Pattern Analysis & Applications (2002)5:296–305

Layered Representation of a Video Shot with Mosaicing

F. Odone[1], A. Fusiello[2] and E. Trucco[3]

[1]*INFM – DISI, Università di Genova, Genova, Italy;* [2]*Dipartimento di Informatica, Università di Verona, Verona, Italy;* [3]*Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, UK*

Abstract: This paper presents a motion segmentation method useful for representing efficiently a video shot as a static mosaic of the background plus sequences of the objects moving in the foreground. This generates an MPEG-4 compliant, layered representation useful for video coding, editing and indexing. First, a mosaic of the static background is computed by estimating the *dominant motion* of the scene. This is achieved by tracking features over the video sequence and using a robust technique that discards features attached to the moving objects. The moving objects get removed in the final mosaic by computing the median of the grey levels. Then, segmentation is obtained by taking the pixelwise difference between each frame of the original sequence and the mosaic of the background. To discriminate between the moving object and noise, temporal coherence is exploited by tracking the object in the binarised difference image sequence. The automatic computation of the mosaic and the segmentation procedure are illustrated with real sequences experiments. Examples of coding and content-based manipulation are also shown.

Keywords: Content-based representation; Mosaicing; Motion segmentation; MPEG-4; Video coding; Video sequence analysis

1. INTRODUCTION

This paper presents a mosaic-based motion segmentation method for a layered, MPEG-4 compliant coding of video shots. A *video shot* is defined as an image sequence captured with a single operation of the camera and presenting a continuous action in time and space [1].

A compact representation of a video shot, useful for video compression [2], coding, editing [3] and indexing [1,4–6], is obtained by computing a mosaic of the background and sequences of the foreground moving objects. This representation achieves high compression rates in the transmission of the sequence, since all the information about the background (which does not change) is processed and sent only once. These ideas meet the requirements of the MPEG-4 standard [7], in which a scene is described as a composition of several *video objects*, encoded separately.

A *mosaic* is a panoramic image obtained by collating all frames of a sequence or set of images after aligning (warping) all the images onto a common reference frame. The result can be regarded as a *panoramic image* acquired by a virtual camera, especially useful where a single, real camera would limit resolution or could not be used at all [8,9]. Besides video compression, video coding and editing, and automatic indexing of video data, mosaicing techniques are useful for image stabilisation [10] and building high quality images with low-cost imaging equipments [11].

Mosaicing is based on the fact that, in some cases, two views of the same scene can be related by a non-singular linear transformation of the projective plane, called *homography* or *collineation*. This happens if the camera performs a pure rotation (like in a panning operation), or the scene can be well approximated by a single plane (that is, the depth range of the scene is small compared to the distance from the camera). By composing all the different homographies between subsequent frames, it is possible to obtain the transformations relating each image of the sequence to a reference frame, chosen arbitrarily.

The *motion segmentation* problem can be stated as follows: given a sequence of images, classify the pixels of each frame as moving either according to camera motion or independently. In many works [3,12–14], object segmentation is obtained by first compensating for camera motion, and then considering the residual motion. Usually it is assumed that the camera is responsible for the *dominant motion*, which is defined as the parametric motion characterising the majority of points of a sequence [15].

Received: 31 August 2000
Received in revised form: 18 April 2001
Accepted: 20 July 2001

In the remainder of this section, the proposed method is outlined, and it is related to previous work on video mosaicing and motion segmentation.

The structure of the rest of the paper is as follows: Section 2 reviews the background notions needed to understand the paper; in particular, the transformations between pairs of images are formalised, and assumptions we made on the scene being considered are specified. Section 3 describes how the transformation between pairs of images is computed, starting from a sparse set of correspondences. Section 4 focuses on mosaic construction, and Section 5 illustrates our mosaic-based motion segmentation method. Section 7 provides examples of MPEG-4 coding and video manipulation. Finally, Section 8 summarises the paper, and describes possible improvements.

1.1. Method Outline

We obtain sparse correspondences through the image sequence by automatically tracking distinctive points, which do not suffer from the aperture problem [16]. A global 2D motion model of the whole image is computed using the reliable motion information at feature points, while keeping the computational complexity low (e.g. by controlling the number of features). The dominant motion is obtained by calculating the homographies with a *robust technique* that treats the points belonging to moving objects as *outliers*. Assuming that camera motion is the dominant motion of the sequence, warping the images according to the inverse of the dominant motion yields a sequence where the background appears fixed (having compensated for camera motion), and the other objects (if any) are still moving. We paste the warped images into a single mosaic image, using the median operator to assign grey levels to the mosaic pixels: in this way, moving objects are removed, and a mosaic of the background is obtained. We achieve the actual segmentation of moving objects by thresholding the grey-level difference between the background and each frame of the sequence. The resulting binary image should represent the silhouettes of moving objects, but in practice it is noisy for several reasons: object or illumination changes, residual misalignments, interpolation errors during warping, and acquisition noise. To extract only relevant moving objects, we exploit temporal coherence by tracking the centroid of the moving object over the sequence.

1.2. Related Work

Direct minimisation of discrepancy in pixel intensities has been widely used to align images [2,3,8,10,13,14,17]. This technique is closely related to computing a dense approximation of the 2D motion field, i.e. the apparent motion of the image brightness pattern (the optical flow) [16].

Another approach to 2D motion estimation is known as *feature-based* [18,19], which identify and match local image as features, such as corners, and produce a sparse 2D motion representation. Sparse approximations are used whenever a low computational complexity is needed, and sparse but reliable results are enough to perform the task required.

