Submitted to Computer Vision and Image Understanding Journal (CVIU), in press.

Efficient, Causal Camera Tracking In Unprepared Environments ⋆

Manolis I. A. Lourakis * and Antonis A. Argyros

Institute of Computer Science, Foundation for Research and Technology - Hellas
Vassilika Vouton, P.O.Box 1385, GR 711 10, Heraklion, Crete, Greece

Abstract

This paper addresses the problem of tracking the 3D pose of a camera in space, using the images it acquires while moving freely in unmodeled, arbitrary environments. A novel feature-based approach for camera tracking is proposed, intended to facilitate tracking in on-line, time-critical applications such as video see-through augmented reality. In contrast to several existing methods which are designed to operate in a batch, off-line mode, assuming that the whole video sequence to be tracked is available before tracking commences, the proposed method operates on images incrementally. At its core lies a feature-based 3D plane tracking technique, which permits the estimation of the homographies induced by a virtual 3D plane between successive image pairs. Knowledge of these homographies allows the corresponding projection matrices encoding camera motion to be expressed in a common projective frame and, therefore, to be recovered directly, without estimating 3D structure. Projective camera matrices are then upgraded to Euclidean and used for recovering structure, which is in turn employed for refining the projection matrices through local resectioning. The proposed approach is causal, is tolerant to erroneous and missing feature matches, does not require modifications of the environment and has computational requirements that permit a near real-time implementation. Extensive experimental results demonstrating the performance of the approach on several image sequences are included.

Key words: camera tracking, plane tracking, egomotion estimation, 3D structure and motion estimation, corner matching, matchmoving, augmented reality

⋆ This work was partially supported by the EU IST - 2001 - 34545 project LifePlus
* Corresponding author. Tel: +30 2810 391716, Fax: +30 2810 391601, URL: http://www.ics.forth.gr/cvrl/

Email addresses: lourakis@ics.forth.gr (Manolis I. A. Lourakis), argyros@ics.forth.gr (Antonis A. Argyros).

1 Introduction

Tracking the 3D position and orientation of a camera using the images it acquires while moving freely in unmodeled, arbitrary environments, is a very challenging problem in visual motion analysis. Its difficulty primarily stems from the fact that the only information that can be extracted from images concerns the observed 2D motion of image points, which depends nonlinearly both on the sought camera motion and the unknown 3D structure of the viewed scene. Thus, small errors in the estimates of 2D motion can have a significant impact on the accuracy of the recovered camera motion and 3D structure. In addition to its theoretical interest, camera tracking has a wide spectrum of important practical applications ranging from robotics and computer assisted surgery to augmented reality and the creation of special effects in the post-production/film-making industry [1]. To provide the versatility required by such applications, very demanding camera tracking requirements, both in terms of accuracy and speed, are imposed [2].

Despite the fact that successful camera tracking solutions have been engineered for certain applications like virtual TV production [3], such custom technologies are quite expensive, typically suffer from range limitations and call for special modifications of the environment (e.g. placement at known locations of artificial beacons such as retroreflective markers, infrared LEDs, or current carrying coils), that render them inapplicable for tracking in unprepared, unstructured scenes, large scale environments or archive footage. Being non-intrusive, passive and capable of covering large fields of view, vision-based approaches provide an attractive paradigm for dealing with camera tracking. During the last fifteen years, numerous research efforts have focused on vision-based camera tracking within the framework of the more general structure from motion problem [4]. Before briefly reviewing a few representative ones, it is pointed out that our requirement for operation in unprepared environments excludes methods such as [5,6] that rely upon the presence of fiducial markers or special calibration objects in the environment. Additionally, due to the problems pertaining to the accurate estimation of optical flow when the inter-frame image motion is not infinitesimal, we have chosen to focus our attention to feature-based approaches only. For more details regarding direct, i.e. flow-based approaches, see [7] and references therein.

