Background mosaic reconstruction

Anton Zachesov
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
azachesov@graphics.cs.msu.ru

Dmitry Vatolin
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
dmitry@graphics.cs.msu.ru

Maxim Smirnov
YUVsoft Corp.
ms@yuvsoft.com

Abstract

In this paper, we present a new fast approach for generation of background panorama from a video sequence based on motion estimation algorithm. The proposed method provides correct matching of background areas in video sequences with complex camera movement. It also performs rough preliminary segmentation which allows removing foreground objects from mosaic on inserting stage. Result sprites have appropriate quality and level of detail to use them in other applications.

Keywords: *panorama, motion-based segmentation, background mosaicing, background restoration.*

1. INTRODUCTION

Nowadays video processing algorithms, for example multiview generation or diminished reality construction, require high-quality background restoration technique. Different approaches such as inpainting, motion-based background completion and mosaicking have been proposed for the solution of this task.

Multi-layer texture inpainting have been used in [2] for objects' removing from the original video. Method provides a good quality of restored area, but has been tested for sequences with static camera and does not provide extrapolation of background outside the frame borders. Real-time background inpainting method has been proposed by [3]. It works fine for uniform background areas, but its work on complex textured background with larger foreground objects is unknown.

Motion based background completion have been proposed in [6]. Method uses tracking window for foreground objects marking. The main disadvantage of this method is the dependence of computational complexity on foreground objects' size. Also method has not been tested on sequences, where foreground occupies a large area of frame. And it does not provide the opportunity for expanding background outside the frame borders.

Mosaicking and background sprites construction is one of widespread approaches. The method of super resolution sprite generation was described in [5]. It provides single image representing background of a video sequence. The main disadvantage of this method is geometrical distortions which may appear on sprite borders. Another approach presented in [1] uses optical flow for frame segmentation and hierarchical insertion coordinates calculation which makes it not fast enough for use as a part of other algorithms.

The main task of the current approach was to create a method for fast background sprites construction, which quality and level of detail is the same as the quality of source frames. Most of existing approaches [1][5] are complicated for use at high resolutions video sequences and as part of other algorithms.

The main problems for panorama construction are connected with accuracy of frame matching during the insertion process and, in particular, with the choice of points for the transformation matrix calculation between background areas of frames. In this approach the coordinates of points in neighbor frames are obtained from motion estimation algorithm and the 8-parameter affine transformation is used to align two frames.

This paper is organized as follows. Section 2 describes motion estimation algorithm and points choosing algorithm. Section 3 describes frames' merging algorithm. Section 4 provides experimental results and Section 5 summarizes the paper.

2. MOTION ANALYSIS

This section describes the algorithm of motion analysis used for background transformation obtaining, background area evaluation and algorithm of choosing points for transformation calculation. Motion estimation algorithm [4] is used for correspondence search between points of analyzed frames. It provides motion vectors between square blocks in two frames with quarter-pixel precision and information about estimation error for every vector.

2.1 Background segmentation

For background area evaluation a simple motion-based segmentation algorithm is proposed. It uses two-dimensional histogram of motion vectors.

The histogram has size $MV_{max} \times MV_{max}$ for all possible motion vectors' coordinates, where MV_{max} is an algorithm parameter, which specifies the maximum length of a motion vector.

Each element with position (x, y) in the histogram corresponds to a motion vector with coordinates:

$(x - center\_x,\ y - center\_y)$,

where $(center\_x, center\_y)$ – coordinates of the center of the histogram.

Each element of the histogram stores:

- list of points in a frame corresponding to this motion vector
- top left and bottom right points from this list
- number of points corresponding to this motion vector
- label – in a clusterized histogram, number of a cluster containing this point

The main idea of histogram clusterization is to distinguish a rough mask of background area for the current frame. This segmentation will be used afterwards in transform matrix calculation.