Zoghlamy et al [19] developed an interesting corner-based method for 2D mosaic construction. They were mainly concerned with obtaining the best homography possible, therefore their first approach was to compute all the possible homographies obtainable by pairs of fourtuples of corners, and then to take the best one (i.e. the one which maximises a similarity function over all the corners). This approach was very accurate, but unfortunately time consuming. They limited the number of possible combinations of corners by representing each corner with a so-called *corner model* [21], which is richer of information and then easier to match reliably. In this case, though, the quality of the homography is not only affected by possible errors during corner extraction, but also by possible errors in the corner model computation.

Our approach, following Morimoto and Chellappa [21], is to track the motion of selected features, and then to compute a global transformation of the image, which will describe the evolution of each pixel. Although popular in other motion analysis applications, there seem to have been few attempts to use feature tracking techniques in mosaicing.

Whereas optical-flow techniques are dense both in time and in space, feature-based techniques, which are sparse in space, are also sparse in time, for they typically use frames with a moderate overlap and rely on feature matching. Instead, we use feature *tracking*, which is sparse in space but dense in time. This makes the feature matching fast and reliable, since the features do not change too much from one frame to the other, and the estimation of their motion is unambiguous, since they do not suffer from the aperture problem. The reason why we decided to use the whole of the sequence is because our segmentation approach, based on eliminating moving objects from the sequence, needs dense time information for the temporal median filtering to be effective.

Approaches to segmentation through camera motion compensation have been used in the field of surveillance and targeting [3,12–14]. Our work improves on closely related work in many respects. To register images, all those approaches are based on the computation of motion at each pixel, closely resembling optical flow techniques. As pointed out by Brunelli, Mich and Modena [1], such algorithms are 'currently too complex to be applied to large video databases'. On the other hand, we are looking for a low parametrical representation of the 2D motion, therefore it is sufficient, and advisable, to compute sparse 2D motion representation, using information only where it is most reliable.

As for the segmentation of moving objects, in Sawhney and Ayer [13], motion is computed at each pixel with a robust technique, and outlier masks correspond to the moving object. In Giaccone and Jones [3], temporal analysis of grey levels, based on probabilistic models and *a priori* information (user-initialised), is carried out in order to segment moving objects. Irani et al [14] use a local misalignment analysis based on the normal flow [15] to compare consecutive frames and extract moving objects.

Our segmentation method, based on image differences

and blob tracking, is less computationally expensive than that of Sawhney and Ayer [13], requires no user initialisation (unlike Giaccone and Jones [3]), and is more appropriate than image flow techniques [14], because of the strong spatiotemporal discontinuity caused by the disappearing moving object. Indeed, since we *first* compute a mosaic free from the objects in motion and *then* we compare it with each frame of the sequence, we can use effectively a simple technique like a pixel-wise difference, while in Irani et al [14], consecutive and hence similar frames were compared, and then a derivative-based comparison (like their *local misalignment analysis*) was needed.

2. BACKGROUND

A *homography* (or *collineation*) is a non-singular linear transformation of the projective plane [22] into itself. The most general homography is represented by a non-singular 3 × 3 matrix **H**:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1}$$

Points are expressed in homogeneous coordinates, that is, 2D points in the image plane are denoted as $\tilde{\mathbf{m}} = (x_1, x_2, x_3) = (x_1/x_3, x_2/x_3, 1)$ with $\mathbf{m} = (u, v) = (x_1/x_3, x_2/x_3)$ being the corresponding Cartesian coordinates.[1] In what follows, since we are modelling a transformation from one image to another, we consider a mapping from finite to finite points. This can be formalised assuming that $x'_3 \neq 0$: $\tilde{\mathbf{m}} = (x'_1, x'_2, x'_3) = (x'_1/x'_3, x'_2/x'_3, 1)$ with $\mathbf{m} = (u', v') = (x'_1/x'_3, x'_2/x'_3)$ being the point corresponding to (u, v). The matrix **H** has eight degrees of freedom, being defined up to a scale factor. The transformation is linear in projective (or homogeneous) coordinates, but it is *nonlinear* in Cartesian coordinates:

$$\begin{cases} u' = \dfrac{H_{1,1}u + H_{1,2}v + H_{1,3}}{H_{3,1}u + H_{3,2v} + H_{3,3}} \\ v' = \dfrac{H_{2,1}u = H_{2,2}v + H_{2,3}}{H_{3,1}u + H_{3,2}v + H_{3,3}} \end{cases}$$

Two images taken by a moving camera are related by a homography if the scene is planar or if the point of view does not change (the camera is rotating around its optical centre).

In general, it can be seen that two points **m** and **m′** that are the projection of the 3D point **w** onto the first and the second view, respectively, are related by

$$\kappa' \tilde{\mathbf{m}}' = \kappa \mathbf{A}' \mathbf{R} \mathbf{A}^{-1} \tilde{\mathbf{m}} + \mathbf{A}' \mathbf{t} \tag{3}$$

where **A** and **A′** are two 3 × 3 matrices encoding the *intrinsic* parameters (focal length, aspect ratio, image centre) of the left and right cameras, respectively, **R** is a 3 × 3 rotation matrix which gives the camera rotation between the two views, and **t** is a 3 × 1 vector representing the translation of the optical centre between the two views. κ and κ' are the distances of the 3D point from the first and second camera focal planes, respectively.

