Active Video Object Extraction

Ye Lu Ze-Nian Li

School of Computing Science
Simon Fraser University
Burnaby, B.C., V5A 1S6

Abstract

This paper addresses the problem of intelligently extracting objects from videos. Our method assumes that the video making process is purposive and attempts to extract those objects that the original authors of the videos intended to capture. We accomplish this by analyzing three types of actions of the author (saccadic movements, smooth pursuits, and multi-baseline pursuits) in an active vision framework using dense 2D disparity vectors computed from successive frames of the video. We demonstrate the effectiveness of our algorithm using real video sequences.

1 Introduction

An important step in automatic video content analysis is the extraction of the *objects of interest* from videos. This task can be simplified if we know the kind of objects that we are trying to extract from the videos. For example, if we wish to detect and extract people from videos, then the approaches described in [6] can be applied to efficiently achieve this purpose. Another example in which prior knowledge of the object can tremendously aid in the extraction process is given in [5] which reliably detects and localizes the passenger cars in the input videos.

In the absence of knowledge about target objects, traditional algorithms take advantage of the motion of rigid objects to identify and extract them. In this approach, each frame in the video is first decomposed into regions according to coherent image features such as color and texture. Then, subsequent frames of the video are used to estimate the motion parameters of these regions. Regions with similar motion are then merged and considered to be a single object. This approach is exemplified by the works of Irani et al. [2, 3] and Altunbasak et al. [1]. It is worth noting that 2D parametric transformations are used to approximate the 3D motion of objects on the image plane. This approximation is only valid when the difference in depth caused by the motion is small relative to the distances of objects from the camera. Three types of motion models are often used by these algorithms: translation, affine motion, and moving planar surface.

In this paper, we attack the object extraction problem from a different perspective. Specifically, we model the author of the video as an active observer in a dynamic environment. The actions taken by this active observer is assumed to be purposive. Thus, the resulting video is not a series of random shots, but a combination of well intended camera movements capturing a set of objects of interest. Our goal is to extract only the objects of interest from a video and leave behind other objects that just happen to be in the shot. We model three types of actions performed by the active observer: saccadic movement, smooth pursuit, and multi-baseline pursuit. Each of these three types of actions result in different characteristics in the disparity maps computed from consecutive frames of the video as well as give important hints as to what objects the active observer is interested in. Therefore, we analyze the disparity maps to determine the objects of interest.

Because estimating accurate disparity maps from uncalibrated cameras under general motion is vitally important to our object extraction process, a major part of our algorithm is devoted to perform this estimation. We will describe a novel uncalibrated cooperative stereo algorithm in Section 2. The active object extraction process will be discussed in Section 3, followed by experimental results in Section 4. The conclusion will be given in Section 5.

2 Computing Dense Disparity Under General Motion

Performing stereo matching on consecutive frames of a prerecorded video is much more difficult than traditional two-framed stereo matching. Since the camera calibration parameters are usually not known in advance, image rectification based on only the estimated fundamental matrix may be very inaccurate. This problem is worsened by the free motion of the camera which can place the epipoles close or even within the image, making most rectification algorithms impractical [4]. To effectively avoid these problems

associated with image rectification, we have developed our new cooperative stereo algorithm to work directly on non-rectified images.

Our cooperative stereo algorithm uses a four dimensional matching score volume parameterized by (x, y, d_x, d_y) to compute dense stereo matchings for non-rectified images obtained by cameras under general motion. The value associated with each element represents the matching score of the disparity d_x and d_y at (x, y). Thus, given two stereo images, $I_1(x, y)$ and $I_2(x, y)$, we wish to find the x and y disparities such that the two images are matched as closely as possible.

$$I_1(x, y) \approx I_2(x - d_x, y - d_y) \quad . \tag{1}$$

Let $L_n(x, y, d_x, d_y)$ be the matching score computed at the nth iteration. The initial values $L_0(x, y, d_x, d_y)$ are calculated using local similarity measures such as normalized cross correlation on windows centered at (x, y) in the first image and $(x - d_x, y - d_y)$ in the second image. For simplicity, we write $L_n(x, y, d_x, d_y)$ using vector notation as $L_n(\mathbf{x}, \mathbf{d_x})$.

