Volume 2, Issue 2, February 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Object Tracking Based on Pattern Matching

V.Purandhar Reddy

Associate Professor, Dept of ECE,

S V College of Engineering,

Tirupati-517501

Abstract— In this paper a novel algorithm for object tracking in video pictures, based on edge detection, object extraction and pattern matching is propsed. With the edge detection, we can detect all objects in images no matter whether they are moving or not. Using edge detection results of successive frames, we exploit pattern matching in a simple feature space for tracking of the objects. Consequently, the proposed algorithm can be applied to multiple moving and still objects even in the case of a moving camera. We describe the algorithm in detail and perform simulation experiments on object tracking which verify the tracking algorithm's efficiency.

I. INTRODUCTION

The moving object tracking in video pictures has attracted a great deal of interest in computer vision. For object recognition, navigation systems and surveillance systems, object tracking is an indispensable first-step.

The conventional approach to object tracking is based on the difference between the current image and the background image. However, algorithms based on the difference image cannot simultaneously detect still objects. Furthermore, they cannot be applied to the case of a moving camera. Algorithms including the camera motion information have been proposed previously, but, they still contain problems in separating the information from the background.

In this paper, we propose edge Detection based Mathod 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.

II. PROPOSED CONCEPT FOR MOVING OBJECTS TRACKING

A. 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

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

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.

B. Boundary Extraction By Connectivity Method

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 as shown in Fig.2(a ,b). There are difficulties associated with these definitions of connectivity, as shown in fig.2(c). 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.

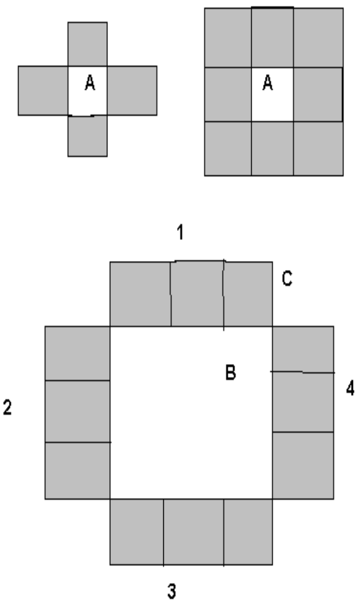


Fig .2: connectivity on a rectangular grid .Pixel A and its (a) 4- connected and (b) 8- connected neighbours; c) connected paradox; "are B and c connected ?

C. Feature Extraction For Objects

In this subsection, we describe the extracted features of Extracted objects. Figure.3 shows an example of a object for explanation purposes.

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

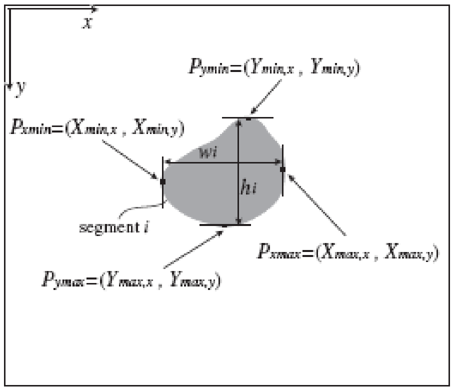


Fig.3: Explanation of the proposed feature extraction from the object extraction result.

*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

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

3) Position: We define the positions of each object in the frame as follows

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

*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.

D. 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.

Further refinements of the proposed algorithm are in order.

(1) We have not specified the distance measure used for matching yet. In the simulation experiments we could confirm that besides the Euclidean distance D_E the simpler Manhattan distance D_M is already sufficient for object tracking purposes.

(2) In order to treat all object features with equal weights, it is necessary to normalize the features. One possible way is dividing them by their maximum values. Dividing by 2^n, where the integer n is determined for each feature so that approximately equal weights results, is another possibility. The second possibility has the advantage that the division can be realized by a shifting operation in a hardware realization. Figure 4 shows a block diagram of proposed method & Figure 5 shows a Detailed description of the proposed object tracking algorithm.

