Tracking Moving Objects In Video Sequences [*]

Yiwei Wang, Robert E. Van Dyck, and John F. Doherty
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA16802

Abstract–Object tracking in video sequences is a very important topic and has various applications in video compression, surveillance, robot technology, etc. In many applications, the focus is on tracking moving objects. In this paper, we propose a method for tracking moving objects in video sequences with the presence of the camera motion. The projective/bilinear model is used to estimate the camera motion and the wavelet decomposition is applied to perform multi-resolution analysis and edge extraction. Experimental results are given to demonstrate the performance of the proposed algorithm.

I. Introduction

As object tracking has a lot of applications in areas such as video compression, surveillance, robot technology and so on, many algorithms have been proposed to solve the problem [1][2][3][4]. However, a lot of existing methods first perform computationally expensive spatial segmentation based on gray-scale values[1][3]. This is not necessary in a lot of applications, where only moving objects need to be tracked. The spatial segmentation also create many problems when the algorithm is applied on outdoor sequences, whose background, unlike most test indoor sequences, are not homogeneous. Another problem for most existing algorithms is that they do not consider the camera motion that is very common in many applications [2][4]. In this paper, we propose a method for tracking moving objects in video sequences with the presence of camera motion. The projective/bilinear model is used to deal with the camera motion, and the wavelet transform is applied to perform multi-resolution analysis and edge extraction.

The rest of this paper is organized as follows: In Section II, we review some relevant work and establish the background required for further discussion. Then, we present the algorithm in Section III. Experimental results are given in Section IV, where we demonstrate the performance of the proposed algorithm. In Section V, we present our conclusions, comments and some possible future study directions.

[*]This paper is partially sponsored by AFRL under contract number F30602-98-0061.

II. Background

Before introducing our algorithm, we want to give a short review of relevant work to establish the necessary background.

(A) Camera Motion Estimation

Between consecutive frames of a video sequence, there are usually both camera motions and object motions. Before tracking moving objects, we need to remove the effects of camera motion, such as translation, rotation, zooming, panning, tilting, etc.

There are several existing models for camera motion. We choose the projective/bilinear model since it is able to characterize almost all possible camera motions [5][6]. The projective model is described by eight parameters $(m_i, i = 1, 2, \ldots, 8)$ through equation

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where (x, y) is the original coordinates. The transformed coordinates (x', y') can be obtained by equations

$$x' = \frac{m_1 x + m_2 y + m_3}{m_7 x + m_8 y + 1} = \frac{u}{w};$$

$$y' = \frac{m_4 x + m_5 y + m_6}{m_7 x + m_8 y + 1} = \frac{v}{w}.$$

The projective model can be approximated by the bilinear model

$$x' = q_1 xy + q_2 x + q_3 y + q_4,$$

$$y' = q_5 xy + q_6 x + q_7 y + q_8;$$

which also uses eight parameters $(q_i, i = 1, 2, \ldots, 8)$ to describe the camera motion.

The assumption for 2-D optical flow is

$$u_f E_x + v_f E_y + E_t \approx 0,$$

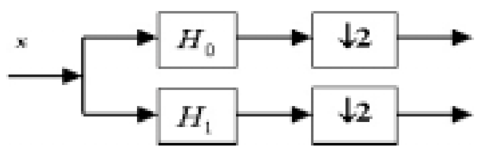
where u_f and v_f are the velocities along x and y directions while E_x, E_y and E_t are the partial derivatives of the gray-scale values with respect to x, y, and t. If we define $u_m = x' - x$ and $v_m = y' - y$, by using the bilinear model and minimizing

$$\epsilon = \sum_x (u_m E_x + v_m E_y + E_t)^2,$$

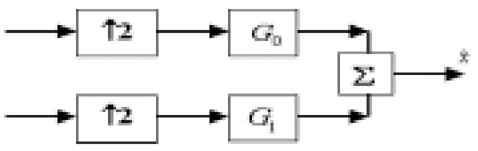
we can get the parameters for the bilinear model. If we know four pairs of coordinates and their corresponding transformed coordinates, we can approximate the parameters for the projective model. The above procedure can be performed on multi-scale pyramid.

(B) Wavelet Transform

The study of the wavelet transform has thrived in the past two decades. The wavelet transform is now widely used in signal analysis, compression, etc., since it can achieve both spatial and frequency localization. A typical 2-channel wavelet decomposition and reconstruction structure is given in Fig. 1 below [7].



