行政院國家科學委員會專題研究計畫　期中進度報告

總計畫(1/3)

計畫類別：整合型計畫
計畫編號：NSC91-2219-E-002-044-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：國立臺灣大學電機工程學系暨研究所

計畫主持人：貝蘇章
共同主持人：陳良基，李枝宏，馮世邁

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92 年 5 月 15 日

智慧型音視訊和傳輸技術及多媒體應用(I)–總計畫

Intelligent audio/video/transmission techniques and multimedia applications (I)

計畫編號: NSC-91-2219-E-002-044
執行期限：91 年 8 月 1 日至 92 年 7 月 31 日

主持人：貝蘇章　台灣大學電機系教授

摘要

由於多媒體的資訊日益遽增，人們對於多媒體的資訊的使用也日漸頻繁，所以必須要有強有力的工具去管理、索引、搜尋、過濾影片，而移動物體擷取則是此工具最基本也最重要的一環，尤其是對於體育節目。而這篇報告主要是針對如何在動態背景下能夠有效的抓出移動的物件，雖然需要人工的介入，但之後藉由一些自動化的步驟，一樣可以把我們想要的移動物件追蹤以及擷取出來。我們針對幾個不同的影片作實驗，其結果亦顯示我們的方法的確能有效擷取出動態背景中的移動物體。

關鍵字: 視訊分析、移動物體擷取、移動估測

Abstract

As the amount of available video data is overwhelming, people prefer to access this information actively. Powerful tools for video indexing, retrieval and filtering are in great demand. Moving object extraction plays an important role in these tools, especially for sports videos. This report mainly discusses a semi-automatic video object extraction algorithm using object tracking by motion and color projection. Experimental results show the good performance of proposed method.

Keywords: Video analysis、moving object extraction、motion estimation

Introduction

Besides the requirement of object-base in mpeg-4 standard, the capability of extracting moving objects from a video sequence is also a fundamental and crucial problem of many vision systems including video surveillance, video editing, event recognition, and so on.

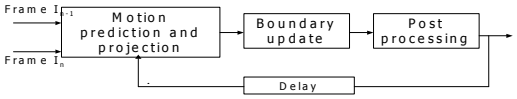


Fig. 4-1 block diagram of video objects extraction in moving background

However, a good video object extraction

algorithm is usually time-consuming and complicated. We introduce a simpler system in this paper to extract moving objects in video with moving background. Figure 4-1 shows the block diagram of our system. Every processing block is very efficient and use simple operation to complete. That will be explained in successive section.

This report is organized as follow: section 2 introduces some related researches recently. Motion estimation and motion projection will be explained in section 3, and a fast search algorithm “spiral search” will also be presented in this section. In section 4 and section 5, we will respectively explain how to keep the object boundary exact and how to eliminate the salient at the boundary and the hole in the video object. And we will give some experimental results in section 6.

Related work

Generally speaking, there are two major methods in the video object extraction scheme: the spatial and the temporal segmentation. The spatial segmentation use spatial information of pixels such as the colors, positions, and relative distances, etc [31], [32]. However, these methods using only spatial information may not converge to the target edge if the distinction between the object and the background is ambiguous.

Temporal segmentation techniques exploit the time-variant property of video sequence. The motion information of objects is especially important when the camera is not static or when we have to track the objects. There are two popular methods for temporal segmentation. One is a change detection-based approach [33][34], where potential foregrounds are detected by applying statistical hypothesis test to interframe differences. Its major characteristic is to automatically detect new appearance of an object in the scene, and it is very useful in surveillance system. However, in case of video sequences with non-stationary background, the difference of frames might be too noisy to extract object easily.

The other approach for temporal segmentation is based on object tracking. Most of them employ a tracking mechanism that projects the objects of previous frame to the current frame [35][36][37]. Since this approach tends to be robust to sequences with moving background, it is now becoming commonly accepted although the adoption of human interaction in the processed might be required. The disadvantage of these methods always result from the complexity of adjusting the projected regions boundaries in the current frame.

