INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Pattern Matching For Tracking Applications

V.Purandhar reddy
Associate Professor, Dept of ECE, S V College of Engineering, India
K Naga sudha
S V College of Engineering,India
B Rajeswari
S V College of Engineering,India
V Amrutha
S V College of Engineering,India
V Sravya rao
S V College of Engineering,India

Abstract:
A New method is developed to improve an existing recognition system for video streams by using knowledge of object features occur and do not occur in subsequent frames was used to filter false objects and to better identify real ones. The recognition ability was tested by measuring how many objects were found and how many of them were correctly identified in two short video files. The tests also looked at the number of object detections. We have developed a system which empowers users to upload MPEG video files and track the object using the web based front end. Evaluations are accomplished by comparing tracking output objects with a set of ground truth objects according to different performance metrics. These metrics are object based as well as speed based. Profiling has been done on the program to evaluate the gain in hardware realization of the algorithm.

Keywords: *pattern, object, matching, feature, vectors.*

1 .Introduction

The increasing use of multimedia streams nowadays necessitates the development of efficient and effective methodologies and systems for manipulating databases storing these streams. Today many applications requiring surveillance is cropping up. The demand for visual surveillance systems is still growing. Airports, train stations, courts and public buildings are only few examples of places where security has an extremely high priority. Security systems must provide a high degree of reliability to be credible. This means that for visual surveillance systems must minimize false alarms as much as possible. This is especially true for systems that provide automatic alarming mechanisms. In the last decade object tracking in video scenes has become very popular because of its applicability to daily problems and ease of production, e.g. surveillance cameras, adaptive traffic lights with object tracking, plane detection etc. The superiority of object tracking to object recognition became apparent after the development in the video processing and motion estimation. One major part of visual surveillance is people tracking. A people tracker must be able to deal with many different situations. In a simple situation we can have only one person appearing in the video scene whereas in a very complex situation we can have many people in scene occluding each other, hence the tracking algorithm should be robust enough to handle such situations. Tracking people in real-time is also at times a very important requirement of a tracking systems.

In this paper, we propose edge Detection based Method for object tracking in video pictures. Our algorithm is based on Edge detection, object extraction and pattern matching. With the edge detection, we can extract all objects in images. The proposed method for tracking uses pattern matching between successive frames. As a consequence, the algorithm can simultaneously track multiple moving and still objects in video pictures.

This paper is organized as follows. The proposed method consisting of stages edge detection, objects extraction, features extraction & object tracking is described in detail.

2. Pattern Matching

2.1.Edge Detection

A problem of fundamental importance in image analysis is edge detection. Edges characterize object boundaries and are therefore useful for segmentation, registration, and identification of objects in scenes. Edge points can be thought of as pixel locations of abrupt gray-level change.

In developed algorithm gradient operator method is used. For digital images these operators also called masks, represent finite difference approximations of either the orthogonal gradients f_x, f_y or the directional gradient $\partial f/\partial r$.

let H denote a p X p mask and define, for an arbitrary image U, their inner product at location (m,n) as the correlation $\langle U,H\rangle_{m,n} \cong \sum_i \sum_I h(i, j)\, u(i+m, j+n) = u(m,n) \otimes h(-m,-n)$ here the symbol $\otimes$ represents the convolution.

Let us consider the pair of sobel masks

$$H_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

And

$$H_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

boxed element indicates the location of an object .The masks H_1,H_2 measures the gradient of the image U(m,n) in two orthogonal direction .

Defining the bidirectional gradients

- $g_1(m, n) \cong \langle U,H_1\rangle_{m,n}$
- $g_2(m, n) \cong \langle U,H_2\rangle_{m,n}$

the gradient vector magnitude and direction is given by

- $g(m, n) = (g_1(m,n)^2 + g_2(m,n)^2)^{1/2}$
- $\theta_g(m, n)= \tan^{-1}(g_2(m,n) / g_1(m,n))$

Often the magnitude gradient is calculated as

- $g(m, n) \cong |g_1(m,n)| + |g_2(m,n)|$

. This calculation is easier to perform and is preferred especially when implemented in digital hardware.

The sobel operator computes horizontal and vertical differences of local sums. This reduces effect of noise in the data. Noting that this operator have a desirable property of yielding zeros for uniform regions.

The pixel location (m, n) is declared an edge location if g(m, n) exceeds some threshold ‘t’. The locations of edge points constitute an edge map ξ(m,n), which is defined as

$$\xi(m,n) = \begin{cases} 0 & (m,n) \varepsilon I_g \\ 1 & \\ & \text{thr} \end{cases}$$

Where

$$I_g \cong \{ (m,n) ; g(m,n) > t \}$$

The edge map gives necessary data for tracing the object boundaries in an image. Typically,‘t’ may be selected using the cumulative histogram of g(m, n) so that 5 to 10% of pixels with largest gradients are declared as edges.