First of all, the histogram represents all possible motion vectors for a pair of frames. If some point in an image has a motion vector with coordinates (x,y), histogram element (x,y) contains this point. Segmentation of a frame is based on analysis of histogram elements that contain at least two points. These elements will be called "valuable elements".

The proposed method considers all connected regions in the histogram as clusters. It means the algorithm marks neighbor valuable elements of the histogram and all valuable elements with the same label from one cluster. A cluster, in which the histogram element with maximum of points is situated (points from the current frame), is assumed to be background.

Figure 1 illustrates example of a clusterized histogram. Colored pixels correspond to valuable elements of histogram. White color marks the background area, blue – foreground objects. Three-dimensional plot represents the number of points in current frame, contained in each histogram element. Red peak corresponds to the background area.

2.2 Sky detection

For outdoor scenes background area often contains sky. The proposed method considers sky in calculations to improve segmentation accuracy. It is assumed, that sky area always belongs to background, so the points' choice is modified if sky detection gives positive result.

The sky detection algorithm is based on a color segmentation of potential sky areas, using pixel colors in HSV color space.

First of all, it is supposed that sky is in the upper region of a frame, so the method searches for it there. If a pixel color corresponds to a comparison criterion, the method checks its neighbor pixels. The algorithm stops if the next pixel doesn't correspond to the color criterion or a difference between neighbor pixels colors is bigger than a constant threshold.

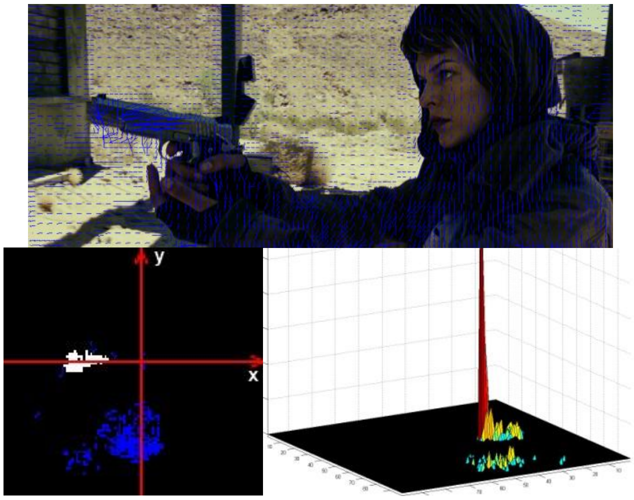

Figure 1: Example of source frame with motion vectors (blue lines) and clusterized histogram. Orthogonal projection (left) and 3D-visualization (right)

The used color criterion has been chosen empirically based on set of ~20 outdoor images containing sky. It consists of 3 inequalities:

- $\begin{cases} hsv.v > 0.7 \\ hsv.h < 0.001 \end{cases}$
- $\begin{cases} hsv.v > 0.3 \\ hsv.h > 0.4 \\ hsv.s < 0.8 \end{cases}$
- $\begin{cases} hsv.v > 0.75 \\ hsv.s < 0.2 \end{cases}$

Figure 2 shows the example of obtained sky mask.

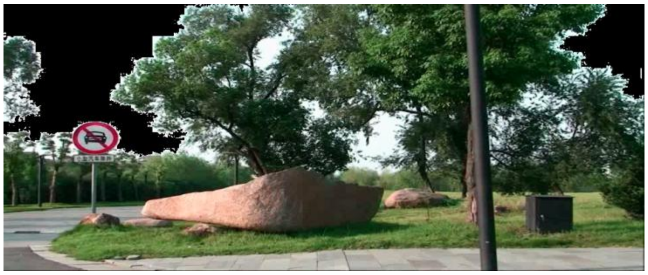

Figure 2: Result of sky detection algorithm

2.3 Equation points choice

Transform matrix is calculated from corresponding points' coordinates. Motion estimation provides large number of points with different coordinate precision, so there is need in choosing appropriate small set of points' pairs for further calculations. This set will be called equation points

The following algorithm is used for selecting equation points. First, points chosen for the previous pair of frames are checked with the current comparison criterion. It means that method checks several properties of a point. Two methods are implemented.

