

## РАСПОЗНАВАНИЕ ДВИЖУЩИХСЯ ОБЪЕКТОВ И ПАРАМЕТРОВ ИХ ДВИЖЕНИЯ

**Н. В. Марченко, К. М. Шестаков**

Одной из важных задач, решаемых в интеллектуальных приборах наблюдения, является задача распознавания движущихся объектов и вычисление параметров их движения, имеющая особое значение в охран-ных системах, системах контроля дорожного движения и т.п.

Пространственно-временное представление последовательности видеокадров [1] можно изобразить в виде куба (Рис. 1).

На передней его стороне показано последнее изображение, т.е. ось времени направлена из глубины изображения к нам. Верхняя грань куба является последовательностью  $xt$  изображений, а правая – последовательностью  $yt$  изображений.

Правая грань может служить хорошим примером двухмерного пространственно-временного изображения. На нем очень хорошо видно, что, когда автомобили стоят на светофоре, то линии горизонтальны. Потом четко различим участок ускорения и участок с постоянной скоростью движения.



Рис. 1. Трехмерное пространственно-временное изображение.

Ось времени направлена из глубины к нам. На правой грани  $yt$  плоскость, выделенная вертикальной белой линией на  $xu$  плоскости. На верхней грани  $xt$  плоскость, выделенная горизонтальной белой линией (из [1])

Для трехмерного случая скорость будет определяться выражением

$$\mathbf{u} = - \begin{bmatrix} \operatorname{tg} \varphi_x \\ \operatorname{tg} \varphi_y \end{bmatrix}, \quad (1)$$

где  $\varphi_x, \varphi_y$  – углы между компонентами  $x$  и  $y$  вектора в направлении постоянства уровней серого и осью времени  $t$ . Достоинством данного метода является то, что вывод алгоритмов распознавания может быть сформулирован в непрерывном пространстве  $xt$  и исследован аналитически, прежде чем будет применена подходящая дискретизация.

Чтобы найти характерную особенность одного изображения  $g_1(t)=g_1$  во втором  $g_2(t)=g_2$ , они сравниваются в пределах определенного диапазона поиска. Два образца схожи, если вектор их различий не выходит за пределы зоны допуска  $\alpha$ , которая отражает вариацию освещения и некоторое допустимое изменение формы объекта.

Коэффициент взаимной корреляции принимает значение нуля для абсолютно несовпадающих (ортогональных) изображений и достигает максимума (единицы) для одинаковых образцов. Вырезающая функция  $w$  выделяет «зону», которая движется по изображению, и в пределах которой считается локальное значение корреляционного коэффициента.

$$r(\mathbf{x}, \mathbf{s}) = \frac{\int_{-\infty}^{\infty} w(\mathbf{x} - \mathbf{x}') g_1(\mathbf{x}') g_2(\mathbf{x}' - \mathbf{s}) d^2 x'}{\left( \int_{-\infty}^{\infty} w(\mathbf{x} - \mathbf{x}') g_1^2(\mathbf{x}') d^2 x' \int_{-\infty}^{\infty} g_2^2(\mathbf{x}' - \mathbf{s}) d^2 x' \right)^{1/2}}, \quad (2)$$

где  $\mathbf{x}$  – точка на изображении,  $\mathbf{x}'$  – точка внутри текущего положения «окна»,  $\mathbf{s}$  – смещение точки  $\mathbf{x}'$  на изображении  $g_2$ .

Предлагаемый алгоритм обрабатывает последовательность кадров

$$X(t) = [I_{k,m}(t)]. \quad (3)$$

Значение  $I_{k,m}$  определяется 8-битным числом ( $I_{k,m} \in [0, \dots, 255]$ ),  $k$  и  $m$  – координаты точки,  $t = 0, 1, 2, \dots, T$  – номер кадра [2].

0-й кадр в последовательности принимается за опорный кадр, с которым в дальнейшем будет происходить сравнение.

Изображения разбиваются на блоки размерами  $K \times M$  (например  $K \times M = 32 \times 24$ ) и для пикселей внутри блока выполняются неравенства:

$$\begin{aligned} 0 < k &\leq K, \\ 0 < m &\leq M. \end{aligned} \quad (4)$$

Обычно, распознаваемый объект превосходит по размерам блок. Поэтому достаточно взять для обработки не все  $N$  пикселей, а только некоторую их часть  $N'$  (например, каждый второй:  $N' = N/2 = 360$ ). Выбранные пиксели вычитаются (по модулю) от соответствующих пикселей того же блока но в опорном кадре. Далее находится среднее значение отклонения в блоке  $\Delta X_{i,j}^{cp}$ ,  $i, j$  – номер блока. Если оно превышает некоторое пороговое значение  $\epsilon_0$

$$\Delta X_{i,j}^{\tilde{n}\tilde{d}}(t) = \frac{\sum_{k=1}^{K'} \sum_{m=1}^{M'} \Delta I_{k,m}^{i,j}(t)}{N'} \geq \epsilon_0, \quad (5)$$

то считаем, что в блоке произошли изменения, связанные с движением объекта [3]. Пороговая величина  $\epsilon_0$  позволяет регулировать чувствительность алгоритма к допустимым не связанным с движением изменениям в изображениях.

