A Compact and Retrieval-Oriented Video Representation Using Mosaics

Gabriele Baldi, Carlo Colombo, and Alberto Del Bimbo

Università di Firenze, Via Santa Marta 3, I-50139 Firenze, Italy

Abstract. Compact yet intuitive representations of digital videos are required to combine high quality storage with interactive video indexing and retrieval capabilities. The advent of video mosaicing has provided a natural way to obtain content-based video representations which are both retrieval-oriented and compression-efficient. In this paper, an algorithm for extracting a robust mosaic representation of video content from sparse interest image points is described. The representation, which is obtained via visual motion clustering and segmentation, features the geometric and kinematic description of all salient objects in the scene, being thus well suited for video browsing, indexing and retrieval by visual content. Results of experiments on several TV sequences provide an insight into the main characteristics of the approach.

1 Introduction

Quite recently, the rapid expansion of multimedia applications has encouraged research efforts in the direction of obtaining compact representations of digital videos. On the one hand, a compact video encoding is required for high quality video storage; on the other hand, an ad hoc video representation needs to be devised at archival time in order to ease video browsing and content-based retrieval. Past approaches to video compression (see e.g., the standards MPEG 1 and 2) have privileged image processing techniques which, taking into account only the signal-level aspects of visual content, emphasize size reduction over retrieval efficiency. More recently, they have been presented browsing-oriented computer vision techniques to represent videos by reconstructing the very process of film making [4]. These techniques are capable to segment the video into a number of "shots," each delimited by film editing effects such as cuts, dissolves, fades, etc. The description of shot content relies then on the extraction of salient "keyframes." However, the above techniques have the limitation of providing only a partial information of video content, being it impossible to reconstruct a video only from its keyframes. The advent of mosaicing techniques [7], [8] paved the way to content-based video representations which are both retrieval- and compression-efficient. Such techniques reduce data redundancy by representing each video shot through a single patchwork image composed using all of its frames.

In this paper, a method to represent video content through image mosaics is described. Mosaics are extracted from video data through corner-based tracking and a 2D affine motion model. An original motion clustering algorithm, called DETSAC, is proposed. The obtained video representation features the image and motion description of all salient objects in the scene, and is well suited to both video browsing and retrieval by visual content.

2 Video Segmentation

The primary task of video analysis is video editing segmentation, i.e. the identification of the start and end points of each shot. Such a task implies solving two problems: *i*) avoiding incorrect identification of shot changes due to rapid motion or sudden lighting change in the scene; *ii*) detect sharp (*cuts*) as well as gradual transitions (*dissolves*). To avoid false shot change detection, a correlation metric based on HSI color histograms is used, which is highly insensitive even to rapid continuous light variations while maintaining reliable to detect cuts. To detect dissolves, a novel algorithm based on *corner statistics* is used, based on monitoring the minima in the number of salient points detected. During a dissolve, while the previous shot gradually fades out and its associated corners disappear, the new one fades in, its corners being still under the saliency threshold [3].

3 Shot Analysis

Once a video is segmented into shots, each shot is processed so as to extract its 2D dynamic content and allow its mosaic representation. Image motion is processed between successive frames of the shot via a three-step analysis: (1) corner detection; (2) corner tracking; (3) motion clustering and segmentation.

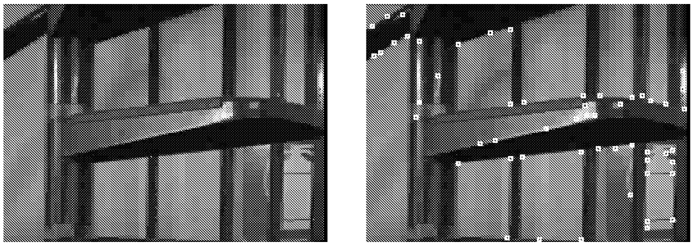

Fig. 1. Corner detection. The pixel locations corresponding to extracted corners are shown in white.