Therefore, we propose an efficient semi-automatic method to deal with the sequences of moving background based on object tracking by motion and color projection. We refer to [38] proposed by j. N. Hwang, we use some simpler process elements to construct whole system. It is different to the existing temporal segmentation methods, since we use the color backprojection and morphological operation are quite low cost in computation.

Motion projection

The object tracking process begins with motion projection to locate a video object with rough object boundary information. Firstly, motion estimation is required for it. We use transformation function to describe the motion field between two successive frames:

(1)

Where , are frame *n* and frame *n-1* respectively, and the relationship between $I_n(x,y)$ and $I_{n-1}(x',y')$ is defined by the transformation functions $x' = f(x,y)$ and $y' = g(x,y)$. There are lots of algorithms of motion estimation, and the fast spin search is used for motion estimation in our procedure because it is fast with reasonable accuracy. Thus the transform functions for the pixel of the image are

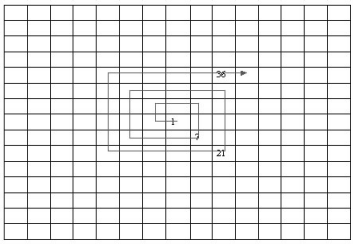
$$f(x,y) = x - u(x,y), \tag{2}$$

$$g(x,y) = y - v(x,y), \tag{3}$$

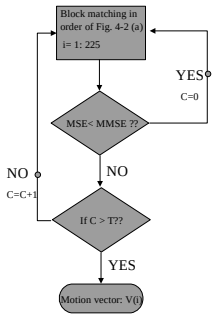
Where is the estimated motion vector at . The spiral search algorithm matches all the checking points inside the search window as the full search algorithm. The search begins at the origin checking point and then moves outwards with a spiral scanning path, as shown in figure 2 (a). This order of searching is to exploit the center-biased motion-vector distribution characteristics of the real-world video sequence [39] [40]. During each block matching, the spiral search algorithm compares each mad with the minimal mad. If there is still no mad is larger than minimal mad until the counter is larger than the threshold, the motion vector will be decided and the process will stop. The procedures of motion-vector decision in spiral search algorithm are summarized in the flowchart as shown in figure 2 (b). Motion projection is performed with video object extracted in the previous frame. The correspondence of video object into frame can be approximately estimated by projecting into according to the motion vector. The projected region p of into is described by

$$P = \{(x + u(x,y), y + v(x,y)) \mid (x,y) \in vo_{n-1}\} \quad (4)$$

Since the projected points by motion information of block-base searching method might not exactly fall onto the video object in the current frame, boundary refinement algorithms have been used to correct the motion projected region. Usually, spatial image segmentation using color/gray level information is conducted to get partitioned image and match projected regions to some of partitions. Difficulty is generated in dealing with this partitioned image, since this image tends to be over-segmented. Region merging based on homogeneous color/motion can then be applied [36]. However, the adopted region merging schemes require complicated algorithms and are time-consuming. In the next section, we show simple and efficient method to correct the boundary locations of motion projected regions.



(a)



(b)

Fig. 2 illustration of spiral search method:

(a) the search path of spiral search method;

(b) the algorithm of motion vector decision in spiral search.

Boundary refinement

In the previous section, we performed motion projection by projecting vo_{n-1} onto the current frame (see figure 3 (a)). However, since we use block-based search not pixel-wise search, the projected points may not be exactly corresponding to the correct positions of video object in the current frame (see figure 3 (b)). In order to correct the motion projected positions, we use the histogram backprojection. The histogram backprojection was originally proposed for solving the location problem by swain and ballard [41]. While it was introduced to locate a target in the projected image, we modify it to extract video objects with pixelwise accuracy. In histogram backprojection the model *p*, which is a motion projected video object from the previous frame, and its spatially swollen region *d* (see figure 3 (c)) are represented by their multidimensional color histograms $H_j(P)$ and $H_j(D)$. A histogram H_j is the count of pixels whose color values fall into the same bin *j*. In our approach, color image were originally digitized at 24-bit rgb, and equally re-quantized to 4 bits in each channel, thus resulting in 2^{12} bins in rgb color space. However, in [38], the author use distance transform [42] to get swollen model *d*, but we use more simple method to get it (see figure 3 (c)). We first use a 5*5 kernel to do dilation, and then use the 5*5 kernel to do the selective erosion. The effect is almost the same with distance transform.

