Using Histograms to Detect and Track Objects in Color Video

Michael Mason and Zoran Duric
Department of Computer Science
George Mason University
Fairfax, VA 22030
mmason1@gmu.edu & zduric@cs.gmu.edu

Abstract

Two methods of detecting and tracking objects in color video are presented. Color and edge histograms are explored as ways to model the background and foreground of a scene. The two types of methods are evaluated to determine their speed, accuracy and robustness. Histogram comparison techniques are used to compute similarity values that aid in identifying regions of interest. Foreground objects are detected and tracked by dividing each video frame into smaller regions (cells) and comparing the histogram of each cell to the background model. Results are presented for video sequences of human activity.

1. Introduction

Many authors have developed methods of detecting people in images [1, 2, 3, 4, 7, 11]. A comprehensive survey [10] reviews most of the relevant references. Most of this work has been based on background subtraction using color or luminance information. Recently, edge information has been used for background subtraction [5, 8, 9] These methods usually use a number of frames to "learn" a model of the background scene which is later used to classify pixels in new images as either a background or a foreground. These methods assume that the camera does not move from frame to frame since any movement of the camera or the background objects could cause static parts of the scene to be classified as a moving foreground. The results frequently suffer from false positives/negatives and require additional post-processing to remove false objects and/or holes. In this paper, we present a novel moving object detection and tracking method that utilizes histogram matching techniques to improve the quality and reliability of the results and overcome some difficulties faced by existing color and/or edge based object detection methods. Our method creates a background model from a single image and can be used with both static and mobile, robot mounted, cameras.

Our approach is based on comparing color and/or edge histograms of small image blocks/cells computed for the background and the foreground images. It is fast and reliable under a wide range of scene conditions. The method is divided into two main parts: (i) building and maintaining the background model, (ii) performing object detection and tracking. We will illustrate our method using the images shown in Figures 1, 2, and 3. Images in Figure 1 were collected at the Keck Laboratory at the University of Maryland in College Park using a SONY progressive scan 3CCD digital camera; the images are 640 × 480 RGB color, and the frame rate was sixty frames per second. The images in Figures 2 and 3 were collected at George Mason University using a Sony CCD TR500 Handycam camera; the images are in 320 × 240 RGB color, and the frame rate was twenty frames per second.

2. Methodology

The goal of this method is to identify regions of a video frame that contain moving objects. We begin our process by using the initial frame of the video sequence to compute a background model. As new frames are encountered they are compared with the background model. In order to determine which areas of the current frame contain foreground objects, we construct our scene models by overlaying a 40 × 40 (see Figure 4) pixel grid on top of each frame, including the background frame. Histograms are computed for each cell in the background grid when the background model is constructed; histograms are computed for the corresponding cells of the current frame at the time of processing. Comparisons between the background and current frames are made on a cell by cell basis by comaparing their histograms to determine which cells contain foreground objects.

Figure 4 illustrates the overlapping grid structure used in our method. This structure allows for the possibility of detecting and tracking parts of foreground objects when they appear in the corner of a cell.

0-7695-1245-3/01 $17.00 © 2001 IEEE

Figure 1. Frames 60, 100, 140, 180, 220, 260, 300, and 340 from a 300-frame sequence of a moving human.

Figure 2. Frames 0, 40, 80, 120, 160, 200, 240, and 280 from a 350-frame sequence of three humans that enter the scene from different directions and interact.

2.1. Computing Edges in Color Images

In color images (RGB) we apply an edge detector to each color band to obtain partial derivatives r_x, r_y, g_x, g_y, b_x, b_y for the (r)ed, (g)reen, and (b)lue bands. Edges in color images can be computed using a standard technique used for processing multi-channel imagery [6]. We first form a matrix S,

$$S = \begin{pmatrix} r_x^2 + g_x^2 + b_x^2 & r_x r_y + g_x g_y + b_x b_y \\ r_x r_y + g_x g_y + b_x b_y & r_y^2 + g_y^2 + b_y^2 \end{pmatrix}.$$

The trace of S corresponds to the edge strength. If there is an edge at point (x, y), the larger eigenvalue of S, λ_1, corresponds to the edge strength. The corresponding eigenvector (n_x, n_y) represents the edge direction. Therefore we can treat color edges in the same manner as we would treat gray level edges. The only difference is that the edge strength and the edge direction correspond to the larger eigenvalue of S and its corresponding eigenvector.

