Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Signal Processing: *Image Communication* 18 (2003) 465–481

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

Panoramic view construction

M. Traka[a,b], G. Tziritas[a,b,*]

[a] *Institute of Computer Science—FORTH, Heraklion, Greece*
[b] *Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece*

Abstract

In this paper, the problem of constructing the whole view of a scene background from an image sequence is considered. First, point or block correspondence between each pair of successive frames is determined. Three parametric motion models are used: 2-D translation with scale change, affine, and projective. Motion parameters are estimated using either robust criteria and the Levenberg–Marquardt algorithm, or affine moment invariants. Then the parametric models are composed and all the frames are aligned, yielding a whole view of the scene background. A new technique is introduced for the correction of accumulated frame alignment errors.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Mosaic construction; Affine motion; Projective motion; Alignment correction

1. Introduction

During the past few decades, developments in audiovisual technology have given rise to new applications involving the processing and exploitation of video information, such as digital libraries. Moreover, the ever-increasing volume of data transmitted over the World Wide Web (WWW) has led to the need for fast on-line access to visual information, including video. However, their effective utilization is still limited because of certain detracting factors, primarily the cost of storage and the transmission time, consequences of the large number of video frames and their size. Furthermore, the lack of content-based indexing, intelligent searching and querying tools lead to partial exploitation of video information. One solution is to find an efficient visual representation of the video scenes, which will facilitate the search, recognition and indexing of objects through queries with visual attributes [5]. A video shot could be represented by key frames [2]. A key frame is a representative frame in a shot, typically the first, middle or last frame or a combination of them. In order to search and retrieve objects, the queries could be performed only on key frames. However, this representation is too poor, since a single frame, the key frame, cannot always represent the contents of all frames in a shot.

Even with rapid delivery of video and efficient querying and indexing tools for visual content searching and retrieval, the lack of efficient ways of representing the video content and inefficient interactive manipulation and video editing may still inhibit the widespread use of video information. The video editing process involves the insertion and removal of objects into the video

*Corresponding address. Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece. Tel.: +30-810-39-3136; fax: +30-810-39-3501.

E-mail address: tziritas@csd.uoc.gr (G. Tziritas).

0923-5965/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0923-5965(03)00032-8

sequence. The video manipulation process may call for the synthesis of some new views of the scene, corresponding to a desired viewing position. Currently these processes are very tedious, as they are done manually frame-by-frame.

There has been a growing interest in the use of panoramic images, called “mosaics”, as an efficient way of representing video shots [12]. The mosaic image is constructed from all frames comprising a video shot, giving a panoramic view of the whole background. The mosaic image construction is based on successive frames which overlap usually by a substantial amount. This image may be used for querying based on static image features, like colour, texture and shape of surfaces or objects. Furthermore, video editing can be performed on the constructed mosaic image rather than on all the frames in the video sequence. Benefits of this approach are the increased efficiency and the reduced temporal cost of the process.

The mosaic construction involves two steps: (1) the alignment of the frames in the sequence and (2) the composition of these frames in order to create the mosaic image. Frame alignment is achieved by motion estimation between successive frames of the sequence or between each frame and the mosaic image incrementally constructed from the previous frames.

Irani et al. [6] defined and described various types of mosaic representations, in particular the *static mosaic* considered in our work, which operates in batch mode by aligning all frames to a fixed coordinate system. Two 2-D motion models are used, the affine and the eight-parameter quadratic, wherein frame alignment is obtained by direct frame registration. 3-D alignment is also considered. Various temporal filtering techniques are employed for the frame integration leading to the mosaic construction. In [12], a method for model-based robust dominant motion estimation is presented using direct image registration. A 3-D model is also considered, defined by a 12-parameter transformation and a point-wise projective depth. In addition, the simultaneous estimation of multiple motions is addressed using an appropriate mixture model. The experimental results of parametric motion estimation are illustrated in video mosaic constructions. In [13], the global consistency of the successive frame-to-frame alignments is obtained using the frame topology. The topology is determined after local coarse image registration. Szeliski [15] uses a projective transformation which is identified by direct image registration. The Levenberg–Marquardt iterative minimization algorithm is employed in order to identify the motion model and construct either a planar or a cylindrical view of the scene. In [14], a technique for long-term global motion estimation is proposed and a hierarchical strategy is applied for parameter estimation. In addition, a closed-loop prediction is adopted for avoiding error accumulation. A method for dynamic mosaicking has been proposed by Nicolas [8].

The remainder of the paper is organized as follows. Section 2 presents the models used for characterizing camera motion, the techniques employed for estimating 2-D displacement vectors and methods for robust motion parameter estimation. We first extract a sparse but validated 2-D displacement vector field for achieving the required robustness in parametric camera motion estimation. Our approach is particularly well-suited to the case of large motion between successive frames or to the case of less textured images. The presence of independently moving objects does not influence the accuracy of camera motion estimation. In addition, we introduce an affine model estimation based on moment invariants. In Section 3, the relations for the frame alignment and the mosaic construction are presented. Two models are considered: the affine and the projective model. The problem of accumulating errors leading to a possible misalignment is also addressed, and a technique is introduced for halting the error propagation and correcting errors of this type. Finally in Section 4, results are shown for four real video sequences and conclusions based on the experiments are given.

2. Camera motion estimation

The frame alignment is based on camera motion estimation. The camera is also called a dynamic observer, since it does not remain stationary during its capture, but rather moves unrestrictedly in a 3-D space. This motion is well known as "egomotion". One of the most difficult tasks in motion analysis is to estimate the camera motion. Various camera movements have to be considered. The camera may pan, translate parallel to the image plane, tilt in a certain direction, zoom, rotate around one of the three axes or undergo any combination of these motion types. Numerous parametric models have been introduced to describe some of the above motions, as they are projected into 2-D space, like the image plane. The parametric motion, which is estimated using a parametric model, represents the motion of a dominant surface in the scene, usually the background scene.

2.1. Parametric models

Mann and Picard [7] present, describe and qualitatively compare various 2-D motion parametric models. They conclude that using a small number of parameters the projective model is the most accurate. From all of these in our implementation three different parametric models are used. Each of the models represents a different kind of camera motion. The parametric models express a coordinate transformation which maps the image coordinates $p = (x, y)$ to a new set of coordinates $p' = (x', y')$. The set of the implemented transformations is represented in Table 1.

In the simplest motion parametric model, it is assumed that the camera translates parallel to the image plane. Although translation is the least constraining and simplest of all the motion types to implement, it is poor at capturing large changes due to camera zoom, rotation, pan, and tilt.