(a) analysis part



(b) synthesis part

Fig. 1. 2-channel wavelet decomposition and reconstruction structure

Digital images can be considered as 2-D signals and we can apply the above 1-D wavelet analysis to the horizontal and vertical directions separately [8]. By cascading the structures, we can create a wavelet pyramid that is suitable for multi-resolution analysis. Fig. 2 is an example of a wavelet pyramid of image "Lenna".

III. Algorithm

In this section, we introduce our algorithm for tracking moving objects in video sequences. A video sequence contains a series of frames. Each frame can be considered as an image. If an algorithm can track moving objects between two digital images, it should be able to track moving objects in a video sequence.

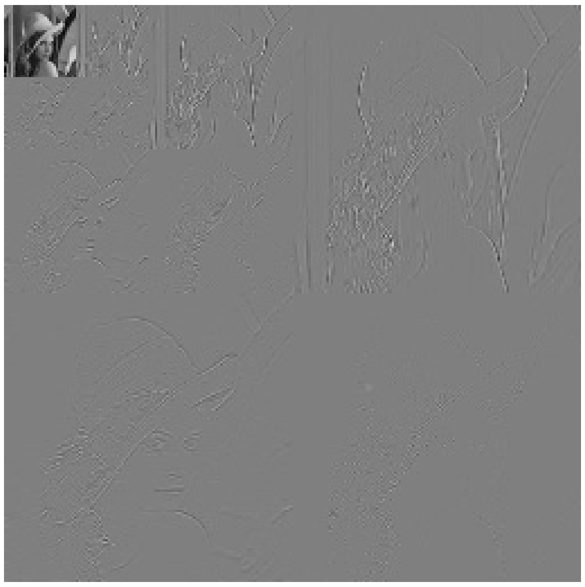


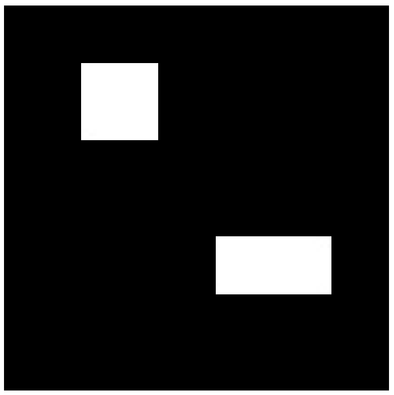
Fig. 2. the wavelet pyramid for "Lenna"

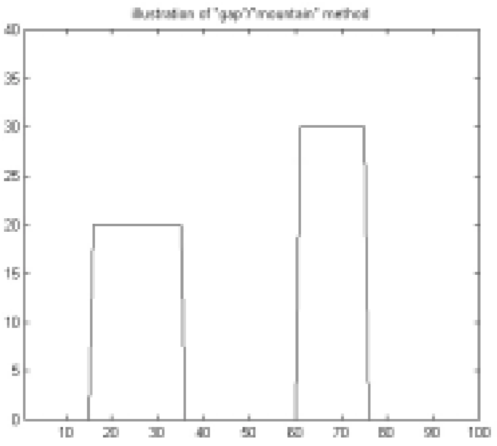
The algorithm starts with two consecutive frames. Each frame is decomposed to create a wavelet pyramid using the wavelet transform. The number of levels and the filters are chosen based on the size of the frame and the content of the video. We choose a wavelet pyramid so that it is possible to incorporate the algorithm into a wavelet compression scheme. First, the camera motion estimation algorithm described in section II is applied to the two DC bands. Second, two edge images are created by adding the magnitudes of the AC bands of the lowest resolution. Then, the motion parameters obtained from the first step are used to align the two edge images. It is obvious that the differences at the location of moving objects will be larger. Hence, by thresholding the difference, we can segment the lowest resolution images into possible object areas and possible background areas. Furthermore, we apply the motion estimation algorithm to only the possible background areas and achieve a better camera motion estimation. After we finish processing the level of the lowest resolution, we repeat the above procedure on higher resolution levels. The DC bands of a higher resolution level can be created from the lower resolution level. And we use the corresponding motion parameters from the lower resolution level as an initial value for a higher resolution value and only consider the parts of images that have been classified as background locations in the lower resolution levels. Basically, the algorithm will select more areas as object areas as the resolution gets higher, until it reaches the highest resolution level.