2.1 Initial Matching and Candidate selection

We exploit both the color consistency and the epipolar constraint to identify an initial set of matching candidates. We first robustly estimate the fundamental matrix $\mathbf{F}$ between the two input images using tracked sparse features. Then, for any pixel location $\mathbf{x}$ in the first image, the matrix $\mathbf{F}$ maps it to an epipolar line $\mathbf{l} = \mathbf{F}\mathbf{x}$ in the second image. With this estimate of epipolar geometry, the correct match for $\mathbf{x}$ is within a narrow band around the epipolar line $\mathbf{l}$. We refer to this band as the *epipolar band*. The width of the epipolar band required to include the correct match depends on the accuracy of the estimated fundamental matrix. An indicator of the quality of the estimated fundamental matrix can be computed as the average distance of inlying features to their respective epipolar lines. We set the width of the epipolar band on either side of the epipolar line to be three times that distance. The initial candidate list consists of candidate matches having positive normalized correlation scores on the epipolar band.

2.2 Matching Score Adjustments

A major advantage of cooperative algorithms is their ability to perform only local computation and yet have behaviors similar to that of global optimization algorithms. The mechanism making this possible is the excitation and inhibition of candidates through local support.

The local support area used in our algorithm is similar to that of the fixed 3D box-shaped local support proposed by Zitnick and Kanade [7]. Because we represent disparity as 2D vectors, our local support areas consists of only sparse elements in a 4D box. Let Φ be set containing the local support elements of $\mathbf{x}$, and let $S_n(\mathbf{x}, \mathbf{d_x})$ be the amount of local support for $\mathbf{x}$ with disparity vector $\mathbf{d_x}$. Then, we calculate the amount of local support as

$$S_n(\mathbf{x}, \mathbf{d_x}) = \sum_{(\mathbf{x}', \mathbf{d}'_\mathbf{x}) \in \Phi} L_n(\mathbf{x}', \mathbf{d}'_\mathbf{x}) \quad , \tag{2}$$

such that $(\mathbf{x}', \mathbf{d}'_\mathbf{x}) \in \Phi$ if and only if $\|\mathbf{d}'_\mathbf{x} - \mathbf{d_x}\| < r$ for some support radius r.

Another well known characteristic of cooperative algorithms is the mutual inhibition of candidate matches. Generally, the correct match will receive higher match values and thus causing the matching values of false candidates to decrease. Let Ψ denote the set of inhibition elements. There are two types of elements in Ψ: those that project to the pixel $\mathbf{x}$ in the first image; and those that project to $\mathbf{x} - \mathbf{d_x}$ in the second image.

The first type of inhibition elements can be easily located, but obtaining the second type of inhibition elements is difficult. Since our parameterized disparity is 2D, performing a full search for these inhibition elements would require an extremely large amount of computation. Therefore, we only perform inhibition on the first type of inhibition elements and delay inhibition of the second type to the final optimization step of the algorithm. Let Ψ' be the set containing only the first type of inhibition elements. For computational simplicity, we adopt the inhibition function used in [7]:

$$R_n(\mathbf{x}, \mathbf{d_x}) = \left(\frac{S_n(\mathbf{x}, \mathbf{d_x})}{\sum_{(\mathbf{x}'', \mathbf{d}''_\mathbf{x}) \in \Psi'} S_n(\mathbf{x}'', \mathbf{d}''_\mathbf{x})} \right)^\alpha \quad , \tag{3}$$

where α is a constant controlling the amount of inhibition per iteration. To ensure a single element within Ψ will converge to 1, α must be greater than 1.

