Intelligent Target Detection and Tracking of Moving Targets From Real Time Video Sequences

Pradyumna K. Mishra, P. K. Biswas

Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology
Kharagpur- 721 302, INDIA
pkb@ece.iitkgp.ernet.in

Abstract

This paper describes a technique for extracting moving targets and robustly tracking them from real time video sequence. It consists of two phases: Initial segmentation, temporal tracking and optimal paraimeter selection. After target detection subsequent tracking is based on a modified Canny edge detection and accelerated Hausdorff measure. The modified canny edge detection algorithm produces closed loop edges. The manual optimization of the thresholds for edge detection is replaced by adaptive inputs based on the initial segmentation. An accelerated Hausdorff measure is used here to make the system computationally inexpensive. This technique differs from other methods in the following features: 1) Non-feature based method, 2) Can effectively track fast moving targets, 3) Modified Canny dege detection and accelerated Hausdorff measure for robust yet computationally inexpensive tracking, 5) dynamic template updation makes it sutable to track any unknown object. The resulting system robustly identifies targets of interest, rejects background clutter, and continually tracks over large distances despite occlusions, appearance changes, and cessation of target motion.

Key words: Moving target tracking, Difference Image Processing, Canny Edge Operator, Hausdorff measure.

1. Introduction

The increasing availability of high performance video processing hardware has opened up new innovative methods for real time video understanding problem. One of the important problems is that of target detection and subsequent tracking from a real time video stream.A number of techniques have been proposed and implemented in the past [2, 3, 4, 5]. Many methods are computationally expensive and are inapplicable to real time situations or require specialized hardware to operate in real time domain. Most use feature based methods or prior knowledge or model of the target [6]. Systems based on kalman filtering are of limited use because they are based on unimodal Gaussian distribution and do not support alternative motion hypothesis. Optical flow methods such as [7] are also unsuitable for real time applications.

The essence behind these techniques is the segmentation of the image into targets of interest vs. non-target or background regions. A requirement of such a system is the need of a large number of pixels on target, making it unsuitable for outdoor surveillance, specially air borne targets like planes and missiles. A better approach as in [1] uses a simple classification metric based on simple rules largely independent of appearance and purely dependant on target’s shape.

This paper presents a much more robust technique in which during the initial segmentation, a history is maintained where the approximate properties of appearance are used as a training template for subsequent shape based tracking. The elimination of false object formed by background movement is achieved by keeping track of all objects in the form of a history database. Hence the tracking relies less on appearance and mainly on shape information but not ignoring the former.

Two of the methods used for real time target tracking are temporal differencing (TD) [8] and template correlation matching. In the TD approach pixelwise difference between consecutive frames are done to detect moving regions. It fails for significant camera motion, target occlusion or cessation of target motion. In template correlation each frame is scanned for a region that best correlates to an image template. This method requires apriori knowledge of the target features, hence is not robust to changes in object size, orientation, or even changing of lighting conditions.

The above handicaps are avoided in this proposed method by using TD for initial target detection and a dynamic target template for subsequent matching. Using edge and shape based properties avoids tracking depending on appearance. However appearance properties during the initial tracking stages are used to arrive at approximate thresholds required for efficient and optimum shape and target determination. Object motion parameter estimation from noisy images was introduced in [10]. The computation time is reduced by using a windowing technique and part prediction. The tracker works in two states: initial Locking State and the long term Tracking state.

2. Initial Segmentation

Initial segmentation is carried out to detect probable moving targets in video sequences. The underlying principle is to take the absolute difference of two consecutive frames. An optimum threshold function is then used to determine the change. If $I_n(u,v)$ is the intensity of the $(u,v)^{th}$ pixel in the n^{th} frame, then the motion region $M_n(u,v)$ can be extracted by obtaining the difference image of I_n with I_{n-1} and then thresholding. To avoide false motion indication each individual frame is low pass filtered to remove noise. In the difference image the moving sections are clustered into probable target regions $T_n(i)$. Each such probable target region is segmented separately to extract probable target and their shape feature.

3. Target Detection and Locking

The principle for target detection is : if a target persists over time, it is a good candidate for locking or classification else it is to be considered as background clutter and rejected.