Table 1
Coordinates transformations

Model	Coordinates transformation	Parameters
Translation	$\mathbf{p}' = \mathbf{p} + \mathbf{b}$	$\mathbf{b} \in \mathbb{R}^2$
Zoom and translation	$\mathbf{p}' = q\mathbf{p} + \mathbf{b}$	$q \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^2$
Affine	$\mathbf{p}' = \mathbf{A}\mathbf{p} + \mathbf{b}$	$\mathbf{A} \in \mathbb{R}^{2\times 2}$, $\mathbf{b} \in \mathbb{R}^2$
Projective	$\mathbf{p}' = \dfrac{\mathbf{A}\mathbf{p} + \mathbf{b}}{\mathbf{c}^{\mathrm{T}}\mathbf{p} + 1}$	$\mathbf{A} \in \mathbb{R}^{2\times 2}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^2$

After the simple 2-D translation, the next simplest transformation is the zooming-translation model. In this form of transformation, the camera zooms and translates parallel to the image plane. The scale in each of the image coordinates x and y is isotropic, meaning that the magnification in the x and y directions is the same. The isotropic scale is described by parameter q, while the translation in the x and y directions is described by vector $\mathbf{b}$.

The affine model is more general and involves six scalar parameters. It assumes a planar surface and an orthographic projection into the image plane. The affine model accurately describes zoom, translation in the x and y directions and pure shear. The scale in the two directions (x and y) may be anisotropic. The parameter $\mathbf{b}$ expresses the two-dimensional translation vector and the $\mathbf{A}$ matrix describes the anisotropic scale and shear in the two directions.

The projective model is the most general used in our work and sometimes the most efficient model. It involves eight scalar parameters, and it describes many possible camera motions for a planar scene with perspective projection. The projective model describes all possible camera motions (3-D translation and rotation) for a planar scene. Therefore, it can be used for any 3-D surface provided the objects in the scene are sufficiently far away from the camera. It can be also used for any 3-D object, if the camera movement is restricted to rotation and zoom.

Comparing the above three parametric models, we conclude that the most complicated is not always the most efficient. The parametric model with the largest number of parameters is usually prone to less precise computation of the parameters, when compared with other models involving fewer parameters. Each of the models however is sufficient and also efficient for certain kinds of camera motions.

The parametric model estimation is based on the accurate computation of a displacement vector for a number of image points. The estimated displacement vector field usually contains errors. On the other hand, the reliability of the resulted parameters depends on the number of points in which the displacement vector field is accurate. The total number of point correspondences must be sufficiently large.

The erroneous data in a displacement vector field are called outliers. These are image points which are not consistent with the dominant camera motion, and are usually due to noisy measurements or are points belonging to independently moving objects. Using these data in parametric model estimation leads to an incorrect camera motion estimation. Therefore, the discrimination and removal of these data from the set of data used for the estimation are necessary. The correct data in a displacement vector field are called inliers. These are points which are consistent with the dominant motion.

We approach the problem of correct motion field estimation by implementing two methods with appropriate criteria in order to separate and remove the outliers. We formulate the problem of dominant motion estimation as that of model-based robust maximum likelihood estimation (M-estimation) with direct methods or using affine moment invariants.

2.2. 2-D motion field estimation

In this section, we shall present two methods for displacement vector field estimation. We shall also refer to the set of criteria that are used in order to discriminate the outliers from inliers. Both methods are based on spatio-temporal variation of intensity.

2.2.1. Block matching method

The first method for displacement vector estimation is the well-known block matching method with sub-pixel accuracy. The criterion to be minimized is the average absolute value of the displaced frame difference, and a typical value for the block size is 16. In order to reduce the computational complexity required by the block matching method, a technique called "increasing accuracy search" is implemented [18]. At first the algorithm searches in an area with low accuracy. The algorithm is iterated with successively increasing accuracy until sub-pixel (equal to $\frac{1}{2}$ pixel in our implementation) accuracy is obtained. Since the displacement vector has sub-pixel accuracy, the intensity involved in the criterion has to be interpolated. For this purpose, the bi-linear interpolation from the four nearest points is used.

A number of extra criteria are used in order to achieve correct displacement vector estimation. The role of these criteria is to separate outliers from inliers and remove the outliers. The set of criteria is:

- Removal of blocks with uniform intensity values (typically with variance less than 30). As the displacement vector estimation is based on the difference of intensity values, the motion estimation within these blocks may be wrong.
- Removal of corresponding blocks with a large displaced frame difference. A typical value for the threshold on the average of the absolute value of the displaced frame difference is 15.
- Removal of blocks with a displacement vector that differs excessively from the average value of displacement vectors of all blocks. These blocks usually belong to independently moving objects. The threshold depends on the amount of zoom and is typically set for both displacement components to 3 for taking into account the zoom component, or to 1 in case of pure translation or panning.
- Smoothing the difference of the displacement vector of each block according to neighbouring displacement vectors of other blocks. The process of smoothing is achieved by applying a filter with weights in a neighbourhood of eight blocks. The filter that is used is

$$u_s(m,n) = \frac{\sum_{k=-1}^{1}\sum_{l=-1}^{1} \Phi(u(m-k,n-l)-u(m,n))h(|k|+|l|)u(m-k,n-l)}{\sum_{k=-1}^{1}\sum_{l=-1}^{1} \Phi(u(m-k,n-l)-u(m,n))h(|k|+|l|)}, \tag{1}$$

where $h(0) = 4, h(1) = 2, h(2) = 1$, and $\Phi(\cdot) = 0$, if the block has no measure, or the difference is more than 1 pixel, and $\Phi(\cdot) = 1$, if the difference is at most one pixel. In Eq. (1), $u(m,n)$ is the initial value of one displacement component at pixel (m,n), while the smoothed value is $u_s(m,n)$.

In order to achieve correct motion field estimation, we use the neighbouring motion estimation as an initial value for the current block motion estimation. The initial value is corrected by applying the block matching method in a smaller search area. This criterion is intended to prevent the neighbouring blocks from having large difference with the estimated displacement vector. In any neighbourhood of blocks, the displacement vectors usually have similar values.

2.2.2. Corner correspondence

A corner detector is first applied to each image to extract feature points of high curvature. A correlation technique is then used to establish candidate correspondences between two images. In our implementation, we use the corner detector described in [16]. Consider a generic image point p and a matrix $\mathbf{C}$, defined as

$$\mathbf{C} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}, \tag{2}$$

where the summations on the two components (I_x, I_y) of the image gradient are performed on a local window, typically of size 7 × 7, around the considered point p. A corner is identified by two strong edges, which are characterized by the eigenvalues λ_1 and λ_2. A corner is a location where both eigenvalues are large enough (typical values range from 1000 to 3000 for the above window size).