2.2. Computing Histograms

The color video frames used as input in our method are 24-bit RGB color. Using color histograms to model scenes can be prohibitively expensive if the full 24-bit representation of the pixels is used. In addition, twenty four bit histograms are very hard to compare since a one bit change in a color value places the corresponding pixel into a different histogram bin. In the interest of building a fast and reliable system, we apply a color depth reduction formula to transform 24-bit color to 12-bit color. Given an (r, g, b) triple we obtain a twelve bit representation C using

$$C = 256\frac{r}{16} + 16\frac{g}{16} + \frac{b}{16}.$$

There are many reasons why it is preferable to use a 12-bit color representation over the original 24-bit color pixel values. First, when using inexpensive cameras that capture

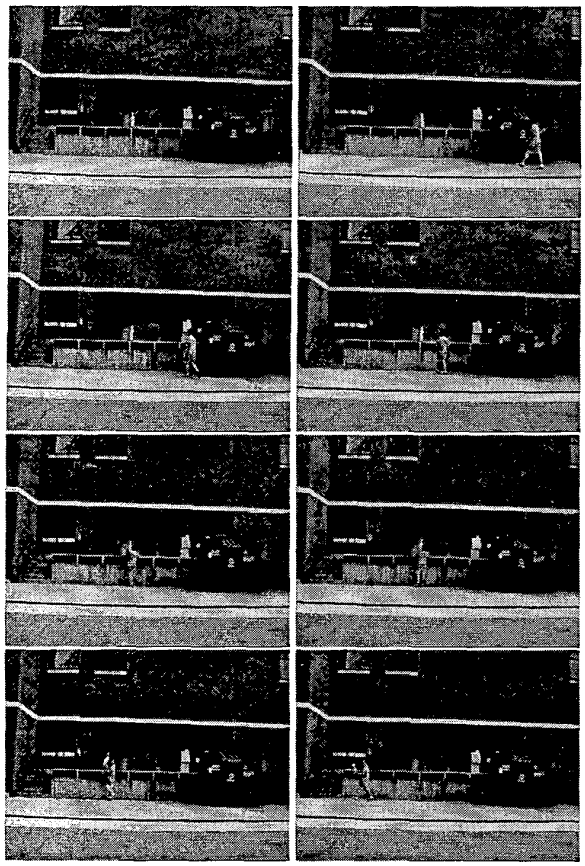

Figure 3. Frames 10, 30, 50, 70, 90, 110, 130, and 150 from a 165-frame sequence of human picking an object.

24-bit video the lower four bits of each pixel may be very noisy. Second, since these 12-bit pixel representations only require that we allocate 4096 bins for each histogram, we are conserving an enormous amount of storage. (Note that if we wanted to store the 24-bit representation in a histogram we would need 2^{24} bins per histogram.) And finally, smaller histograms are much easier to build and compare.

Edge histograms are composed of thirty six bins. For each edge pixel we compute the bin index using the edge orientation; the resolution of the histogram is $10°$ per bin. When the bin index is determined the bin is incremented with the edge magnitude.

2.3. Comparing Histograms

We use two different formulas to compute histogram similarity. The first formula (see Eq. (1)) computes the intersection $f(h_c, h_b)$ of histograms h_c (current image) and h_b (background image).

$$f(h_c, h_b) = \frac{\sum_i \min\{h_c(i), h_b(i)\}}{\sum_i h_b(i)} \quad (1)$$

The second formula (see Eq. (2)) computes the chi-squared measure of similarity $X^2(h_b, h_c)$ for histograms h_c and h_b.

$$X^2(h_b, h_c) = 2 \sum_i \frac{(h_c(i) - h_b(i))^2}{h_c(i) + h_b(i)} \quad (2)$$

Applying these two formulas to the histograms shown in Figure 5 yields $f = 0.92$, $X^2 = 61$ for the histograms on the left and $f = 0.22$, $X^2 = 828$ for the histograms on the right. In our experiments, we have used thresholds $T_f = 0.6$ and $T_{X^2} = 500$ to distinguish between similar and different histograms.

