots)$ — вектор описания i -го признака. Первая компонента этого вектора $s_i \in \mathbf{N}$ соответствует номеру класса, которому принадлежит признак, а все

остальные компоненты есть вещественные числа, описывающие геометрические характеристики этого признака. Алгоритм A_2 формирует по \mathbf{L} вектор \mathbf{V} .

Пример. Рассмотрим следующую задачу обнаружения квадрата на изображении. Предполагается, что стороны квадрата параллельны осям координат изображения. Требуется найти центр квадрата. В качестве A_1 можно взять алгоритм обнаружения углов четырех классов с помощью поиска локальных экстремумов соответствующих четырех шаблонов. Каждый найденный признак характеризуется вектором $\mathbf{L}_i = (s_i, x_i, y_i)$, где $s_i = 1, \dots, 4$ — номер класса признака, x_i, y_i — координаты его центра. В качестве A_2 можно взять алгоритм, который ищет в \mathbf{L} такие четверки признаков $(\mathbf{L}_a, \mathbf{L}_b, \mathbf{L}_c, \mathbf{L}_d)$, что $s_a = 1, s_b = 2, s_c = 3, s_d = 4$, и их координаты лежат с заданной точностью на соответствующих углах некоторого квадрата. Если такая четверка в точности одна, то в качестве \mathbf{V} выдается вектор

$$\frac{1}{4}(x_a + x_b + x_c + x_d, y_a + y_b + y_c + y_d),$$

в противном случае $\mathbf{V} = 0$.

2. Вероятностная модель обнаружения признаков

Для исследования качества работы алгоритма $A = A_2 \circ A_1$ предлагается построить модель, описывающую обнаружение алгоритмом A_1 признаков на изображении как случайный процесс. Желательно, чтобы модель имела несколько управляющих параметров $\mathbf{H} = (H_1, \dots, H_m)$. Эти параметры могут характеризовать различные типы искажений входных изображений, в том числе структурные. Например, это могут быть степень расфокусировки и количество посторонних объектов на изображении.

Построение точной модели прикладного алгоритма — трудная задача, превышающая по сложности разработку самого алгоритма. Поэтому предлагается использовать упрощенную модель, в которой построение признаков ведется последовательно.

Обозначим через $\mathbf{U} = (x_0, y_0, \varphi_0, \dots)$ вектор фактических геометрических характеристик объекта на изображении. (Компоненты вектора \mathbf{U} соответствуют компонентам выходного вектора \mathbf{V} в определении алгоритма A .) Тогда случайный процесс $\mathcal{L}(\mathbf{U}, \mathbf{H})$, который по заданным \mathbf{U} и \mathbf{H} строит множество \mathbf{L} , можно описать так:

- 1) $n = n(\mathbf{U}, \mathbf{H})$;
- 2) $\mathbf{L}^0 = 0$;
- 3) $\mathbf{L}_1 = \mathbf{f}(\mathbf{L}^0, \mathbf{U}, \mathbf{H}), \quad \mathbf{L}^1 = \{\mathbf{L}_1\}$;
- ...
- 4) $\mathbf{L}_k = \mathbf{f}(\mathbf{L}^{k-1}, \mathbf{U}, \mathbf{H}), \quad \mathbf{L}^k = \mathbf{L}^{k-1} \cup \{\mathbf{L}_k\}$;
- ...
- 5) $\mathbf{L}_n = \mathbf{f}(\mathbf{L}^{n-1}, \mathbf{U}, \mathbf{H}), \quad \mathbf{L}^n = \mathbf{L}^{n-1} \cup \{\mathbf{L}_n\}$;
- 6) $\mathbf{L} = \mathbf{L}^n$.

Здесь $n = n(\mathbf{U}, \mathbf{H})$ — случайная величина, определяющая число шагов процесса и одновременно — число элементов множества \mathbf{L} . Множество \mathbf{L} строится индуктивно, за n шагов. На очередном шаге k к нему добавляется вектор $\mathbf{L}_k = (s_k, x_k, y_k, \varphi_k, \dots) = \mathbf{f}(\mathbf{L}^{k-1}, \mathbf{U}, \mathbf{H})$, где $\mathbf{f}(\mathbf{L}^{k-1}, \mathbf{U}, \mathbf{H})$ — случайный вектор, распределение которого зависит от \mathbf{L}^{k-1} , \mathbf{U} и \mathbf{H} . Таким образом, для построения модели алгоритма A_1 требуется задать распределения случайных величин $n(\mathbf{U}, \mathbf{H})$ и $\mathbf{f}(\mathbf{L}', \mathbf{U}, \mathbf{H})$ для всех $\mathbf{L}', \mathbf{U}, \mathbf{H}$.