If the camera is rotating, then $\boldsymbol{t} = 0$, and we get

$$\frac{\kappa'}{\kappa} \tilde{\mathbf{m}}' = \mathbf{A}' \mathbf{R} \mathbf{A}^{-1} \tilde{\mathbf{m}} \tag{4}$$

The 3 × 3 matrix $\mathbf{H}_\infty = \mathbf{A}' \mathbf{R} \mathbf{A}^{-1}$ represents a homography, and does not depend upon the 3D structure. In the other case, if the camera undergoes a general rigid motion, but the scene points lie on a plane Π with Cartesian equation $\mathbf{n}^\top \mathbf{w} = d$, Eq. (3) can be specialised, obtaining:

$$\frac{\kappa'}{\kappa} \tilde{\mathbf{m}} = \left(\mathbf{H}_\infty + \frac{\mathbf{A}' \mathbf{t}\, \mathbf{n}^\top \mathbf{A}^{-1}}{\mathrm{d}} \right) \tilde{\mathbf{m}} \tag{5}$$

Therefore, there is a projective plane transformation between the two views induced by the plane Π, given by $\mathbf{H}_\Pi = \mathbf{H}_\infty + \mathbf{A}' \mathbf{t} \dfrac{\mathbf{n}^\top}{d} \mathbf{A}^{-1}$. **The** $\mathbf{H}_\infty$ homography, obtained in the previous case, can be interpreted as the homography induced by a very special plane, *the infinity plane*, as can be seen by letting $d \rightarrow \infty$ in Eq. (5).

In the general case (full 3D scene and arbitrary camera motion), the relationship between the two views can be cast in terms of a homography plus a *parallax* term [23], depending on the scene structure and camera translation. If the depth range of the scene is small compared to the distance from the camera, or the translation is small, then the parallax can be neglected.

3. HOMOGRAPHY COMPUTATION

Let us suppose that we are given an image sequence with a negligible parallax, and that point correspondences through the image sequence have been obtained by feature tracking. In this section we shall see how homographies are computed, and how to cope with moving objects.

3.1. Estimating a Unique Motion

Four points, provided that no three of them are collinear, determine a unique homography. Indeed, eight independent parameters are required to define the homography. Each point correspondence in the plane provides two equations in the unknown entries of **H**:

$$\begin{cases} u'(H_{3,1}u + H_{3,2}v + H_{3,3}) = H_{1,1}u + H_{1,2}v + H_{1,3} \\ v'(H_{3,1}u + H_{3,2}v + H_{3,3}) = H_{2,1}u + H_{2,2}v + H_{2,3} \end{cases} \tag{6}$$

It is then necessary to find at least four point correspondences to define the transformation matrix uniquely, up to a scale factor.

Equation (6) can be rearranged in matrix form. For $n \geq$

[1] We shall henceforth use the symbol ˜ above a vector to indicate homogeneous coordinates.

4 points, we obtain a rank-deficient system of homogeneous linear equations, which has the form $\mathbf{Lh} = \mathbf{0}$. If $n > 4$ there are more equations than unknowns, and, in general, only a least-squares solution can be found. Let $\mathbf{L} = \mathbf{UDV}^\top$ be the Singular Value Decomposition (SVD) [24] of $\mathbf{L}$. One Least Squares (LS) solution is the column of $\mathbf{V}$ corresponding to the least singular value of $\mathbf{L}$. The computational cost of SVD is $O(n^3)$.

As pointed out by Hartley for the problem of fundamental matrix estimation, the conditioning of the problem is dramatically improved by *data standardisation* [25]. The points are translated so that their centroid is at the origin, and then scaled so that the average distance from the origin is $\sqrt{2}$. Let $\mathbf{T}$ and $\mathbf{T}'$ be the resulting transformations in the two images, and $\tilde{\mathbf{m}}^\star = \mathbf{T}\tilde{\mathbf{m}}$, $\tilde{\mathbf{m}}'^\star = \mathbf{T}'\tilde{\mathbf{m}}'$ the transformed points. Using $\tilde{\mathbf{m}}^\star$ and $\tilde{\mathbf{m}}'^\star$ in the homography estimation algorithm, we obtain a matrix $\mathbf{H}^\star$ that is related to the original one by $\mathbf{H}^\star = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}$.

3.2. Estimating a Dominant Motion

In the case of a static scene with a moving camera, a least squares estimate could be accurate enough. However, when objects are moving in the scene, features attached to different objects move with different motions, and a single homography cannot cater for all of them. Therefore, a robust method must be employed to estimate the homography that explains the motion of the *majority* of the features, i.e. the *dominant motion*. Unless the scene is cluttered with many moving objects, this is usually the relative motion of the camera with respect to the background (as we are assuming).

We use Least Median of Squares (LMedS) [26], a robust regression technique which has been used in many computer vision applications [27,28]. The principle behind LMedS is the following: given a regression problem, in which d is the minimum number of points determining a solution (four, in our case), compute a candidate model based on a randomly chosen d-tuple from the data; estimate the fit of this model to *all* the data, measured by the median of the squared *residuals*; if the current fit is better than the previous one, update the homography; choose a new random d-tuple and repeat the previous steps. In our case, the residuals are defined, for each point correspondence, as the distances between the warped and the actual point in the second image. In formulae, let $\hat{\mathbf{H}}$ be an approximate solution of Eq. (6), then the residuals are

$$s_j = \|\mathbf{m}'_j - \hat{\mathbf{H}}\mathbf{m}_j\| \quad j = 1, \ldots, n \tag{7}$$

where n is the number of point correspondences.

The optimal model represents the majority of data. Data points that do not fit into this model are *outliers*. The *breakdown point*, i.e. the smallest fraction of outliers that can yield arbitrary estimate values, is 50%. Although, in principle, all the d-tuples should be evaluated, in practice a Monte Carlo technique is applied, in which only a random sample of size m is considered. Assuming that the whole set of points may contain up to a fraction ϵ of outliers, the probability that at least one of the m d-tuples consists of d inliers is given by [28]

$$P = 1 - (1 - (1 - \epsilon)^d)^m \tag{8}$$

In our case, $\epsilon = 0.5$, $P = 0.99$, and $m = 72$.

