BACKGROUND RECOVERY FROM VIDEO SEQUENCES USING MOTION PARAMETERS

Srenivas Varadarajan[*], *Lina J. Karam*[*], *and Dinei Florencio*[+]
(*) Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706
(+) Microsoft Research, One Microsoft Way, Redmond, WA 98052
svarada2@asu.edu , karam@asu.edu, dinei@microsoft.com

ABSTRACT

This paper presents a novel scheme for extracting a still background occluded by a number of foreground objects, moving in different directions and velocities in a video sequence, such that every background pixel is exposed in at least one of the frames. Each identified foreground object is decomposed into blocks. The proposed scheme is able to efficiently estimate, for each foreground block, a source frame from which the occluded background pixels can be extracted. The pixels of the identified source frames are used to populate the co-located occluded pixels in the initial frame. The efficacy and the simplicity of the algorithm lie in its capacity to recover the background directly from the estimated source frames instead of performing a foreground-background classification for every frame. The proposed algorithm is robust to variations in lighting and is effective in removing both rigid and deformable foreground objects. Simulation results are presented to illustrate the performance of the proposed scheme.

Index Terms— Occlusion Removal, Object Removal, Background Extraction, Motion, Video

1. INTRODUCTION

Background extraction is at the heart of many object tracking, surveillance, content-based retrieval and object-based video coding applications. Occlusions can be predominantly of two types. Firstly, a background can be occluded by moving foreground objects, and secondly a wider continuous background may be occluded by a static foreground object like a statue or an information board. Only the former types of occlusions are addressed in this paper. Background extraction algorithms typically use techniques like image inpainting [1] on a single image, or object-tracking techniques [2,3] on a sequence of images or a combination of both [4].

Several approaches have been proposed in the past for locating the foreground regions in video sequences [5,6]. Though these techniques perform well in the basic foreground-background classification on a per-image basis, they do not recover the occluded background. Inpainting techniques [1] have been used to fill in the occluded areas by suitably interpolating the neighboring pixels or using texture synthesis [7] to remove foreground objects from still images. These techniques can result in a degraded visual quality when applied to video sequences because, instead of extracting the real value of the exposed pixel, they try to form an estimate. In [8], a background recovery from a set of images that share an identical background is proposed. The method of [8] can only be applied to a video sequence on a frame-by-frame basis as it does not exploit the temporal correlation that is present in the video sequence. When applied to a video sequence, the method of [8] would consider every frame as a potential source for un-occluding every occluded region and would lead to excessive computations. Segmentation-based approaches have also been proposed [9, 10]. However, the method presented in [9] is restricted to rigid moving objects, and the method of [10] relies on differential texture regions to refine the segmentation. A set of motion-based approaches [2, 3, 11, 12] have also been proposed, many of which do simultaneously foreground tracking and background updating. But these methods are computationally very expensive as the foreground-background classification is done for every frame.

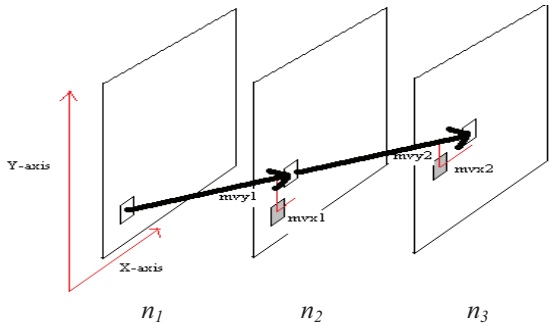


Fig. 1: Tracking of occluding foreground blocks.

This paper presents a video-based background-extraction scheme. The proposed scheme estimates a source frame for every foreground block, from which the occluded background pixels can be extracted and is, hence, suitable for real-time background recovery.

This paper is organized as follows. The proposed video-based background extraction algorithm is described in Section 2. Performance results are presented in Section 3 followed by a conclusion in Section 4.

2. PROPOSED ALGORITHM

Using an image sequence, it is possible to extract the background pixels that are occluded by the moving foreground if they are exposed in at least one other frame. The motion vectors of each moving object can be used to indicate how fast the corresponding region can be recovered. As shown in Fig 1, if the horizontal motion vectors are mvx1 and mvx2 for a block in two consecutive frames n_1 and n_2, respectively, then the background of the block in the frame n_1 is exposed at a rate of (mvx1 + mvx2)/2 pixels per frame assuming, locally, a linear motion model.