2.2.Boundary Extraction

Boundaries are linked edges that characterize the shape of an object. They are useful in computation in computation of geometry features such as size or orientation. For extracting the boundaries of an object connectivity method is used.

Conceptually, boundaries can be found by tracing the connecting edges. On a rectangular grid a pixel is said to be four- or eight-connected when it has the same properties as one of its nearest four or eight neighbors, respectively. There are difficulties associated with these definitions of connectivity. Under four-connectivity, segments 1, 2, 3, and 4 would be classified as disjoint, although they are perceived to form a connecting ring. Under eight-connectivity these segments are connected, but the inside hole (for example pixel ‘B’) is also eight-connected to the outside (for instance, pixel ‘C’). Such problems can be avoided by considering eight-connectivity for object and four-connectivity for background. An alternative is to use triangular or hexagonal grids, where three- or-six-connectedness can be defined. However, there are other practical difficulties that arise in working with non rectangular grids.

2.3. Feature Extraction For Objects

In this subsection, we describe the extracted features of Extracted objects.

2.3.1 <u>Area</u>

By counting the number of pixels included in object i of the t-th frame, we calculate the area of the object $a_i(t)$.

2.3.2 . Width And Height

We extract the positions of the pixel P_{xmax} (P_{xmin}) which has the maximum (minimum) x-component:

$P_{xmax} = (X_{max,x},X_{max,y})$,

$P_{xmin} = (X_{min,x},X_{min,y})$,

where $X_{max,x}$, $X_{max,y}$, $X_{min,x}$, and $X_{min,y}$ are the x- and y coordinates of the rightmost and leftmost boundary of object i, respectively. In addition, we also extract

$P_{ymax} = (Y_{max,x}, Y_{max,y})$,

$P_{ymin} = (Y_{min,x}, Y_{min,y})$.

Then we calculate the width w and the height h of the objects as follows

$w_i(t) = X_{max,x} - X_{min,x}$,

$h_i(t) = Y_{max,y} - Y_{min,y}$.

2.3.3.Position

We define the positions of each object in the frame as follows

$x_i(t) = (X_{max,x} + X_{min,x})/2$

$y_i(t) = (Y_{max,y} + Y_{min,y})/2$

2.3.4 Color

Using the image data at P_{xmax}, P_{xmin}, P_{ymax} and P_{ymin}, we define the color feature of each object for the R (Red) component from original color frame

$R_i(t) = [R(P_{xmax}) + R(P_{xmin}) + R(P_{ymax}) + R(P_{ymin})]/4$,

as well as by equivalent equations for the G and B components.

2.4.Objects Tracking And Distance Measure

The proposed algorithm for object tracking exploits pattern matching with the features above and makes use of the minimum distance search in the feature space. We now go into more details of our algorithm.

Using the edge detection result of the object i in the t-th frame, we first extract the features of the object (N+1, i). Here, the notation (N+1, i) stands for the objects i in the t-th frame. Then we perform the minimum distance search in the feature space between (N+1, i) and (N, j) for all objects j in the preceding frame. Finally, the object (N+1, i) is identified with the object in the preceding frame which has the minimum distance from

(N+1,i). Repeating this matching procedure for all objects in the current frame, we can identify all objects one by one and can keep track of the objects between frames.
[Objects Tracking Algorithm]

2.4.1.Feature Extraction

- Extract the Features(Area, width, height & color features) of object to track in N^{th} Frame(ie previous frame)
- Extract the Features (Area, width, height & color features) of object to track in $N+1^{th}$ Frame(ie previous frame

2.4.2.Pattern Matching in the Feature Space

- Calculation of distances Search for the minimum Distance among the distances.
- Apply Feature match of N^{th} Frame object with minimum distance object of $N+1^{th}$ Frame object if not matched perform the feature match next minimum distance object and so on .
- After matching remove the data of N^{th} Frame and store the data of $N+1^{th}$ Frame.
- Increase the value of N by N+1 Repeat the steps 1 to26

3. Results

Some experimental results, obtained on real sequences, have been presented in this section. These results illustrate the tracking algorithm based on motion vectors and feature vectors. The tracking algorithm was used in video sequences shot in our institute premises using a Sony handy cam held in static condition. All sequences are captured at a constant frame rate of 25 frames per sec. The videos were converted to standard MPEG video with 352 X 288 pixel size frame. The sequence of frames in GOP has also kept as standard, i.e. I-B-B-P-B-B-P-B-B-P-B-B-I. In addition to this we also have used few standard video clipping (like Coastguard, tennis and Container). We performed all the tests in a PC powered with Intel Pentium IV 1.8 GHz processor and 256 MB RAM. To find the first performance metric i.e. the % of frames in which the algorithm tracks correctly, we need to do a frame-by-frame evaluation. This means, tracking output and ground truth are compared on a frame-by-frame basis. The rectangle coordinates of the tracked objects in the sequence are marked on the frames while playing the video by the front-end applet. These window locations were obtained from a file which is the output of the back-end. The ground truth hence can be seen in the marked sequence of frames. The tracked frame sequences of above captured video of a moving man are shown below as per the figure 1.