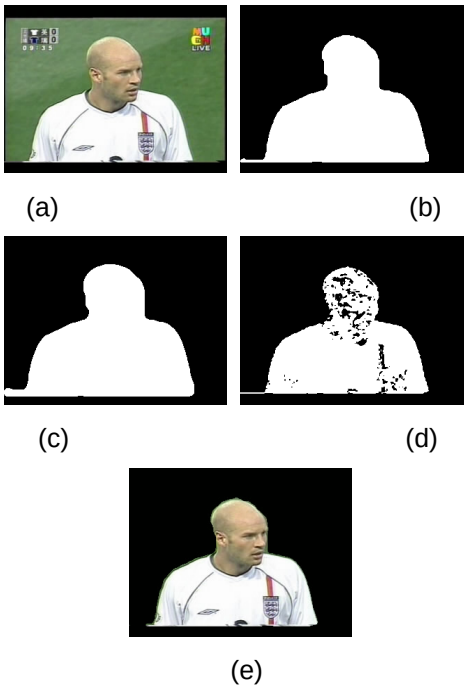
The selective erosion acts like general erosion but it doesn't change the shape unless the area is smaller than the threshold T_{AREA}. For example, figure 4 (a) is the chosen kernel of erosion. By setting T_{AREA} to 4 (pixels), the result of the traditional erosion and selective erosion are shown in figure 4 (c) and (d), respectively. Gray cells are pixels eroded after operation. Note that if we use two 1-d erosion kernels, for example, horizontal and vertical, the same function can be achieved by iterate those two 1-d erosion several times.

After getting a swollen model *d*, a ratio histogram *r* is defined as .

$$R^j = \min\left(\frac{H^j(P)}{H^j(D)}, 1\right) \quad (5)$$

It is this histogram *r* that is backprojected onto the image, i.e., the image values are replaced by the values of *r*. Let *h(c)* be the histogram function that maps a color *c* (a

re-quantized rgb color value) to a histogram bin. The $R_{h(c(x,y))}$ is calculated for each position *(x, y)*. This backprojected image is then



(a) (b)

(c) (d)

(e)

Fig. 3 illustration of the process in our proposed method:

(a) original image;

(b) motion projection region p;

(c) the swollen model d;

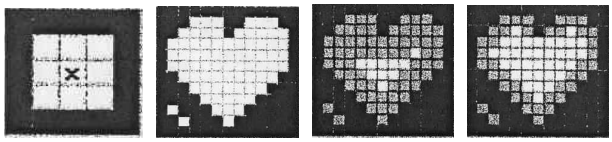
(d) refined object boundaries using histogram backprojection;

(e) final vop after post processing.

Thresholded to be a binarized image b (see figure 3 (d)).

$$\begin{cases} B(x,y) = 1 & \text{if } R_{h(c(x,y))} > \bar{}, \\ B(x,y) = 0 & \text{otherwise} \end{cases} \quad (6)$$

Where $\bar{}$ is a threshold value and $0 < \bar{} < 1$. Median filtering can selectively be used before thresholding to reduced the false small holes in region of video object, but it will cost much time to calculate due to the nonlinear filtering. The binary image b is further processed in the final step, called object post-processing, to be explained in the next section.



