A novel video object tracking approach using bidirectional projection

Zhi Liu (刘 志) and Jie Yang (杨 杰)

Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030

Received January 12, 2004

This paper proposes a novel video object tracking approach using birdirectional projection. Forward projection is exploited to locate the current video object with rough boundary information. Watershed segmentation is applied to the simplified gradient image of the current frame to obtain a reasonable partition. An improved backward projection, which incorporates pixel classification with region classification, is performed on some segmented regions in a rather small search range, and the tracking performance is enhanced in respect to both reliability and efficiency. Experimental results for various types of the MPEG-4 (moving picture experts group) test sequences demonstrate an efficient and faithful segmentation performance of the proposed approach.

OCIS code: 100.2960.

As an important issue for the implementation of many content-based multimedia applications supported by MPEG-4 (moving picture experts group), video object segmentation remains a challenging research topic till now. At present, efficient algorithms for automatic video object segmentation only apply to moving objects or some kind of objects with *a prior* knowledge. Therefore, a more practical solution, the so-called semi-automatic video object segmentation[1−5], draws more and more attention in recent years. A typical paradigm of semi-automatic video object segmentation consists of two steps: the user defines the interested video object in the first frame, and then the defined video object is automatically tracked in the rest frames of the sequence.

For video object tracking, many existing approaches adopt a two-step configuration to track the video object[1−3], i.e., firstly project the previous object to the current frame using some kind of parametric motion model, and then refine the projected object boundary. The underlying tracking mechanism is forward projection, which works well for rigid objects with translation motion. For non-rigid objects with multiple motions, inevitable refinements are needed to smooth irregular boundaries and fill uncertain holes in the video objects. In contrast with forward projection, backward projection[4,5] is suitable to deal with non-rigid objects, and needs no further refinements. Each segmented region in the current frame is projected to the previous frame, and then it is assigned to the current video object if the majority of the projected region overlaps the previous video object. In nature, it is a region classification approach rather than a tracking approach. However, it is not an efficient way to backward project all segmented regions for classification. Another problem may occur when a segmented region overlaps the video object and the background, which causes peninsulas or gaps to appear on the video object no matter what classification is assigned to the region. In this paper, we propose a bidirectional projection approach mainly as an extension of backward projection[4], which is more efficient due to the combination with forward projection, and ensures the visual quality of the tracked video objects by incorporating pixel classification with region classification.

Our tracking scheme can be defined as obtaining the video object vo_n of the current frame I_n, based on the motion information related with the previous video object vo_{n-1}, and the spatial segmentation information of the current frame. The proposed tracking approach consists of three steps: forward projection, spatial segmentation, and backward projection.

Forward projection: The objective of forward projection is to locate the video object with rough boundary information, which is derived from the motion estimation. For each contour pixel $I_{n-1}(x,y)$ of the previous video object vo_{n-1} (see Fig. 1(a)), the motion vector $(u(x,y), v(x,y))$ is estimated using the three-step searching method[6] to minimize the following prediction error,

$$e(x,y) = \min_{u,v} \sum_{i=-N}^{N} \sum_{j=-N}^{N} \| I_{n-1}(x+i, y+j) - I_n(x+u(x,y)+i, y+v(x,y)+j) \|. \quad (1)$$

The size of the matching block equals $(2N+1) \times (2N+1)$. In our experiments, N is set to 2, and the search range for $(u(x,y), v(x,y))$ is set to $[-7,7]$ for all sequences. Forward projection is performed on all contour pixels of vo_{n-1}, denoted by a pixel set ct_{n-1}. The projection of ct_{n-1} in the current frame I_n can be denoted by another pixel set p_n (see the black pixels in Fig. 1(b))

$$p_n = \{(x+u(x,y), y+v(x,y)) \,|\, (x,y) \in ct_{n-1}\}. \quad (2)$$

All pixels in p_n are then dilated with a disk-shaped structuring element E_d to obtain a band area B_n (see Fig. 1(b)) to accommodate the rotation, scale change, and deformation of the video object. The radius of E_d is set to 15 by the experiment, which is enough for most video sequences to ensure that the true contour ct_n locates in B_n. The approximate translation vector (T^u_{n-1}, T^v_{n-1}) for the video object is estimated using the average of motion

