MOSAIC OF A VIDEO SHOT WITH MULTIPLE MOVING OBJECTS

A. Fusiello, M. Aprile, R. Marzotto, V. Murino

Dipartimento di Informatica, University of Verona
Strada Le Grazie 15, 37134 Verona, Italy
{fusiello, murino}@sci.univr.it

ABSTRACT

In this paper we describe an application which takes a video shot as input and produces a compact representation composed by a background layer and segmented moving objects. We deal with the problems of global registration, super-resolution mosaicing, objects segmentation and tracking. Global registration is achieved with a graph-based technique that exploits situations when the camera returns to a previously seen area. Objects segmentation is based on motion analysis using a robust statistical model of the background. Tracking is based on blob matching using Singular Value Decomposition.

1. INTRODUCTION

Since the introduction of MPEG-4, extracting moving objects (MOs) from video sequences has attracted a growing attention. By exploiting the object-based representation offered by MPEG-4, video shots can be encoded as a stationary background mosaic – obtained after compensating for camera motion – plus MOs represented individually. The challenge is to create a system that is able to do this segmentation automatically and with great accuracy, both in terms of resolution and sharpness of the background mosaic and in the trimming of the MOs silhouettes.

In this paper we describe a complete application which produces an object-based representation of a video shot, and in particular, we address the problems of the global registration, super-resolution mosaicing, and multiple MOs segmentation and tracking.

This paper builds on a previous work [1], where only one MO was allowed, no global optimization was performed and super-resolution was not considered. In this paper we improve radically the MOs tracking algorithm as well as the quality of the background mosaic. A simple statistical model of the background is also introduced.

Global registration refers to the alignment of video frames taking into account (ideally) all the overlapping frames, and not just the consecutive ones. Many approaches have been proposed in the last years. In [2] the global consistency of the inter-frame alignment matrices is enforced by solving a linear system of equations. The system is linear only if an affine model is used. In [3] global registration is achieved by minimizing differences between ray directions going through corresponding points. As far as the global registration is concerned, the most closely related work are [4, 5]. Both uses a graph representation and [5] cast the problem as a shortest path.

The MOs tracking approach is inspired by [6], where a graph is used to represent objects and both shape and color features are used to match them.

2. BACKGROUND AND NOTATION

Two pictures of the same scene are related by a (non-singular) linear transformation of the projective plane in two cases: i) the scene is planar or ii) the point of view does not change (pure rotation). In these cases, which can be summarized by saying the there must be no *parallax*, images can be composed together to form a *mosaic*.

Points are expressed in homogeneous coordinates, that is, 2-D points in the image plane are denoted as $\tilde{\mathbf{x}} = (x, y, 1)$ with $\mathbf{x} = (x, y)$ being the corresponding Cartesian coordinates.

A linear transformation of the projective plane, called a *homography*, is represented by a 3×3 matrix H:

$$\tilde{\mathbf{x}}_i = H_{ij}\tilde{\mathbf{x}}_j \tag{1}$$

where $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ are corresponding points in frame i and j respectively.

3. MOSAICING

Inter-frame homography computation is based on the Kanade-Lucas-Tomasi (KLT) tracker [7, 8], initialized with phase-correlation to reduce search range. Features are tracked through the video sequence and correspondences are used to compute homographies $H_{i+1,i}$ between each consecutive pair of frames. As in [1], Least Median of Squares is used in order to be robust against tracking errors and features attached to moving objects. These homographies are then combined to produce a mosaic of the background (assuming that the majority of the tracked features belong to the background). Any frame can be chosen as the reference one onto which register all the others. The *pairwise alignment* consists in computing, for each frame i the homography $H_{k,i} \triangleq H_i^t$ that maps frame i onto the reference frame k, using recursively the homographies that links consecutive frame pairs:

$$\begin{cases} H_k = I \\ H_i^t = H_{i-1}^t H_{i-1,i} & \text{if } i > k \\ H_i^t = H_{i+1}^t H_{i+1,i} & \text{if } i < k \end{cases}$$

When the video sequence is long enough, this straightforward way to compose homographies yields an appreciable misalignment for the frames more distant from the reference one. This is especially evident when the camera goes back onto a scene part previously seen. In this case, registration can take advantage from homographies linking non-consecutive frames and reduce the global misalignment error.