The output of segmentation of a particular frame n are the I probable targets $T_n(i)$. A history is maintained for each probable target $T_n(i)$. A target that appears consistently over a few frames is passed to the Tracking state.

The first step is to build a history for each target $T_n(i)$ for the n^{th} frame. This history contains the perceptually significant features of the objects. To reduce the computational complexity of the feature extraction the significant features are limited to object positional and shape information. Each one of the probable targets is tracked in the next frame in a limited neighborhood. If the target matching fails the target is discarded as noise or background clutter. If a matching exists then the history is updated. The history also contains the number of frames in which the target was present. Also segmentation of a new frame introduces some new targets. So after the history of a probable target shown that it has existed continuously for some particular k frames it is passed on to a long Term Tracker.

Before the target is passed on to the tracker, it needs to decide the optimal parameters required for Canny Edge detection, they are σ (the standard deviation of the gaussian smoothing filter.) , ξ_1 (the low value to use in the hysteresis function) and ξ_h (the high value to be used in the hysteresis function).

So far in previous methods, It required manual optimization of these thresholds for given image groups in order to derive desired results. This was undesirable, as manual imput would mean custom made trackers, which would require prior knowledge of the nature of the object to be tracked. Further for object identification it was required to obtain closed loop edges, which previous canny edge detection algorithms failed to do.

For this purpose before a particular target is passed to the Long Term tracker, the object intensities will be processed. A modal intensity segmentation method is used to determine the initial thresholds. This process determines the minimum between the intensity peak maxima and then set the appropriate threshold. The hysteresis thresholds ξ_h , ξ_l are then set at some suitable value below this initial threshold.

4. Long Term Tracker

The inputs to this stage are the target positions, target dimensions or more precisely the window dimensions that contain the target wholly, the velocity of the target, and the optimum parameters for canny edge detection. Long Term tracker is responsible for the real time identification and tracking of the locked target. The detection and matching in the tracking state is based on edge detection and corresponding Hausdorff distance measure. A target is confirmed if the Hausdorff distance measure is smaller than a particular threshold. Then the next window location is predicted using a predictor, which is large enough to contain the target in the next frame and the process is repeated as long as the target exists.

4.1 Canny Edge Detection

The edge detection process is very important. It is desired that after edge detection, the output be a binary image, which contains the closed edges of the target, and all other noise and background are discarded. The entire robustness of the tracker will depend upon the versatility of the edge detector. J. Canny in his paper [13] introduced a versatile method of edge detection.

The Canny edge detector is a multi stage process. The image is smoothened by Gaussian convolution. It is a 2-D convolution operator that is used to 'blur' images and remove detail and noise. The gradient of the smoothed array is computed using 2 x 2 first difference approximations to produce two arrays, P[i,j] and Q[i,j] , for the x and y partial derivatives. These gradients are averaged over a 2 x 2 square so that the x and y partial derivatives are computed at the same point in the image.

The magnitude M[i,j] and orientation θ[i,j] of the gradient are computed by converting to polar coordinates.

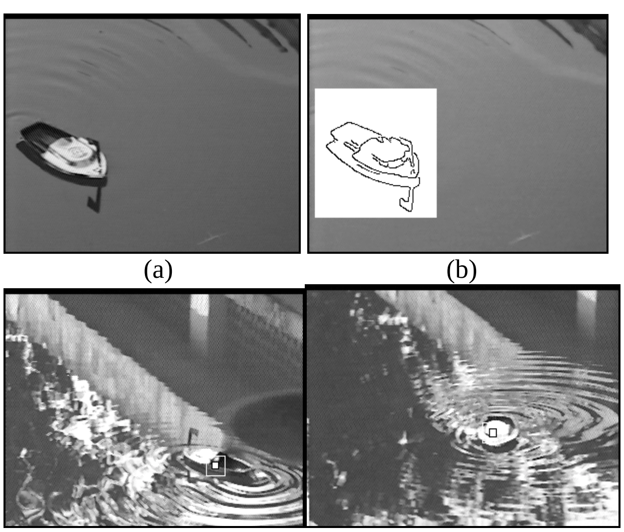
Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks the pixels along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non maximal suppression. The tracking process exhibits hystersis controlled by two thresholds. Tracking edges can only begin at a point on a ridge higher than the higher threshold. Tracking edges then continues in both directions out form that point until the height of the ridge falls below the lower threshold.