When Gaussian noise is present in addition to outliers, the *relative statistical efficiency* (i.e. the ratio between the lowest achievable variance for the estimated parameters and the actual variance) of the LMedS is low. To increase the efficiency, it is advisable [26] to run a weighted LS fit after LMedS, with weights depending on the residuals of LMedS S_j, $j = 1, \ldots, n$, as follows. First, a robust standard deviation estimate [26] is computed as

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n - d}\right) \sqrt{\operatorname{med}_j^2} \tag{9}$$

where d is the number of parameters (four in our case). Secondly, a weight is assigned to each point correspondence, such that

$$w_j = \begin{cases} 1 & \text{if } |s_j|/\hat{\sigma} \leq 2.5 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The computational cost of LMedS with Monte Carlo speed-up is $O(mn \log n)$. This technique works well under the following conditions:

1. The dominant motion is the relative motion of the camera with respect to the background, i.e. more tracked points are attached to the background than to moving objects.
2. The parallax for the background is negligible.

Residuals analysis, though, can tell whether the input sequence fulfills these requirements. In the latter case, results are meaningless. As in any data fitting problem, residual analysis gives a measure of the goodness of fit.

4. MOSAIC CONSTRUCTION

The mosaic construction is usually achieved by aligning the images of the sequence with respect to a common reference frame, and by blending them into a single mosaic image. Assuming that images can be transformed into each other by homographies, the alignment of all image frames in the sequence can be performed in the following ways:

- *Frame to frame:* homographies are first computed between successive frames for the entire sequence. Homographies can be composed to obtain the alignment homographies between *any* two frames of the sequence, and in particular, between the current image and the reference one.
- *Frame to mosaic:* homographies are computed between a temporary mosaic and the current frame. This homographies are directly used to update the temporary mosaic with the current warped frame.

In the first approach, a temporary mosaic can be built as soon as a new image of the sequence is processed, or else all the images of the sequence can be warped according to the homographies and only one mosaic is built, as a final step. With the second approach a temporary mosaic must be

upgraded at each step, because the homography is computed between it and the current frame.

Our method is based on the first approach, and the global registration precedes the actual mosaic construction. This is done to allow us to use the median operator to render the mosaic. In Section 5 this choice will be motivated in detail.

Our mosaic construction method is then accomplished in three stages, described in more detail in the remainder of this section: *motion estimation* along the sequence, *global registration* with respect to a common reference frame, and *blending* of the images into a single mosaic.

We use a 2D motion model, but more complicated 3D methods can be utilised when parallax is not negligible [13,14].

4.1. Feature-based Motion Estimation

A sparse approximation of the 2D motion field between adjacent images is obtained using a feature tracking technique. We chose the well known Shi–Tomasi–Kanade tracker [29–31], which selects optimal image features and tracks them as they move from frame to frame. Features are points where the image gradient is strong in two directions (corners). The tracker produces a list of corresponding points for each pair of images. New features are extracted as soon as there are too few left. Once a sparse 2D motion field is known, a global 2D motion model, i.e. homographies can be obtained. These homographies represent the 2D motion model of the sequence, specifying the frame-to-frame motion of each pixel.

4.2. Global Registration

Once all the homographies between pairs of adjacent images have been computed, we perform a global alignment of all the images with respect to a unique point of view.

Changing reference frame, the resulting mosaic changes, but the results of the segmentation and coding methods described in the next sections will not be affected. In most cases, we choose the frame of the first image of the sequence as a reference. To produce the global alignment, since the homography defined in Eq. (1) is a linear operator, the transformation between non-contiguous frames can be obtained by multiplying the transformation matrices of the in-between image frames. The transformation $H_{i,j}$ between images I_i and I_j, where $i < j$, is

$$H_{i,j} = \prod_{k=i}^{j-1} H_{k,k+1} \tag{11}$$

Equation (11) can be used to obtain the homography $H_{ref,i}$ between any sequence frame I_i and the reference frame I_{ref}.

4.3. Mosaic Rendering

Assume global alignment has been completed. If we imagine piercing all the aligned frames with a temporal line (see Fig. 1), we will intersect pixels that, ideally, correspond to the same world point. The grey level in each pixel of the mosaic will be computed by applying an appropriate temporal operator to the corresponding points. Several temporal operators can be used to construct the mosaic image. The most common are the *use-first*, the *use-last*, the *average* and the *median*. They all work on the intensity values belonging to the temporal line of each pixel. The use-first method adds to the mosaic only the parts of the current image that did not appear in the previous images. On the contrary, the use-last technique pastes each image, once it is warped, into the mosaic. The average is effective in removing temporal noise, but if the sequence contains objects in motion they will appear blurred, with 'ghost-like' traces in the resulting mosaic. The median operator removes temporal noise and also moving objects whose intensity patterns are stationary for less than half of the frames. The moving objects are treated as outliers. Other operators that can be found in the literature are the mode, weighted average and trimmed mean. Since we are interested in building a still mosaic of the background, a median-like temporal operator is appropriate.