Assuming a mostly rigid scene, vision-based camera tracking methods that avoid relying on modifications of the environment exploit geometric constraints arising from the automatic extraction and matching of appropriate 2D image features such as corner points. Depending on their mode of operation, proposed approaches can be classified into two categories. The first category consists of methods designed for off-line use on pre-recorded image sequences. Such methods simultaneously process all image data in a batch mode and therefore are

non-causal, employing both past and future frames for deducing the camera motion corresponding to the current frame. Viéville et al [8], for instance, rely upon a set of point and line features extracted from an uncalibrated monocular image sequence and recover structure and motion by formulating a large non-linear optimization problem that is solved by alternating between structure and motion estimation. Sturm and Triggs [9] recover projective shape and motion from multiple images of a scene by factoring a matrix containing the images of all points in all views. The factorization method is inexact since it relies on approximate estimates of "projective depths" of image points and requires that all points are visible in all images. Cornelis et al [10] describe a system that relies on pairwise fundamental matrices for simultaneously recovering structure and motion for augmented reality applications. Camera motions for all frames are determined in a single final step using bundle adjustment.

Fitzgibbon and Zisserman [11] take a different direction and recover structure and motion using a hierarchical bundle adjustment approach based on image triplets and associated trifocal tensors. Rother and Carlsson [12] assume that a reference plane is visible in all images and present a linear algorithm that simultaneously computes 3D point structure and camera positions. The reconstruction and camera recovery is achieved in a single step by finding the null-space of a matrix built from image data using SVD. Kaucic et al [13] propose a similar approach that employs image-to-image homographies for linearizing the problem of estimating the camera motion. Commercially available camera tracking software products such as BOUJOU, MATCHMOVER, 3D-EQUALIZER and PFTRACK also fall into this category [1] . Typically, batch techniques share the use of global bundle adjustment as their last step. Global bundle adjustment involves the solution of high dimensional, non-linear optimization problems [14], therefore, and despite the use of careful implementations exploiting its sparse structure, it is computationally demanding. This, plus the requirement of operating on the whole sequence at once, makes batch methods inappropriate for use in on-line, time-critical applications where camera tracking must proceed in parallel with image acquisition.

Methods operating in a continuous mode, in which images are processed incrementally as acquired, constitute the second class of camera tracking techniques. Such methods exploit the natural ordering of images and are causal, i.e. they rely only on past frames for estimating the camera motion for the current image. An inherent shortcoming of all incremental techniques is that they are strongly dependent on an initial reconstruction encompassing a small subset of views. The work of Beardsley et al [15], who estimate camera matrices from 3D structure that is recovered incrementally, was one of the first

[1] See http://www.2d3.com, http://www.realvis.com, http://www.3dequalizer.com and http://www.thepixelfarm.co.uk respectively.

to propose such an approach. In some cases, structure recovery is completely avoided. For example, Simon et al [16] describe a camera tracking system that relies on continuous tracking of a 3D plane that is assumed to be present in the scene. The plane is tracked by estimating pairwise planar homographies with the aid of tracked interest points. The major disadvantage of this technique is that it requires the tracked plane to be continuously visible and its image segmented from the rest of the scene. Moreover, the tracking scheme employed requires manual intervention to bootstrap and cannot incorporate information from points off the plane or from more than two images. McLauchlan [17] presents a formulation of sequential least-squares for structure and motion reconstruction that is based on the Variable State Dimension Filter (VSDF). An appealing characteristic of this approach is that the VSDF is used both for obtaining an initial reconstruction in batch mode and for recursively updating this reconstruction by incorporating new data as they become available.

Avidan and Shashua [18] follow a direct approach for recovering a set of consistent projective camera matrices without reconstructing the 3D scene. The main contribution of [18] is a "threading" operation on two consecutive fundamental matrices that uses the trifocal tensor as the connecting thread. Their method is based on tracking a scene plane along an image sequence and provides, as a byproduct, the homography matrices it induces between adjacent views. However, owing to the use of constraints involving algebraic distances, the estimated homographies are not statistically optimal. Besides, the experimental results provided mainly focus on the performance of plane tracking rather than on the recovery of the underlying Euclidean camera motion and its accuracy. After restricting the permissible camera motions to pure rotations, Prince et al [19] propose a camera tracking algorithm for augmented reality applications. Regardless of its computational efficiency, however, its applicability is severely limited by the restricted camera motion model it employs. Employing calibrated images, Nistér [20] suggests a novel solution to the five-point relative pose estimation problem that is amenable to an efficient numerical implementation and uses it in a preemptive RANSAC framework to achieve real-time structure and motion estimation. Note, however, that the low latency of his implementation owes a lot to the use of MMX instructions for feature detection and matching. By introducing probabilistic simultaneous localization and mapping (SLAM) techniques in computer vision, Davison [21] has recently proposed an interesting approach to camera tracking. By employing a camera motion model and explicitly modeling uncertainty, his method is capable of determining in real-time the camera position/orientation and recovering sparse 3D information regarding the environment. Still, the method relies upon certain prior knowledge regarding the imaged scene, requires manual intervention for bootstrapping and employs image features that are of limited viewpoint-invariance, thus restricting its application in small scale environments.