This normal Canny edge detection algorithm does not guarantee closed edges. To obtain close edges for better target identification a simple arc extrapolation could be applied to edge ends, but this may not be necessary since we had initially obtained the optimum parameters for the particular target from the Locking state .

4.2 Hausdorff Distance Computation

Hausdorff distance has been used for comparing images as in [9]and [11].The magnitude of the Hausdorff distance gives the measure of instances of a “model” edge map M in some edge map I. So from every window we extract a suitabale template and after extracting the edged image of the next frame we measure the Hausdorff distance between the previous mode template and the new edge image. If the Hausdorff distance is less than a threshold τ then we state that the target exists and the tracking continues by finding the Center of Gravity of the binary edge image. We compute the Voronoi Surface Array D’(which is the distance transform) of the new predicted window and then used the K^{th} partial Hausdorff measure array $F_B[x,y]$ for a translation t given by:

$$F_B[x,y]=H_K(M\oplus t,I) =K^{th}_{b\in M}D'[b_x+x,b_y+y]$$



(a) (b)

(c) (d)

Fig. 1

A parallel mask was used to compute the distance transform. To reduce computational complexity, integer approximations were made Here the speed of computation, was of primary concern we observed that the edge image was very robust allowing us some freedom for this computation. For speed considerations a (3-4)3x3 mask was preferred. A major kpart of the efficiency depended upon the K^{th} maximum value sorting. Direct calculation using a singly linked list, with insertion sort, is of computational complexity $O(N^2)$. A K-max algorithm was used having O(Nlog(N))complexity

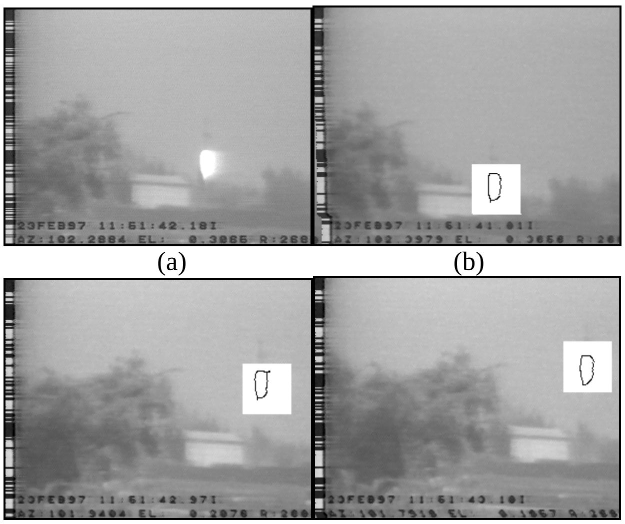
Methods to accelerate the procedure have been introduced in [12]. We obtain a skpeedup by a method called ‘Early Scan Termination’. This is by not computing $F_B[x,y]$ completely if we can deduce partway through the computation that it will be greater than τ . We probe D’in q places and maintain a count kof the number of these values D’ that exceed τ. If this count exceeds q-K, we know that the K^{th} – ranked value must be greater than τ and therefore, $F_B[X,Y]>\tau$; therefore, we need not probe any more locations for the translation (x,y).

Another method used in parallel to the above –mentioned method to accelerate the Hausdorff measure is ‘Ruling Out Circles’. Consider two images A and B. Let the bounds on A be $0\leq k<m_a, 0\leq 1<n_a,$ and the bounds on B be $0\leq K<m_b$ and $0\leq 1 <n_b$ Although the array F[x,y] is , in principle of infinite evtent, its minimum value must be attained when the translated model overlaps the image in at least one location. Therefore we only consider the portion where $-m_b + 1 \leq x < m_a$ and $-n_b + 1 \leq y < n_a$. This is intuitive as we expect only target to exist in the predicted window. One property of F_B [x,y] is that the slope cannot exceed one, that is, the function does not decrease rapidly than linearly. Thus, if $F_B[x_1, y_1] = V$ (where $V > \tau$), then F_B [x,y] cannot be less than τ in circle of radius $v - \tau$ about the point (x_1, y_1). In other words, if the value of $F_B[x_1, y_1]$ is large at some location, then it cannot be small in a large area around that location. This fact can be used to rule our possible translations near (x_1, y_1).