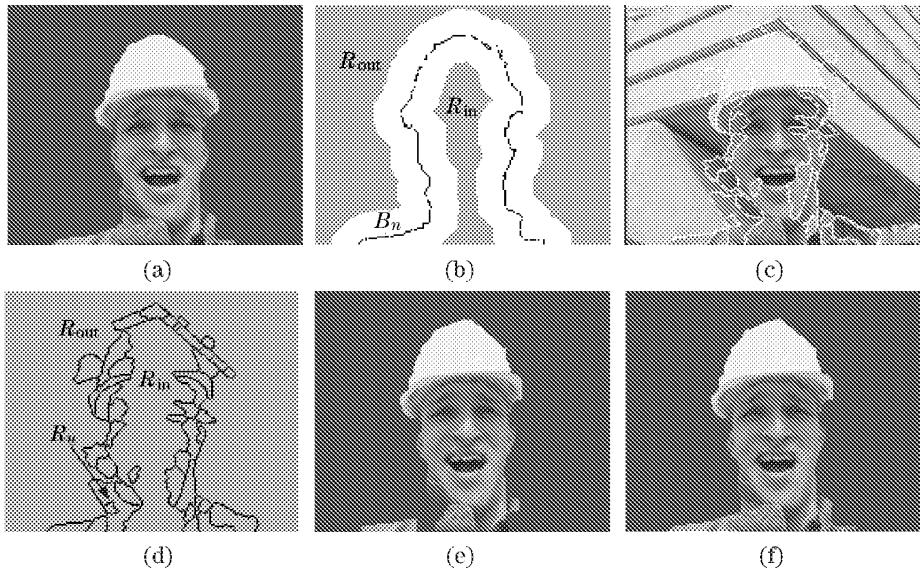

Fig. 1. A pictorial description of the proposed bidirectional projection approach.

vectors for all the pixels in ct_{n-1}

$$T_{n-1}^{u} = \sum_{(x,y)\in ct_{n-1}} u(x,y) \Big/ \sum_{(x,y)\in ct_{n-1}} 1,$$
$$T_{n-1}^{v} = \sum_{(x,y)\in ct_{n-1}} v(x,y) \Big/ \sum_{(x,y)\in ct_{n-1}} 1. \quad (3)$$

This vector $\left(T_{n-1}^{u}, T_{n-1}^{v}\right)$ reflects a global translation movement of the video object if an apparent translation exists, which will be used in backward projection.

Spatial segmentation: The watershed segmentation algorithm[7] is exploited to partition the current frame I_n into a set of regions (see Fig. 1(c)). In fact, only the band area B_n needs to be partitioned into some regions, which need to be backward projected to determine whether they belong to the current video object or not. The area R_{in} inside B_n definitely belongs to the current video object, while the area R_{out} outside B_n belongs to the background (see Fig. 1(b)).

The gradient image g of the color image f in YUV space is estimated by the delicate method proposed by Di Zenzo[8]. However, the noise in the gradient image causes the inevitable problem of over-segmentation when directly applying watershed segmentation. In order to obtain a moderate segmentation result, we propose a simplification step to remove insignificant local minima in the gradient image. Firstly, g is dilated with a 3×3 cross-shaped structuring element E, and the dilated image is elevated by a height h to get the marker image, $g_{\text{m}} = (g \oplus E) + h$. Then, the reconstruction of g from g_{m} by geodesic erosion[9] is performed to obtain the simplified gradient image, $g_{\text{s}} = \varphi^{(\text{rec})}(g_{\text{m}}, g)$. Now we can apply the watershed segmentation algorithm to the image g_{s} to obtain a reasonable partition of the original image f. The only one parameter h is set to 2 by the experiment, a smaller value that leads to a reasonably finer partition (see Fig. 1(c)).

Backward projection: The objective of backward projection is to find the true contour ct_n of the current video object vo_n in B_n. The segmented regions, excluding the regions R_{in} and R_{out} (see Fig. 1(d)), are backward projected to determine their classifications. For each region R_i, the backward motion vector (u_i, v_i) is estimated to minimize the following prediction error,

$$e_i = \min_{u_i,v_i} \sum_{(x,y)\in R_i} \|I_n(x,y) - I_{n-1}(x+u_i, y+v_i)\|. \quad (4)$$

A small search range $\left[-T_{n-1}^{u} - 3, -T_{n-1}^{v} + 3\right]$ is set for (u_i, v_i), because the vector $\left(T_{n-1}^{u}, T_{n-1}^{v}\right)$ already reflects a possible apparent translation of the video object. The backward projected region R_i' in the previous frame I_{n-1} can be denoted by

$$R_i' = \bigcup_{(x,y)\in R_i} (x+u_i, y+v_i). \quad (5)$$

The classification of R_i can be determined from the intersecting area of R_i' and vo_{n-1}. However, region classification[4] is not suitable for such a segmented region that overlaps the video object and the background at the same time. If such a region (see the region R_u in Fig. 1(d)) is classified into the video object, a peninsula appears on the video object; otherwise a gap appears (see Fig. 1(e)). In order to deal with such a problem, we propose a robust approach to improve region classification[4].

The ratio of the intersecting area of R_i' and vo_{n-1} to the area of R_i' is defined as

$$\theta_i = \frac{A[R_i' \cap vo_{n-1}]}{A[R_i']}, \quad (6)$$

where $A[\cdot]$ denotes the area operation. The value of θ_i indicates three different types of region, that is, a fairly higher value that shows the region R_i belongs to the video object, a fairly lower value that shows R_i is a part of the

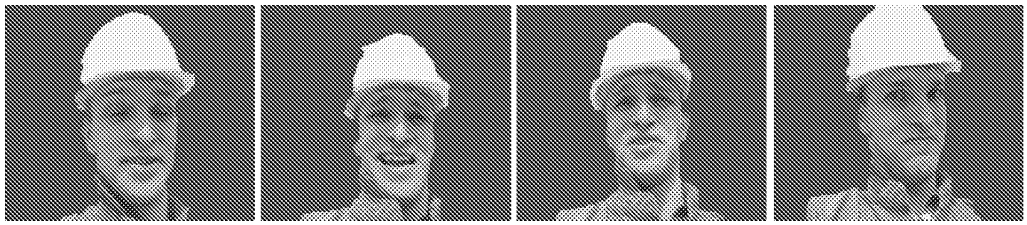

Fig. 2. Experimental results for the sequence "Foreman" (frames 1, 20, 60, and 100).