It is well-known that the fundamental matrix, trifocal and quadrifocal tensor can be used to directly recover the camera motion from two, three of four images respectively. When, however, the number of images exceeds four, the corresponding multiview tensors do not provide closed-form solutions regarding the camera motion. Therefore, in the absence of any further information, most incremental camera tracking algorithms resort to recovering the scene structure and using it repetitively to register together the motion parameters estimated from pairs, triplets, or quadruples of successive images. This paper presents a novel feature-based approach to camera tracking that alleviates the need for relying on scene structure to recover an initial estimate of the camera motion. The method is based on tracking a 3D plane through a homography "chaining" operation that is applied to triplets of consecutive images through a sliding time window and exploits the fact that all images of a planar surface acquired by a rigidly moving observer depend upon the same 3D geometry. Since the tracked plane is not required to be physically present in the scene, a virtual one can be used instead. Plane tracking is achieved by tracking the 2D projections of points from all over the scene. By doing so, all information conveyed by matching points is taken into account, without the need for continuously maintaining an image-based segmentation of the tracked plane. Missing and erroneous point matches are tolerated, the motion model estimated for the tracked plane is exact and fully projective (i.e. a homography) and no camera calibration information or 3D structure recovery is necessary.

Knowledge of the homographies induced by the virtual 3D plane between each pair of successive images allows the corresponding projection matrices encoding camera motion to be expressed in a common projective frame and therefore to be recovered directly, without the need for retrieving structure. Then, 3D structure is recovered from the projection matrices via triangulation and used for refining them through local resectioning. Intended for use in close to online applications such as video see-through augmented reality and vision-based control, the proposed method is designed to operate in a continuous mode. The method follows a strategy similar to [18]. However, it is based on much simpler constraints whose derivation is shorter and does not involve either the trifocal tensor or tensorial notation. Obviating the unwieldy, time consuming process of trifocal tensor estimation [22], has apparent computational implications for continuous camera tracking. Moreover, our method tracks 3D planes by minimizing a geometrically meaningful criterion with respect to a set of four free parameters, which, according to the subspace constraint of [23], is a theoretically minimal one. Compared to [18] and [16] which estimate twelve and eight parameters respectively, the estimation of just four parameters is both faster and more accurate.

The rest of the paper is organized as follows. Section 2 explains the notation that will be used throughout all equations and provides some background knowledge. Section 3 describes plane tracking and section 4 builds upon it for

solving the problem of camera tracking. Since the tracked plane is not required to be physically present in the scene, any virtual 3D plane suffices for the purposes of camera tracking. Section 5 explains how can such a virtual plane be selected. Implementation issues and sample experimental results are reported in section 6. The paper is concluded with a brief discussion in section 7.

2 Notation and Background

In the following, vectors and arrays appear in boldface and are represented using projective (i.e. homogeneous) coordinates. 3D points are written in uppercase, while their image projections in lowercase (e.g. $\mathbf{X}$ and $\mathbf{x}$). Transposition is denoted by ^{T}. The symbol $\simeq$ denotes equality of vectors up to an arbitrary scale factor. The fundamental matrix and the epipole pair pertaining to two images are respectively denoted by $\mathbf{F}$ and $\mathbf{e}$, $\mathbf{e}'$. Also, $\mathbf{H}$ is used to designate the interimage homography induced by a 3D plane. A detailed treatment of the application of projective geometry to computer vision can be found in [4].