After we complete the procedure above, we will have different sizes of blocks classified as objects. Some of the blocks are obtained because of moving objects, while others are just the result of noise. Therefore, we need to group the blocks of the same object and eliminate the noise effect. We start with the edge image at the highest resolution level and set all pixels classified as background to zero. Then we add the values for each row to get a column vector. A zero element in the column vector means that there are no object pixels in the corresponding row. If the number of consecutive zeros is larger than a preset threshold, we call it a "gap". The group of elements between two "gaps" is called a "mountain" and the element with the largest value is called a "peak". If the width of a "mountain" is larger than a preset threshold and its "peak" is high enough, we conclude that there is at least one object in the "mountain". If we regard an image as a $p \times q$ matrix, after the above procedure, all the possible objects are in m smaller matrices and each of them has q columns. If we apply the same idea to these matrices and add the values for each row instead of column, the total size of the matrices that contain objects will become smaller. The same algorithm is applied iteratively until the total size of the matrices does not change any more. Each final matrix will be considered as an object location. In summary, by adding along columns and rows repeatedly, we are able to isolate every object.

To explain the "gap"/"mountain" method described above more clearly, we will give a simple example. A 100 by 100 binary image is shown in Fig. 3.(a). The gray-scale value for the background is zero and there are two object rectangular of gray-scale one in the image. The sizes of the objects are 20 by 20 and 15 by 30. By adding the value for each row, we get a column vector of 100 elements. We plot it in Fig. 3.(b). If we set the thresholds for "gap" width and "mountain" width to be 10, we can see that there are three "gaps" and two "mountains" and the "mountains" are from row 16 to 35 and from row 61 to 75. Hence, instead of knowing the possible objects are in the initial 100 by 100 matrix, now we know that they are in two smaller matrices. The first one is 20 by 100 and the second is 15 by 100. Then, we apply the same idea to these two smaller matrices one by one. However, instead of adding the values for each row, we add along columns and detect the "gaps" and "mountains" in the corresponding row vectors. The algorithm is applied iteratively until the matrices do not change anymore. In this example, we only need three steps. The left, right, top, bottom boundary location for the first object are 21, 40, 16 and 35 respectively and those for the second object are 56, 85, 61 and 75.

The object areas obtained from the above step usually contain the locations of the moving object in the current frame and the previous frame. To get more accurate information about the location, the described comparison should be performed between the current frame and the next frame as well. The mutual areas detected in the current frame are then considered the object locations.


(a) the example image



(b) the plot of the column vector

Fig. 3. an example

When the object locations are determined in each frame, we need to track the objects between the frames. Most temporal tracking methods can be used at this point [1][4]. In our experiment, we use a simple yet effective method based on the assumption that the size of the objects do not change much between adjacent frames. If we define the centroid of an object (c_x, c_y) as

$$c_x = (\sum_{(i,j)\in\mathbf{O}} p_{i,j} \cdot i)/(\sum_{(i,j)\in\mathbf{O}} p_{i,j}),$$

$$c_y = (\sum_{(i,j)\in\mathbf{O}} p_{i,j} \cdot j)/(\sum_{(i,j)\in\mathbf{O}} p_{i,j});$$

and the dispersion of an object f as

$$f = (\sum_{(i,j)\in\mathbf{O}} \sqrt{(i-c_x)^2 + (j-c_y)^2} \cdot p_{i,j})/(\sum_{(i,j)\in\mathbf{O}} p_{i,j});$$

where $\mathbf{O}$ is the set of coordinates of an object area and $p_{i,j}$ is the value of the edge image at position (i,j), then we can track the objects by comparing the dispersion f. Besides simplicity, another advantage of this tracking algorithm is that it can track fast movement (e.g., swapping places) while some existing methods [1] have difficulties.

To illustrate the whole algorithm, a block diagram is given in Fig. 4.