The matching of feature points which are already extracted is established through a correlation technique that differs somewhat from the method described in [19]. For each feature point p_1 of the first image, we use a correlation window of size $(2M+1) \times (2N+1)$ centred at point p_1. A rectangular search area of size $(2d_u+1) \times (2d_v+1)$ is also used, centred at the same point (p_1) in the second image. The correlation process is applied between the correlation window of point p_1 and the corresponding window of each feature point lying within the search area in the second image. The distance criterion used in the correlation process is the squared intensities difference between the corresponding windows of two feature points,

$$D(p_1, p_2) = \frac{1}{(2M+1) \times (2N+1)} \sum_{m=-M}^{M} \sum_{n=-N}^{N} \times (I_1(x_1+m, y_1+n) - I_2(x_2+m, y_2+n)^2. \tag{3}$$

If $D(p_1, p_2)$ does not exceed a threshold, typically about 25, the corresponding feature points are defined as candidates matches. In order to establish correct feature points correspondence, we apply an extra criterion such as the one given in [19]. The correlation coefficient is the second criterion that is used. In our application, we accept the candidates' corresponding feature points only if the correlation coefficient is very close to 1.

After this process, it is possible for a feature point in the first image to have more than one candidate match in the second image and vice versa. In order to avoid the many-to-one correspondence between two images we apply the following process. If several points in the first image are found to correspond to a single point in the second image, we accept as candidate corresponding feature point the point in the first image and the point in the second image which give the largest value of correlation coefficient. It is also clear that, if we reverse the image roles, taking the first image as the second and vice versa, the same pair of points will be selected. Therefore, the most similar feature points in two images are accepted as corresponding features.

2.3. Parametric model estimation

After the block or corner matching process, a number of correspondences between the two images is obtained. Some of these correspondences are correct, without being consistent with dominant motion. Our goal is to estimate the parameters of the motion model, which describe the dominant motion of the scene, using the previously computed point correspondences. One of the most popular methods which recover the structure that best fits the majority of the data, while identifying and rejecting "outliers" or "deviating substructures", is the robust estimation method. In our work, we use the M-estimation technique as the most suitable among all robust techniques for solving this form of problem [4]. The second method proposed in our work, which achieves motion model estimation, is based only on moment invariants. The presentation and analysis of these methods follows.

2.3.1. Robust estimation of a motion model

The M-estimator addresses the problem of finding the values for parameters (θ) of one of the motion models (Table 1) that best fits the majority of the data. The data set is the result of motion field estimation. Therefore, the data are the pairs of corresponding points of the two images (p'_i, p_i). Given two images I_1 and I_2, after the 2-D motion field estimation method a set of vectors (p'_i, p_i) is obtained, where p_i is a 2-D vector of image I_1 coordinates and p'_i is the corresponding 2-D vector of image I_2 coordinates. It is assumed that the camera motion is modeled according to one of the transformations (T) already presented. Then the point correspondence of the two images according to transformation T is

$$p'_i(\theta) = T(p_i),$$

where $p'_i(\theta)$ is the point in image I_2 corresponding to p_i as determined by the transformation T, while p'_i is the point in image I_2 as already determined by the displacement vector estimation method.

In the M-estimation formulation, the unknown parameters of the transformation T are estimated by solving a minimization problem where the objective function is a weighted sum of the residual errors. In particular, the following minimization problem is solved:

$$\min_\theta f(\theta) = \min_\theta \sum_i \rho(r_i, \sigma), \quad r_i = p'_i(\theta) - p'_i, \quad (4)$$

where $\rho(r, \sigma)$ is the objective function defined over residual r and scale factor σ. The residual error r is based only on the geometric position of points in the plane. The scale factor $\sigma = c_0 \,\mathrm{median}\{|r_i|\}$, where $c_0 = 1.4826$ [11]. In this work, we use the Geman–McClure function [1]

$$\rho(r, \sigma) = \frac{r^2}{r^2 + \sigma^2}.$$

In the case of an affine model, the minimization is achieved using an iterated least-squares method. In Fig. 1, we plot the global horizontal displacement as computed from the affine model parameters for our method (dashed line) and for the robust differential method of Black and Anandan [1] applied to the whole *Stefan* sequence. In view of these results, the two methods are equivalent, both

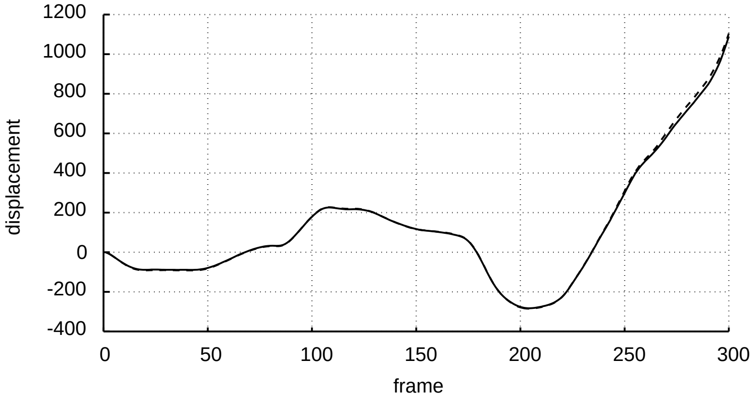

Fig. 1. The estimated global horizontal displacement for the Stefan sequence using the robust differential method and our method (dashed line).

are robust, rejecting the independent movement and accurately estimating large motions.

In the case of a projective model, the Geman–McClure function is used, leading to a non-linear system for resolving the resulting minimization problem. To solve this system, we use the Levenberg–Marquardt iterative non-linear minimization algorithm. The Levenberg–Marquardt algorithm computes an approximate Hessian matrix H and the weighted gradient vector g with components

$$g_k = \sum_i \frac{\partial \rho}{\partial r_i}\frac{\partial r_i}{\partial \theta_k}, \qquad H_{kl} = \sum_i \frac{\partial^2 \rho}{\partial r_i^2}\frac{\partial r_i}{\partial \theta_l}\frac{\partial r_i}{\partial \theta_k}.$$

Then the motion model parameters θ are updated by an amount $\delta\theta$ according to the equation

$$(H + \mu H_{\mathrm{d}})\delta\theta = -g, \tag{5}$$

where H_{d} is the diagonal of H, and μ is a stabilization parameter.

The steps of the Levenberg–Marquardt method are:

1. Compute the error $f(\theta)$.
2. Initialize $\mu = 0.001$.
3. Solve system of equations (5). Compute the estimation error with the updated parameters.
4. Compare with the previous error value. If the error has increased, the parameter μ is increased by a factor of 10, otherwise the parameter μ is decreased by a factor of 10 and the parameters θ are updated $\theta \leftarrow \theta + \delta\theta$. We then return to Step 3.

These steps are iterated until the relative error difference between successive iterations is less than a threshold that in practice may be roughly 0.001.