It is shown in [23] that the fundamental matrix and plane homographies are tightly coupled. More specifically, the entire group of all possible homography matrices between two images lies in a subspace of dimension 4, i.e. it is spanned by 4 homography matrices. These 4 homography matrices are such that their respective planes do not all coincide with a single point. Shashua and Avidan show in [24] that given the fundamental matrix $\mathbf{F}$ and the epipoles $\mathbf{e}$ and $\mathbf{e}'$ in an image pair, a suitable basis of 4 homography matrices $\mathbf{H}_1, \ldots, \mathbf{H}_4$, referred to as "primitive homographies", is defined as follows

$$\mathbf{H}_i = [\epsilon_i]_\times \mathbf{F}, \quad i = 1, 2, 3 \quad \text{and} \quad \mathbf{H}_4 = \mathbf{e}' \delta^T, \tag{1}$$

where ϵ_i are the identity vectors $\epsilon_1 = (1, 0, 0)$, $\epsilon_2 = (0, 1, 0)$ and $\epsilon_3 = (0, 0, 1)$, $[.]_\times$ designates the skew symmetric matrix representing the vector cross product (i.e. for a vector $\mathbf{a}$, $[\mathbf{a}]_\times$ is such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}, \quad \forall\ \mathbf{b}$) and δ is a vector such that $\delta^T \mathbf{e} \neq 0$. This last requirement can, for example, be satisfied by defining vector δ so that each of its elements has an absolute value of 1 and a sign identical to that of the corresponding element of $\mathbf{e}$. The first three homography matrices (i.e. $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$) are of rank 2 and span the subgroup of homography matrices whose underlying 3D planes contain the center of projection O' of the second camera. On the other hand, $\mathbf{H}_4$ by definition corresponds to a 3D plane not coincident with O' but going through the center of projection O of the first camera, thus having rank 1. Knowledge of the 4 primitive homographies allows any other homography $\mathbf{H}$ to be expressed as a

linear combination

$$\mathbf{H} = \sum_{i=1}^{4} \lambda_i \mathbf{H}_i, \quad \lambda_i \in \Re. \tag{2}$$

Next, a result due to Shashua and Navab [25] that plays a central role in the development of the proposed method is presented. Let Π be an arbitrary 3D plane inducing a homography $\mathbf{H}$ between two images. Let also $\mathbf{X}_0$ be a 3D point not on Π projecting to image points $\mathbf{x}_0$ and $\mathbf{x}_0^{'}$ and assume that $\mathbf{H}$ has been scaled to satisfy the equation $\mathbf{x}_0^{'} \simeq \mathbf{H}\mathbf{x}_0 + \mathbf{e}^{'}$. Then, for any 3D point $\mathbf{X}$ projecting onto $\mathbf{x}$ and $\mathbf{x}^{'}$, there exists a scalar κ such that

$$\mathbf{x}^{'} \simeq \mathbf{H}\mathbf{x} + \kappa \mathbf{e}^{'}. \tag{3}$$

Equation (3) suggests that the position of projected points in the second image can be decomposed into the sum of two terms, the first depending on the homography induced by Π and the second involving *parallax* due to the deviation of the actual 3D structure from Π. The scalar κ in Eq. (3) depends on $\mathbf{X}$ but is invariant to the choice of the second image and is termed as *relative affine structure* in [25]. Given $\mathbf{x}$, $\mathbf{x}^{'}$, $\mathbf{H}$ and $\mathbf{e}^{'}$, the term κ corresponding to $\mathbf{X}$ can be computed by cross-multiplying both sides of Eq. (3) with $\mathbf{x}^{'}$, which after some algebraic manipulation yields

$$\kappa = \frac{(\mathbf{H}\mathbf{x} \times \mathbf{x}^{'})^T (\mathbf{x}^{'} \times \mathbf{e}^{'})}{||\mathbf{x}^{'} \times \mathbf{e}^{'}||^2}. \tag{4}$$

3 Chaining Homographies for 3D Plane Tracking

Suppose that three consecutive images I_1, I_2 and I_3 are available and that a planar homography between I_1 and I_2 has been estimated. Considering the two pairs (I_2, I_1) and (I_2, I_3) formed by the three images, a key observation is the fact that image I_2 is shared by both of these pairs. Hence, the relative affine structure defined when I_2 assumes the role of the first image in Eq. (3) is insensitive to the choice of the second image (i.e. I_1 or I_3) completing the pair. This allows one to estimate the relative affine structure from the pair (I_1, I_2) and the corresponding homography and then use this estimate for computing the plane homography for the pair (I_2, I_3). This, in effect, constitutes a chaining operation involving plane homographies. The process just outlined is explained in more detail in the following subsections.