The first is fully based on the result of segmentation results and also considers the variance of the block containing analyzed point (point does not suit if the variance is smaller than a threshold which depends on the average variance of the frame and is calculated for every frame). The second considers the sky mask and segmentation results. It takes only points from the background area which motion vectors are similar to the average motion vector of the sky area.

$$P = P(x, y, variance) \cdot label \cdot sky\_label$$

$$label = \begin{cases} 1, \; pixel \in background \\ \quad 0, \; otherwise \end{cases}$$

$P(x, y, variance)$ – probability for point with coordinates (x, y) situated in block with $variance$ variance

If a point suits, information about it is updated and used in further calculations. If the number of suitable points from the previous step is less than predefined threshold (algorithm parameter, differs from 10 to 30), the method chooses other points from the current frame, using 32x32 pixels' grid. From each grid cell the candidate point with the smallest motion vector error is chosen. Then array of suitable points is sorted using points' motion vector error as a criterion. ¼ of points with the largest errors are deleted from the list. Then remaining points are decimated by removing every second point from the points list until the list contains ~25 points for the transformation calculation.

3. FRAME WARPING

After obtaining the set of equation points with appropriate coordinates, frame is inserted into the mosaic. 8-parametric affine transform is used to warp frame to his actual coordinates in mosaic. This section describes transformation matrix obtaining and inserting process. Such matrix is calculated for all neighbor frames in video sequence.

3.1 Transformation calculation

The used model can be represented by the following matrix:

$$H = \begin{pmatrix} h00 & h01 & h02 \\ h10 & h11 & h12 \\ h20 & h21 & 1 \end{pmatrix}.$$

This model provides accurate background matching in cases of plane-parallel, rotational camera motion, zoom and more complex combinations of such motion. The transformation is applied to every pixel of inserted frame. This means the matrix is multiplied by a pixel coordinate vector (z coordinate always equals to 1).

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} h00 & h01 & h02 \\ h10 & h11 & h12 \\ h20 & h21 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}; \quad \begin{pmatrix} x'' \\ y'' \end{pmatrix} = \frac{1}{z'} * \begin{pmatrix} x' \\ y' \end{pmatrix},$$

Here (x, y) are pixel coordinates in the current frame, (x'', y'') are pixel coordinates in the reference frame.

After equation points have been chosen, system of equations for the projective transformation matrix calculation is configured. From every point 2 equations are obtained:

$$X * h00 + Y * h01 + 1 * h02 + 0 * h10 + 0 * h11 + 0 * h12 - X * X' * h20 - Y * X' * h21 - X' = 0$$

$$0 * h00 + 0 * h01 + 0 * h02 + X * h10 + Y * h11 + 1 * h12 - X * Y' * h20 - Y * Y' * h21 - Y' = 0$$

where (X, Y) are coordinates of a pixel in the current frame, (X', Y') – in a reference frame, hxx – the corresponding element of 3x3 transformation matrix. Then obtained system is reduced to the square form by applying least squares and solved.

3.2 Frame insertion

Before frame insertion into mosaic the final transformation matrix between frame and mosaic is obtained. It equals to product of all consecutive transformation matrices between frames, starting from frame, which was the initial for the current mosaic. Good precision of consecutive transformation matrix calculation minimizes the result error of matrix multiplication for a long sequence.

If camera rotation angle for final transformation matrix is large, new sprite generation starts. Otherwise frame distortion on sprite boundaries will be very notable and spoil the quality of result panorama. This will make a sprite inapplicable for further use.

If frame transform increases frame area or its dimensions comparing to source dimensions frame is upscaled before the insertion into the mosaic by bicubic interpolation. Every frame changes only unfilled regions of mosaic except of frame boundaries. Linear blending of pixels is performed for boundary pixels in mosaic. It is performed to make these boundaries not notable in final sprite. Blending is based on bilateral filtering.