Пример. Для вышеописанного примера обнаружения квадрата построим модель следующим образом. В качестве вектора управляющих параметров возьмем $\mathbf{H} = (H_1, H_2)$, где H_1 — параметр расфокусировки, H_2 — число посторонних объектов на изображении. Естественно определить $n(\mathbf{U}, \mathbf{H})$ так, чтобы его математическое ожидание было пропорционально H_2 . Если $\mathbf{U} \neq 0$, это значит, что \mathbf{U} есть вектор (x, y) , характеризующий центр квадрата. В этом случае определим первые четыре признака так, чтобы они располагались в соответствующих углах некоторого квадрата с центром в (x, y) с точностью, обратно пропорциональной H_1 . Так как размер D квадрата не задан, будем считать D равномерно распределенной случайной величиной, принимающей все возможные значения, при которых квадрат целиком помещается на изображении. Для $k = 1, \dots, 4$ положим класс признака i равным $s_k = k$, а его положение (x_k, y_k) — равным значению случайного вектора с нормальным распределением, центр которого лежит в соответствующей точке $(x \pm D, y \pm D)$, а дисперсия пропорциональна H_1^2 .

Фактически, мы задали распределение $\mathbf{f}(\mathbf{L}', \mathbf{U}, \mathbf{H})$ для случая, когда число элементов \mathbf{L}' не превосходит трех и $\mathbf{U} \neq 0$. Для всех остальных случаев будем считать, что в векторе $(s, x, y) = \mathbf{f}(\mathbf{L}', \mathbf{U}, \mathbf{H})$ значение s распределено равномерно на множестве $\{1, 2, 3, 4\}$, а вектор (x, y) равномерно распределен на дополнении к множеству уже занятых векторов $\{(x', y') \mid (s', x', y') \in \mathbf{L}'\}$.

3. Оценка качества алгоритма методом Монте-Карло

Для фиксированных \mathbf{U}, \mathbf{H} описанный выше процесс $\mathcal{L}(\mathbf{U}, \mathbf{H})$ дает возможность построения случайной выборки множеств признаков $(\mathbf{L}^1, \mathbf{L}^2, \dots, \mathbf{L}^N)$ произвольного размера N . Если подать каждое множество \mathbf{L}^i на вход алгоритма A_2 и обозначить выход \mathbf{V}^i , то получится выборка $(\mathbf{V}^1, \mathbf{V}^2, \dots, \mathbf{V}^N)$. Фактически, это выборка результатов моделирования работы алгоритма A в ситуации, когда искомый объект расположен в \mathbf{U} , а параметры съемки есть \mathbf{H} . Чтобы оценить по этим выборкам качество работы алгоритма, необходимо задать критерий сравнения истинной локализации \mathbf{U} и вычисленной локализации \mathbf{V} . Обозначим такой критерий $\delta(\mathbf{U}, \mathbf{V})$. Если $\mathbf{U} \neq 0 \neq \mathbf{V}$ и расстояние в некоторой метрике между \mathbf{U} и \mathbf{V} не превышает заданного порога либо $\mathbf{U} = \mathbf{V} = 0$, то $\delta(\mathbf{U}, \mathbf{V}) = 1$. В противном случае $\delta(\mathbf{U}, \mathbf{V}) = 0$. В этих обозначениях качество работы алгоритма может быть оценено следующим образом. Если $\mathbf{U} \neq 0$, то

$$\begin{aligned} F_{ta}(\mathbf{U}, \mathbf{H}) &= \frac{1}{N} \text{card}\{i : \mathbf{V}^i \neq 0\}, \\ F_{loc}(\mathbf{U}, \mathbf{H}) &= \frac{1}{N} \sum_i \delta(\mathbf{U}, \mathbf{V}^i), \\ F_{II}(\mathbf{U}, \mathbf{H}) &= 0, \end{aligned}$$

иначе

$$\begin{aligned} F_{ta}(\mathbf{U}, \mathbf{H}) &= F_{loc}(\mathbf{U}, \mathbf{H}) = \frac{1}{N} \sum_i \delta(\mathbf{U}, \mathbf{V}^i), \\ F_{II}(\mathbf{U}, \mathbf{H}) &= \frac{1}{N} \text{card}\{i : \mathbf{V}^i \neq 0\}, \end{aligned}$$

где $\text{card}(X)$ — мощность множества X .