Assume that N triplets of matching points $(\mathbf{x}_i,\ \mathbf{x}_i^{'},\ \mathbf{x}_i^{''})$, $i = 1, \ldots, N$, are available across the three images I_1, I_2 and I_3 respectively and that the homography $\mathbf{U}$ from image I_1 to I_2 due to some 3D plane has been estimated. In the remainder of this section, a procedure for estimating the plane homography $\mathbf{V}$ induced by this 3D plane between images I_2 and I_3 will be described. $\mathbf{V}$ will be shown here to depend on nine parameters; later on, section 3.2 will demonstrate that four parameters suffice for determining it.

Equation (3) presupposes that the relative affine structure corresponding to a certain point $\mathbf{X}_0$ is equal to one. Let us begin by explaining the role of $\mathbf{X}_0$ in the development of Eq. (3). Recall that $\mathbf{H}$ and $\mathbf{e}^{'}$ are homogeneous entities, defined up to an arbitrary scale factor. Therefore, by fixing $\mathbf{H}$'s scale, $\mathbf{X}_0$ serves to establish a common relative scale between $\mathbf{H}$ and $\mathbf{e}^{'}$. Notice, however, that in the case that $\mathbf{H}$ has not been scaled with the aid of $\mathbf{X}_0$, Eq. (3) continues to hold for some $\kappa^{'}$ that is a scaled version of κ given by Eq. (4). In addition, in this case κ is not invariant to the choice of the second view. What remains invariant though, is the ratios of κ's computed from different image pairs.

From the set of matching pairs $(\mathbf{x}_i,\ \mathbf{x}_i^{'})$ the epipolar geometry for images I_1 and I_2 and thus the epipole $\mathbf{e}$ in image I_1 can be estimated. In a similar manner, the epipole $\mathbf{e}^{''}$ in I_3 for the camera motion corresponding to frames I_2 and I_3 can be estimated from the set of matching pairs $(\mathbf{x}_i^{'},\ \mathbf{x}_i^{''})$. Recalling that the homography from image I_2 to I_1 is simply $\mathbf{U}^{-1}$, for all point matches in those two images Eq. (3) becomes

$$\mathbf{x}_i \simeq \mathbf{U}^{-1}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}. \tag{5}$$

By employing Eq. (4), the κ_i can then be estimated as

$$\kappa_i = \frac{(\mathbf{U}^{-1}\mathbf{x}_i^{'} \times \mathbf{x}_i)^T(\mathbf{x}_i \times \mathbf{e})}{||\mathbf{x}_i \times \mathbf{e}||^2}. \tag{6}$$

Taking into account point matches in frames I_2 and I_3, Eq. (3) gives

$$\mathbf{x}_i^{''} \simeq \mathbf{V}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}^{''}. \tag{7}$$

In order for Eq. (7) to hold for the κ_i given by Eq. (6), $\mathbf{V}$ and $\mathbf{e}^{''}$ in it have to be defined up to the same scale factor. In other words, $\mathbf{V}$'s scale is determined by the scale of $\mathbf{e}^{''}$'s estimate. For this reason, $\mathbf{V}$ in Eq. (7) is no longer a homogeneous 3×3 matrix but rather an ordinary, inhomogeneous one. Equation (7) is thus a vector equation linear in $\mathbf{V}$, providing three linear constraints

on the nine unknown elements of $\mathbf{V}$. Due to the presence of an arbitrary, unknown scale factor, only two of those three constraints are linearly independent. Denoting the i-th row of matrix $\mathbf{V}$ by $\mathbf{v}_i^T$, writing $\mathbf{x}_i^{''} = (x_i^{''}, y_i^{''}, 1)^T$ and $\mathbf{e}^{''} = (e_x^{''}, \ e_y^{''}, \ e_z^{''})^T$, those two constraints can be explicitly expressed as [2]

$$\begin{aligned} \mathbf{v}_3^T \mathbf{x}_i^{'} x_i^{''} - \mathbf{v}_1^T \mathbf{x}_i^{'} &= \kappa_i e_x^{''} - \kappa_i e_z^{''} x_i^{''} \\ \mathbf{v}_3^T \mathbf{x}_i^{'} y_i^{''} - \mathbf{v}_2^T \mathbf{x}_i^{'} &= \kappa_i e_y^{''} - \kappa_i e_z^{''} y_i^{''} . \end{aligned} \tag{8}$$