3.1. Global registration

The first step is to establish which frame overlaps which. The pairwise alignment gives a good approximation of the registration ma-

0-7803-7750-8/03/$17.00 ©2003 IEEE.

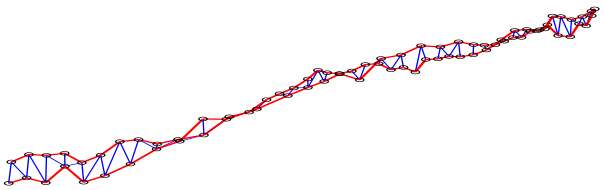

Fig. 1. Graph of the sequence. The red (bold) edges links consecutive frames in the sequence. Vertices position is given by the centroid of the corresponding frame in the mosaic reference system (y-axis is stretched). Blue (thin) edges are those added by our algorithm.

trices $\{H_i^t\}$, which allow to estimate the degree of overlap between each frame pair.

A graph is then constructed, whose vertices are the frames and edges links frame pairs for which an homography can be computed directly. Edges are weighted with the mean squares residual of the homography computation. Initially, only consecutive frames in the sequence are connected.

Not all overlapping frames will be linked in the graph, but only those that i) have a significant overlap, sufficient to yield a correct alignment and ii) reduce significantly the shortest path between two vertices. The latter condition contributes to increase efficiency. Finally, for each new edge (i, j) the corresponding homography H_{ij} is computed directly from feature correspondences and the weight is assigned to the edge. As the two frames i and j are not consecutive, features can undergo severe perspective distortion. To overcome this problem and obtain a more accurate matching (KLT tracker is based on a translational model) the two frames are first transformed onto the reference frame (with $\{H_i^t\}$ and $\{H_j^t\}$ respectively), thereby compensating the distortion.

In the final graph we can compute, for each frame i, the transformation H_i^s that aligns it with the reference frame by chaining homographies along the shortest (weighted) path from i to k. H_i^s is less affected by errors accumulation than H_i^t because it is the product of (possibly) fewer low-residual factors.

The subsequent global optimization finds the $\{H_i\}$ that simultaneously minimizes the misalignment of a pre-defined set G of grid-points on the mosaic. Let $\mathbf{x}_k$ be a grid-point and let L_k be the set of edges (i, j) such that $\mathbf{x}_k$ belongs to overlap region between frame i and frame j. The error at the grid-point $\mathbf{x}_k$ is defined as:

$$E(\mathbf{x}_k) = \frac{1}{|L_k|} \sum_{(i,j)\in L_k} ||\mathbf{x}_k - \Pi(H_i H_{ij} H_j^{-1} \tilde{\mathbf{x}}_k)||^2 \quad (2)$$

where Π transforms homogeneous coordinates into Cartesian (pixel) coordinates. Since we want to simultaneously minimize the error at all the grid points, we end up with a system on non-linear equations that can be cast as a least-squares problem:

$$\min_{\{H_i\}} = \sum_{\mathbf{x}_k \in G} E^2(\mathbf{x}_k) \quad (3)$$

The Levemberg-Marquardt algorithm[1] is used to solve Eq. 3, using $\{H_i^s\}$ as the starting solution. Usually this is already a good solution and few iterations are needed to get to the global minimum.

[1]Available through the `lsqnonlin` MATLAB function

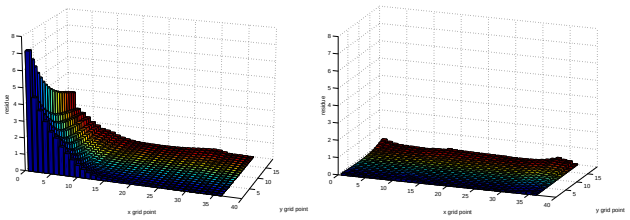

Fig. 2. Alignment residuals on mosaic grid-points before (left) and after (right) global optimization.

