Real-time Multiple Tracking Using A Combined Technique

Hui-Huang Hsu, Timothy K. Shih, Chia-Tong Tang, Yi-Chun Liao
Multimedia Information Network Lab.
Department of Computer Science and Information Engineering
Tamkang University, Taiwan, ROC
E-mail: hhsu@cs.tku.edu.tw, tshih@cs.tku.edu.tw, g8190221@tkgis.tku.edu.tw

Abstract. This paper considers a real-time multi-object tracking algorithm for rigid and non-rigid objects. The major components of the tracking object system are extraction of the background image, adaptation of the background image, and identification of the extracted object. In each component, we improved existing methods without increasing the complexity of computation. The system was tested on our fish tank experiment that solves dynamic occlusions problems.

Keywords: Object tracking, Occlusion detection, Object identification.

1. Introduction

One of the current research trends in the video processing field is object tracking. The most important components of object tracking are the background, the movement of object, and the number of the tracking objects. But due to the limited of the power of computer, it is difficult to have good performance in real time. In multimedia standards, for example, MPEG-4, the extraction of an interested object is essential for a high compression rate. In this paper, we will address a real-time algorithm for identifying and tracking objects.

The Kalman Filter (KF) has been used for automatic object tracking from video sequences for video surveillance or scene understanding. The new adaptive Kalman filter (EKF) was proposed to suit for nonlinear systems and to extend the model defined by KM [1]. Based on linear prediction, the accurate and robust tracking algorithm, called maximum entropy method (MEM), was used in common cases [2]. A fuzzy clustering method was applied in surveillance video analysis [3].

Some existing systems focus on the occlusions problem by using multiple camera [4][5][6] or shape matching method [7]. And other methods, for example, contour-based, region-based, and mesh-based tracking, are applied in different environment.

This paper proposes and evaluates a frame-based method for object tracking in real time. In our method, it is possible to extract shape contour rapidly and dynamically update the background image. The complex function is not needed in our algorithm to reach good performance. The paper is organized as follows. Section 2 contains a description of the proposed system. The architecture of tracking system includes three modules: background and object extraction, background update, and object tracking. In Section 3, we have experiment results on real-time tracking of multiple objects. In Section 4, a brief conclusion is drawn.

2. The Proposed Method

The proposed method consists of three major components: background building & object classification, background adaptation, and object tracking. The challenge is to get a whole background that will influence the validity of object extracting. With the real-time restraint, only two source images are needed. Using our inpainting method for the background image, the background can be obtained rapidly. According to the background, we can get the object data with missing data. The image enhancement process is then used this paper to recover the lost data. Due to the effects caused by the illumination and the object movement, the background adaptation is necessary. Each module is described in the following subsections.

2.1 Background Separation and Contour Extraction

At first, the illumination difference method is used to build an approximate background. The difference of lightness in each pixel between the current image and preceding image is then computed. If the result is larger than the threshold, the system sets the pixel value equal to 255. Otherwise, it sets the value to 0.

$$P_j(x,y)=\begin{cases}255 & \text{if } |I_j(x,y)-B_j(x,y)|>T\\ 0 & \text{Otherwise,}\end{cases}$$

T: the threshold to determine that the difference of lightness is in the user-defined range.

An approximate background is shown in Figure 1 with many noises. Although the noises can be erased by increasing the threshold, but that will increase the number of lost pixels of the extracted object. To conquer this problem, there are two major solutions:

- Averaging Method [8]

  Compute the average value of the pixel value of the background bitmap and the current image at the same position.

- Inpainting Method

  Get two images continuously and compute the lightness difference to find out the pixels which belong to background. If the difference is smaller than the threshold (set by the user), the pixel is filled into the background bitmap at the same position.

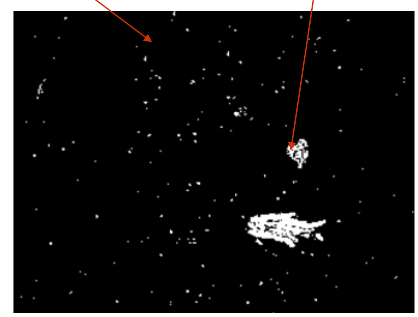


Figure1. An approximate background

If there are too many tracking objects, the background image will be inaccurate by the averaging method. Because the pixels of the extracting object may be involved into the averaged pixels, we proposed a new inpainting method that combines the averaging method and the original inpainting method. Here we describe the inpainting method in details:

Step 1. Get two images continuously.