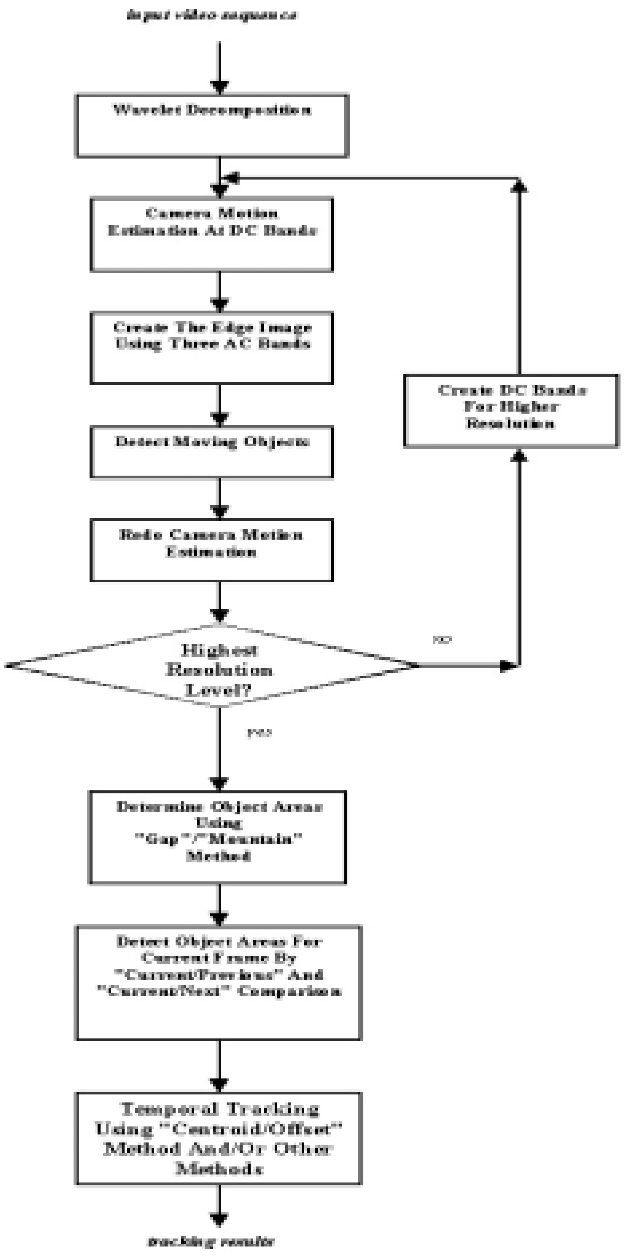

input video sequence
Wavelet Decomposition
Camera Motion Estimation At DC Bands
Create The Edge Image Using Three AC Bands
Create DC Bands For Higher Resolution
Detect Moving Objects
Redo Camera Motion Estimation
Highest Resolution Level?
no
yes
Determine Object Areas Using "Gap"/"Mountain" Method
Detect Object Areas For Current Frame By "Current/Previous" And "Current/Next" Comparison
Temporal Tracking Using "Centroid/Offset" Method And/Or Other Methods
tracking results

Fig. 4. algorithm block diagram

IV. Experimental results

In this section, we will show the experimental results of our algorithm.

The algorithm is first tested on the sequence "toy vehicle". The frame size is 512 by 512. Two adjacent frames (frame 4 and frame 5) are shown in Fig. 5(a) and Fig. 5(b). By applying the algorithm described in section III, two object areas are detected in each frame. We choose the number of wavelet decomposition levels to be four and use Daubechies' 5/7 biorthogonal filters. By adding special effects, we highlight the object areas in Fig. 6(a) and Fig. 6(b). The centroids and dispersions are listed in Table. 1. The objects can be tracked based on the dispersions and the final results are given in Fig. 7.

Table. 1. Centroids and dispersions

Frame 4						
Boundary				Centroid		Dispersion
Left	Right	Top	Bottom	c_x	c_y	
33	116	285	356	72.8	321.0	27.6
225	404	305	352	328.5	328.6	40.8

Frame 5						
Boundary				Centroid		Dispersion
Left	Right	Top	Bottom	c_x	c_y	
1	132	285	386	75.8	329.8	31.1
193	338	289	356	267.5	324.7	39.3

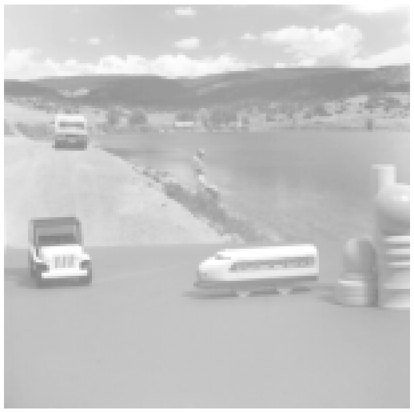
Because the sequence of "toy vehicle" does not have much camera motion, the algorithm is also tested on the sequence "tank". The frame size is 240 by 320. Three consecutive frames are shown in Fig. 8. with obvious camera motion between frames. We choose the number of decomposition levels to be three and use Daubechies' 5/7 biorthogonal filters. The tracking result of the middle frame is given in Fig. 9.