4. EXPERIMENTAL RESULTS

The main criterion for result quality measure is to avoid discontinuities on boundaries of inserted frames and to preserve precise level of detail.

The algorithm has been tested on fragments from movies such as “James Bond” and “Resident evil”. Scenes with both simple and complex motion have been used. The number of used test sequences is 44. The example of algorithm result is shown in Figure 3 and 4.

Use of frame segmentation during insertion process provides a mosaic with removed foreground objects, as shown in Figure 4. It is shown that the proposed method allows correct inpainting of massive foreground objects. Scaled fragment shows that the proposed method allows correct matching for complex background with large number of small details. This makes method useful in tasks where quality of restored background is critical.

The speed of background mosaic construction depends on algorithm mode. It is possible to create a set of mosaic sprites for analyzed video sequence or to create panorama for every frame in both directions. For the first mode algorithm speed exceeds 4 fps for 960×400 resolution sequence. Second mode has much more computational complexity and requires huge amount of memory operations. It also depends on number of frames added to mosaic for each frame. Creating panorama for one frame (960×400) in this mode takes approximately 40-45 seconds with 25 frames range in both directions. Speed tests were made on Intel Core2 Quad Q9450 2.66 GHz.

5. CONCLUSION

This work presents a fast approach of creating background mosaic for input videos. The proposed method is suitable for fully automatic background restoration and provides sprites of good level of detail. Used foreground-background pre-segmentation allows full or partial removing of foreground objects from the result sprite. Method can also be used for background generation for stereo and multiview generation.

6. ACKNOWLEDGMENTS

This work was supported by RFFI grant 10-01-00697-a.

7. REFERENCES

[1] A. Krutz, A. Glantz, T. Borgmann, M. Frater, and T. Sikora, “*Motionbased object segmentation using local background sprites*,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, April 2009.

[2] I. Cheng Chang and Chia-We Hsu, “*Video Inpainting Based on Multi-Layer Approach*,” Proceedings of APSIPA Annual Summit and Conference, Sapporo, Japan, Oct. 2009

[3] Jan Herling and Wolfgang Broll, http://www.tu-ilmenau.de/journalisten/pressemeldungen/einzelnachricht/newsbeitrag/5784/

[4] K. Simonyan, S. Grishin, D. Vatolin, D. Popov, “*Fast video super-resolution via classification*,” in Proc. IEEE ICIP, pp. 349-352, San Diego, Oct. 2008.

[5] M. Kunter, J. Kim, and T. Sikora, “*Super-resolution mosaicking using embedded hybrid recursive flow-based segmentation*,” in IEEE Int. Conf. on Information, Communication and Signal Processing (ICICS’05), Bangkok, Thailand, Dec. 2005.

[6] Soon-Yong Park1, Chang-Joon Park2, and Inho Lee, “*Moving Object Removal and Background Completion in a Video Sequence*,” 2008

About the authors

Dmitriy Vatolin (M’06) received his M.S. degree in 1996 and his Ph.D. in 2000, both from Moscow State University. Currently he is Head of the Video Group at the CS MSU Graphics & Media Lab. He is author of the book Methods of Image Compression (Moscow, VMK MGU, 1999), co-author of the book Methods of Data Compression (Moscow, Dialog-MIFI, 2002) and co-founder of www.compression.ru and www.compression-links.info. His research interests include compression techniques, video processing and 3D video technics: optical flow, depth estimation - from motion, focus, cues, video matting, background restoration, high quality stereo generation. His contact email is dmitriy@graphics.cs.msu.ru.

Maxim Smirnov is the chief technology officer at YUVsoft IT company. Maxim has graduated from Saint Petersburg State University of Aerospace Instrumention in 2001 and received a Ph.D. in data compression from the same university in 2005. His research interests include forecasting of processes in technical and economic systems, video and universal data compression, stereo video. His contact email is ms@yuvsoft.com.