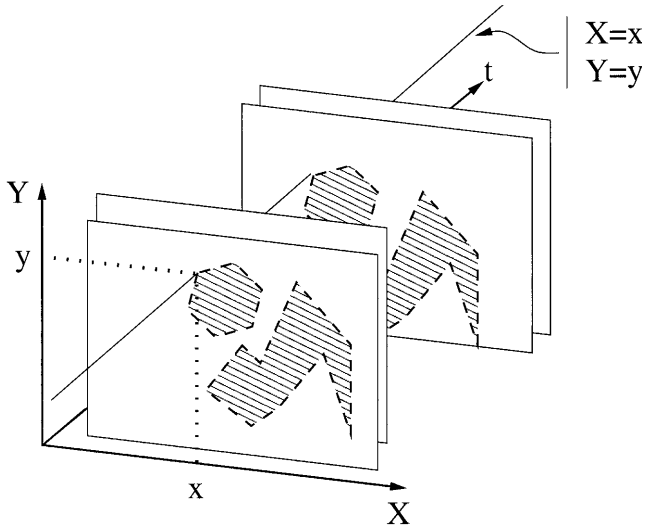

Fig. 1. Temporal alignment: once all the frames are aligned, a temporal straightline will intersect each image in corresponding points.

The method is clearly offline, since the median filter requires the whole image sequence, and it is effective in deleting the moving objects only if, for each pixel of the mosaic, the majority of the contributions come from the background.

5. SEGMENTATION OF MOVING OBJECTS

This section describes a method to segment moving objects in image sequences using a mosaic-based technique. After constructing the mosaic as described in the previous sections, moving objects are segmented out by computing the grey-level differences between the current frame and the background mosaic, where moving objects have been deleted. To this end, a synthetic sequence of the background is obtained by warping the final mosaic into each image sequence frame I_i using the inverse of the $H_{ref,i}$ homography.

The foreground is segmented by comparing each frame of the virtual sequence with the corresponding frame of the original one. This can be done using well-known techniques for change detection and motion segmentation [32,33,15]. Irani et al [14] perform motion analysis using a *local misalignment measure* [15]. Their method relies on temporal continuity between the frames compared, in our case the mosaic and the current frame. Indeed, a dynamic mosaic containing all the dynamic aspects of the video sequence [2] is updated at each step with all the information of the latest frame, and also with possible moving objects. This implies a temporal coherence between the mosaic and current frame. Instead, if the mosaic is not dynamic, this coherence no longer exists, since a moving object in the mosaic can be blurred or removed, as in our case. There is, thus, a strong spatio-temporal discontinuity between mosaic and frame that decreases the significance of the misalignment measure in itself. A difference-based technique therefore seems more effective for our purposes.

A grey-level difference is performed between each original frame and the equivalent virtual one, and the result is thresholded to obtain a binary map. The binary motion map obtained by differencing contains the blobs produced by the moving objects and other smaller blobs due to misalignments, or changes in illumination and noise. We assume that only one object is moving in the scene (a generalisation is currently under investigation). We detect the object in the first frame by choosing the area of the binary map containing the largest connected region of moving pixels. The centroid of this area is computed. The connected component chosen in the $(i+1)$th binary map is the closest one to the centroid of the previous step. At each step, the centroid needs to be updated. This is an elementary form of tracking with a zero-order prediction (i.e. with a constant position assumption), coupled with an elementary data association algorithm, namely the *closest neighbour* strategy [34].

Post-processing is also applied on the resulting maps, to improve segmentations. We use the morphological operator *closure* [35], (i.e. *dilation* and *erosion* in cascade) to produce more compact regions, without adding noise and without altering the original shapes.

6. RESULTS

This section shows some experimental results, obtained from video shots acquired with a commercial hand-held camcorder; no special set-up nor calibration were used. Figures 2 and 3 show selected frames of the 'Super5' and 'Manuel' sequences, respectively. The first sequence, 'Super5', is an outdoor scene with a car driving from the left to the right of the field of view. The camera motion is mostly rotational, with a small translational component. In the sequence, 'Manuel' the object (a person) in motion is bigger, and a lot of shadows are present. The depth of the scene changes throughout the sequence.

Figures 4 and 5 show the mosaics of the background obtained with the method explained in Section 4. In spite of the fact that the camera motion is not exactly rotational and the scene not planar, the registration obtained is very satisfactory. Note also that moving objects have been *automatically* removed without artifacts.

Figure 6 shows the results of residual analysis, performed between each frame of the original sequence and the background (mapped onto the same frame). Figure 6 (left) shows the results obtained by using a thresholded difference between the 28th frame of the sequence 'Manuel' and its background. Figure 6 (right) shows the results obtained with the local misalignment analysis described in Irani et al [14]. This shows well that, as pointed out in Section 5, differences are more suitable to our purposes than the local misalignment analysis.

Figure 7 illustrates results of segmentation, showing selected frames of the foreground sequences. The moving object in 'Manuel' is not as sharp as in 'Super5', yet the quality of segmentation is still satisfactory.

More examples and sequences are available at: http://www.cee.hw.ac.uk/~franci/mosaic_demo/mosaic.html

7. APPLICATION EXAMPLES

Our technique produces a layered representation of a video sequence, which is useful for automatic indexing of video data, video coding and video editing. We provide here examples of MPEG-4 coding and editing.

7.1. Video Coding

The last MPEG standard, MPEG-4 [2], relies on a layered representation of the video data. A scene is considered to be composed of several Video Objects (VOs). Each VO is characterised by *intrinsic properties* such as shape, texture and motion. In this context, 'object' has a very general interpretation, and it is not necessarily a physical object. For example, the background region may be considered as one VO. A *sprite* consists of those regions of a VO that are present in the scene throughout the whole video segment. An obvious example is the 'background sprite', i.e. the mosaic of the background in a camera-panning shot.

Notice that the MPEG-4 standard does not prescribe the method for creating VOs; it simply provides a standard convention for describing them, so that all compliant decoders are able to extract VOs from an encoded bit stream.