Обычно представляет интерес оценка качества алгоритма при фиксированных параметрах \mathbf{H} , вне зависимости от \mathbf{U} . Для этого требуется задать вероятностное распределение $P_0(\mathbf{U})$ возможных положений объекта на изображении. Например, если нет априорных данных о локализациях объекта, можно положить вероятность появления \mathbf{U} во всех допустимых положениях на изображении одинаковой. А вероятность события $\mathbf{U} = 0$ оценить на основе набора тестовых примеров.

Для оценки качества алгоритма сгенерируем с помощью распределения $P_0(\mathbf{U})$ выборку $(\mathbf{U}^1, \dots, \mathbf{U}^M)$ размера M . Обозначим $E = \{j : \mathbf{U}^j \neq 0\}$. Тогда

$$\begin{aligned} F_{ta}(\mathbf{H}) &= \frac{1}{\text{card}(E)} \sum_{j \in E} F_{ta}(\mathbf{U}^j, \mathbf{H}), \\ F_{loc}(\mathbf{H}) &= \frac{1}{\text{card}(E)} \sum_{j \in E} F_{loc}(\mathbf{U}^j, \mathbf{H}), \\ F_{II}(\mathbf{H}) &= \frac{1}{M - \text{card}(E)} \sum_{j \in \{1, \dots, M\} \setminus E} F_{II}(\mathbf{U}^j, \mathbf{H}). \end{aligned}$$

Данную процедуру оценивания можно провести для нескольких наборов параметров \mathbf{H} . В частности, можно выбрать набор \mathbf{H}^0 , соответствующий тестовой базе примеров, и рассчитать F_{ta} , F_{loc} , F_{II} для \mathbf{H} из некоторой окрестности \mathbf{H}^0 . По степени изменчивости полученных значений можно судить об устойчивости алгоритма при условиях \mathbf{H}^0 . А именно, если градиент вектор-функции $(F_{ta}, F_{loc}, F_{II})$ в точке \mathbf{H}^0 имеет большую норму, это свидетельствует о том, что алгоритм резко меняет свои свойства при малых изменениях параметров съемки, что ставит под сомнение практическую применимость алгоритма. Для получения характеристик устойчивости, инвариантных к масштабу изменения параметров, можно использовать отношения частных производных F_{ta} , F_{loc} , F_{II} по параметрам из \mathbf{H} . Например, величина

$$\frac{\partial F_{loc}}{\partial H_k} / \frac{\partial F_{II}}{\partial H_k}$$

позволяет установить, что чувствительней к изменению параметра H_k — оценка вероятности правильной локализации или ложной тревоги. Такая оценка позволяет обнаружить слабые места в алгоритме и контролировать результат их устранения.

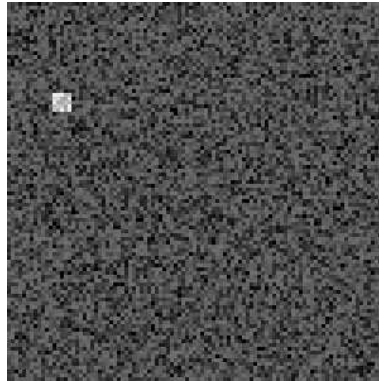
Далее рассматривается пример простого порогового алгоритма, для которого доказывается неустойчивость в отмеченном выше смысле.

4. Неустойчивость порогового алгоритма

Пусть изображение сцены имеет размеры $N_x \times N_y = 100 \times 100$ пикселей, а значения пикселей изображения $\mathbf{I}[x, y]$ являются целыми числами от 0 до 255, где $1 \leq x \leq N_x$, $1 \leq y \leq N_y$. Пусть изображение искомого объекта — это квадрат размером $n_x \times n_y = 5 \times 5$ пикселей, причем значения яркости в этих пикселях не менее $H^0 = 99$. Центр квадрата является случайным вектором $\mathbf{U} = (x_0, y_0)$, распределенным равномерно на множестве

$$\{3, \dots, 31\} \times \{3, \dots, 98\},$$

т. е. в левой трети изображения сцены.