2.3.2. Motion estimation based on affine moment invariants

The second method for estimating motion model parameters is based on affine moment invariants. The moments are generally features that describe succinctly an object shape or a surface. In this work, we use the moments in order to estimate an affine transformation between two image regions. The moments are defined by

$$m_{kl} = \sum_x \sum_y x^k y^l.$$

In our work, the moment evaluation is limited only on the boundary points of the corresponding regions in the two images. It is obviously necessary to determine the corresponding regions in the two images. The affine transformation is then obtained by estimating appropriate moments in these regions. This method is faster than that of the previous section, and it is sufficiently accurate if the initial correspondences are correct. The moments computed on polygons can take in consideration the fact that under affine transformations polygon regions are maintained. In addition, the summation on the region boundaries make the estimated moments less noise-sensitive.

Region correspondence results from point correspondence. In particular, after 2-D motion field estimation, which gives correspondences between points in the two images, points that are not consistent with dominant motion are rejected. We select some points in the set of corresponding points in two images, using criteria that are based on the position of these points in the images. Then the selected points are used to construct the contours of the corresponding regions in the two images. The corresponding contours in two images are described by using moments. Without loss of generality, we assume that the origin is placed at the centre of region in the initial view before the transformation. Therefore, the translation vector results from the first-order moments of the region in the second view.

For estimating the transformation matrix **A** we use affine invariants. There are several functions of moments, which are invariant features in affine transformations. The method that is usually used to derive moment invariant features is the normalization method [10], in which a standard position is defined. The standard position of an object or a region uniquely characterizes the object and it is the same for all affine transformations. Using this standard position the affine transformation, and therefore an affine motion model, between two views of the same object or image region can be determined. The normalization transformations are calculated separately for the two views of the object or image region. Let $\mathbf{s}$ denote the standard position and let $\mathbf{T}_1$, $\mathbf{T}_2$ be the two affine transformations, from the object position to the standard position. The following relations hold true:

$$\mathbf{p}_2 = \mathbf{A}\mathbf{p}_1, \quad \mathbf{p}_1 = \mathbf{T}_1\mathbf{s} \quad \text{and} \quad \mathbf{p}_2 = \mathbf{T}_2\mathbf{s}.$$

The affine transformation between the two views is estimated by composing the two evaluated affine transformations between each view and the standard position

$$\mathbf{A} = \mathbf{T}_2\mathbf{T}_1^{-1}. \tag{6}$$

The steps of the whole algorithm for estimating the affine model parameters are shown in Fig. 2.

The transformation matrix $\mathbf{T}_i$ $(i = 1, 2)$ is decomposed in x-shear, anisotropic scaling and rotation,

$$\mathbf{T} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & \delta \end{bmatrix} \begin{bmatrix} 1 & \beta \\ 0 & 1 \end{bmatrix}. \tag{7}$$

For simplicity, the index i is suppressed in the above equation. By using this decomposition, we calculate the decomposition parameters $\phi, \alpha, \delta, \beta$ [10,17] for $\mathbf{T}_1$ and $\mathbf{T}_2$ and use them to calculate the decomposition parameters for $\mathbf{A}$ (the affine transformation between the two views). The decomposition parameters are calculated by successively obtaining invariants for each of the decomposition steps.

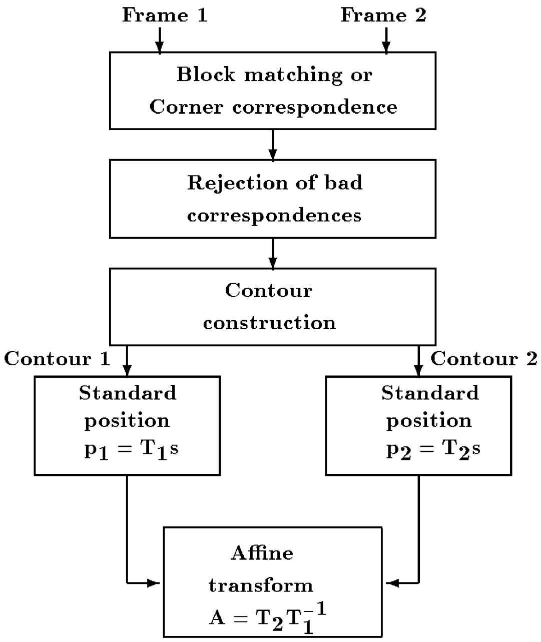

Fig. 2. The steps of the estimation method based on the affine invariants.

In this section, we focus on contour construction from a set of selected points in the two images. After applying the displacement vector field estimation and M-estimation a number of corresponding points has been selected. The contour construction procedure is first applied to the selected points in the first image. After constructing the first contour, the second is obtained from the points of the second image which correspond to the points of the first image contour. Usually the number of selected points is too large, rendering the computational cost of contour construction unacceptable. In order to avoid the large cost of contour construction, we select a fixed number of corresponding points in the two images. This limitation does not influence the parameter estimation accuracy, because under affine transformations a polygon remains a polygon.

We select the points so that the surface built by concatenating them covers a large enough area of the scene plane. Therefore, the affine transformation, which will be estimated by the affine invariant method, will better represent the real camera motion. If the selected points cover a small surface of the image plane, then the estimated camera motion could differ from the real one, as it is only projected in this portion of the scene. The problem is more pronounced in the case of images with more than one motion planes. For this reason, the selection procedure chooses the most distant points. The algorithm used for the selection of points is as follows:

1. If N is the number of points in each image, after applying the method of point correspondence and the method of inliers detection, there are $N(N-1)/2$ possible point pairs. For each pair of points (p_1, p_2) their euclidian distance d is computed.
2. The pair (p_1, p_2) with the largest distance taken from the set of all possible point pairs is

selected. These points are inserted in the list of selected points L.

3. The selected pair is removed from the set of possible pairs. We also remove from the set of possible point pairs those pairs for which at least one point is a neighbour of one of the points that define the selected pair. The neighbourhood is a square area with size taken to be 10 pixels around of each point of the selected pair.
4. If the number of selected points of the list L is smaller than K, we return to Step 2, otherwise the algorithm terminates.

K is the preset number of contour points. One of the most obvious solutions is constructing the convex hull circumscribing all points. The possibility of constructing a uniform convex hull is not dismissed; however, the use of a non-convex hull helps to better estimate the camera motion because the complex shape passing through all selected points contains more information about the motion in the scene.

A non-convex hull is constructed as a solution to the well-known traveling salesman problem (TSP), that of identifying the shortest tour which a traveling salesman will follow in order to visit K cities exactly once. In our application, this problem is to find the shortest tour visiting all the points provided there are no edge sections. This algorithm is a tour construction procedure which builds an approximately optimal tour starting from the original distance matrix. For the solution of this problem there is no fast accurate algorithm. Fortunately, there are some heuristics which give good approximations. One of these, the convex hull insertion procedure [3], is used in our work. This algorithm uses the convex hull as an initial sub-tour. It then successively adds the other points not belonging to the convex hull, following some distance-related criterion. Graham's scan [9] is used in order to construct the convex hull. The steps of the convex hull insertion procedure, known as the Stewart algorithm, are:

Step 1: Form the convex hull of the set of points. The hull gives an initial sub-tour.