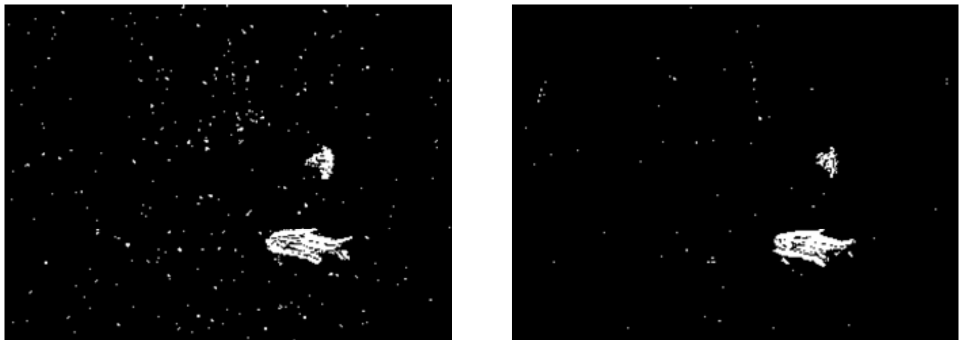
Step 2. Compute the difference of the pixel values at each position.

Step 3. Determine whether the difference is smaller than the threshold value or not. If the difference is smaller than the threshold value, it is viewed as a background point and the average existed value is filled in background bitmap at the same position.

Step 4. Repeat Step 1 to Step 3 until the complete background bitmap is built.

After the background image is obtained, it is used to compare with the current receive image for separating out foreground objects. First, the difference between the current image and the background image is calculated to extract objects. Then the median filter and

morphological operations are used to eliminate noises and join the fractions of the extracted objects.
We compare the current image with the background image and compute the difference of lightness at each point. If the difference is bigger than the threshold, the point is marked as a part of the object (as shown in Figure 2).



(a) diff > 10 (b) diff > 20

Figure 2. Foreground image with difference threshold (Background point: black, foreground point: white, diff: difference of lightness).

- **Median filter**

All the pixels values (gray-level) in an N*N mask are arranged in a sequential order (from the smallest to the largest), then the middle value is selected from the ordered set to replace the value of the central pixel. After using the Median filter to remove the isolation points and smooth the current image, we adopt the morphological operations to eliminate the noise spikes, fill in small anomalies, and connect the fragments of the extracting object.

- **Closing operation**

The closing operation performs the dilation followed by the erosion operation. Usually, it is used to fill in small holes or gaps and connect object's fragments.

Dilation A structuring element looks like Figure 3. The original is the central point. If the origin of the structuring element coincides with '0' in the 3*3 image, no change is needed; If the origin of the structuring element coincide with '1' in the 3*3 image, the OR logic operation is performed on all pixels of the structuring element. A single pixel will be inserted to the boundary of the extracting object.

Erosion If the origin of the structuring element coincides with '0' in the 3*3 image, no change is needed; If the origin of the structuring element coincides with '1' in the 3*3 image, the AND logic operation is performed on all pixels within the structuring element. A single pixel will be removed from the boundary of extracting object

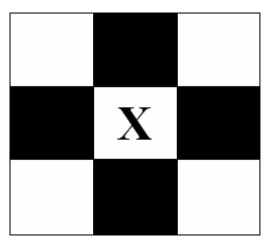


Figure 3. A structuring element within which X is the origin

After the closing operation is performed, the complete object bitmap is obtained with the needed object information (as shown in Figure 4).