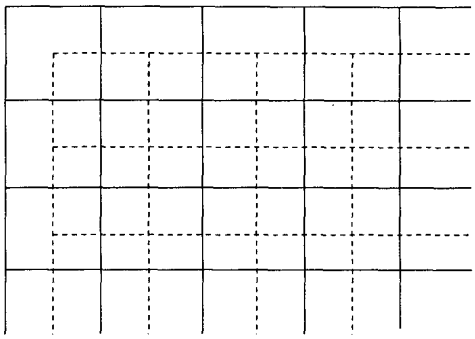

Figure 4. Grid structure used in our algorithm. Each interior cell has twelve neighbors; it shares a corner or an edge with eight cells that belong to the same half-grid and it overlaps four cells belonging to the other half-grid.

3. Experiments

We have tested our method on several image sequences of indoor and outdoor scenes. Figure 6 shows the results of applying our method to the sequence in Figure 1; in this example we used color histograms. Figure 7 shows the results of applying our method to the same sequence using edge histograms. It can be seen that the edge based method yields superior results; it detects the outline of the human much more accurately than the color based method.

Figure 8 shows the results of applying our edge based method to the outdooor sequence shown in Figure 2. In this sequence three people approach the scene from different directions, meet, and leave the scene separately. The method detects and tracks them accurately. Note that the results

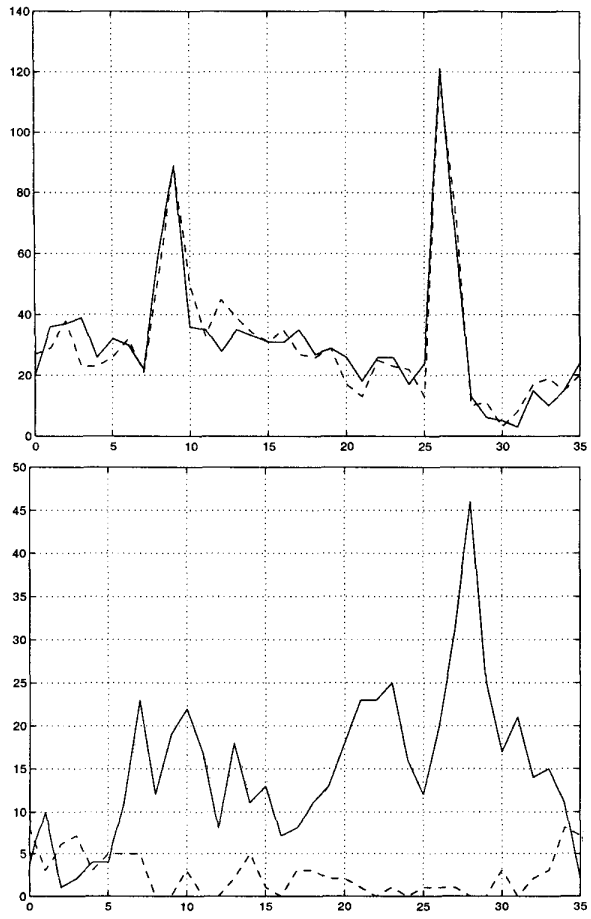

Figure 5. Examples of similar edge histograms (top) and different edge histograms (bottom).

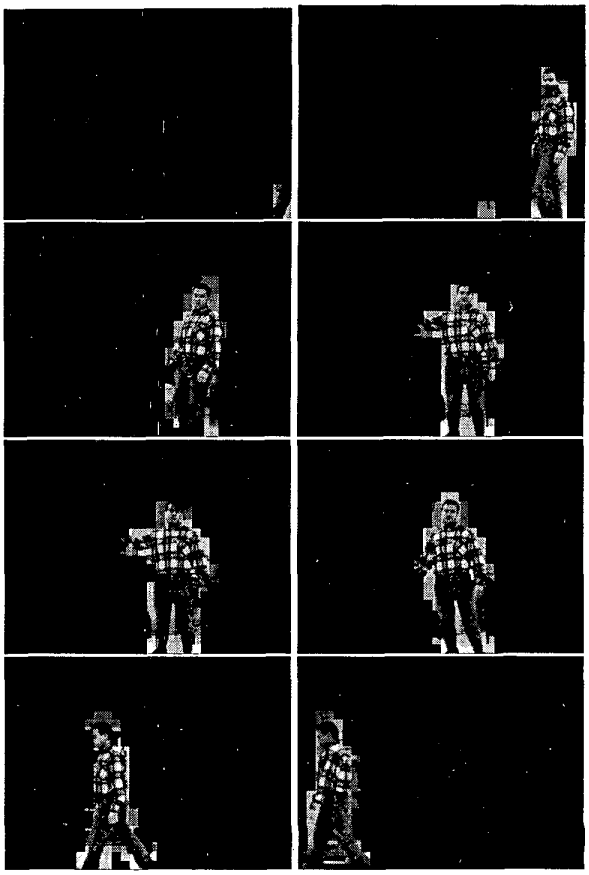

Figure 6. Results of moving object detection using color histograms for frames 70, 100, 140, 180, 220, 260, 300, and 340 of the sequence in Figure 1.

are slightly inferior to the results in Figure 7; this can be explained by the relative sizes of the objects and the grid cells, and the lower resolution and inferior quality of images.