Corner Detection. An image location is defined as a corner if the intensity gradient in a patch around it is not isotropic, i.e. it is distributed along two preferred directions. Corner detection is based on the algorithm originally presented by Harris and Stephens in [6], classifying as corners image points with large and distinct values of the eigenvalues of the gradient auto-correlation matrix. Figure 1(*right*) shows the corners extracted for the indoor frame of Fig. 1(*left*).

Corner Tracking. To perform intra-shot motion parameters estimation, corners are tracked from frame to frame, according to an algorithm originally proposed by Shapiro *et al.* in [9] and modified by the authors to enhance tracking robustness. The algorithm optimizes performance according to three distinct criteria, namely:

Frame similarity: The image content in the neighborhood of a corner is virtually unchanged in two successive frames; hence, the matching score between image points can be measured via a local correlation operator.

Proximity of Correspondence: As frames go by, corner points follow smooth trajectories in the image plane, thus allowing to reduce the search space for each corner in a small neighborhood of its expected location, as inferred based on previous tracking results.

Corner Uniqueness: Corner trajectories cannot overlap, i.e. it is not possible that at a same time two corners share the same image location. Should this happen, only the corner point with higher correlation would be maintained, while the other would be discarded.

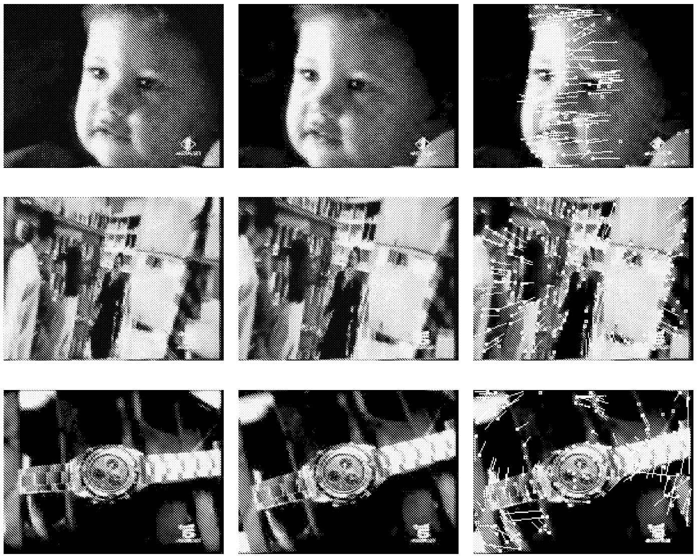

Fig. 2. Corner tracking examples. *1st row*: translation induced by camera panning; *2nd row*: divergence induced by camera zooming; *3rd row*: curl induced by camera cyclotorsion.

Since the corner extraction process is heavily affected by image noise (the number and individual location of corners varies significantly in successive frames; also, a corner extracted in one frame, albeit still visible, could be ignored in the next one), the modified algorithm implements three different corner matching strategies, ensuring that the above tracking criteria are fulfilled:

- *strong match*, taking place between pairs of locations classified as corners in two consecutive frames;
- *forced match*, image correlation within the current frame, in the neighborhood of a previously extracted corner;
- *backward match*, image correlation within the previous frame, in the neighborhood of a currently extracted corner.

These matching strategies ensure that a corner trajectory continues to be traced even if, in some instants, the corresponding corner fails to be detected. Fig. 2 shows corner tracking examples from three different commercial videos, and featuring diverse kinds of 2D motions induced by specific camera operations. Each row in the figure shows two successive frames of a shot, followed by the traced corners pattern.

Motion clustering and segmentation. After corner correspondences have been established, an original motion clustering technique is used to obtain the most relevant motions present in the current frame. Each individual 2D motion of the scene is detected and described by means of the affine motion model

$$\begin{pmatrix} x' - x \\ y' - y \end{pmatrix} = \begin{pmatrix} a_0 & a_1 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_2 \\ a_5 \end{pmatrix} \tag{1}$$

characterizing image points displacements – (x, y) and (x', y') denote the coordinates of a same point in the previous and current frames, respectively. Motion clustering takes place starting from the set of corner correspondences found for each frame. A robust estimation method is adopted, guaranteeing on the one hand an effective motion clustering, and on the other a good rejection of false matches (clustering outliers).