Step 2 (*Insertion*)*:* For each point p_k not yet contained in the sub-tour, decide between which points p_i and p_j on the sub-tour to insert point p_k. That is, for each such p_k, find $\{p_i, p_j\}$, such that $d_{ik} + d_{kj} - d_{ij}$ is minimal.

Step 3 (*Selection*)*:* From all (p_i, p_k, p_j) found in Step 2, determine the (p_i^*, p_k^*, p_j^*), such that $(d_{i^*k^*} + d_{k^*j^*})/d_{i^*j^*}$ is minimal.

Step 4: Insert point p_k^* in sub-tour between points p_i^* and p_j^*.

Step 5: Repeat Steps 2–4 until a Hamiltonian cycle is obtained.

3. Mosaic construction

As mentioned above, the mosaic image is constructed from information from all frames in the sequence. The mosaic construction is performed in two steps: *frame alignment* and *frame composition*. Frame alignment is based on motion estimation, while frame composition uses the frame alignment in order to compose all frames into a single image. Both of these steps depend on the mosaic's representation.

In our implementation, we have focused on the *static mosaic* representation, which is constructed from each shot of the sequence and represents the view of the scene background over the whole sequence. The frames of the shots are aligned to a fixed coordinate system, which can be chosen by the user and is nominally the coordinate system of the first frame of the shot. The aligned frames are then composed in two ways: by using some type of temporal filter or by simply adding the information of each new frame to the mosaic without filtering. The mosaic image constructed in the first method reveals a sharp background scene and ghost-like moving objects, while the image constructed in the second way reveals a sharp background and the moving objects in their position in the frame whose coordinate system is used (the *reference frame*). The most important residuals are simply computed as the difference which results by comparing each frame with the static mosaic. The most important residuals usually represent individual moving objects or changes in the scene that occur over the elapsed time. The static mosaic is a compact scene representation. It is well suited to video storage and to rapid browsing in large digital video libraries. It is also used to obtain efficient access to individual frames of interest.

3.1. Frame alignment

The alignment of all sequence frames is generally achieved by estimating the camera motion. The ways of alignment vary according to the chosen coordinate systems and the images which take part in motion estimation. Each different coordinate system defines a different representation that requires a different alignment. Images are selected in such a way as to obtain motion estimation as accurate as possible and, therefore, to establish correct alignment between all the frames in the sequence. In what follows we describe in detail the *frame-to-frame* alignment, and we mention briefly the *frame-to-mosaic* alignment.

Frame-to-frame: The alignment parameters are first computed between successive frames for the whole sequence. The result of motion estimation between successive frames are the alignment parameters themselves. These parameters are composed to compute the global frame-to-mosaic alignment parameters. The mosaic image is created by warping each frame to a reference coordinate system using the computed parameters. When constructing a static mosaic, all the frames are aligned to a fixed coordinate system. This can be the system of one particular frame called the reference frame or can be a virtual coordinate system. In the second case, it is necessary to use the transformation between the virtual coordinate system and each one of the input frames. The alignment process requires one pass over the sequence in order to compose all the successive transformations.

If camera motion is described by the affine parametric model, then the alignment between successive frames is given by the following relation:

$$\mathbf{p}_{t+1} = \mathbf{A}_t\mathbf{p}_t + \mathbf{b}_t, \tag{8}$$

where $\mathbf{p}_{t+1}$ and $\mathbf{p}_t$ are the 2-D real pixel coordinates of the successive frames $t+1$ and t. $\mathbf{A}_t$ and $\mathbf{b}_t$ are the transformation and translation matrices of frame $t+1$ in order that it be aligned in the coordinate system of the frame t. Generally, the alignment of each frame t in the coordinate system of reference frame s is obtained by the following relations:

$$\mathbf{p}_s = \mathbf{A}_{s-1}\cdots\mathbf{A}_t\mathbf{p}_t + \sum_{i=t}^{s-2} \mathbf{A}_{s-1}\cdots\mathbf{A}_{i+1}\mathbf{b}_i + \mathbf{b}_{s-1} \quad \text{if } t<s, \tag{9}$$

$$\mathbf{p}_s = (\mathbf{A}_{t-1}\cdots\mathbf{A}_s)^{-1}\left(\mathbf{p}_t - \sum_{i=s}^{t-2} \mathbf{A}_{t-2}\cdots\mathbf{A}_{i+1}\mathbf{b}_i - \mathbf{b}_{t-1}\right) \quad \text{if } t>s, \tag{10}$$

where $\mathbf{p}_s$ are the coordinates of the points in frame t according to the coordinate system of reference frame s. The above relations hold if the mosaic coordinate system is exactly the coordinate system of frame s. In the case that we select a virtual coordinate system as the mosaic coordinate system, the above coordinates $\mathbf{p}_s$ are transformed to the virtual coordinate system

$$\mathbf{p}_\mathrm{V} = \mathbf{A}_\mathrm{V}\mathbf{p}_s + \mathbf{b}_\mathrm{V}, \tag{11}$$

where $\mathbf{A}_\mathrm{V}$ and $\mathbf{b}_\mathrm{V}$ express the transformation and the translation of each frame in order to be projected into the virtual coordinate system, while $\mathbf{p}_\mathrm{V}$ corresponds to the coordinates of each frame in the virtual coordinate system.

If camera motion is described by the projective parametric model, the alignment of the successive frames corresponds to the relation

$$\mathbf{P}_{t+1} = \mathbf{M}_t\mathbf{P}_t,$$

where P_{t+1} and P_t are the projective coordinates of frames $t+1$ and t. M_t corresponds to the projective transformation of frame t in the coordinate system of frame $t+1$. The projective transformation corresponds to the matrix

$$\mathbf{M} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix},$$

where a_i are the parameters of the projective parametric model. The alignment of each frame t according to coordinate system of frame s

corresponds to the relations

$$\mathbf{P}_s = \left(\prod_{\substack{i=s-1 \\ t<s-1}}^{t} \mathbf{M}_i \right) \mathbf{P}_t \quad \text{if } t<s, \tag{12}$$

$$\mathbf{P}_s = \left(\prod_{\substack{i=t-1 \\ s<t-1}}^{s} \mathbf{M}_i \right)^{-1} \mathbf{P}_t \quad \text{if } t>s, \tag{13}$$

where P_s are the projective coordinates of t frame in the coordinate system of s frame. The product of the above relations expresses the composition of successive projective transformations and also corresponds to a projective transformation (M_{ts}). The inversion of the product of the projective transformations is given by

$$\mathbf{M}'_{ts} = |\mathbf{M}_{ts}|\mathbf{M}_{ts}^{-1},$$

where $|\mathbf{M}_{ts}|$ is the determinant of $\mathbf{M}_{ts}$ and $\mathbf{M}_{ts}^{-1}$ is its multiplicative inverse. In the case where the chosen coordinate system is a virtual one, the alignment of each frame corresponds to the relation

$$\mathbf{P}_\mathrm{V} = \mathbf{M}_\mathrm{V}\mathbf{P}_s,$$

where the P_s coordinates of frame t are transformed to the virtual coordinate system according to the projective transformation M_V. P_V are the new coordinates of frame t projected into the virtual coordinate system.