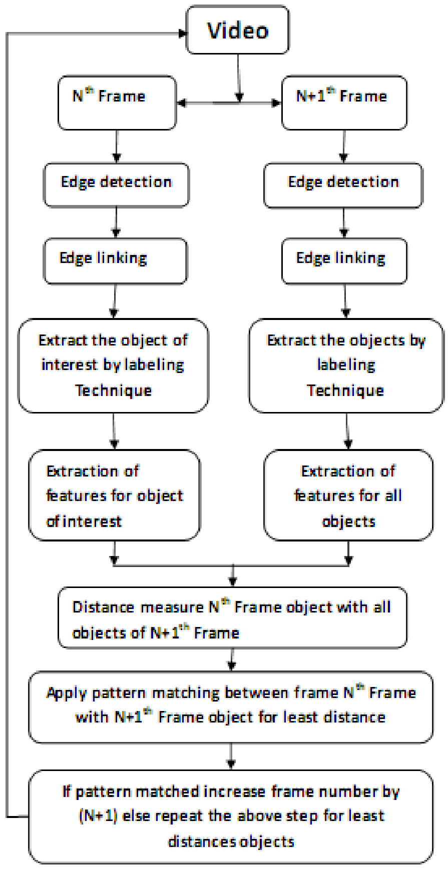


Fig. 4: Block diagram of proposed object tracking method.

[Objects Tracking Algorithm]

1) Convert the color image to gray scale image
2) Perform the edge detection by sobel edge detection.
3) Dilate the image by boundary connectivity.
4) Extract all objects by labeling method
5) Feature Extraction
   a) Extract the Features(Area, width, height & color features) of object to track in N^{th} Frame(ie previous frame)
   b) Extract the Features (Area, width, height & color features) of object to track in $N+1^{th}$ Frame(ie previous frame
6) Pattern Matching in the Feature Space
   a) Calculation of distances Search for the minimum Distance among the distances.
   b) 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 .
   c) After matching remove the data of N^{th} Frame and store the data of $N+1^{th}$ Frame.
   d) Increase the value of N by N+1 Repeat the steps 1 to 6

.

Fig. 6: Detailed description of the proposed object tracking algorithm.

III. SIMULATION

The proposed algorithm is tested using Matlab 7.1. For experimental verification two different video sequences were taken from moving camera. Then frames were extracted from the video sequences. Since all the processing is done on gray scale images, 24 bit color image frame is initially converted into gray scale frame of 8 bit size. By giving frames one by one to the Matlab program of the proposed algorithm, the tracked object segmentation is extracted. The dimension of the processing image is 320×240.

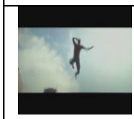
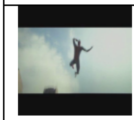
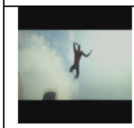
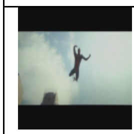
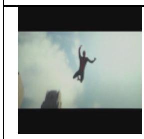
Frame number	**Object tracked result**	**Result of the tracked object**
 FRAME 1	----------	------------
 FRAME 2	Pattern matched	
 FRAME 3	Pattern Matched	
 FRAME 4	Pattern matched	
 FRAME 5	Pattern matched	

Fig. 7: The tracked object results from successive frames**.**

TABLE-I: EXTRACTED FEATURES FOR SAMPLE **1**

F. No	A	W	H	P		R	G	B
1	5	3.2	2.2	4.3	6.1	**4.6**	5.1	0.03
2	4.75	2.9	2.2	4.3	6.1	4.4	4.8	0.03
3	4.2	2.7	2.1	4.2	6.1	4.3	4.7	0.03
4	3.95	2.7	2	4.1	5.9	4.7	5.1	0.03
5	3.5	2.6	2.1	4	5.9	4.2	4.4	0.03
6	3.21	2.5	2	4	5.9	4.2	4.5	0.03
7	2.8	2.3	2	4	5.9	4.3	4.7	0.03
8	2.6	2	2.1	4.1	5.9	4	4.3	0.03

IV. CONCLUSION

We have proposed an object tracking algorithm for video pictures, based on Edge detection and pattern matching of the Extracted objects between frames in a simple feature space. Simulation results for frame sequences with moving objects verify the suitability of the algorithm for reliable moving object tracking. We also have confirmed that the algorithm works very well for more complicated video pictures including rotating objects and occlusion of objects.

In order to extract color features of Extracted objects, we used four boundary pixels color features from the original image. Thus, correct color features of an object that has gradation or texture is not extracted. Nevertheless, the mean value turns out to sufficiently represent the object's color features for the tracking purpose.

REFERENCES

[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 algorithm Kuo-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

[10] Image Segmentation and Pattern Matching Based FPGA/ASIC Implementation Architecture of Real-Time Object Tracking K. Yamaoka, T. Morimoto, H. Adachi, T. Koide, and H. J. Mattausch Research Center for Nanodevices and Systems, Hiroshima University.

.