To limit the amount of over-smoothing, we use the initial matching score $L_0(\mathbf{x}, \mathbf{d_x})$ computed using normalized cross correlation to further restrict the current match values. Putting this restriction together with the inhibition function, we have the update equation:

$$L_{n+1}(\mathbf{x}, \mathbf{d_x}) = L_0(\mathbf{x}, \mathbf{d_x}) \cdot R_n(\mathbf{x}, \mathbf{d_x}) \quad . \tag{4}$$

2.3 Final Disparity Estimates and Occlusion Detection

After the matching scores have converged within the matching score volume, we need to generate final disparity assignments. In our algorithm, we jointly formulate occlusion detection and final disparity assignment as a maximum weight bipartite matching problem on a graph constructed

using matching scores computed during the iterative updating process. Let $G = (V_1, V_2, E)$ be a bipartite graph with partite sets V_1 and V_2. We identify the nodes in V_1 with pixels in the first image and nodes in V_2 with pixels in the second image. There is an edge between $v \in V_1$ and $v' \in V_2$ if the pixel represented by v' is a matching candidate of the pixel represented by v. The weight on the edge is the respective converged matching scores in the matching score volume. The result of this maximum weighted bipartite matching is a graph $G' = (V_1, V_2, E')$ of maximum weight such that $E' \subseteq E$ and no two edges in E' share a common node. This effectively enforces strict two-way uniqueness. We label pixels that do not have a match as occluded pixels.

3 Active Object Extraction

We share many common grounds with the active vision framework. In active vision, objects of interest are being actively acquired by cameras mounted on computer controlled platforms to simulate the visual acquisition process of humans; the output is a piece of video. In our task, we assume that an active observer already acquired the video and analyze the motion patterns. As a result, we extract the objects of interest from the video.

The three types of actions performed by the active observer can be readily identified from the estimated disparity maps. In particular, the magnitudes of the 2D disparity vectors reveal much of the needed information. We perform foveated analysis of the 2D disparity magnitude map. If the input video has dimension $N \times M$, then we define the foveal region as the rectangle of size $\frac{N}{4} \times \frac{M}{4}$ centered on the image.

The saccadic movements consist of rapid motions of the camera aimed at focusing an object on the fovea. The resulting magnitudes of the 2D disparity vectors are usually large. In addition, the closer an object is to the camera, the larger the magnitude of the disparity. This is particularly true for the foveal region. Thus, if the average disparity magnitude in the fovea is greater than some threshold, then we can classify the two consecutive frames in which the disparity map is derived from as part of a saccadic movement. Smooth pursuits are accurate movements of the camera to keep the target on the fovea. By examining the magnitudes of the disparity vectors, we see that disparity magnitudes of the object of interest are very small since the camera compensates for the object's motion and tries to keep it at the same location on the fovea. Other areas in the image usually contain large disparity magnitudes. The multi-baseline pursuits are by far the most complex movements. It generally involves a combination of camera movements such as horizontal and vertical displacements (i.e., tracking and booming) coupled with necessary panning and/or tilting in order to keep the object of interest in view. Similar to smooth pursuit movements, the multi-baseline pursuits result in small disparity magnitudes within the region containing the object of interest and large disparity magnitudes elsewhere.

From the above observations, the object of interest can be readily detected in a video. We summarize our object extraction algorithm as follows:

1. Compute the dense 2D disparity maps for consecutive frames in the video using the algorithm described in Section 2.
2. Examine magnitude of the disparity vectors in the foveal regions of each disparity map.
3. If the average disparity magnitude in the fovea is less than a threshold τ, then the current action must be either smooth pursuit or multi-baseline pursuit. Grow a region from the fovea containing only pixels having disparity magnitudes less than τ.
4. Fill holes in the region computed in the previous step and output the resulting object.

4 Experimental Results

We have implemented the proposed active object extraction algorithm using C++ on a PC platform. We have tested our algorithm on a number of video sequences taken by a freely moving camera in the hands of a human user. Because of space constraints, we only present the result on two such sequences.

Since the threshold τ depends on the speed of the camera motion, we set it to be 20% of the maximum disparity magnitude in each frame. Fig. 1 shows the camera performing smooth pursuit of a moving toy truck. From this figure, we can clearly see the disparity magnitudes within the region containing the toy truck are very small. As a result, the truck is correctly extracted from both disparity maps. Fig. 2 contains a longer video which pans horizontally to the left until the monk figurine is centered on the fovea. Then, the camera rotates around the monk figurine in the last three frames. This video has several interesting characteristics. First, a number of objects are present throughout the video. It therefore tests the algorithms ability to extract only the object of interest. Second, a combination of saccadic and multi-baseline pursuit are used to make this video. This tests the algorithm's ability to distinguish between these two types of actions. In both cases, our algorithm performs correctly by extracting only the monk figurine from the entire video sequence.