The proposed video-based background extraction algorithm is based on this principle and tries to extrapolate the motion of each of 8x8 blocks in an occluding foreground component along the horizontal (*X*) and vertical (*Y*) directions, after an initial tracking step. It assumes that rotation and scaling are very limited in the small interval over which the background is recovered and that static scaling would not fully expose the background. Hence, a generic affine model is not necessary.

A block diagram of the proposed algorithm is shown in Fig. 2. The algorithm first performs a pixel-based foreground-background classification using a small number of frames. The foreground pixels are then clustered into separate occluding components (OCs), corresponding each to a moving foreground object. Each identified OC is encapsulated in a rectangular box and is divided into 8x8 blocks. For each of these foreground 8x8 blocks, the index of the nearest source frame in which the pixels that were occluded by the considered block just become uncovered, is determined using the block-wise computed motion vectors. The occluded pixels are then replaced using the uncovered background pixels from the determined source frames. Since every foreground block can be independently tracked and the corresponding region recovered, the algorithm is quite effective in recovering occlusions due to deformable foreground objects. Details about the foreground-background classification, occluding components formation, and the source frames selection are presented Sections 2.1 to 2.3, respectively.

2.1 Foreground-Background Classification

The algorithm employs the simple approach of "change-detection" based on the co-located pixel values of *N* consecutive frames, $F_1, F_2, \ldots, F_N$, to segment the moving foreground from the background (*N=5* was used in our implementation). In order to account for fluctuations and changes due to lighting conditions, a modified approach based on hysteresis classification is proposed.

The considered *N* consecutive frames, $F_1, F_2, \ldots, F_N$, are first lowpass-filtered to remove any noise. The goal here is to classify the pixels in the first frame F_1 as foreground or background. This is achieved by first classifying the pixels into three classes corresponding to Strong Foreground (SF), Weak Foreground (WF), and Background (B) as follows:

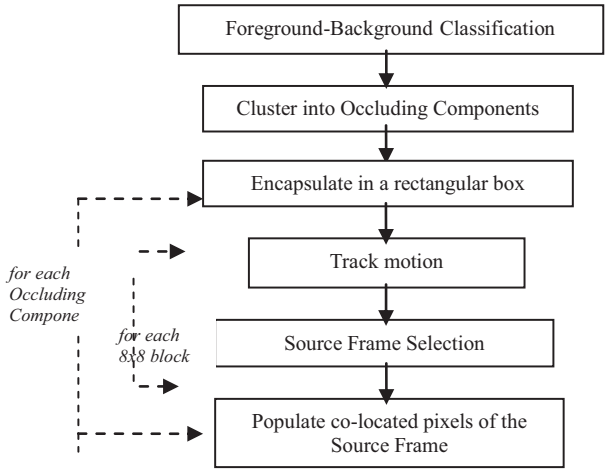


Fig. 2: Block diagram of the proposed algorithm.

$$C_{x,y,1} = \begin{cases} SF, & if \ \left\| p_{x,y,1} - p_{x,y,n} \right\|_{L_1} > t_1 \ for \ any \ n = 2, \ldots, N \\ WF, & if \ t_2 < \left\| p_{x,y,1} - p_{x,y,n} \right\|_{L_1} < t_1 \ for \ any \ n = 2, \ldots, N \\ B, & else \end{cases} \quad (1)$$

where $p_{x,y,n} = (Y_{x,y,n}, Cr_{x,y,n}, Cb_{x,y,n})$ corresponds to the luminance and chrominance values of the pixel at position (x,y) in frame F_n, t_1 and t_2 ($t_1 > t_2$) are positive integers corresponding to a high threshold and a low threshold value, respectively. In our implementation, t_1 = 20 and t_2=3 were used.

After performing an initial classification based on (1), misclassified pixels (outliers) are detected as follows. If none of the 8-connected neighbors surrounding a SF pixel is a SF pixel, the SF pixel is considered an outlier and is changed to a B pixel. Similarly, if none of the 8-connected neighbors surrounding a B pixel is a B pixel, the B pixel is considered an outlier and is changed to a SF pixel. Starting from every SF pixel, the neighboring pixels are scanned along the left, right, top and bottom directions and any WF pixel connected to a SF pixel, is changed to a SF pixel in a recursive way. So, the SF pixels propagate over the connected WF pixels. At the end of the scan, all SF pixels are classified as foreground and the remaining as background.