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

Fig. 4-4 selective erosion [43]

(a) Erosion kernel
(b) Original object
(c) Result of selective erosion by kernel (a) (t_{area} = 4)
(d) result of traditional erosion by kernel (a)

Post processing

Although the extracted video objects produce the overall shape of moving object, there may still exist some errors such as black holes in video objects or scraggly boundaries of objects. These errors are visually annoying and may cause the temporal error propagation. The goal of object post-processing step is to remove those errors and at the same time to smooth the object boundaries. Figure 5 shows the flow of the post processing.

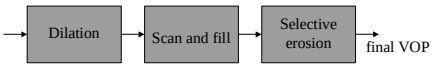


Fig. 5 the flow chart of post processing

In order to solve the problem of the holes in video objects, we will scan the binary image b and fill the inside of the video object.

Assume figure 6 (a) is the result after motion projection and histogram backprojection. Before scan and fill, we dilate the binary image first (shown in figure 6 (b). The structuring usually is 3*3 square). We check each point (1-pixel) of the video object by scanning the image row by row. If the distance between two successive 1-pixels smaller than T_{GAP}, then we fill the space (0-pixels) between them. The horizontal scanning result is shown in figure 6 (c). Note that T_{GAP} useless if it is too small to fill the object. On the other hand, it is also improper if T_{GAP} is too large to fill many wrong pixels. General speaking, it is proportional to the size of the object.

After the horizontal scanning, we apply vertical scanning (column by column) with $0.5 * T_{GAP}$. The result of scan-fill is shown in figure 6 (d).

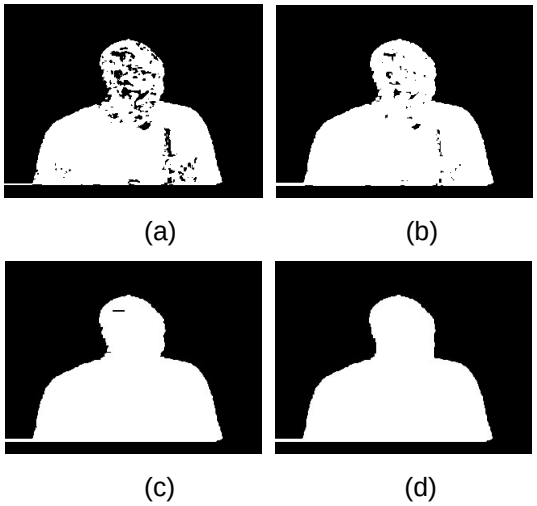


Fig. 6 illustration of post processing:

(a) the result of motion projection and color histogram backprojection

(b) the result of dilating (a)

(c) the result of horizontal scanning and filling

(d) the result of vertical scanning and filling

After the process of scan-fill, there still exist some peninsulas created at the boundaries of objects. So, morphological operation can be adopted to smooth the objects boundaries. We can use different size of structuring elements depending on the size of video objects. There is a tradeoff of conflicting performance requirements, such as noisy background region elimination and accurate object boundary approximation. We final decide to use selective erosion and check the pixel about to erosion whether $P(I(x,y)) > \overline{\quad}$. The filtered map represents

the final outcome of the proposed algorithm (see figure 3 (e)).

Experimental results

In this section, we show the results using the proposed video object extraction system for several sports video sequences. These sequences represent different degrees of difficulties in real situations. The three selected color video sequences are all recorded by vhs and transformed to mpeg-1. The frame size is 360*240 and the frame rate is 30hz in these three sequences. The first video objects are extracted manually in all the implementation results.

The first sequence “shaven-headed” is a soccer player’s close-up scene. The camera has little zoom and panning left-and-right. The motion of the player is relatively small. However, this motion is a non-rigid body motion because the human bodies contain moving and still parts at the same time. We want to extract the player bodies, which is semantically meaningful from the background field. Figure 4-7 shows some frames in the sequence “shaven-headed”. The result of the vops extracted by only motion projection (shown in figure 8). We can find that the video objects cannot be extracted accurately if only motion projection without any boundary refinement. Therefore, in figure 9, the vops extracted by our system are seemed to quite good.