The clustering technique, called DETSAC ("DETerministic SAmple Consensus"), is an adaptation of the "RANdom SAmple Consensus" (RANSAC) algorithm ([5], see also [2]) to the problem of motion clustering. DETSAC operates as follows. For each trajectory obtained by corner tracking, the two closest corners trajectories are used to compute the affine transformation (i.e., the 6 degrees of freedom $a_0, \ldots, a_5$) which best fits the trajectory triplet (each corner trajectory provides two constraints for eq. (1), hence three non collinear trajectories are sufficient to solve for the six unknown parameters). The number of trajectories "voting" for each obtained transformation candidate determine the consensus for that candidate. Iterating the candidate search and consensus computation for all possible corner triplets, the dominant motion with maximum consensus is obtained. All secondary motions are iteratively computed exactly in the same way, after the elimination of all the corner points with dominant motion. The RANSAC algorithm is conceived to reject well outliers in a set of data characterized by a unimodal population. Yet, in image motion segmentation, it is highly probable that two or more data populations are presented at a given time instant, corresponding to independently moving objects. In such cases, RANSAC is likely to produce grossly incorrect motion clusters. As an example, Fig. 3 shows that, when attempting to cluster data from two oppositely translating objects, RANSAC wrongly interprets the two motions as a single rotating motion.

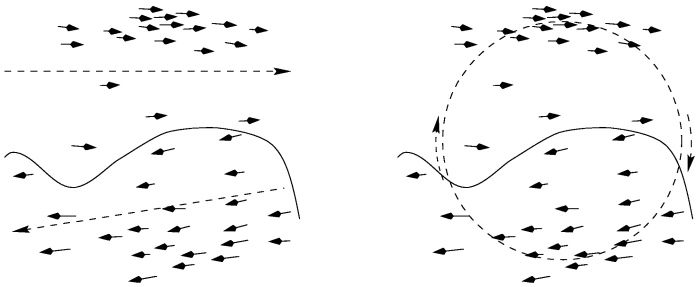

Fig. 3. Motion clusters for two translating objects. *Left*: Ground truth solution. *Right*: RANSAC solution.

Although DETSAC is conceived to solve the multimodal distribution problem, however, it achieves this at the cost of a diminished parameter estimation accuracy (nearby trajectories tend to amplify ill-conditioning). Therefore, DETSAC is only meant to provide a rough motion estimate, to be refined later via an iterative weighted least squares strategy, where the higher is the departure of each individual observation from the current estimate, the lower is its associated weight. In such a way, the robustness to outliers of DETSAC is efficiently coupled with the estimation accuracy of least squares. Besides, the above clustering algorithm would exhibit a somewhat tendency to fragment each cluster into sub-clusters. To avoid that, a further cluster merging step is performed, to ensure that each new cluster is distant enough from previous clusters in the space of affine transformations [10]; if this does not happen, clusters below a minimum threshold are merged together. The distance between two affine transformations $A = (a_0, \dots, a_5)$ and $B = (a'_0, \dots, a'_5)$ is defined as

$$d = \sqrt{(p_0^2 + p_1^2 + p_3^2 + p_4^2)\left(\frac{l}{2}\right)^2 + p_2^2 + p_5^2} \quad , \tag{2}$$

where $p_i = |a_i - a'_i|$ for $i = 0, \dots, 5$, and $l = (w + h)/2$ is the average frame size. Qualitatively, eq. (2) expresses the displacement (in pixels) produced in the frame's periphery as the effect of the difference between the motions A and B. Indeed, each addend under the square root expresses the contribution of each individual parameter to the overall displacement.

Another important feature of the clustering algorithm is temporal subsampling. In fact, by limiting initially the motion analysis to every 16 or 32 frames, slow motions or motions very similar to each other can be succesfully detected and differentiated. Only in a second phase, the motion analysis is refined by iteratively halving the frame interval, until all the frames of a sequence are processed. The motion clusters obtained at higher subsampling levels are used as constraints for clustering refinement, so as to avoid that previously formed clusters are incorrectly merged.