V. Conclusion, comment and future study

In this paper, we present an algorithm to track moving objects in a video sequence. Our experimental results show that the algorithm can track the moving objects with and without the presence of camera motion.

Although we use a simple algorithm to do the temporal tracking, other methods can also be applied after the object areas are determined. One can even develop an algorithm to weigh in different tracking methods to achieve more accurate results.

Since the projective model for camera motion is only suitable for: 1) images taken from the same location of an arbitrary 3-D scene with the camera free to pan, tilt, rotate and zoom; or 2) images of a flat scene taken from arbitrary locations [5]; the proposed algorithm, like a lot of existing algorithms, cannot solve the problem of occlusion. In future study, we will try to incorporate occlusion-killer algorithms in our method.


(a) frame 4

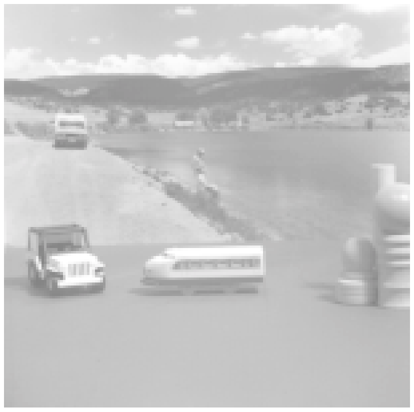
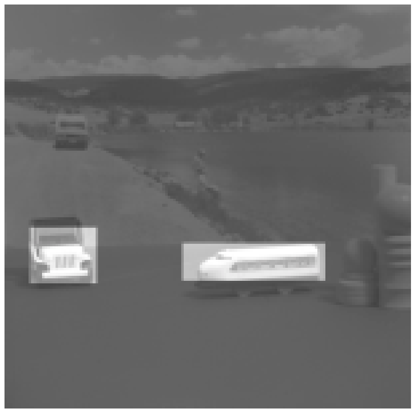

(b) frame 5

Fig. 5. two frames from the “toy vehicle” sequence


(a) frame 4

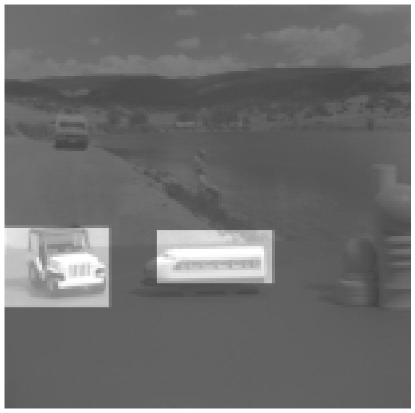
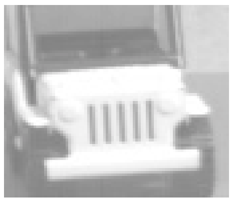

(b) frame 5

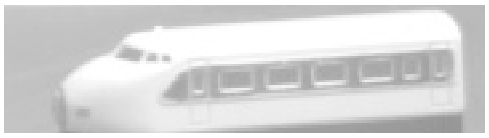
Fig. 6. detected object areas in the two frames