Notice that Eqs. (8) do not require that the employed point matches have been identified as lying on the tracked plane or not. Therefore, they do not require that the tracked plane has been segmented from the rest of the scene and are applicable even in the case of tracking an out of view or a virtual plane (i.e. a plane not physically present in the scene). Since $\mathbf{v}_j^T \mathbf{x}_i^{'} = \mathbf{x}_i^{'T} \mathbf{v}_j$, Eqs. (8) can be written in matrix form as

$$\begin{bmatrix} -\mathbf{x}_i^{'T} & \mathbf{0}^T & \mathbf{x}_i^{'T} x_i^{''} \\ \mathbf{0}^T & -\mathbf{x}_i^{'T} & \mathbf{x}_i^{'T} y_i^{''} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \kappa_i e_x^{''} - \kappa_i e_z^{''} x_i^{''} \\ \kappa_i e_y^{''} - \kappa_i e_z^{''} y_i^{''} \end{bmatrix} . \tag{9}$$

Thus, each triplet of corresponding points provides two equations in the elements of $\mathbf{V}$. By concatenating the equations arising from five triplet correspondences, a matrix equation of the form $\mathbf{M}\mathbf{v} = \mathbf{b}$ is generated, where $\mathbf{M}$ is a 10×9 matrix, $\mathbf{v}$ is a 9×1 vector equal to $(\mathbf{v}_1^T, \ \mathbf{v}_2^T, \ \mathbf{v}_3^T)^T$ and $\mathbf{b}$ is a 10×1 vector. Omitting any row of matrix $\mathbf{M}$, yields a 9×9 system that may be solved using linear algebra techniques. In the case that more than five triplet matches are available, Eq. (9) gives rise to an over-constrained system from which $\mathbf{V}$ can be estimated in a least squares manner.

According to the terminology of [4], ch. 3, the estimation of $\mathbf{V}$ as described up to this point, is achieved with a Direct Linear Transformation (DLT) algorithm. It is well-known that DLT algorithms are not invariant to similarity transformations of the image but depend on the coordinate system in which image points are expressed. To alleviate this and, at the same time, improve the condition number of the DLT constraints, therefore ameliorating the accuracy of results, the normalization technique of [26] is applied to matching points prior to feeding them to the DLT algorithm. Independently for each image, this normalization consists in translating image coordinates so that the centroid of points is brought to the origin of the coordinate system, followed by an isotropic scaling that maps the average point to $(1, 1, 1)^T$. The normalizing transformation for image I_n is expressed by a 3×3 linear transformation

[2] Notice that all available point matches are assumed to originate from actual image points (i.e. corners); no ideal points whose third coordinate is zero exist among them.

$\mathbf{L}_n$. Notice that in this case, the normalized version $\bar{\mathbf{U}} = \mathbf{L}_2\mathbf{U}\mathbf{L}_1^{-1}$ of $\mathbf{U}$ must be employed in Eq. (6) along with the normalized points and epipole. The normalized epipole can be recovered from the normalized fundamental matrix $\bar{\mathbf{F}} = \mathbf{L}_2^{-T}\mathbf{F}\mathbf{L}_1^{-1}$. After the application of DLT, the computed homography estimate $\bar{\mathbf{V}}$ needs to be denormalized using ${\mathbf{L}_3}^{-1}\bar{\mathbf{V}}\mathbf{L}_2$.