If we think of the mosaic of the background and the foreground sequence as VOs, the idea described in the previous sections can be seen as an MPEG-4 compliant content-based encoding technique. A mosaic of the background of a video sequence is built, and moving objects are segmented. The background sprite is transmitted to the receiver only once. The moving foreground object is transmitted separately as a separate VO, its position being described in the mosaic reference frame. All transformations between mosaic and original sequence are also needed; actually, it is sufficient to transmit all the homographies between consecutive frames, which allows us to relate any two frames

Fig. 2. Frames 0, 20, 40 from the 'Super5' sequence.

Fig. 3. Frames 0, 50, 99 from the 'Manuel' sequence.

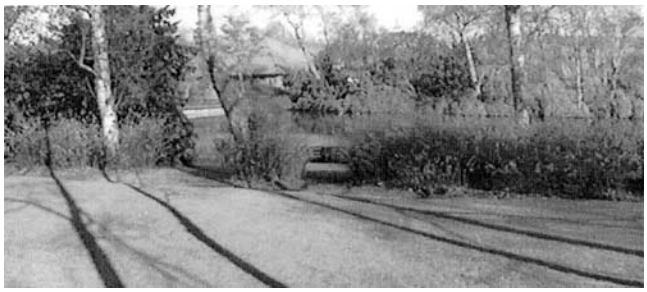

Fig. 4. Mosaic of 'Manuel' (background sprite).

Fig. 5. Mosaic of 'Super5' (background sprite).

in sequence. When decoding, to rebuild the original sequence, all we have to do is map the mosaic onto the frame of each image and paste the foreground onto it.

To assess our coding technique, we encoded and decoded the 'Super5' and 'Manuel' sequences and compared the result with the original one. Figure 8 (left) is a frame of the coded-decoded 'Super5' sequence, whereas Fig. 8 (right) visualises the differences between the same frame and the original one. As an image quality measure we computed the *Peak Signal-to-Noise Ratio* (PSNR) between the original sequence and the coded-decoded one. Let I, $N \times M$ be the original image, and J the result of encoding and decoding I:

$$\mathrm{PSNR}(I,J) = 20 \log_{10} \frac{255}{\left(\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (I(i,j) - J(i,j))^2 \right)^{1/2}} \tag{12}$$

The plots in Fig. 9 show that the coding/decoding does not degrade too much throughout the sequence.

7.2. Video Editing

This section describes a particular content-based manipulation of a video sequence, in which the layered representation is exploited to add a synthetic object (an advertising poster), to the background.

We first synthesise a fronto-parallel view of the background plane from the mosaic. This is known as *metric rectification* [36] of a perspective image. A 3D plane and its perspective image are related by a homography, which is fully defined by the relative position of four points in the world plane, specified by the operator. Once the homography is determined, the image can be backprojected onto the object plane. After inserting the synthetic object into the rectified mosaic, the mosaic is warped back onto its original plane. Then we use the decoding procedure described in Section 7.1 to create a new sequence with the modified background.

Figure 10 presents an example of video editing, where the 'Heriot-Watt University' sign is inserted into the background of the original sequence.

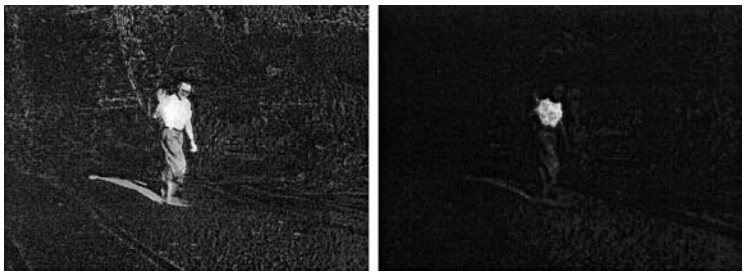

Fig. 6. Residual analysis with differences (left) and normal flow (right).

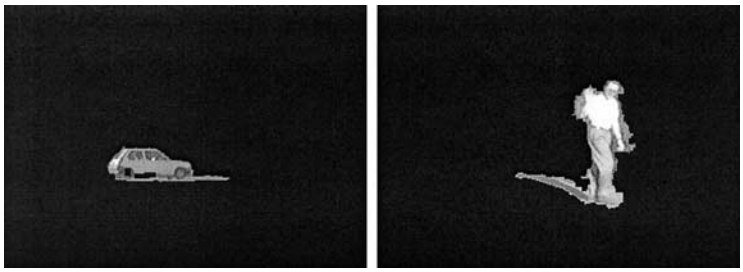

Fig. 7. Example of moving objects extracted from the sequences 'Super5' and 'Manuel'.

Fig. 8. Example of a frame from encoded/decoded 'Super5' and differences with the original one (right).

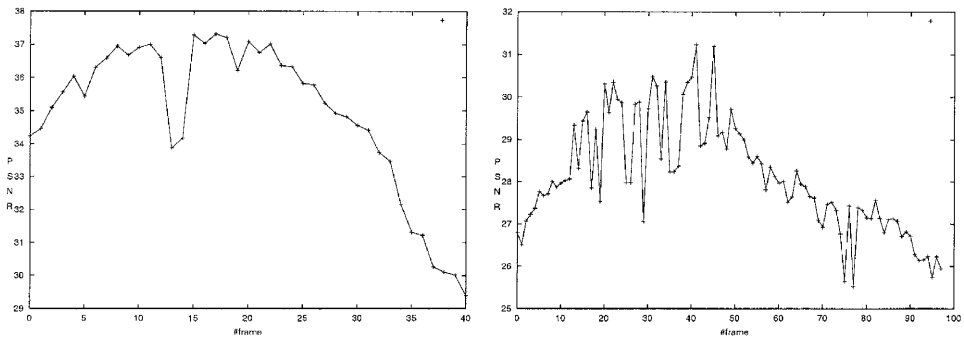

Fig. 9. Peak signal to noise ratio of the decoded videos: 'Super5' on the left and 'Manuel' on the right.