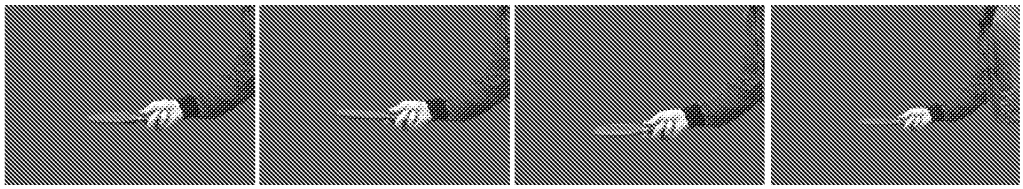

Fig. 3. Experimental results for the sequence "Table Tennis" (frames 1, 15, 25, and 40).

Table 1. Average Processing Time for a Frame of the Two Sequences (ms)

Test Sequence	Proposed Bidirectional Projection Approach				Backward Projection Approach[4]		
	Forward Projection	Segmentation	Backward Projection	Total	Segmentation	Backward Projection	Total
Foreman (176×144)	43	42	68	153	58	275	333
Table Tennis (352×240)	95	131	181	407	154	981	1135

background, and a moderate value that shows R_i may overlap the video object and the background at the same time. For the first and the second cases, the whole region is assigned to the video object or the background based on the following criterion,

$$\begin{cases} R_i \in vo_n, & \text{if } \theta_i > T_h \\ R_i \notin vo_n, & \text{if } \theta_i < T_l \end{cases} . \tag{7}$$

For the third case, $T_l \le \theta_i \le T_h$, pixel classification in the region R_i is performed using the criterion of

$$\begin{cases} (x,y) \in vo_n, & \text{if } (x+u_i, y+v_i) \in vo_{n-1} \\ (x,y) \notin vo_n, & \text{if } (x+u_i, y+v_i) \notin vo_{n-1} \end{cases} . \tag{8}$$

The two parameters T_l and T_h are set to 0.15 and 0.65 in our experiments, and these two criteria demonstrate a reliable tracking performance (see Fig. 1(f)). Since the boundaries (watershed lines) are not classified in backward projection, a closing morphological operation is performed to fill the watershed lines in the video object.

We use several MPEG-4 test sequences to test the proposed approach, and the experimental results for two of them are shown in Figs. 2 and 3. The user-defined video object is shown in the first image of each figure, and the tracking results are shown in the latter three images respectively. As shown in the figures, the accurate and reliable video objects are obtained for the two sequences with different levels of spatial detail and movement.

These experiments are performed on a low-end AMD Athlon XP1800 (1.53 GHz) PC. The average processing time per frame using our bidirectional projection approach and the backward projection approach[4] is shown in Table 1. The same values are set to the related parameters in both approaches. Compared with the backward projection approach[4], our approach needs to consume some time on forward projection, but sharply reduce the time on backward projection, and spatial segmentation to some extent. For the two sequences, the total processing time of our approach is 46% and 36% of the backward projection approach[4], which demonstrates the improved segmentation efficiency of our approach.

In this work, a bidirectional projection approach is proposed for video object tracking, which extends backward projection with the combination of forward projection. The proposed tracking approach produces more reliable video objects for different types of video sequences, and improves the segmentation efficiency by a factor of two.

Z. Liu's e-mail address is liuzhi@sjtu.edu.cn.

References

1. M. Kim, J. G. Jeon, J. S. Kwak, M. H. Lee, and C. Ahn, Image and Vision Computing **19,** 245 (2001).
2. C. Gu and M. C. Lee, IEEE Trans. Circuits Systems for Video Technology **8,** 572 (1998).
3. J. Lim, H. K. Cho, and J. Beom Ra, in *Proc. IEEE ICIP'2000* 339 (2000).
4. C. Gu and M. C. Lee, in *Proc. IEEE ICIP'98* 643 (1998).
5. D. Gatica-Perez, M. T. Sun, and C. Gu, in *Proc. IEEE ICIP'99* 145 (1999).
6. A. M. Tekalp, *Digital Video Processing* (Tsinghua University Press, Beijing, 1998) p. 104.
7. L. Vincent and P. Soille, IEEE Trans. Pattern Analysis Machine Intelligence **13,** 583 (1991).
8. S. Di Zenzo, Computer Vision, Graphics, and Image Processing **33,** 116 (1986).
9. L. Vincent, IEEE Trans. Image Processing **2,** 176 (1993).