5 Conclusion

In this paper, we have presented a novel active object extraction algorithm. This algorithm extracts only the object of in-

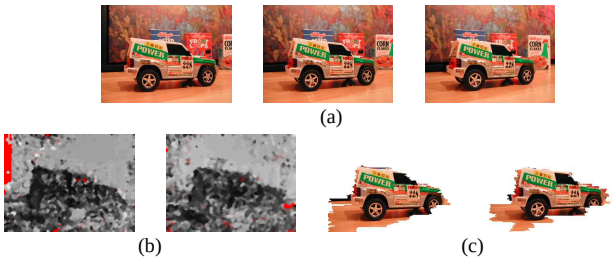

Figure 1: Object extracted from smooth pursuit. (a) Input frames, (b) Computed disparity magnitudes from consecutive frames (red pixels indicate occlusion), and (c) the extract object of interest.

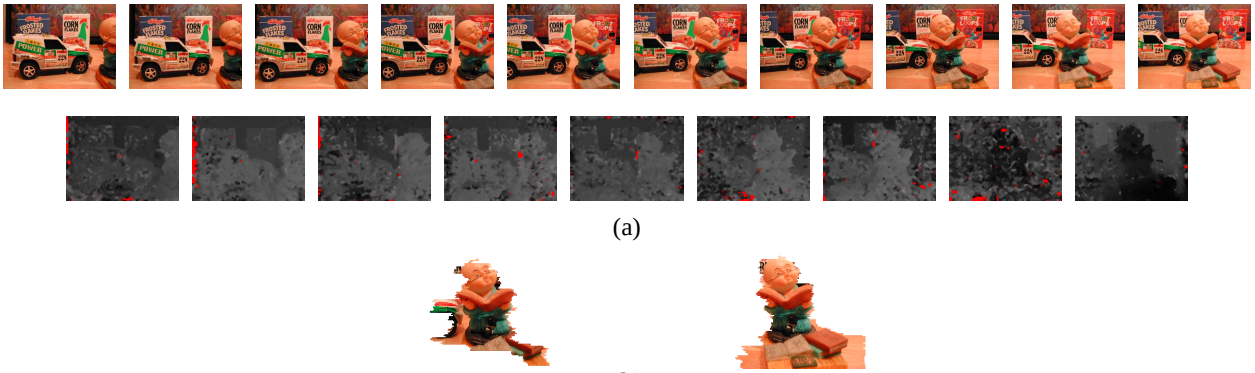

Figure 2: Object extracted from a combination of saccadic and multi-baseline pursuit. (a) The input frames and the computed disparity magnitudes (red pixels indicate occlusion), and (b) the extracted object of interest.

terest from a given video by performing analysis of motion patterns on the foveal region of the 2D disparity magnitudes based on the analysis of three types of camera movements by the author. In particular, the multi-baseline pursuit allows uncalibrated camera under unrestricted motion. This is significantly different from previous approaches such as multi-baseline stereo in which camera displacements are regular and controlled. Because accurate estimation of disparity is an important factor for the success of this algorithm, we have also developed a new uncalibrated cooperative stereo algorithm that can compute 2D disparity estimates directly on non-rectified images.

We are currently working on improving this object extraction algorithm by considering combining low level image features such as edge maps and textures with disparity magnitudes to increase the accuracy of object boundaries. In addition, we are seeking ways to incorporate different views of the object of the interest into a single unified representation.

References

[1] Y. Altunbasak, P. E. Eren, and A. M. Tekalp. Region based parametric motion segmentation using color information. *Graphical Models and Image Processing*, 60(1):13–23, 1998.

[2] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proceedings of the European Conference on Computer Vision*, pages 282–287, 1992.

[3] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[4] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *Proceedings of the International Conference on Computer Vision*, 1999.

[5] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[6] M. H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[7] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, 2000.