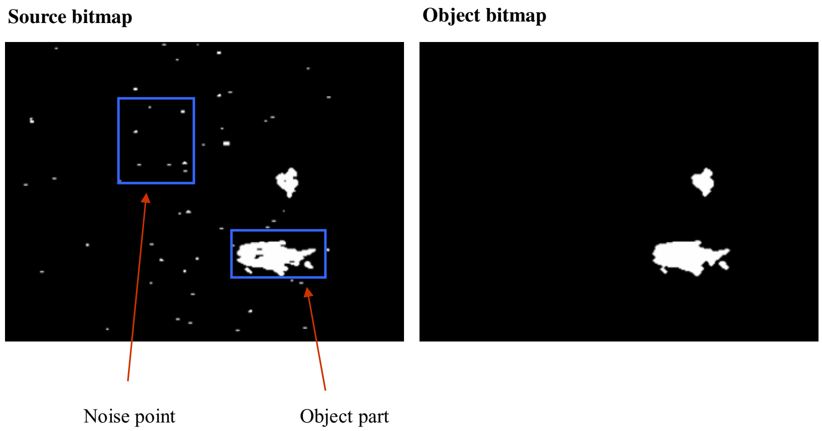


Figure 4. The complete extracted object

2.2 Dynamical Background Adaptation

The illumination change and the background change are the major factors of getting an inaccurate background image. Thus a dynamical background update mechanism is necessary. The major component of background adaptation is to calculate the difference between the current frame and the successive frame. If the difference exceeds a threshold, the background image must be re-found by the inpainting method.

2.3 Object Classification and Object Tracking

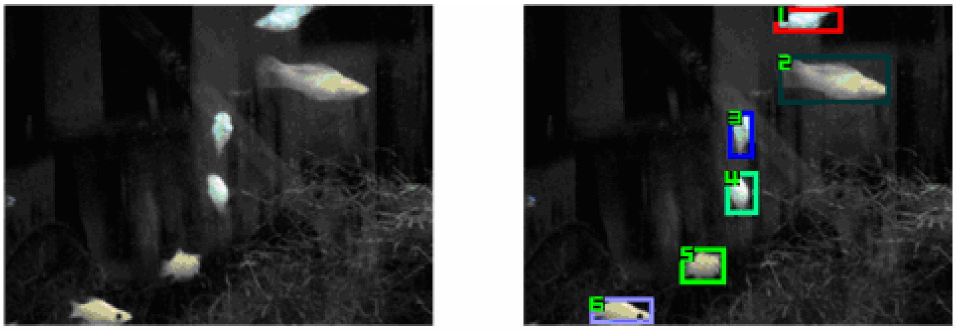
We use a recursive method to estimate the profile information for classifying each object and collect the motion information of the tracking object during the classifying process. In our system, the object table is built to contain the object profile. We label different objects with different colors. The following description shows our labeling process:

```
IF pixel(x, y) belongs to object THEN
    assign a specific color value to pixel(x, y);
    search four neighboring pixels of the (x, y);
    //up, down, left, right
    record the position of the center of gravity ,
    object's color value and the area of object;
  IF area > Threshold THEN
        draw a rectangle according to the center of
        gravity of the extracted object ;
  END IF
END IF
```

After the classified process on the current image, we can get each object's new information (the position of the matching, the color histogram in the RGB color space) and use the movement of object's mass point to compute the average of the motion vector.

Figure 5 (a) is the source bitmap, and each object is classified and shown in Figure 5 (b). Then we collect all object information to build an object table, and use in the following tracking process.

There are several algorithms, e.g., the Kalman filter, for reference motion vector detection [9][10], and the Condensation algorithm about reference color information, to solve the problem of object tracking. But if the system only adopts one kind of those algorithms, it will be restricted to a special environment. Therefore our method combines the movement of the object and the color information of the object. At same time, we solve the problem of multi-objects with an unfixed moving direction.



(a) object's bitmap (b) After classify each object

Figure5. The object after classify process

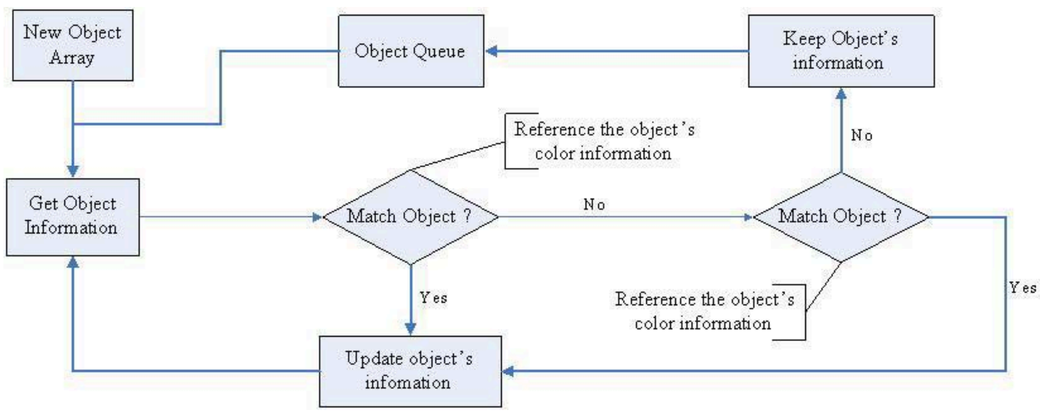


Figure 6. Overview of the tracking method

In the closed loop of the Figure 6, the prediction of the position of the object is based on the motion and the color information in the object table is used to confirm the matched object. The following steps show our method by using the color information and the motion vector.