The second sequence “dribbling” is a scene that a soccer player dribbles. In our experiment, the semantically meaningful video objects are the soccer player and the ball. Comparing with the first sequence, the motion is faster between the background and the foreground. In addition, the object is smaller than it in the first sequence. Figure 10 and figure 11 are respectively the original sequence and the extracted vops.

The last sequence “ball-passing” is the most complex sequence of these three. The sequence is the scene that the second baseman catches the ball and pass to first baseman. The meaningful objects we extracted are the players dressed in blue. The two players will occlude and then apart from each other. Figure 12 and figure 13 are respectively the original sequence and the extracted vops.

There are some parameters that need to be set for each sequence: the threshold value to make binarized image b, the threshold value of the gap size in scan and fill, and the parameter of the final selective erosion table 1 shows the parameters setting in the implementation of these sequences.

	Shaven-headed	Dribbling	Ball-passing
$\overline{\ }$	0.6	0.6	0.5
T_{GAP}	15	20	10
Structuring size T_{AREA}	5*5 12	5*5 12	3*3 6

Table 1 the parameters in the implementation

(a) (b)

(c) (d)

(e) (f)

Fig. 7 original frames in sequence “shaven-headed”
(a) frame01; (b) frame07; (c)frame10;
(d) frame21; (e) frame32; (f) frame40

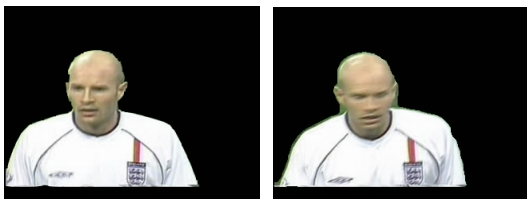
(a) (b)

(c) (d)

(e) (f)

Fig. 8 the result of vops extracted by only motion projection in sequence “shaven-headed”

(a) frame01; (b) frame07; (c)frame10;
(d) frame21; (e) frame32; (f) frame40



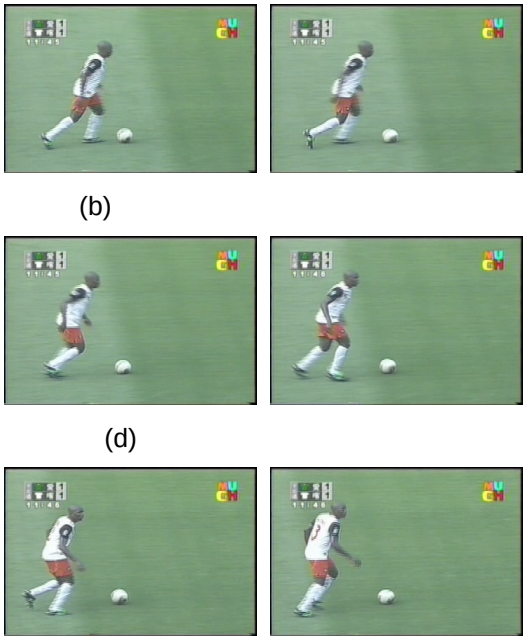
(a) (b)

(c) (d)

(e) (f)

Fig. 9 the result of vops extracted by proposed method in sequence “shaven-headed”

(a) frame01; (b) frame07; (c) frame10;
(d) frame21; (e) frame32; (f) frame40



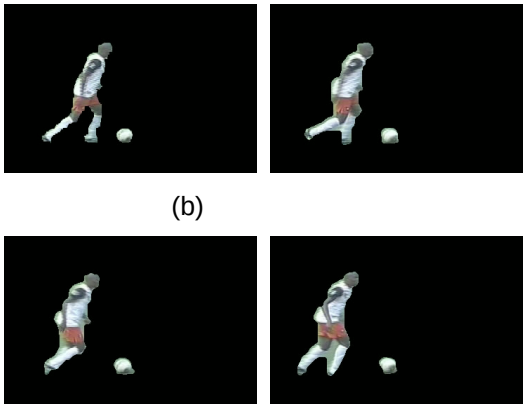
(a) (b)