8. CONCLUSIONS

This papers described a mosaic-based motion segmentation method that can be used to get a layered representation of video sequences in terms of static background plus moving foreground objects.

Motion estimation was performed on a sparse (in space) set of features to obtain fast and reliable results, but on a dense (in time) sequence of images, to fulfill the requirements of the motion segmentation method devised. Features were tracked over the sequence, and a robust technique allowed us to discard features attached to the moving objects. Once a sparse estimation of the motion field was available, an approximation of a global transformation from each

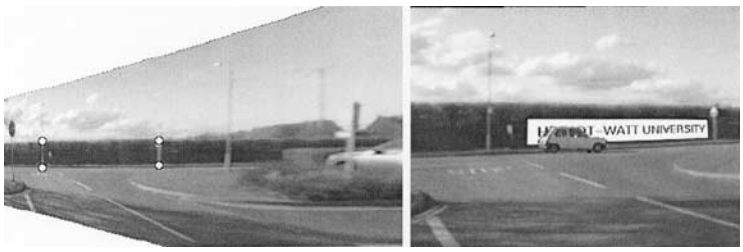

Fig. 10. On the left the metrically rectified mosaic of the sequence 'Super5': the four points that have been used to compute the homography are highlighted. On the right, a sample frame of the synthetic advertisement sequence.

pixel of one image to each pixel of the next image could be computed.

Once the motion had been estimated, all the frames of the sequence were aligned accordingly. Using the median as the grey levels blending operator, the moving objects were removed in the final mosaic. Then, segmentation was obtained by taking the pixel-wise difference between each frame of the original sequence and the mosaic of the background. To discriminate between the moving object and noise, temporal coherence was exploited by tracking the object silhouette in the binarised difference image.

At present, we assume that only one object is moving in the scene, but further work will address multiple object tracking with data association [34]. We reckon that it could be done without changes to the present structure of the algorithm.

The segmentation technique described fits perfectly into the MPEG-4 standard for video coding. A number of experiments have been carried out to verify the quality of the sequences obtained after decoding. Very promising results have been obtained, where image quality is well preserved throughout the sequence.

Acknowledgements

We wish to thank Francesco Isgrò for the useful comments he made during the course of this work, and Costas Plakas for providing the feature tracker code. Francesca Odone was supported by a 'Marie Curie' Training and Mobility Grant (ERB4001GT-97–3072), Andrea Fusiello by a EPSRC Visiting Fellowship (GR/M40844) and by a grant from the University of Verona (*Progetto Giovani Ricercatori*).

References

1. Brunelli R, Mich O, Modena CM. A survey on the automatic indexing of video data. Journal of Visual Communication and Image Representation 1999; 10: 78–112
2. Irani M, Anandan P, Hsu S. Mosaic based representations of video sequences and their applications. International Conference on Computer Vision 1995: 605–611
3. Giaccone PR, Jones GA. Segmentation of global motion using temporal probabilistic classification. British Machine Vision Conference 1998: 619–628
4. Baldi G, Colombo C, Del Bimbo A. Automatic video representation using mosaicing. Proceedings of the Joint Workshop of AI*IA and IAPR-IC, Ferrara, April 1998: 152–157
5. Chang SF, Chen W, Meng HJ, Sundaram H, Zhong D. VideoQ: An automated content based video search using visual cues. Fifth ACM Multimedia Conference, Seattle, November 1997
6. Irani S, Hsu M, Anandan P. Video indexing based on mosaic representations. Proceedings of the IEEE 1998; 86(5): 905–921
7. Koenen R, Pereira F, Chiariglione L. MPEG-4: Context and objectives. Signal Processing: Image Communications 1997; 9(4): 295–304
8. Szeliski R. Video mosaics for virtual environments. IEEE Computer Graphics and Applications 1996; 16(2): 22–30
9. Trucco E, lot YR, Tena Ruiz I, Plakas K, Lane DM. Feature tracking in video and sonar subsea sequences with applications. Computer Vision and Image Understanding, (special issue on 'Underwater Computer Vision and Pattern Recognition') 2000
10. Hansen M, Anandan P, Data K, Wal G, Burt P. Real-time scene stabilization and mosaic construction. Proceedings of IEEE Workshop on Applications of Computer Vision 1994
11. Capel D, Zisserman A. Automated mosaicing with super-resolution zoom. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1998; 885–891
12. Cohen I, Medioni G. Detecting and tracking moving objects in video surveillance. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1999; II: 319–325
13. Sawhney H, Ayer S. Compact representations of videos through dominant and multiple motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996; 18(8): 814–830
14. Irani M, Anandan P, Bergen J, Kumar R, Hsu S. Efficient representations of video sequences and their applications. Signal Processing: Image Communication 1996; 8(4): 327–351
15. Irani M, Rousso B, Peleg S. Computing occluding and transparent motions. International Journal of Computer Vision 1994; 12(1): 5–16
16. Trucco E, Verri A. Introductory Techniques for 3-D Computer Vision. Prentice-Hall, 1998
17. Rousso B, Peleg S, Finci I, Rav-Acha A. Universal mosaicing using pipe projection. International Conference on Computer Visione (ICCV98) 1998
18. Dani P, Chaudhuri S. Automated assembling of images: Image montage preparation. Pattern Recognition 1995; 28(3): 431–445
19. Zoghlami I, Faugeras O, Deriche R. Using geometric corners to build a 2D mosaic from a set of images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1997: 420–425
20. Deriche R, Blaszka T. Recovering and characterizing image features using an efficient model based approach. Proceedings of the International Conference on Computer Vision and Pattern Recognition 1993: 530–535
21. Morimoto C, Chellappa R. Fast 3D stabilization and mosaic construction. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1997: 660–665
22. Semple JG, Kneebone GT. Algebraic Projective Geometry. Oxford University Press, 1952
23. Shashua A, Navab N. Relative affine structure: Canonical model