Step 1. Get the object information from the object table.

Step 2. Predict the object's center by using average of the motion vectors of successive frames.

Step 3. If the object is predicted, using the color histogram of the object to determine whether it is a correct object or not.

Step 4. If the color histogram is matched, update the object information of the object table.

Step 5. If the color information is not matched, Search the x-axis from LCM-LM to LCM+LM, and the y-axis from LCM-LM to LCM+LM. (LMP: last center of mass. LM: last average moment.)

Step 6. If the object is found in Step 5, update the object information of object table. If the object can not be found in Step 5, keep the object information and search it in the next frame.

In occlusion situation, the object can not be tracked temporarily in the current image. The object information is kept in the object table for a searching process until it can be tracked again. Over a limited time, if the overlapped objects cannot separate from each other, the overlapped objects will be regarded as one object. A new object will then be defined in the object table.

3. Experimental results

Three videos are used to test the qualitative and quantitative performance in our experiment. Our system runs on a P4 1.8GHz PC. In Table1, the frame rate is 15 frames per second. Comparing to other systems [10], it is better than the other methods in which the frame rate is 7~10 frames per second. We use a fish tank with 20 fishes of different sizes. Because the fish has properties of fast and unfixed moves, we also can apply it in other environment.

Table1. The performance of experiments

	Test # of Frame	Time (sec)	Frame Rate (frames/sec)
Movie 1	1000	67.2	14.88
Movie 2	1000	66.4	15.06
Movie 3	1000	68.1	14.68

4. Conclusions

Classical object tracking method is restrained easily by the number of objects, the speed of object, and the unfixed motion vector. We proposed a frame based algorithm to overcome the problems. In the experimental result, we showed that it is acceptable in the real-time environment. In addition, the interaction with the user will be addressed in our future work.

References

1. P. Wira, and J. P. Urban, "A new adaptive Kalman filter applied to visual servoing tasks," Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Sept. 2000.
2. P. Y. Yeoh, and S. A. R. Abu-Bakar, "Accurate real-time object tracking with linear prediction method," Proceedings of the 2003 International Conference on Image Processing, Sept. 2003.
3. D. Xie, Hu Weiming, Tieniu Tan, and Junyi Peng, "A multi-object tracking system for surveillance video analysis," Proceedings of the 17th International Conference on Pattern Recognition, Aug., 2004.
4. S. L. Dockstader, A. M. Tekalp, "Multiple camera fusion for multi-object tracking," Proceedings of the 2001 IEEE Workshop on the Multi-Object Tracking, July 2001.
5. S. Yonemoto, A. Matsumoto, D. Arita, and R. I. Taniguchi, "A real-time motion capture system with multiple camera fusion," Proceedings of International Conference on Image Analysis and Processing, Sept. 1999.
6. S. L. Dockstader, A. M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," Proceedings of the IEEE , Volume: 89 , Issue: 10 , Oct. 2001.
7. L. Marcenaro, M. Ferrari, L. Marchesotti, and C.S. Regazzoni, "Multiple object tracking under heavy occlusions by using Kalman filters based on shape matching," Proceedings of the 2002 International Conference on Image Processing , Volume 3, June 2002.
8. H. T. Chen, H. H. Lin, and T. L. Liu, "Multi-object tracking using dynamical graph matching," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognitio. Dec. 2001.
9. Huang Yu, T.S. Huang, and H. Niemann, "Segmentation-based object tracking using image warping and Kalman filtering," Proc. International Conference on Image Processing, June 2002.
10. L. Marcenaro, M. Ferrari, L. Marchesotti, and C.S. Regazzoni. "Multiple object tracking under heavy occlusions by using Kalman filters based on shape matching," Proc. International Conference on Image Processing, June 20.