Frame-to-mosaic: Frame-to-mosaic alignment requires motion estimation between the current mosaic image and the current frame of the sequence. To handle the problem of large displacements between the current mosaic image and new frames, the alignment parameters are computed between the mosaic image and the new frame using the alignment of the previous frame as the initial value for the displacement.

3.2. Frame composition

When the frames have been aligned, they must be composed. This is the last step in the construction of the mosaic image. In our implementation, different ways of frame composition are considered. These are separated into two categories, depending on whether they use temporal filtering or simply “stick” the further information of each frame onto the current mosaic image. The filters that are used in the first category are the median and the average intensity value. To reduce visual misalignments occurring in the mosaic image area containing the borders of the frames, we use the average intensity value filter in a zone around the borders of each frame.

3.3. The frame misalignment problem

One of the previous approaches to the problem of frame alignment, as we have already seen, was the frame-to-frame alignment. In this approach, the parameters of the transformations between successive frames are composed to compute the global frame-to-mosaic alignment parameters. The main problem with this approach is that the parameters of the transformation between successive frames are not accurate enough to guarantee the construction of a correctly aligned mosaic image. Small errors in parameter estimation of successive frames or small deviations of the supposed parametric model from the real motion accumulate. Therefore, global frame-to-mosaic alignment parameter computation is susceptible to accumulating errors. The extent to which errors accumulate depends significantly on the duration of the video shot that the mosaic image represents. When the length of a video shot is large enough, the errors in each of the frame-to-frame alignment parameters are large as well, because the number of frames is increasing.

An improvement over frame-to-frame alignment is the frame-to-mosaic alignment. In this approach, accumulating errors do not exist. The results are better under some conditions. Constructing the mosaic image with the entire information of the previous frame and the additional information of each new frame leads to misalignment. This problem is the result of the great difference of the information between each of the new frames and the area of the mosaic image that corresponds to the same information. The information difference appears because of the changes in the scene in these areas. These changes result either from the presence of a moving object or from changes in illumination. If mosaic images are constructed in the same way, problems are caused by previous misalignments. The areas in which misalignments occur take part in the process of motion estimation between the mosaic image and the new frame. Consequently, mosaic-to-new-frame alignment parameters are incorrectly computed. Also, the misalignment problem appears when the difference in resolution between the mosaic image and the new frame is large enough. The large difference in resolution leads to incorrect mosaic-to-frame alignment parameters, when the parametric model is not general enough to describe it. There is no direct solution to the above problems. Therefore, in order to construct a well-aligned mosaic image we use the previous frame alignment approach and try to find techniques for correcting possible misalignments. In our implementation, we propose the local fine correction of the already-computed global frame-to-mosaic alignment parameters.

3.4. Local fine correction of the frame alignment

In order to construct a well-aligned mosaic image, we use a local fine correction technique. Assuming that frame-to-frame alignment parameters have already been computed, the local fine correction technique takes place during the mosaic image construction. The process of mosaic construction involves the frames' transformation according to global frame-to-mosaic alignment parameters, which are obtained by composing the frame-to-frame alignment parameters. The frame transformation is achieved using the bi-linear interpolation method. The fine correction technique is applied to a subset of previously transformed frames. The obtained parameters are exactly the camera motion correction. For obtaining the correction parameters we use the M-estimator on corresponding corners of the transformed frames according to the initial estimation. Composing the directly new parameters with the global frame-to-mosaic alignment parameters of the corresponding frame, the corrected global frame-to-mosaic alignment is generated. The new global frame-to-mosaic alignment parameters approximate better the real camera motion parameters. The two basic issues for the local fine correction technique implementation are the following:

1. the frequency determination of the motion parameters correction,
2. the choice of the images which take part in the correction process.

The local fine correction could be active in each pair of successive frames of the sequence. As this has a high computational cost, its frequent utilization should be avoided. The correction frequency is inversely proportional to the quality of the mosaic image construction, and also to the cost of its construction. Therefore, we must choose a reasonable trade-off in the value of the correction frequency.

In our implementation, we propose the use of the frames topology of a shot in order to determine the correction frequency and select the frames which take place in this process. The topology describes the relative position of each frame in the mosaic image. This depends on the direction of the camera motion. In Fig. 3, a frame topology is depicted. The camera first pans to the right of the scene and then returns again to the left, capturing a larger portion of the scene than before. The problem of misalignments in the mosaic image construction appears when the camera returns to areas which have been already captured. The frames which are captured during the return of the camera are wrongly placed in the mosaic image, and intense misalignments are the result of this frame location. Therefore, in sequences where the camera changes motion direction, we propose the local fine correction application to one of the new direction frames.

It is also necessary to apply the correction process to frames that are topologically neighbours of the reference frame. The misalignments express the deviation of each transformed frame from the coordinate system of the reference frame. Therefore, the correction of motion estimation between each of the frame of the sequence and the reference frame is required. Because of differences in the visual information of each frame with that of the reference frame, the correction will not be trustworthy.

The neighbouring frames depend on the sequence topology and they are not necessarily

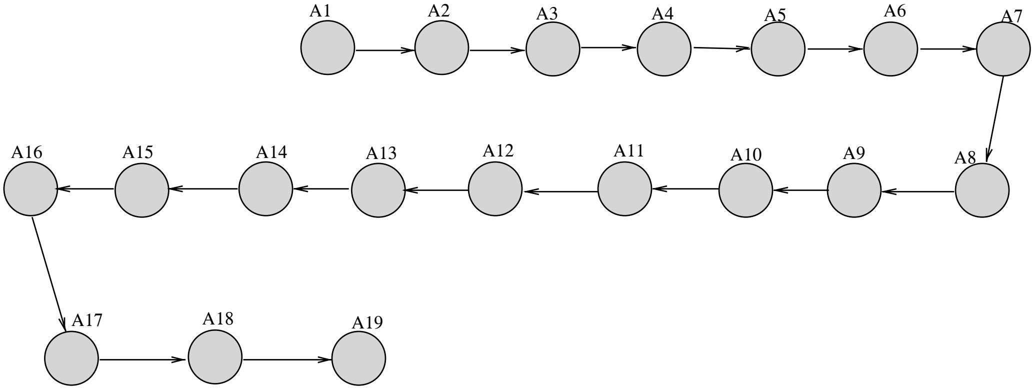

Fig. 3. Topology of an image sequence.