Рис. 1. Пример изображения сцены для $H = 50$

Далее, пусть значения яркостей пикселей, не принадлежащих изображению объекта, — независимые случайные величины, принимающие значения, равномерно распределенные на множестве

$$\{0, 1, \dots, H - 1\},$$

где H — параметр, характеризующий уровень зашумления изображения (рис. 1). Требуется оценить центр изображения объекта.

Рассмотрим следующий алгоритм $A = A_2 \circ A_1$ решения этой задачи.

Алгоритм 1. Пороговый алгоритм обнаружения квадрата

Вход: изображение \mathbf{I} .

Выход: $\mathbf{V} = (x^*, y^*)$.

A_1 (поиск признаков)

Все признаки $\mathbf{L}_k = (s_k, x_k, y_k)$ принадлежат одному классу, т. е. $s_k = 1$, а их координаты таковы, что $\mathbf{I}[x_k, y_k] \geq H^0$:

$$\mathbf{L} = \{(1, x, y) : \mathbf{I}[x, y] \geq H^0\}.$$

A_2 (анализ геометрического расположения признаков)

Рассчитать координаты габаритного прямоугольника, в котором содержатся все \mathbf{L}_k :

$$X_0 = \min_k(x_k), \quad Y_0 = \min_k(y_k),$$

$$X_1 = \max_k(x_k), \quad Y_1 = \max_k(y_k).$$

Вычислить его центр и выдать его в качестве ответа:

$$x^* = \frac{X_0 + X_1}{2}, \quad y^* = \frac{Y_0 + Y_1}{2},$$

$$\mathbf{V} = (x^*, y^*).$$

Результат работы алгоритма считается правильным, если

$$(x^* - x_0)^2 + (y^* - y_0)^2 \leq \delta^2, \quad \delta = 2.$$

Согласно предложенной методике, исследуем алгоритм на устойчивость по отношению к уровню шума H . Если $H \leq H^0$, то вероятность правильной локализации будет равна 1, так как L будет состоять в точности из пикселей, соответствующих изображению объекта. Если $H > H^0$, то значения пикселей, не принадлежащих объекту, будут меньше H^0 с вероятностью

$$p = \frac{H^0}{H} \leq 1 - \frac{1}{H}.$$

В частности, вероятность того, что все пиксели (x, y) из области $W = \{67, \dots, 100\} \times \{1, \dots, 100\}$ будут иметь значение, меньшее H^0 , равна

$$p^{100 \cdot 34} < 10^{-14}.$$

С практической точки зрения такое событие наступить не может. А значит, в области W найдется пиксел со значением, больше либо равным H^0 . Следовательно, при работе первого этапа алгоритма он будет выделен как признак и попадет в L , поэтому на этапе A_2 получится $X_1 \geq 67$, значит:

$$x^* = \frac{X_0 + X_1}{2} \geq 34.$$

Так как по предположению $x_0 \leq 31$, то

$$|x^* - x_0| \geq 3.$$

Эта величина больше δ , поэтому результат алгоритма будет считаться неверным.

Таким образом, при уровне шума $H < H^0$ алгоритм работает всегда правильно, а при $H \geq H^0$ результат его работы с практической точки зрения всегда неверен. Следовательно, алгоритм неустойчив относительно изменения параметра H в точке $H = H^0$.

5. Исследование алгоритма обнаружения автосцепки

Рассмотрим алгоритм решения прикладной задачи [8] обнаружения и локализации изображения автосцепки двух сцепленных железнодорожных вагонов. Съемка изображений производилась стационарно закрепленной видеокамерой. Примеры изображений автосцепок показаны на рис. 2.



Рис. 2. Несколько примеров изображений автосцепки

Т а б л и ц а 1. Результаты тестирования алгоритма обнаружения автосцепки

| | |
|--|------------------|
| Частота правильного обнаружения $F_{ta} = 666/685$ | ≈ 0.9723 |
| Частота ложной тревоги $F_{II} = 1888/23211$ | ≈ 0.0813 |
| Частота правильной работы $F_{correct} = F_{Obj}F_{ta} + (1 - F_{Obj})(1 - F_{II})$ | ≈ 0.9202 |

Работа алгоритма основана на том факте, что около центра изображения автосцепки имеется четыре темных прямоугольных области. Алгоритм работает в два этапа. На первом этапе (поиск признаков) на изображении выделяются темные прямоугольные области заданного размера. На втором этапе (анализ конфигураций признаков) производится геометрический анализ найденных прямоугольников. Подробное описание алгоритма приведено в статье [8].