3.2. Blending and background modeling

Starting from a single mosaic pixel P, if we imagine to pierce all the aligned frames with a temporal line, we will intersect pixels that correspond to the background and pixels belonging to MOs. We model the color histogram of these pixels as a Gaussian distribution corrupted by outliers corresponding to the MOs. Therefore, the median of the distribution – being a robust estimate of the mean – is taken as the background color and assigned to P:

$$\bar{c} = \underset{i}{\text{med}}\{c_i\}. \quad (4)$$

Moreover, we attach to each mosaic pixel P an estimate of the background color variability at that point. A robust estimator of the spread of the distribution is given by the median absolute difference (MAD):

$$\text{MAD} = \underset{i}{\text{med}}\{|c_i - \bar{c}|\}. \quad (5)$$

It can be seen [9] that, for symmetric distributions, the MAD coincides with the *interquartile range*:

$$\text{MAD} = \frac{\xi_{3/4} - \xi_{1/4}}{2}, \quad (6)$$

where ξ_q is the qth quantile of the distribution (for example, the median is $\xi_{1/2}$). Hence, a pixel with color c, is deemed to belong to the background with 99.9% confidence if

$$|c - \bar{c}| < 5.2MAD \quad (7)$$

This is the X-84 outlier rejection rule [9].

3.3. Super-resolution

Our approach is inspired by [10], where sub-pixel motion information of a global motion model is used to create mosaics with a resolution that is higher than the resolution of each single video frame that composes the mosaic. In [10] *Source-scan* is performed and only if a pixel is mapped close enough to an integer position (± 0.2) in the mosaic, its color is assigned. This procedure can leave holes in the mosaic, which are then filled by interpolation.

Our super-resolution mosaic is built using *destination-scan* warping: for each pixel in the mosaic (which has a resolution greater than the single frames), find the corresponding position in each frame by backward mapping (with properly scaled transformations) and pick the color of the nearest pixel, over all the frames.

This "nearest pixel" strategy works well only if the registration is *very* accurate. In practice a weighted strategy gives usually better results: backward-map the mosaic pixel in each frame, find the closest pixel and weigh its color with the inverse of the distance. Assign to the mosaic pixel the weighted average of the colors.

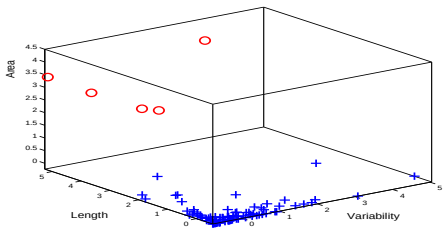

Fig. 3. Tracks represented in the features space. Red circles are the good tracks.

4. SEGMENTING MOVING OBJECTS

MOs are obtained from the original video shot by differencing with the background. For each frame, the mosaic of the background is back-warped onto the frame and each pixel is labeled as belonging to MO or not according to the rule given by Eq. 7.

Then, the resulting binary image is cleaned with morphological filtering and connected components (blobs) are identified as candidate MOs. In order to discriminate between true MOs and noise, the next step is to exploit temporal coherence.

4.1. Tracking Moving Objects

A layered graph is built, where each layer correspond to a frame and each vertex is a blob. An edge links two blobs from consecutive layers if they represent the same MO (or part of it) at different time. A trajectory of an object is a multi-path in the graph, i.e., a path that can split and merge. Initially there are no edges, and the goal of the tracking is to find multi-paths in the graph.[2]

A similarity measure between blobs is defined taking into account the appearance (shape and color) of the blob and its position. In particular, each blob is described by a feature vector composed by: centroid, area, solidity, eccentricity, orientation[3], average color, contrast (standard deviation of the color). Following [6], the similarity of blobs I_i and J_j is computed as

$$\mathrm{sim}(I_i, J_j) = \frac{\mathrm{asim}(I_i, J_j)}{1 + d(I_i, J_j)^2} \tag{8}$$

where $d(I_i, J_j)$ is the distance of the centroids in the mosaic reference frame and $\mathrm{asim}(I_i, J_j)$ is the *appearance similarity* between the two blobs. The latter is computed as a weighted sum of the similarity value for each component of the feature vector (without centroid). If s is a scalar component of the feature vector (e.g. solidity), $\mathrm{asim}(I_i, J_j)_s = e^{-k_s(s_i - s_j)^2}$.

Path in the graph are constructed by matching blobs from one layer to the next. We used the matching technique introduced by Longuet-Higgins [11], who proposed an algorithm (based on the singular value decomposition (SVD)) for associating the features of two images.