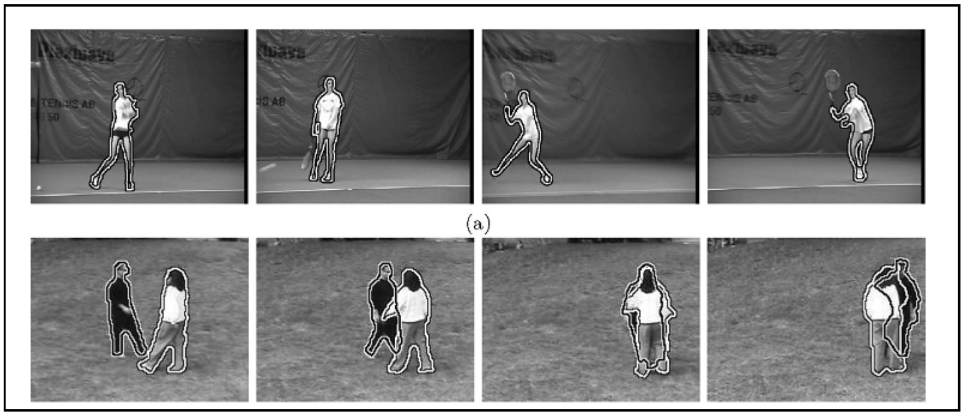

Figure 1: Tracked results

4.Conclusion

In this article, we present an extensive survey of object tracking methods and also give a brief review of related topics. We divide the tracking methods into three categories based on the use of object representations, namely, methods establishing point correspondence, methods using primitive geometric models, and methods using contour evolution. Note that all these classes require object detection at some point. For instance, the point trackers require detection in every frame, whereas geometric region or contours-based trackers require detection only when the object first appears in the scene. Recognizing the importance of object detection for tracking systems, we include a short discussion on popular object detection methods. We provide detailed summaries of object trackers, including discussion on the object representations, motion models, and the parameter estimation schemes employed by the tracking algorithms. Moreover, we describe the context of use, degree of applicability, evaluation criteria, and qualitative comparisons of the tracking algorithms. We believe that, this article, the first survey on object tracking with a rich bibliography content, can give valuable insight into this important research topic and encourage new research.

5.Reference

1. A new appearance model based on object sub-region for tracking Shu-Peng Wang;Hong-BingJi; Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07.
2. A moving object tracking method based on sequential detection scheme Qiu Xuena; Liu Shirong; Liu Fei; Zhu Weitao;DuFangfang;Control Conference (CCC), 2010 29th Chinese.
3. A stream field based partially observable moving object tracking algorithmKuo-ShihTseng;Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference.
4. W. G. Kropatsch and H. Bischof, “Digital Image Analysis,” Springer, 2001.
5. G.L. Foresti, “A real-time system for video surveillance of unattended outdoor environments,” IEEE Trans. Circuits and Systems for Vid. Tech., Vol. 8, No. 6, pp. 697–704, 1998.
6. C. Stauffer and W.E.L. Grimson, “Learning patterns of activity using real-time tracking,” IEEE Trans. PAMIL, Vol. 22, No. 8, pp. 747–757, 2000.
7. H. Kimura and T. Shibata, “Simple-architecture motion-detection analog V-chip based on quasi-two-dimensional processing,” Ext. Abs. of the 2002 Int. Conf. on Solid State Devices and Materials (SSDM2002), pp. 240– 241, 2002.
8. S. W. Seol et al., “An automatic detection and tracking system of moving objects using double differential based motion estimation,” Proc. of Int. Tech. Conf. Circ./Syst., Comput. and Comms. (ITC-CSCC2003), pp. 260– 263, 2003.
9. An embodiment of stereo vision system for mobile robot for real-time measuring distance and object tracking Ik-Hwan Kim; Do-Eun Kim; You-Sung Cha; Kwang-hee Lee; Tae-Yong Kuc; Control, Automation and Systems, 2007. ICCAS '07. International Conference