Тестирование алгоритма было проведено с использованием 23 896 изображений, причем на 685 из них присутствовало изображение автосцепки. Анализировался только факт нахождения алгоритмом сцепки, безотносительно правильности локализации. Результаты тестирования приведены в табл. 1. Параметр $F_{Obj} = 685/23\,896 \approx 0.0287$ обозначает частоту появления искомого объекта на изображении. Из таблицы видно, что алгоритм сработал правильно в 92%. Это позволяет сделать вывод о работоспособности алгоритма поиска автосцепки при данных условиях съемки.

Согласно предлагаемой методике, исследуем алгоритм на устойчивость к структурным искажениям с помощью построения вероятностной модели обнаружения признаков. В качестве параметра модели будем использовать величину H , характеризующую число посторонних объектов в кадре.

Положение автосцепки на изображении обозначим символом \mathbf{U} . В соответствии с тестовыми данными, примем вероятность отсутствия автосцепки в кадре равной

$$P(\mathbf{U} = 0) = 1 - 0.0287 = 0.9713.$$

Будем считать, что если $\mathbf{U} \neq 0$, то вектор $\mathbf{U} = (x, y)$ распределен равномерно на множестве

$$\{r, \dots, N_x - r + 1\} \times \{[N_y/2]\},$$

где $N_x \times N_y = 170 \times 34$ — размер входного изображения, $r = 3$.

Построенные гистограммы числа признаков n при работе алгоритма с тестовыми изображениями в случае присутствия и отсутствия автосцепки были аппроксимированы методом максимального правдоподобия нормальными распределениями:

$$G(n; n_j, \sigma_j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{(n - n_j)^2}{2\sigma_j^2}\right), \quad j = 1, 2,$$

где $n_1 = 7.2886$, $\sigma_1 = 1.8578$, $n_2 = 5.1567$, $\sigma_2 = 2.8979$. Построим модель $n(\mathbf{U}, H)$, которая включает в себя эту информацию. Внесем ряд упрощений. Из практических соображений можно ограничиться рассмотрением $n \leq 30$. Кроме того, в случае $\mathbf{U} \neq 0$

на изображении будет обнаружено не менее трех признаков, т. е. $n \geq 3$. Будем считать, что при $\mathbf{U} = 0$ вероятность обнаружения n признаков равна

$$P(n; \mathbf{U}, H) = \frac{G(n; n_2 + H, \sigma_2)}{\sum_{i=0}^{30} G(i; n_2 + H, \sigma_2)}, \quad 0 \leq n \leq 30,$$

а при $\mathbf{U} \neq 0$

$$P(0; \mathbf{U}, H) = P(1; \mathbf{U}, H) = P(2; \mathbf{U}, H) = 0,$$

$$P(n; \mathbf{U}, H) = \frac{G(n; n_1 + H, \sigma_1)}{\sum_{i=3}^{30} G(i; n_1 + H, \sigma_1)}, \quad 3 \leq n \leq 30.$$

Хотя в реальной ситуации признаки-прямоугольники имеют различные размеры, для упрощения будем считать, что все прямоугольники имеют одинаковый размер 16×5 пикселей. Алгоритм поиска прямоугольников устроен таким образом, что найденные прямоугольники не пересекаются. Будем считать, что если $\mathbf{U} = 0$, то n центров прямоугольников генерируются последовательно и они распределены равномерно по всем возможным положениям, при которых не возникает пересечений с уже построенными прямоугольниками.

Если же $\mathbf{U} \neq 0$, тогда $n \geq 3$ и первые три прямоугольника генерируются в соответствующих точках рядом с центром автосцепки \mathbf{U} . Для моделирования влияния дефокусировки к координатам этих точек прибавляются значения равномерно распределенной случайной величины, принимающей значения из множества $\{-1, 0, 1\}$.

Используя построенную модель, методом Монте-Карло вычисляем оценку качества алгоритма для значений параметра H из отрезка $[-3, 3]$ с шагом 0.5. Для каждого H проводилось 20 000 испытаний — по 10 000 испытаний на случаи наличия и отсутствия объекта в сцене. Критерием допустимой погрешности между истинной \mathbf{U} и вычисленной \mathbf{V} локализациями являлось условие $\|\mathbf{V} - \mathbf{U}\|_2 \leq 4$. Графики с полученными результатами изображены на рис. 3.

При $H = 0$ параметры модели соответствуют данным оценки тестовых изображений. Сравнительные результаты приведены в табл. 2.