Let $\{I_i\}_{1\ldots n}$ and $\{J_j\}_{1\ldots m}$ the two sets of blobs which we want to put in one-to-one correspondence. The first stage is to build a *proximity matrix* G of the two sets of features: $G_{ij} = \mathrm{sim}(I_i, J_j)$. The next stage is to perform the SVD of G

$$G = USV^{\mathsf{T}} \tag{9}$$

where U and V are orthogonal and S is a non-negative $m \times n$ diagonal matrix. Finally, S is converted into a new $m \times n$ matrix D by replacing every diagonal element S_{ii} with 1, thus obtaining another matrix $P = UDV^{\mathsf{T}}$ of the same shape as the original proximity matrix and whose rows are mutually orthogonal. The element P_{ij} indicates the extent of pairing between the blobs I_i and J_j. If P_{ij} is both the largest element in its row and the largest element in its column, then we regard I_i and J_j as corresponding with each other.

By iterating matching over each layer of the graphs, chains of nodes (*tracks*) are obtained. Tracks are classified by unsupervised clustering in the feature space composed by: average blob size, trajectory temporal length and trajectory variability, defined as

$$\mathrm{var} = \left(\frac{\sum_i (x^*(i) - x(i))^2}{\sum_i x(i)^2}\right)^{1/2} + \left(\frac{\sum_i (y^*(i) - y(i))^2}{\sum_i y(i)^2}\right)^{1/2}$$

where $(x(i), y(i))$ is the trajectory of the centroid of the blob and $(x^*(i), y^*(i))$ is the mobile-averaged trajectory (window size is 3).

This feature space proved to be adequate to discriminate good tracks in many real sequences (see Fig. 3 for example). We experimented several standard unsupervised clustering algorithm, and the *Complete Link* [12] algorithm was selected. This is a hierarchical clustering algorithms where the distance between two clusters is defined as the maximum of all pairwise distances between patterns in the two clusters. The resulting dendrogram is cut in order to get two classes. Bad tracks are marked but not discarded, yet.

Up to this point, only chains have been obtained. If paths are to be allowed to split and merge (imagine an object partially occluded), a further processing is necessary.

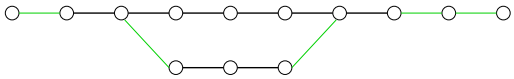

Fig. 4. Two simple paths merges into a multi-path, corresponding to a partially occluded object, like a person walking behind a pole. The thin (green) edges are those added in the second phase.

At both ends of each path a local search is carried out to find blobs that could prolong the path. The search area depends on the blob area and it is centered in the predicted position of the centroid, basing on the last 3 frames. All the blobs are candidate, also those already belonging to a path. The search is repeated recursively, until either it fails or it finds a blob belonging to a path. At this point we consider the response of classification step and remove all the noisy tracks that did not merged with any good one. In this way, beside recovering blobs that were not in a good track, we can merge paths representing fragments of the same MO.

This technique is quite general, and can take into account occlusions between MOs, occlusions between a MO and a background object, MOs entering and leaving the scene at any point.

5. RESULTS

We report here some results on a video shot taken with a digital hand-held camera (Fig. 6). The two persons enter the scene from the opposite side and cross each other. The camera does a panning motion, following first the man from left to right and then the woman from right to left.

[2] Strictly speaking, this is a *data association* task.

[3] See `regionprops` in the MATLAB Image Processing Toolbox

In Fig. 5 the background mosaics are shown. The improvement of the global registration is particularly evident in the framed area at the bottom left of the mosaic.