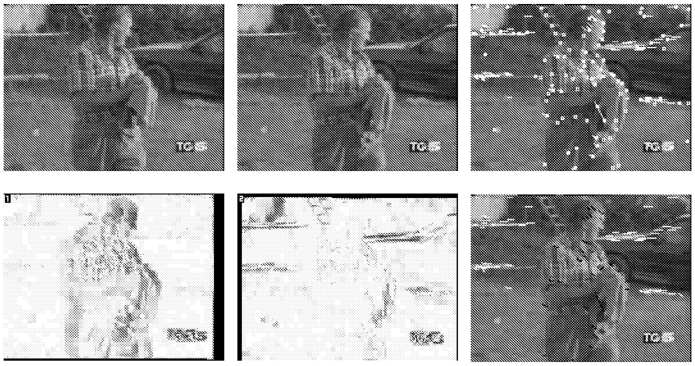

Fig. 4. Residual errors and motion segmentation. The two motion clusters obtained are shown using white and black arrows, respectively.

Figure 4 provides a clustering example obtained from a news video, including a man moving in an exterior. As the camera pans rightwards to track the man, the image background moves leftwards; hence, the shot features two main image motions (man, background). The upper row of Fig. 4 reports two successive frames, and the result of corner tracking; in the lower row, they are shown the residual segmentation errors relative to the background (dominant motion) and the man (secondary motion), respectively, together with the final clustering result.

The actual motion-based segmentation is performed by introducing spatial constraints to the classes obtained via the previous motion clustering phase. Compact image regions featuring homogenous motion parameters – thus corresponding to single, independently moving objects – are extracted by region growing [1]. The motion segmentation algorithm is based on the computation of an a posteriori error obtained by plain pixel differences between pairs of frames realigned according to the extracted affine transformations.

4 Representation Model

The mosaic-based representation includes all the information required to reconstruct the video sequence, namely,

- location of each shot in the overall sequence;
- type of editing effect;
- mosaic image of the background;

– 2D motion due to the camera (for each frame);
– 2D motion and visual appearance of each segmented region (also for each frame).

Figure 5 shows some frames of a video sequence featuring composite horizontal and vertical pan camera movements, and the mosaic obtained. The mosaic image captures all the background details which are present at least in one frame of the sequence. Also, the mosaic composes individual frame details into a global description of the background (notice, e.g., that the babies are never visible all together in just one frame). Figures 6 and 7 show two more complicated examples, featuring camera panning and zooming and the presence of an independently moving objects. Notice, again, that both the car in Fig. 6 and the farm in Fig. 7 is almost integrally visible in the mosaic image, although it is not so in each individual frame. The mosaic representation also allows to "erase electronically" an object from a video (e.g., in Fig. 6, the man is segmented out from the background mosaic image).

References

1. D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, 1982.
2. T.-J. Cham, and R. Cipolla. A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *Proc. Int'l Conf. on Computer Vision and Pattern Recognition CVPR'98*, pages 442–447, 1998.
3. C. Colombo, A. Del Bimbo, and P. Pala. Retrieval of commercials by video semantics. In *Proc. Int'l Conf. on Computer Vision and Pattern Recognition CVPR'98*, pages 572–577, 1998.
4. J.M. Corridoni and A. Del Bimbo. Structured digital video indexing. In *Proc. Int'l Conf. on Pattern Recognition ICPR'96*, pages (III):125–129, 1996.
5. M.A. Fischer and R.C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 24:381-395, 1981.
6. C.G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 147–151, 1988.
7. M. Irani, P. Anandan and S. Hsu. Mosaic based representation of video sequences and their applications. In *Proc. Int'l Conference on Computer Vision ICCV'95*, pages 605–611, 1995.
8. H.S. Sawhney and S. Ayer. Compact representations of videos though dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:814–830, 1996.
9. L.S. Shapiro, H. Wang and J.M. Brady. A matching and tracking strategy for independently moving, non-rigid object. In *Proc. British Machine Vision Conference*, pages 306–315.
10. J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Transaction on Image Processing*, 3(5):625–638, 1994.

This article was processed using the LaTeX macro package with LLNCS style