References

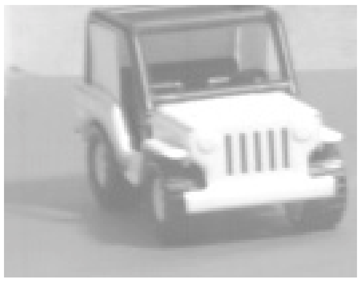
[1] Wang, D.; *“Unsupervised Video segmentation Based On Watersheds And Temporal Tracking”*; Trans. Circuits Syst. Video Technol., vol 8 5, Sept. 1998, pp539-46;
[2] Foresti, G.L.; *“Object Recognition And Tracking For Remote Video Surveillance”*; Trans. Circuits Syst. Video Technol., vol 9 7, Oct. 1999, pp1045-62;
[3] Salembier, P.; Marqués, F.; Pardàs, M.; Morros, J.R.; Corset, I.; Jeannin, S.; Bouchard, L.; Meyer, F.; Marcotegui, B.; *“Segmentation-Based Video Coding System Allowing The Manipulation Of Objects”*; Trans. Circuits Syst. Video Technol., vol 7 1, Feb. 1997, pp60-74;
[4] Lipton, A.J.; Fujiyoshi, H.; Patil, R.S. ; *“Moving Target Classification And Tracking From Real-time Video”*; Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on, 1998, pp8-14;
[5] Mann, S.; Picard, R.W.; *“Video Orbits of The Projective Group: A Simple approach To Featureless Estimation Of Parameters”*; IEEE Trans. on Image Processing, vol 6 9, Sept. 1997, pp1281-95;
[6] Lertrattanapanich, S.; Bose, N.K.; *“Latest Results On High-resolution Reconstruction From Video Sequence”*; Technical Report Of IEICE.; DSP99-140; Dec. 1999, pp59-65;
[7] Vetterli, M.; Kovacevic, J.; *“Wavelets And Subband Coding”*; Prentice-Hall,INC., 1995;
[8] Antonini, M.; Barlaud, M.; Mathieu, P.; Daubechies, I.; *“Image Coding Using Wavelet Transform”*; IEEE Trans. on Image Processing, vol 1 2, Apr. 1992, pp205-20.



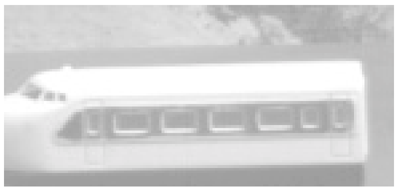
(a) frame 4, object 1: centroid $(c_x, c_y) = (72.8, 321.0)$, dispersion $f = 27.6$;



(b) frame 4, object 2: centroid $(c_x, c_y) = (328.5, 328.6)$, dispersion $f = 40.8$;

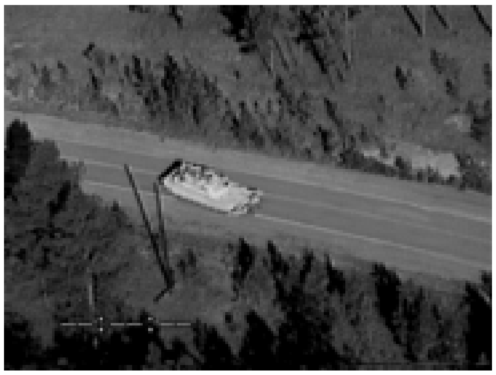


(c) frame 5, object 1: centroid $(c_x, c_y) = (75.8, 329.8)$, dispersion $f = 31.1$;

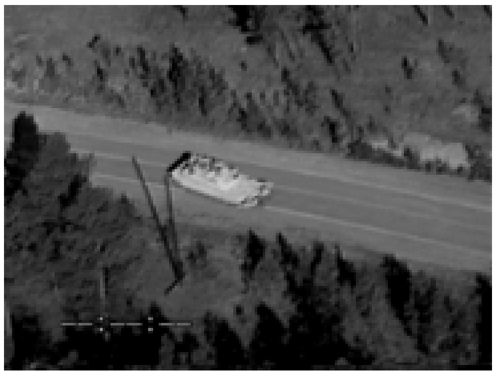


(d) frame 5, object 2: centroid $(c_x, c_y) = (267.5, 324.7)$, dispersion $f = 39.3$.

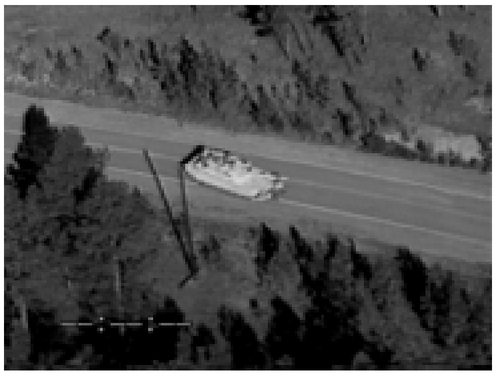
Fig. 7. tracking results of the two "toy vehicle" frames



(a) frame 1



(b) frame 2



(c) frame 3

Fig. 8. three consecutive frames from the "tank" sequence

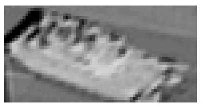


Fig. 9. Tracking result of the middle frame in fig. 8. centroid $(c_x, c_y) = (140.9, 114.9)$, dispersion $f = 20.5$;