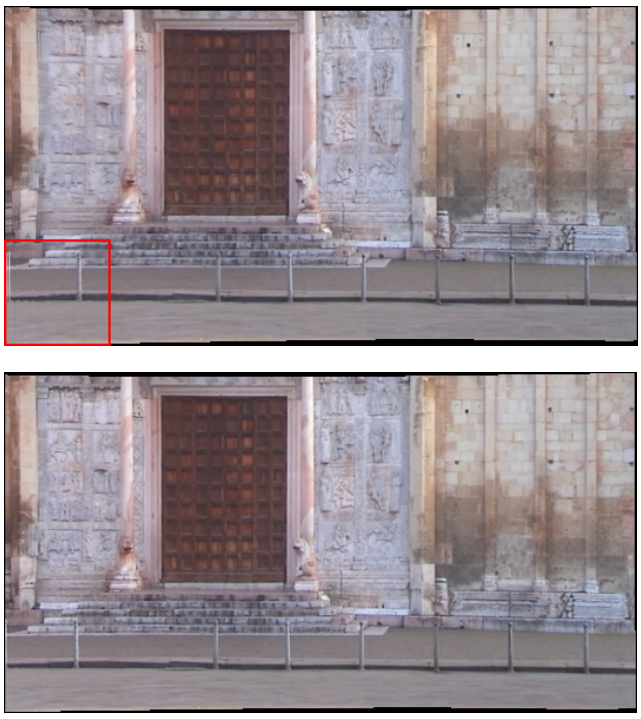

Fig. 5. Mosaics of the background, obtained with pairwise alignment (top), and global optimization (bottom).

The quality evaluation of mosaics is usually subjective, and its based on the perceived blurring. We propose to use an *objective* blurring measure, taken from the vast literature on focusing [13]. In particular, we chose the power of the image Laplacian ∇^2 as it is smooth and has a sharp maximum:

$$\mathrm{LP}(I) = \sum_{x,y} (\nabla^2 I(x,y))^2.$$

The LP is 719.347 for the mosaic after pairwise alignment, 770.057 after graph-based alignment and 773.758 after global optimization.

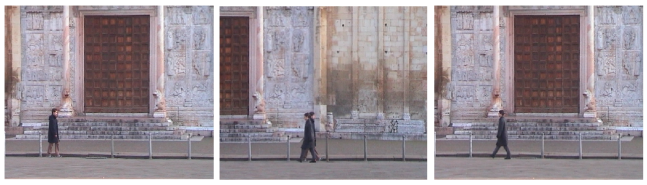

Fig. 6. Frames number 1, 45 and 79 (last) of the "Lorena" sequence. Dimensions are 338×280.

Figure 7 shows some MOs extracted form the sequence. If we paste the MOs onto the mosaic and warp it back with $\{H_i^{-1}\}$ we obtain again the original sequence (decoding), with an PSNR that is always above 28dB.

More examples are available on the web at
`http://profs.sci.univr.it/~fusiello/demo/motseg`.

Fig. 7. Sample MOs estracted from the video sequence

Acknowledgments

The present work is based on a previous paper co-authored by F. Odone, A. Fusiello, and E. Trucco [1]. A. Colombari provided the implementation of the KLT tracker and contributed with useful discussions. Thanks to M. Bicego for his advice on clustering.

6. REFERENCES

[1] F. Odone, A. Fusiello, and E. Trucco, "Layered representation of a video shot with mosaicing," *Pattern Analysis and Applications*, vol. 5, no. 3, pp. 296–305, 2002.

[2] J. Davis, "Mosaics of scenes with moving objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 354–360.

[3] H.-Y. Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," *International Journal of Computer Vision*, vol. 36, no. 2, pp. 101–130, 2000.

[4] H. S. Sawhney, S. Hsu, and R. Kumar, "Robust video mosaicing through topology inference and local to global alignment," in *Proceedings of the European Conference on Computer Vision*, 1998, vol. 1407, pp. 103–119.

[5] E. Kang, I. Cohen, and G. Medioni, "A graph-based global registration for 2D mosaics," in *Proceedings of the International Conference on Pattern Recognition*, 2000, pp. 257–260.

[6] I. Cohen and G. Medioni, "Detecting and tracking moving objects in video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. II:319–325.

[7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981.

[8] C. Tomasi and T. Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA, 1991.

[9] F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel, *Robust Statistics: the Approach Based on Influence Functions*, John Wiley & Sons, 1986.

[10] A. Smolic and T. Wiegand, "High-resolution video mosaicing," in *Proceedings of the IEEE International Conference on Image Processing*, 2001.

[11] G. Scott and H. Longuet-Higgins, "An algorithm for associating the features of two images," in *Proceedings of the Royal Society of London B*, 1991, vol. 244, pp. 21–26.

[12] A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, Prentice Hall, 1988.

[13] M. Subbarao, T. Chio, and A. Nikzad, "Focusing techniques," *Optical Engineering*, pp. 2824–2836, 1993.