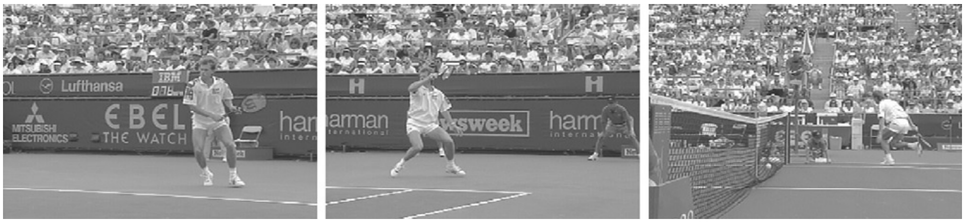

Fig. 4. Three frames of the Stefan sequence.

temporally successive frames. In the above figure, neighbours of frame A1 are considered to be frames A2, A13, A14 and A19. A solution to the problem of misalignments caused by accumulating errors during the camera motion estimation of successive frames would be the application of the local fine correction each the time the camera is translated by T pixels. This translation is measured by the displacement of image centres, as shown for example in Fig. 1 for the Stefan sequence. This means that the whole parametric model is taken into account.

In brief, we propose the application of local fine correction:

- in the frames in which the camera changes motion direction,
- in the frames which are topologically neighbours of the reference frame,
- each time the accumulated camera translation exceeds T pixels.

The frames to place in the correction process are determined by the frame topology. The first frame (F) is selected by the frequency of correction, while the second frame (G) is determined by the following criteria:

- Frame G must belong to the neighbourhood of Frame F. The neighbourhood is also determined by the topology. The neighbourhood is an area around the frame F of size $K \times K$. The correctional process is performed only if G belongs to the neighbourhood of F, because then the amount of common information between the two frames is significant.
- Frame G must be the temporal oldest of all frames in the neighbourhood. This choice helps to decrease cumulative errors.

4. Experimental results and conclusions

We have implemented and compared all the techniques described above on simulated and real image sequences. The comparison between the block-matching and the corner correspondence techniques shows that the first could fail in case of

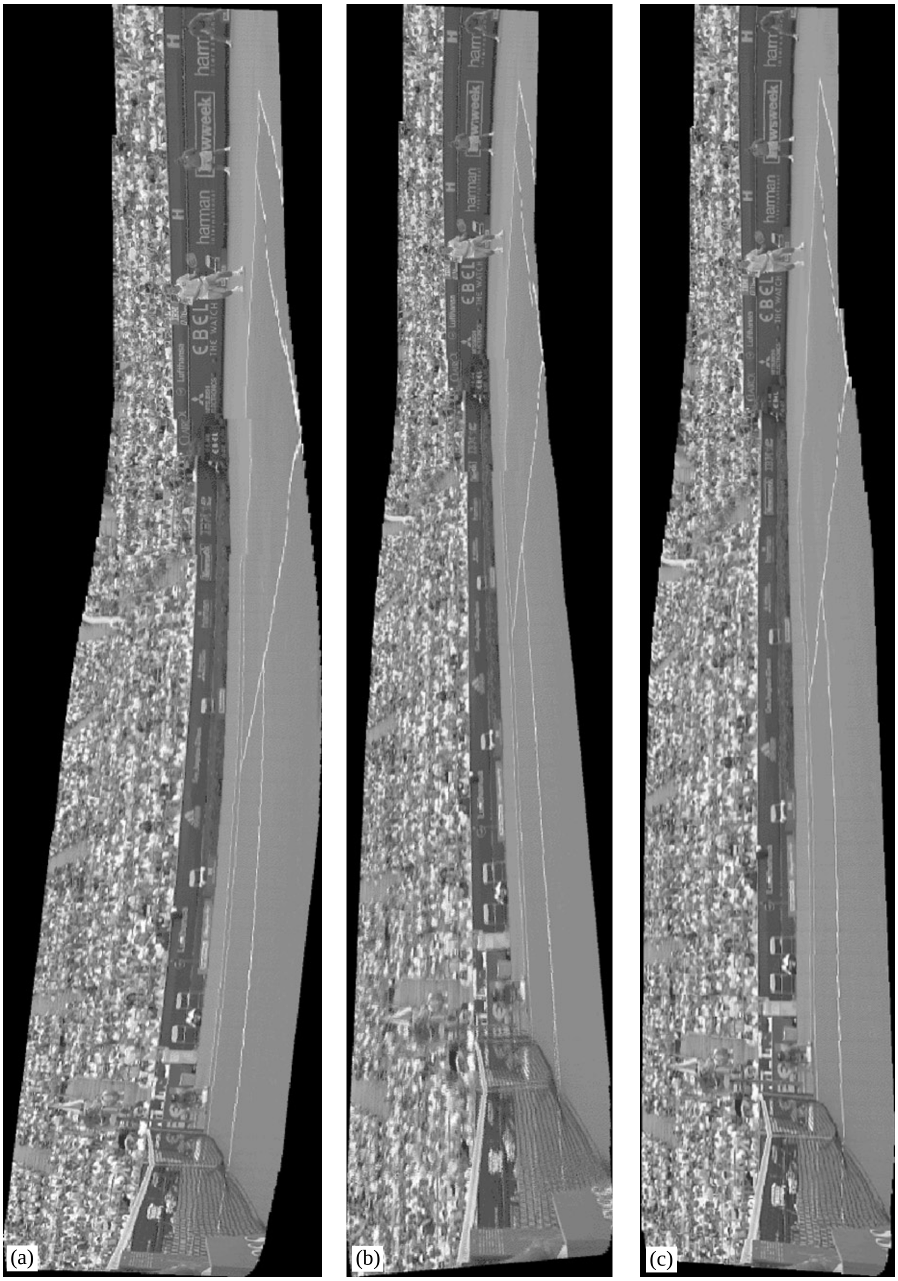

Fig. 5. Panoramic view of the Stefan sequence using: (a) an affine motion model; (b) a projective motion model and (c) the correction procedure.

substantial scale factor changes, as in the case of a significant zoom-in or zoom-out. When the geometric deformation is not very pronounced, the block-matching method appears to exhibit better results and greater efficiency than the corner correspondence. This comparison was done on