5. Results and Conclusion

The proposed tracking algorithm was implemented on a pentium 200 Mhz system under a Microsoft windows NT OS with a Matrox Meteor Digitizer. The system could track targets at approximately 20 frames/sec. This system was tested on around 35 real sequences including aerial, ground and water sequences. The tracker was able to smartly detect targets and robustly tracked them over large number of sequences during its life span.

Figure 1a. shows a frame of dimension 320 x 240 pixels containing a moving boat. The corresponding edge image of the predicted window is shown in figure 1b. The next few images (Figures 1c-1d) demonstrates robust tracking where the target changes orientation and position significantly in the presence of high background clutter in the form of significant ripples. This is possible because of dynamic template updating, where the changes in shape of the target in each frame are used in the next frame. Here the target was tracked for more than 1000 frames after which the target goes outside the field of view of the camera. Here the processing window is of size 120 x 120. Here the thresholds are very accurately decided so that the neighboring clutters are rejected.



(a) (b) (c) (d)

Fig. 2

Figure 2a. shows an aerial target coming out from behind the trees. The corresponding images Figures(2b-2d) shows the edge image of the predicted window. Here we notice that the target is detected correctly and the edge images are very consistent. Further tracking does not fall off track even though target appears almost static during some frames. The processing window is of dimension 50x50 pixels. Due to manual movement of the camera; the frames are jerky in nature and the target often went outside the field of view, but it was successfully detected every time it reappeared.

The tracker performs extremely robustly and is able to track various targets without having any prior information in the presence of background clutter. It does not fail due to cessation of target motion, camera jerks, significant appearance changes and noise. Target models used for dynamic template updating are simple and rely on target shape hence applicable to real world situations.

6. References

[1] A. Lipton H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real time video", IEEE Proc. Of the Workshop on Application of Computer Vision, October, 1998, pp. 47-58.

[2] C. Wren, A. Azarbayejani, T. Darell, A. Pentland, "Pfinder : Real-Time Tracking of the Human Body", IEEE Transactions on Pattern Analysis and Intelligence, July 1997, vol 19, no 7pp. 780-785

[3] S.M. Smith and J.M. Bardy," ASSET-2: Real-Time Motion Segmentation Shape Tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-17, No 7 , Aug 1995, pp.814-820.

[4] N.C. Mohanty, "Computer Tracking of Moving Targets in Space", IEEE Transactions on Pattern Analysis and Intelligence, Vol. PAMI-3, No 5, Sept. 1981, pp.606-610.

[5] Alton L. Gilbert, Michael K. Giles, Gerald M. Flachs, Robert B. Rogers, Yee Hsun U, "A Real Time Video Tracking System", IEEE Transactions on, PAMI, vol. PAMI-2, No. 1 , January 1980.

[6] D. Koller, K. Danilidis, H.H. Nagel, " Model Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes", International Conference on Computer Vision, 10-3, pp. 257-281, 1993.

[7] Negahdaripour, S and Lee, S., "Motion Recovery from Image Sequences using only First Order Optical Flow Information", International Journal of Computer Vision, 9(3):163-184.

[8] C. Anderson, P. Burt, G. Van der Wal, "Change detection and tracking using pyramid transformation techniques", SPIE- Intelligent Robots and Computer Vision , Vol. 579, pp. 72-78, 1985.

[9] D. Phuttnlocher, G. A. Kalanderman and W.J. Rucklidge, "Comparing Images using Housdroff Distance", IEEE Transactions in PAMI, vol. 15, No. 9, pp. 850-863, Sept. 1993.

[10] Broida, T.J. and Chellappa, R., "Estimation of Object Motion Parameters from Noisy Images", IEEE Transactions on Pattern Analysis and Intelligence, 8(1), pp. 90-99, 1986.

[11] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler, " Calculating the Housdroff Distance between Curves", Information Processing Letters, 64(1):17-22, 14th October, 1997.

[12] Huttenlocher, D.P. and Kedem, K., "Efficiently Computing the Housdroff Distance for Point Sets under Translation", Sixth ACM Symposium on Computational Geometry, 1990, pp 340-349.

[13] Canny, J.,"A Coputational Approach to Edge Detection", IEEE PAMI, 8(6), pp. 679-698, November 1986.