2.2 Occluding Components Formation

Once the foreground pixels are determined, they are clustered into Occluding Components (OCs), with each OC corresponding to a single foreground object. This clustering is performed based on the connectivity of the foreground pixels. Each set of connected foreground pixels forms an OC. The resulting OCs are arbitrary-shaped. In order to simplify the processing, each identified OC is encapsulated into a rectangular box. In the remainder of this paper, an OC refers to an occluding component that is encapsulated in a rectangular box.

2.3 Source Frame Selection

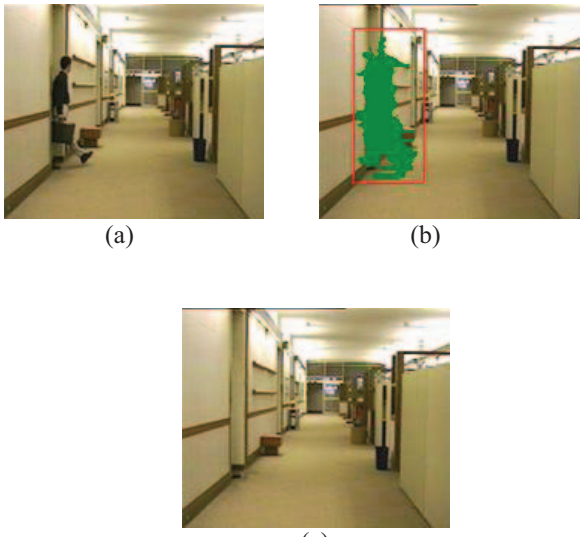
Since each OC may move along different directions and at different speeds, it may happen that a background region, which gets exposed in an intermediate frame, could get occluded by another OC in a later frame. Also, in order to account for deformable regions, each OC is divided into 8x8 blocks and, for each 8x8 occluding block, the index of the nearest frame, from which the corresponding background block can be recovered, is computed. To this end, each 8x8 foreground block is tracked across P frames (P=11 was used in our implementation), $F_1, F_2, \ldots, F_P$, by computing $(P-1)/2$ forward motion vectors MV_i, $i=1,3,5,\ldots,P-2$. The best match for a block in frame F_i is found in the reference frame F_{i+2} using a fast block-wise motion estimation method (e.g., three-step search). The reference frame is 2 frame-delays with respect to the considered current frame, which reduces the computations and accounts for sub-pel motion. While MV_1 was computed by finding in F_3 the block that best matches the original considered block in F_1, the remaining motion vectors, MV_i ($i=3,\ldots,P-2$) were computed by finding in F_{i+2} the block that best matches the corresponding motion-compensated block in F_i. An average motion vector MV_A is finally computed by averaging the obtained $(P-1)/2$ motion vectors MV_i ($i=1,3\ldots,P-2$).

To prevent erroneous zero motion vectors due to texture-less, smooth foreground blocks, a weighted average of the MVs of the neighboring blocks is computed for every zero motion-vectored block that has neighboring non-zero motion-vectored blocks on either side. Let MV_x and MV_y denote, respectively, the horizontal and vertical components of a motion vector MV. Consider a block with a zero motion vector MV (MV_x =0 and MV_y =0). If that block is located at distances d_L and d_R from its nearest non-zero motion-vectored blocks in the left and right directions, respectively, and at distances d_T and d_D from its nearest non-zero motion-vectored blocks in the top and down directions, respectively, then the zero-valued MV_x and MV_y are changed to be:

$$MV_x = (d_L MV_{x,L} + d_R MV_{x,R})/(d_L + d_R)$$
$$MV_y = (d_T MV_{y,T} + d_D MV_{y,D})/(d_T + d_D) \quad (2)$$

where $MV_{x,L}$ and $MV_{x,R}$ are, respectively, the horizontal components of the nearest left and right blocks with non-zero motion vectors, and $MV_{y,T}$ and $MV_{y,D}$ are, respectively, the vertical components of the nearest top and down blocks with non-zero motion vectors.