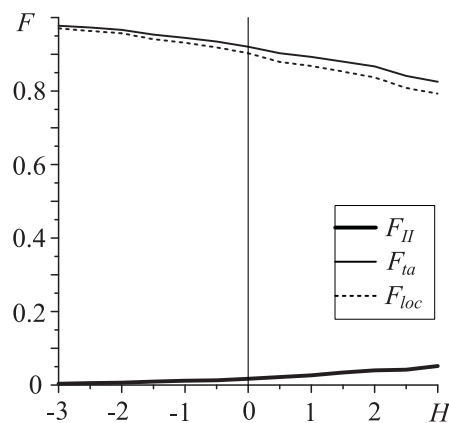


Рис. 3. Зависимость качества работы алгоритма поиска автосцепки от числа посторонних объектов

Т а б л и ц а 2. Оценки качества алгоритма на тестовых изображениях и на его модели

| Производные | Тестовые изображения | Модель |
|-------------|----------------------|------------------|
| F_{ta} | ≈ 0.9723 | ≈ 0.9086 |
| F_{loc} | — | ≈ 0.8868 |
| F_{II} | ≈ 0.0813 | ≈ 0.0186 |

Видно, что значения величин в тесте и модели получились различными, что связано с упрощениями, внесенными в модель.

Исследуем устойчивость алгоритма в точке $H = 0$. Для этого рассчитаем производные $F'_{loc}(0)$ и $F'_{II}(0)$ с помощью симметричной формулы численного дифференцирования:

$$F'(0) \approx \frac{F(h) - F(-h)}{2h},$$

с шагом $h = 0.5$:

$$F'_{loc}(0) \approx -0.0328,$$

$$F'_{II}(0) \approx 0.0069,$$

откуда

$$\frac{|F'_{loc}(0)|}{F'_{II}(0)} \approx 4.75.$$

Таким образом, при росте числа посторонних объектов на изображении способность алгоритма точно локализовать интересующий объект ухудшается быстрее, чем способность правильно указывать на отсутствие интересующего объекта. При дальнейшем усовершенствовании алгоритма следует обратить основное внимание на ситуации, когда на входном изображении присутствует изображение автосцепки и имеется достаточно много посторонних объектов.

Заключение

В настоящее время подходы к исследованию алгоритмов с помощью методов Монте-Карло изучаются и применяются достаточно широко. Достоинства предлагаемого подхода — его применимость в практических задачах, простота используемой математической модели и возможность учета структурных искажений изображения. К недостаткам можно отнести то, что в силу упрощений, внесенных в модель, количественные значения получаемых оценок могут отличаться от реальных. Таким образом, данный метод предназначен в первую очередь для качественного исследования устойчивости алгоритмов к различным типам искажений входных данных.

Автор благодарит В.Б. Костоусова за постановку задачи и плодотворное обсуждение настоящей работы.

Список литературы

- [1] МЕТОДЫ компьютерной обработки изображений / Под ред. В.А. Сойфера. М.: Физматлит, 2003. 784 с.
- [2] NATIONAL instruments vision concepts manual, <http://www.ni.com>

-
- [3] ГОНСАЛЕС Р., ВУДС Р. Цифровая обработка изображений: пер. с англ. М.: Техносфера, 2005. 1072 с.
- [4] COURTNEY P., THACKER N.A. Performance characterisation in computer vision: The role of statistics in testing and design // Imaging and Vision Systems: Theory, Assessment and Applications". N.Y.: Nova Science Publishers, 2001.
- [5] COURTNEY P., THACKER N.A. Tutorial given to the British Machine Vision Conference, overheads, 2001, <http://peipa.essex.ac.uk/benchmark/tutorials/bmvc2001/bmvc2001-tutorial.pdf>
- [6] SCHMID C., MOHR R. Local grayvalue invariants for image retrieval // PAMI. 1997. Vol. 19, N 5. P. 530–534.
- [7] NIXON M.S., AGUADO A.S. Feature extraction and image processing. Oxford, GB: Newnes, 2002.
- [8] ПЕРЕВАЛОВ Д.С. Вероятностная модель и эффективный алгоритм распознавания изображения автосцепки // Тр. 37-й регион. молодеж. конф. "Проблемы теоретической и прикладной математики". Екатеринбург, 2006. С. 468–472.

*Поступила в редакцию 3 сентября 2007 г.,
в переработанном виде — 7 октября 2008 г.*