Anton Zachesov is a student at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is azachesov@graphics.cs.msu.ru.

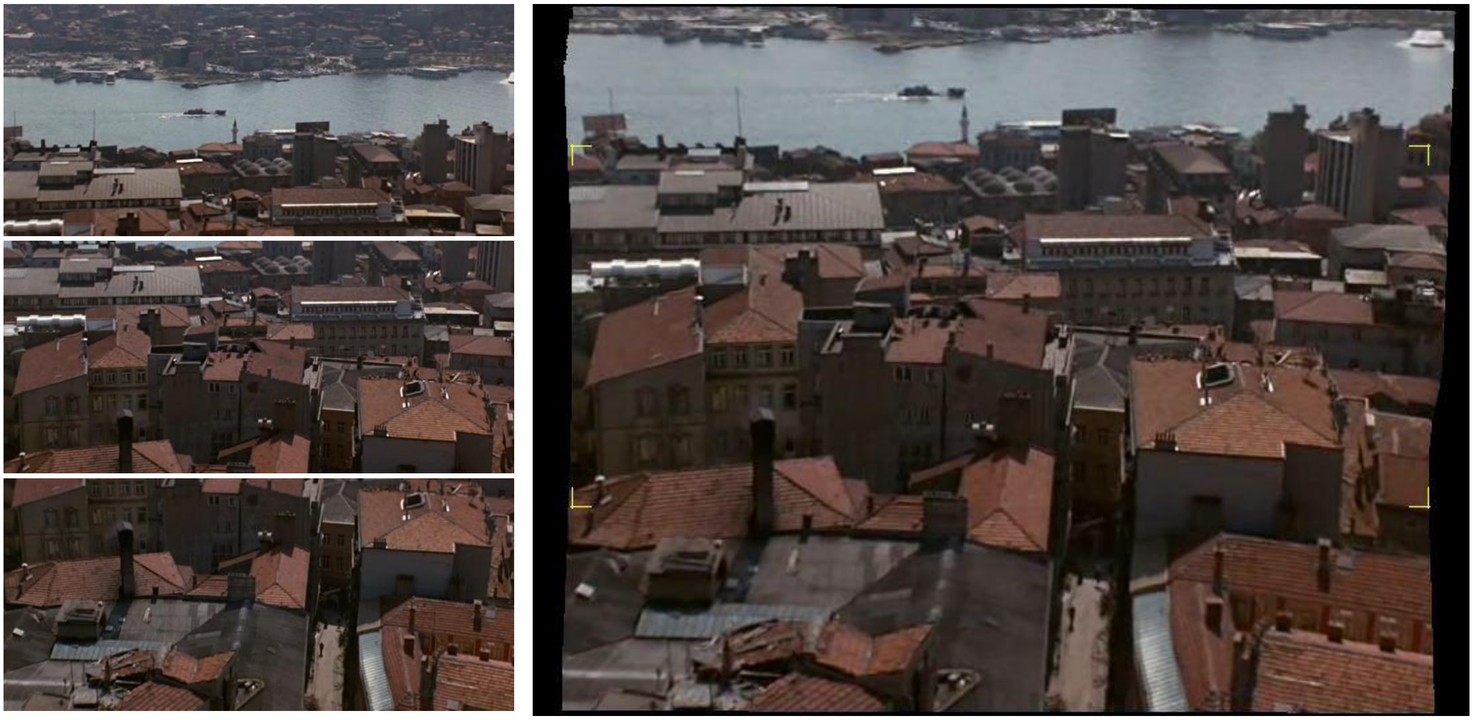

Figure 3: Frames 28, 93, 135 of the original sequence and panorama based on frame 93