for 3D from 2D geometry and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996; 18(9): 873–883
24. Golub GH, Van Loan CF. Matrix Computations. The John Hopkins University Press, 1996
25. Hartley RI. In defence of the 8-point algorithm. Proceedings of the IEEE International Conference on Computer Vision 1995
26. Rousseeuw PJ, Leroy AM. Robust Regression & Outlier Detection. Wiley, 1987
27. Meer P, Mintz D, Kim DY, Rosenfeld A. Robust regression methods in computer vision: a review. International Journal of Computer Vision 1991; 6: 59–70
28. Zhang Z. Parameter estimation techniques: a tutorial with application to conic fitting. Image & Vision Computing 1997; 15(1): 59–76
29. Shi J, Tomasi C. Good features to track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1994: 593–600
30. Tomasi C, Kanade T. Detection and tracking of point features. Technical Report CMU-CS-91–132, Carnegie Mellon University, Pittsburg, PA, April 1991
31. Fusiello A, Trucco E, Tommasini T, Roberto V. Improving feature tracking with robust statistics. Pattern Analysis and Applications 1999; 2(4): 312–320
32. Hsu YZ, Nagel HH, and Rekers G. New likelihood test methods for change detection in image sequences. Computer Vision, Graphics, and Image Processing 1984; 26: 73–106
33. Huang TS. Image Sequence Analysis. Springer-Verlang, 1981
34. Bar-Shalom Y, Fortmann TE. Tracking and Data Association. Academic Press, 1988
35. Serra J. Image Analysis and Mathematical Morphology. Academic Press, 1982
36. Liebowitz D, Zisserman A. Metric rectification for perspective images of planes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1998: 482–488

Francesca Odone received a laurea cum laude (MSc) in Computer Science in 1997 from the Università di Genova, Italy. She is a PhD student in Computer Science at the Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Italy. In 1997 she worked as a Research Assistant at the Computer Vision Laboratory of Heriot-Watt University, Edinburgh, UK. She is currently visiting Heriot-Watt University, Edinburgh, UK, supported by a two years Marie Curie training and mobility research grant. Her present research interests are in 3D computer vision and machine learning with application in industrial environments. Further information can be found at http://www.disi.unige.it/person/OdoneF.

Andrea Fusiello received his Laurea (MSc) degree in Computer Science from the Università di Udine, Italy in 1994 and his PhD in Information Engineering from the Università di Trieste, Italy in 1999. He worked with the Computer Vision Group at IRST (Trento, Italy) in 1993–94, and with the Machine Vision Laboratory at the Università di Udine from 1996 to 1998. He was a Visiting Research Fellow in the Department of Computing and Electrical Engineering of Heriot-Watt University (UK) in 1999. As a Research Associate, he is now with the Dipartimento di Informatica, Università di Verona. He has published papers on real-time task scheduling, autonomous vehicles navigation, stereo and feature tracking. His present research is focused on 3D computer vision, with applications to underwater robotics. He is a member of the International Association for Pattern Recognition (IAPR) and IEEE. Further information can be found at http://www.sci.univr.it/~fusiello.

Emanuele Trucco received his Laurea cum laude (MSc) in 1984 and the Research Doctorate (PhD) degree in 1990 from the University of Genoa, Italy, both in Electronic Engineering. Dr Trucco has been active in machine vision research since 1984, at the EURATOM Joint Research Centre of the Commission of the European Communities (Ispra, Italy), the Universities of Genoa and Edinburgh, and as a consultant. He is currently a Senior Lecturer in the Department of Computing and Electrical Engineering of Heriot-Watt University. His current research interests are in 3D machine vision and its applications, particularly to telepresence and underwater robotics. Dr Trucco has published more than 70 refereed papers and a book on 3D vision with A. Verri. He is a member of IEEE, the British Machine Vision Association (BMVA), AISB, and a committee member of the British Machine Vision Association (Scottish Chapter). Further information can be found at http://www.cee.hw.ac.uk/~mtc/mtc.html

Originality and Contribution

Our work improves on closely related ones in many respects. In order to register images, all those approaches are based on the computation of motion at each pixel, closely resembling optical flow techniques. As pointed out by Brunelli, Mich and Modena, such algorithm are "currently too complex to be applied to large video databases". On the other hand, we are looking for a low parametrical representation of the 2D motion, therefore it is sufficient, and advisable, to compute a sparse 2D motion representations, using information only where it is most reliable.

As for the segmentation of moving objects, in motion is computed at each pixel with a robust technique, and outliers masks correspond to the moving object. In temporal analysis of grey levels, based on probabilistic models and a priori information (user-initialized), is carried out in order to segment moving objects. Irani et al. use a local misalignment analysis based on the normal flow to compare consecutive frames and extract moving objects.

Our segmentation method, based on image differences and blob tracking, is less computational expensive than Sawhney and Ayer [1] requires no user initialization (unlike Giaccone and Jones [2]), and is more appropriate than image flow techniques, because of the strong spatio-temporal discontinuity caused by the disappearing moving object. Indeed, since we first compute a mosaic free from the objects in motion and then we compare it with each frame of the sequence, we can use effectively a simple technique like a pixel-wise difference, while in Irani et al. [3], consecutive and hence similar frames were compared and then a derivative-based comparison (like their it local misalignment analysis) was needed.

Correspondence and offprint requests to: Francesca Odone, DISI Università di Genova, DISI, Via Dodecaneso 35, 16146 Genova, Italy.