Figure 9 shows the results of applying our edge based method to the outdooor sequence shown in Figure 3. In this sequence a person enters the scene and removes a shiny object. The method detects and tracks the person reliably; it also detects that the scene has changed because of the missing object. The part of the scene where the missing was located remains highlighted.

4. Conclusions

We presented two histogram-based methods for detecting and tracking objects in video. These methods are fast, reliable and do not require extensive prior training in order to "learn" the model of the background. In our experiments it was found that the edge histogram method typically yielded better results than the color histogram approach. The training phase of our system is very simple since it only requires that we compute histograms of 40×40 pixel cells of the background image. In the detection phase of our algorithm we compare histograms computed for the cells of the current image with histograms of the corresponding cells of the background image. The tracking is performed by checking the overlap of the (cell) connected components for the current and the previous frames; cell overlap shows that these connected components correspond to the same moving object(s). Because of its simplicity and reliability this method can be used with both static and mobile, robot mounted, cameras (we are assuming that the robot makes periodic stops to observe its environment). Future work will include different histogram matching methods and more so-

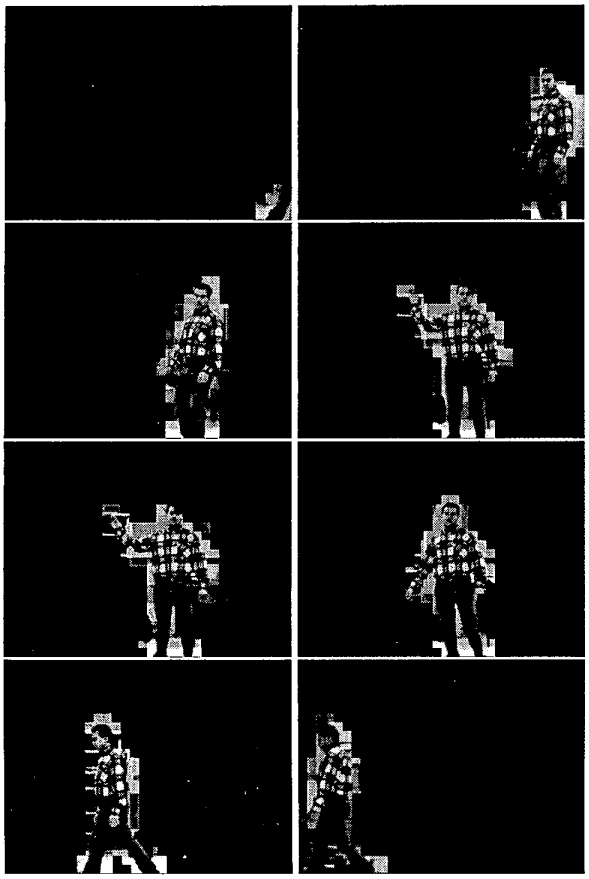

Figure 7. Results of moving object detection using edge histograms for frames 70, 100, 140, 180, 220, 260, 300, and 340 of the sequence in Figure 1.