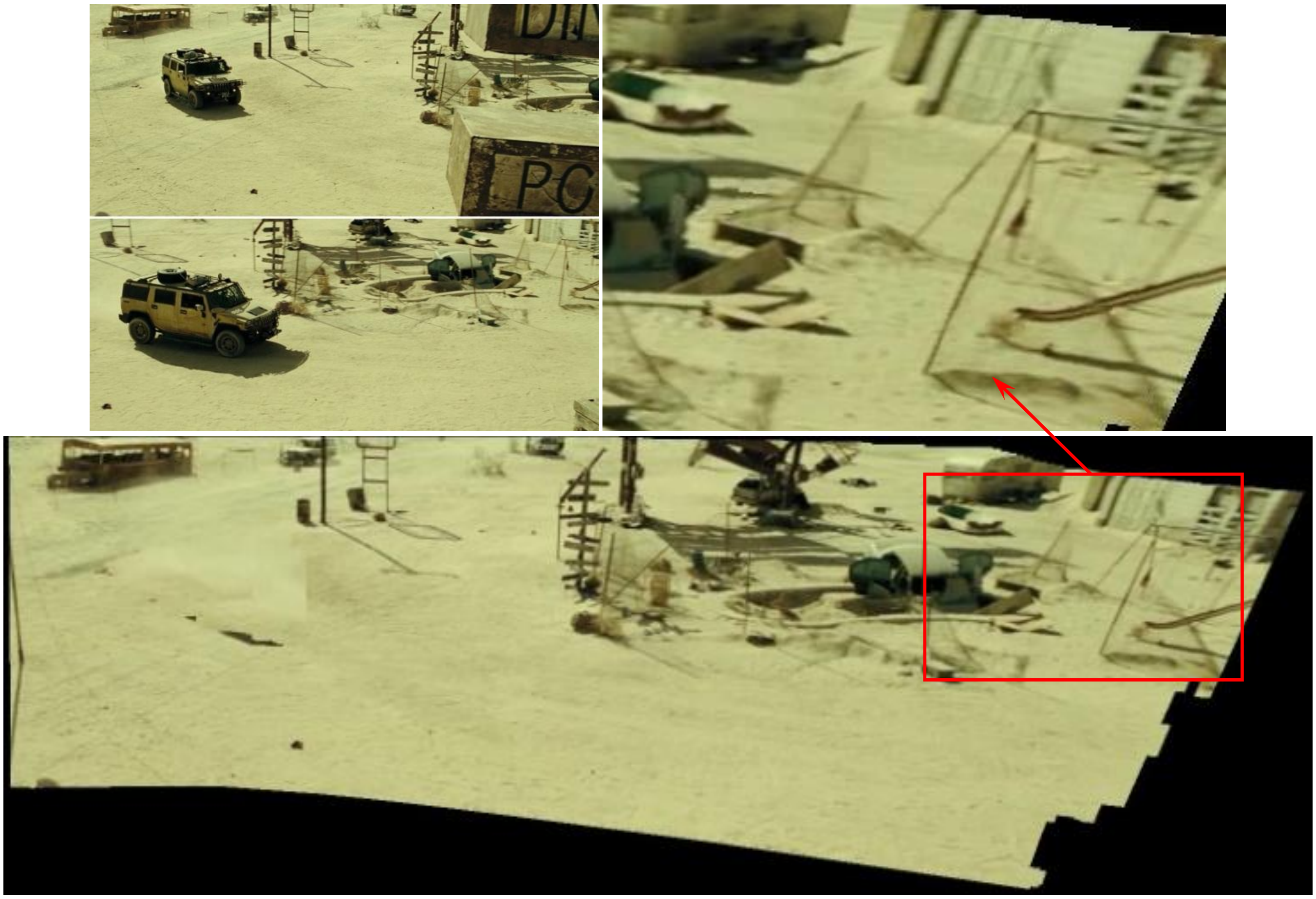

Figure 4: Frames 4, 54 of original sequence and panorama based on frame 4