В блоках, где были зафиксированы изменения и смежных с ними, заново происходит сравнение с опорным, но уже используются пиксели, не участвующие в предыдущем сравнении. В итоге получаем изображение движущегося объекта, который вырезан из кадра. Объект заключается в прямоугольник, который в дальнейшем представляет *область интереса* на изображении [4]. Через каждые  $P$  кадров (например,  $P = 16$ ) опорный кадр обновляется блоками, не входящими в область интереса. Это позволяет избежать накопления ошибок, связанных с плавным изменением освещения.

Рассчитывается центр масс объекта. За массу берется значение  $I_{k,m}(t)$ .

$$G_k(t) = \frac{\sum I_{k,m}(t) \cdot k}{\sum_k \sum_m I_{k,m}(t)},$$

$$G_m(t) = \frac{\sum I_{k,m}(t) \cdot m}{\sum_k \sum_m I_{k,m}(t)}.$$
(6)

Одновременно считается, какое количество пикселей «содержится» в объекте – величина  $S(t)$ , т.е. площадь поперечного сечения объекта.

На следующем шаге простым и надежным способом рассчитывается скорость движения объекта по изображению:

$$v_k(t) = \frac{G_k(t) - G_k(t-1)}{\Delta t},$$

$$v_m(t) = \frac{G_m(t) - G_m(t-1)}{\Delta t}.$$
(7)

А также скорость движения объекта в перпендикулярном к камере направлении:

$$v_{\perp} = \frac{\sqrt{S(t) - S(t-1)}}{\Delta t}.$$
(8)

Используя результаты из (7) и (8), рассчитываем соответствующие ускорения для данного кадра.

$$w_k(t) = \frac{v_k(t) - v_k(t-1)}{\Delta t},$$

$$w_m(t) = \frac{v_m(t) - v_m(t-1)}{\Delta t},$$

$$w_{\perp}(t) = \frac{v_{\perp}(t) - v_{\perp}(t-1)}{\Delta t}.$$
(9)

Применяя полученные в (7) – (9) значения к прямоугольной области интереса, перемещаем ее на новое место и изменяем соответственно ее размер. Именно в этой прямоугольной области начнется поиск движущегося объекта на следующем  $t + 1$  кадре. Если объект не окажется локализованным в предполагаемой области, то поиск перейдет в смежные с ней блоки. Поиск будет продолжаться до тех пор, пока объект не будет полностью выделен. Далее, если требуется, изменяем размеры прямоугольника и, если требуется детектировать более одного движения в кадре, начинаем сравнение вне области интереса для данного объекта. В ре-

зультате получится несколько независимых областей, соответствующих различным объектам, для которых будут применяться в дальнейшем свои соответствующие параметры движения.

Выражение (10) дает действительные значения скоростей объекта, с которыми он проецируется на плоскость изображения.

$$V_x = \sigma v_k, V_y = \sigma v_m, V_z = \sigma v_{\perp}. \quad (10)$$

Коэффициент  $\sigma$ , как раз, и должен выражать масштаб отображения объекта на кадре. Т.е. требуется знать или размеры объекта или же расстояние до него и фокусное расстояние объектива.

Однако (10) не отражает реальную мгновенную скорость объекта в трехмерном мире. Для ее нахождения можно воспользоваться следующим выражением.

$$V = \sqrt{\alpha V_x^2 + \beta V_y^2 + \gamma V_z^2}. \quad (11)$$

Коэффициенты  $\alpha, \beta, \gamma$  определяются, если известно, как соотносятся направления движения на изображении и в реальном мире. На практике обычно это известно заранее, т.к. камеру устанавливают в определенном месте (например, над дорогой), и эти соотношения для всех движущихся объектов остаются постоянными. Или же, иногда, используют несколько камер, которые детектируют только компоненты вектора скорости.

Общая скорость находится как векторное сложение компонент.

В работе была рассмотрена последовательность из 18 изображений, взятых из AVI-файла [5] с частотой 3 кадра в секунду.

Расчеты показали хорошие результаты при детектировании движения в двухмерном пространстве для последовательности изображений с разрешением CIF (352×288) и частотой 3 кадра в секунду. Производилось искусственное изменение яркости отдельных кадров, тем не менее, объект был распознан правильно. Значение скорости равномерно движущегося объекта также было найдено без значительных скачков.

К преимуществам данного алгоритма следует отнести и его гибкость. Регулируя параметры  $K, M, \epsilon_0, P, N'$ , можно добиться удовлетворяющих результатов по надежности детектирования движения. Данный алгоритм, в целом, не предъявляет чрезмерных требований к аппаратной части.

### Литература

1. *Jähne B.*, Digital Image Processing. Springer-Verlag, Berlin Heidelberg, 2002. Разд.14, С. 375–412

2. *Xirouhakis Y., Tsechpenakis G., Delopoulos A.*, Fast Mobile Object Detection and Localization in Video Sequences/ Computer Vision and Image Understanding. March 2002. <http://www.image.ece.ntua.gr/~gtsech>
3. *Xirouhakis Y., Tsechpenakis G., Delopoulos A.*, A Multiresolution Approach for Main Mobile Object Localization in Video Sequences/ In Proc. International Workshop on Very Low Bitrate Video Coding, Athens, Greece, 2001. <http://www.image.ece.ntua.gr/vlbv01/proceedings>
4. *Xirouhakis Y., Avrithis Y., Kollias S.*, Efficient Moving Object Detection and Tracking in Video Sequences/ Image Vision and Computing. March 2002. <http://www.image.ece.ntua.gr/~gtsech>
5. *Кенцл Т.*, Форматы файлов в Internet/Перев. с англ. – СПб: Питер, 1997. Разд. 40, С. 270–272