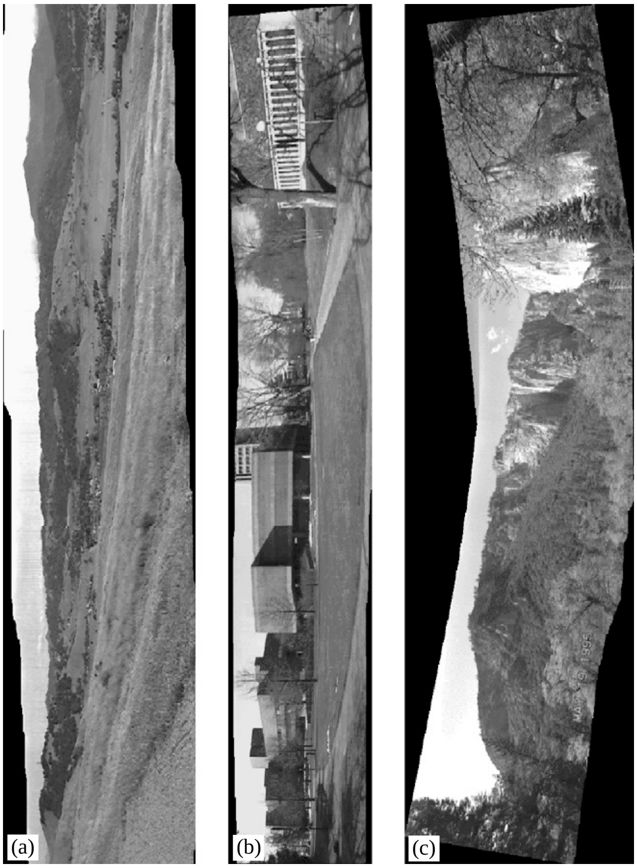

Fig. 6. Panoramic view of the: (a) Almaden; (b) Princeton and (c) Yosemite sequences.

a sequence with synthetic 2-D translation, rotation and zoom.

Comparing the M-estimation method to the invariant moment method, we conclude that the first is less sensitive to 2-D motion field errors. In the case of a sufficiently good initial point correspondence, the moment invariant method gives better camera motion estimation. In any case, the M-estimator is robust to various kinds of outliers, while the moment invariance method has a lower computational cost. The moment invariance method is also limited to the affine or any simpler model, since projective moment invariants are not known to exist.

We have observed the results of our algorithmic approaches mainly on the Stefan sequence (Fig. 4), which presents many different kinds of motion: zoom-in, zoom-out, rotation around the vertical or the horizontal axis. In addition, in a large portion of the viewed scene the image intensity is uniform, and so no block or corner correspondences could be found in these regions. Thus, the estimated motion model is mainly valid on the upper part of the scene. It should also be noted that the displacement between successive frames is up to 25 pixels for a resolution of 350 pixels per line, while the scale change may be as much as 5%. For this complex sequence we have compared two motion models: the affine and the projective. The 2-D motion field was obtained by corner correspondence, and the model parameters were robustly estimated using the Geman–McClure M-estimator. The two panoramic views obtained from 300 frames are shown in Figs. 5(a) and (b), respectively. Clearly, the projective model gives better results and captures well the whole camera movement, provided the local fine correction technique is used for limiting the propagation of image registration errors, as shown in Fig. 5(c). The correction technique was used after a change in the direction of panning and after 40 pixels of image centre displacement.

Globally, we have experimented with three motion parametric models: (a) translation with isotropic scaling, (b) affine and (c) projective. In all cases, the best results are achieved by the model most closely approximating the real 3-D motion. An example of the first model is given in Fig. 6(a) for the *Almaden* sequence, where the movement is mainly translational. The block-matching technique was used for estimating the displacement vectors, and the M-estimator for obtaining the three model parameters, for the whole sequence of 1000 frames. For the construction of the panoramic view, the frame-to-mosaic alignment was used, while the frame-to-frame alignment gave a similar result.

The affine model was used for the construction of the panoramic view from the 350 frames of the *Princeton* sequence (Fig. 6b). Here the camera undergoes 3-D translation, rotation around the vertical axis and zooming. Corner correspondence was used for estimating the 2-D motion field, and the M-estimator provided values for the model parameters.

An example of the projective model is shown in Fig. 6(c). In the *Yosemite* sequence of 1000 frames, the camera motion is 3-D translational and rotational around the vertical axis with some zooming. As in the previous case, corner correspondence and M-estimation are used for obtaining the eight motion parameters.

A possible solution to the model selection problem is a hierarchical model estimation and the choice of the more reduced model with sufficient accuracy. As the models employed are mostly good approximations for planar scene surfaces, in case of a scene composed from distinct planar surfaces, the best approach should be the motion-based image segmentation, in order to obtain a better image registration.

Applications which could be foreseen with the constructed static mosaic are: background storage, video compression, scene summary, video indexing and video retrieval.

References

[1] M. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, Comput. Vision Image Understanding 63 (1) (January 1996) 75–104.

[2] M. Flickner, et al., Query by image and video content: the QBIC system, Computer 28 (9) (September 1995) 23–32.

[3] B. Golden, W. Stewart, Empirical analysis of heuristics, in: E. Lawler, et al., (Eds.), The Traveling Salesman Problem, Wiley, New York, 1984.

[4] P. Huber, Robust Statistics, Wiley, New York, 1981.

[5] M. Irani, P. Anandan, Video indexing based on mosaic representations, Proc. IEEE 86 (May 1998) 905–921.

[6] M. Irani, P. Anandan, J. Bergen, R. Kumar, S. Hsu, Efficient representations of video sequences and their applications, Signal Processing: Image Communication 8 (1996) 327–351.

[7] S. Mann, R.W. Picard, Video orbits of the projective group: a simple approach to featureless estimation of parameters, IEEE Trans. Image Process. 6 (9) (September 1997) 1281–1295.

[8] H. Nicolas, New methods for dynamic mosaicking, IEEE Trans. Image Process. 10 (8) (August 2001) 1239–1251.

[9] F. Preparata, M.I. Shamos, Computational Geometry, Springer, Berlin, 1985.

[10] I. Rothe, H. Susse, K. Voss, The method of normalization to determine invariants, IEEE Trans. Pattern Anal. Machine Intell. 18 (4) (April 1996) 366–375.

[11] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, Wiley, New York, 1987.

[12] S.H. Sawhney, S. Ayer, Compact representation of videos through dominant and multiple motion estimation, IEEE Trans. Pattern Anal. Machine Intell. 18 (8) (August 1996) 814–830.

[13] H.S. Sawhney, S. Hsu, R. Kumar, Robust video mosaicing through topology inference and local to global alignment, in: European Conference on Computer Vision, Springer, Berlin, October 1998, pp. 103–118.

[14] A. Smolic, T. Sikora, J.-R. Ohm, Long-term global motion estimation and its application for sprite coding, content description, and segmentation, IEEE Trans. Circuits Systems Video Technol. 9 (December 1999) 1227–1241.

[15] R. Szeliski, Video mosaics for virtual environments, IEEE Comput. Graph. Appl. 16 (2) (March 1996) 22–30.

[16] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1998.

[17] G. Tzanetakis, M. Traka, G. Tziritas, Motion estimation based on affine moment invariants, in: Proceedings of the IX European Signal Processing Conference, Vol. II, Rhodes, Greece, September 1998, pp. 925–928.

[18] G. Tziritas, C. Labit, Motion Analysis for Image Sequence Coding, Elsevier, Amsterdam, 1994.

[19] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, Artificial Intell. 78 (1995) 87–119.