Let (b_x,b_y) be the position of the top left corner of the considered foreground block in the initial frame F_1. Let (TL_x, TL_y) and (BR_x, BR_y) denote, respectively, the position in F_1 of the top-left and bottom-left corners of the OC to which the considered block belongs. Using the computed average motion vector MV_A=($MV_{A,x}$, $MV_{A,y}$) for the considered foreground block, the index I_S (relative to the initial frame F_1) of the nearest source frame from which the background pixels that are occluded by the considered block can be recovered, is computed as follows:



(a) (b) (c)

Fig. 3: Simulation Results for the Hall Monitor video sequence. (a) Initial occluded frame. (b) Detected occluded component with rectangular bounding box. (c) Recovered background.

$$I_S = \min\ (I_{S,x}, I_{S,y}) \quad (3)$$

where

$$I_{S,x} = (D_x + 8)/MV_{A,x}$$
$$I_{S,y} = (D_y + 8)/MV_{A,y} \quad (4)$$

In (4), D_x and D_y are given by:

$$D_x = \begin{cases} b_x - TL_x, & if\ MV_{A,x} > 0 \\ BR_x - b_x, & if\ MV_{A,x} < 0 \end{cases}$$
$$D_y = \begin{cases} b_y - TL_y, & if\ MV_{A,y} > 0 \\ BR_y - b_y, & if\ MV_{A,y} < 0 \end{cases} \quad (5)$$

For each pixel of the considered 8x8 foreground block, the co-located pixel in the corresponding source frame with index I_S, is fetched and populated to recover the background.

3. SIMULATION RESULTS

Simulation results for the "Hall Monitor" video sequence are shown in Fig. 3. Even though the camera is fixed in this sequence, the background pixel intensities fluctuate a lot due to change in lighting conditions and shadow effects. The proposed foreground-background classification alleviates this problem to some extent in locating the foreground. Figs. 3(a), 3(b), and 3(c) show, respectively, the initial frame (27th frame of Hall Monitor), the located foreground pixels (green) with the encapsulating rectangular box, and the recovered background using the proposed algorithm. As shown in Fig 3 (c), most of the background was accurately recovered by the proposed scheme. The black spot seen in the background is due to the inability of the algorithm to spot some foreground pixels accurately due to their stationarity in the initial frames. This can be easily corrected

by using more sophisticated foreground classification algorithms.

4. CONCLUSION

A new approach for recovering a still background from a video sequence is proposed in this paper. The proposed block-based tracking helps in removing the deformable parts of the foreground. The proposed foreground-background classification takes into account lighting changes in foreground classification. The weighted prediction for the zero motion vectors helps in tracking "texture-less" foreground regions. The proposed scheme can successfully extract the background from a video sequence when one or more foreground objects are present. Future work will include extracting the background from multi-view videos.

5. REFERENCES

[1] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 721 -728, June 2003.

[2] A. Kokaram, B. Collis and S. Robinson, "A Bayesian framework for recursive object removal in movie post-production," *IEEE International Conf. on Image Processing*, vol. 1, pp. 937-40, Sept. 2003.

[3] O. Rostamianfar, F. Janabi-Sharifi, and I. Hassanzadeh, "Visual tracking system for dense traffic intersections," *Canadian Conference on Electrical and Computer Engineering,* pp. 2000 – 2004, May 2006.

[4] K.A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Transactions on Image Processing*, vol. 16, issue 2, pp. 545 – 553, Feb. 2007.

[5] S.-C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Proceedings of the SPIE*, vol. 5308, pp. 881-892, 2004.

[6]A.-N. Lai, H. Yoon, and G. Lee, "Robust background extraction scheme using histogram-wise for real-time tracking in urban traffic video," *IEEE Conf. on Computer and Information Technology*, pp. 845-850, July 2008.

[7] H.-J. Hsu, J.-F. Wang, and S.-C. Liao, "A hybrid algorithm with artifact detection mechanism for region filling after object removal from a digital photograph," *IEEE Transactions on Image Processing*, vol. 16, issue 6, pp. 1611–1622, June 2007.

[8] C. Herley, "Automatic occlusion removal from minimum number of images," *IEEE International Conf. on Image Processing*, vol 2, pp. 1046-1049, Sept. 2005.

[9] P.M.Q. Aguiar and J.M.F. Moura, "Joint segmentation of moving object and estimation of background in low-light video using relaxation", *IEEE International Conf. on Image Processing*, vol. 5, pp. 53-56, Sept 2007.

[10] Y. Lu, W. Ga, and F. Wu, "Automatic video segmentation using a novel background model," *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 807–810, May 2002.

[11] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," *7th IEEE workshop on Application of Computer Vision*, vol. 1, pp. 516-521, Jan 2005.

[12] F.-F. Meng, Z.-S. Qu, and Q.-S. Zeng, and L. Li, "Traffic object tracking based on increased-step motion history image," *IEEE Int. Conference on Automation and Logistics*, pp. 345 – 349, Aug 2007.