(c) (d)

(e) (f)

Fig. 10 original frames in sequence “dribble”

(a) frame01; (b) frame03; (c) frame06;
(d) frame09; (e) frame13; (f) frame16

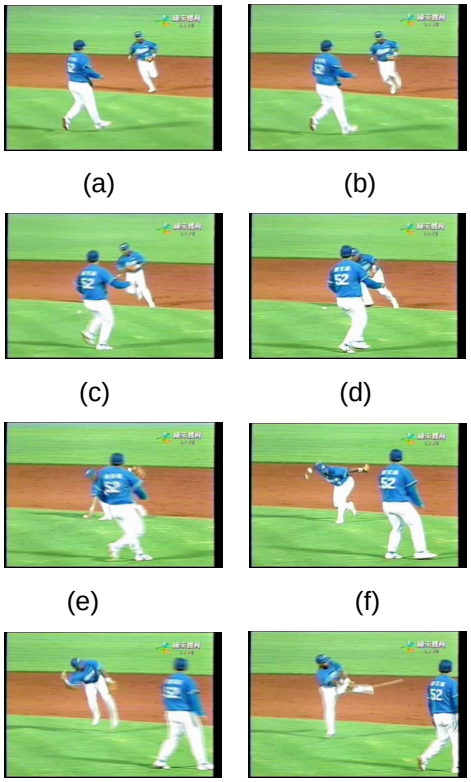


(a) (b)

(c) (d)

(e) (f)

Fig. 11 the result of vops extracted by proposed method in sequence “dribble”
(a) frame01; (b) frame03; (c) frame06; (d) frame09; (e) frame13; (f) frame16



(a) (b)

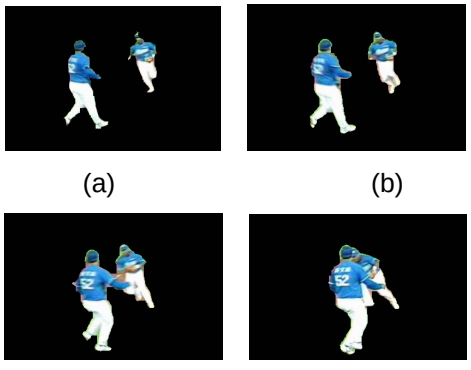
(c) (d)

(e) (f)

(g) (h)

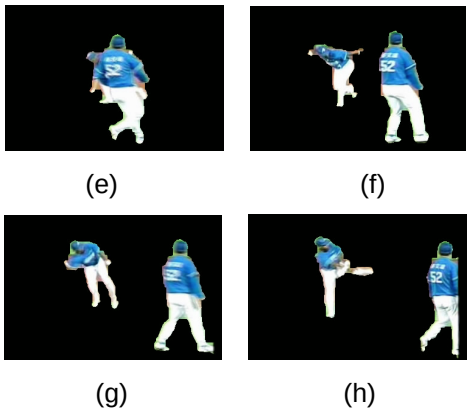
Fig. 12 original frames in sequence “ball-passing”

(a) frame01; (b) frame09; (c) frame14; (d) frame22;
(e) frame41; (f) frame57; (g) frame69



(a) (b)

(c) (d)



(e) (f)

(g) (h)

Fig. 13 the result of VOPs extracted by proposed method in sequence “ball-passing”

(a) frame01; (b) frame09; (c) frame14; (d) frame22; (e) frame41; (f) frame57; (g) frame69

Reference

[1] T.Sikora, “The MPEG-4 Video Standard Verification Model” *IEEE Trans. Circuit Syst. Video Technology*, vol. 7, 1997, pp.19-31

[2] C.Kim and J.N.Hwang, “Video Object Extraction for Object-Oriented Applacation” in *Journal of VLSI Signal Processing 29*, 2001.pp.7-21

[3] M.J. Swain and D.H. Ballard, “Color Indexing” *International Journal of Computer Vision*, vol. 7, no. 1,1991, pp. 11-32.