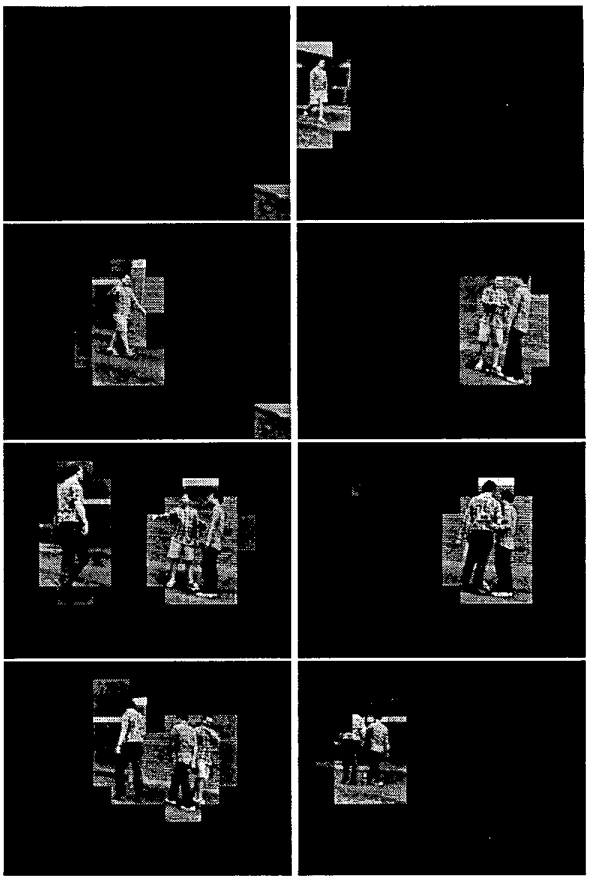

Figure 8. Results of moving object detection using edge histograms for frames 10, 40, 80, 120, 160, 200, 240, and 280 of the sequence in Figure 2.

phisticated tracking methods.

Our method proved to be tolerant to camera noise and slight illumination changes. In addition, since edges are used in detection and tracking, this approach makes use of, and indeed favors, clutter in both the scene and the moving objects. It could be used as a first step toward more task-specific research such as automated surveillance, human gesture recognition, and very low bandwidth communication.

References

[1] J.W. Davis and A.F. Bobick. The representation and recognition of human movement using temporal templates. In *Proc. Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[2] L.S. Davis, D. Harwood, and I. Haritoaglu. Ghost: A human body part labeling system using silhouettes. In *Proc. ARPA Image Understanding Workshop*, pages 229–235, 1998.

[3] D.M. Gavrila and L.S. Davis. 3D model-based tracking of humans in action: A multi-view approach. In *Proc. Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[4] I. Haritaoglu, D. Harwood, and L. Davis. W4S: A real-time system for detecting and tracking people. In *Proc. Computer Vision and Pattern Recognition*, pages 962-968, 1998.

[5] S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Detection and location of people in video images using adaptive fusion of color and edge information. In Proc. International Conference on Pattern Recognition, 2000.

[6] B. Jähne. *Digital Image Processing*. Springer-Verlag, Berlin, Germany, 1997.

[7] S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

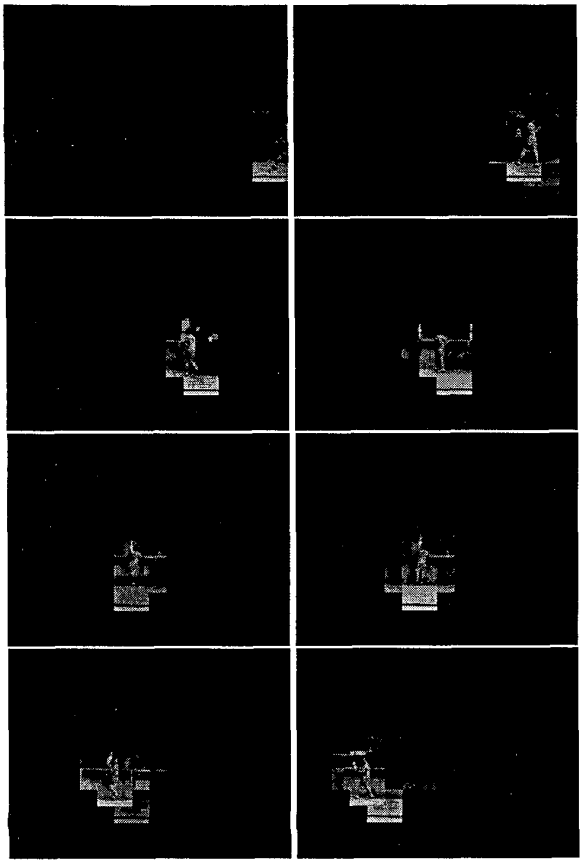

Figure 9. Results of moving object detection using edge histograms for frames 15, 30, 50, 70, 90, 110, 130, and 150 of the sequence in Figure 3.

[8] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In Proc. International Conference on Automatic Face and Gesture Recognition, Grenoble, France, pages 348-353, 2000.

[9] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking Groups of People. Computer Vision and Image Understanding, 80:42-56, 2000.

[10] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. Computer Vision and Image Understanding, 81:231-268, 2001.

[11] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE PAMI*, 19:780–785, 1997.