In practice, the set of available matching point triplets is almost certain to contain errors due to false matches and errors in the localization of image corners. Consequently, in order to prevent such errors from corrupting the computed homography estimate, the group of DLT constraints should be employed within a robust regression framework. In our case, the Least Median of Squares (LMedS) [27] robust estimator is employed to iteratively sample random sets of nine constraints, recover an estimate of matrix $\mathbf{V}$ from each of them and find the estimate that is consistent with the majority of the available constraints. To ensure that those random sets arise from points having a good spatial distribution over the image, random sampling is based on the bucketing technique of [28]. Finally, $\mathbf{V}$ is recomputed using least squares on the set of constraints having the largest support, i.e. the LMedS inliers.

3.2 Reducing the DOFs Involved in Homography Chaining

In the following, the basic method of the previous section will be revised, aiming to derive a model having fewer, therefore easier to estimate, degrees of freedom (i.e. free variables). As already mentioned, the entire group of all possible homography matrices between two images lies in a subspace of dimension four, spanned by the primitive homographies of Eq. (1). According to Eq. (2), knowledge of those homographies allows any other homography $\mathbf{H}$ to be expressed as a linear combination encompassing four scalars λ_i. This implies that when the primitive homographies for frames I_2 and I_3 have been computed, the rows $\mathbf{v}_i^T$ of matrix $\mathbf{V}$ in Eqs. (8) depend on four rather than nine parameters. The process described in section 3.1 can be slightly modified to estimate the coefficients λ_i making up $\mathbf{V}$ instead of directly estimating the latter. In other words, the linear estimation process that has been described above can be performed with four rather than nine unknowns. This reduction in the dimensionality of the problem is very important since fewer degrees of freedom (DOFs) entail less computation time for the homography and typically more accurate estimates. Fewer DOFs involve the solution of smaller systems and require less iterations (i.e. samples taken) to find a solution with a given confidence level when embedded within random sampling schemes such as LMedS or RANSAC [29]. It was found experimentally that the execution time for plane tracking using the formulation involving the λ_i is by an order of magnitude shorter than that required when estimating $\mathbf{V}$ directly.

The procedure outlined in sections 3.1 and 3.2 for estimating the tracked plane homography from an overdetermined set of linear constraints minimizes an algebraic error term with no physical meaning. Therefore, this homography estimate can be improved by refining it using a non-linear minimization process that involves a geometric criterion. Letting $d(\mathbf{x},\ \mathbf{y})$ represent the Euclidean distance between the inhomogeneous points represented by $\mathbf{x}$ and $\mathbf{y}$, the non-linear refinement minimizes the following sum of squared distances

$$\sum_{i=1}^{N}\left(d(\mathbf{x}_i^{''},\ \mathbf{V}\mathbf{x}_i^{'}+\kappa_i\mathbf{e}^{''})^2 \quad + \quad d(\mathbf{x}_i^{'},\ \mathbf{V}^{-1}\mathbf{x}_i^{''}-\frac{||\mathbf{x}_i^{''}||}{||\mathbf{V}\mathbf{x}_i^{'}+\kappa_i\mathbf{e}^{''}||}\kappa_i\mathbf{V}^{-1}\mathbf{e}^{''})^2\right) \quad (10)$$

with respect to the four coefficients making up $\mathbf{V}$ as a linear combination of the primitive homographies of Eq. (1). This criterion involves the symmetric transfer error between actual and transferred points in the two images and is minimized by applying the Levenberg-Marquardt iterative algorithm as implemented by MINPACK's LMDER routine [30], initialized with the least squares estimate computed from the LMedS inliers. To safeguard against point mismatches, the non-linear refinement is performed using only the point features that correspond to inliers of the LMedS homography estimate. The complete algorithm for homography chaining after incorporating the non-linear refinement step, is summarized in pseudocode in Fig. 1.

Having presented the basic 3-frame chaining operation, it is straightforward to extend it to handle a sequence of more than three views. For example, in order to track the plane in a new image I_4, the homography $\mathbf{V}$ computed in the previous step between frames I_2 and I_3 becomes the new $\mathbf{U}$ for the triplet I_2, I_3 and I_4. Note also that the epipolar geometry of frames I_2 and I_3 has been computed during the previous iteration, therefore only the epipolar geometry between frames I_3 and I_4 needs to be estimated during this step. A final remark concerning the extension of the chaining operation to more than three frames is that the estimation of $\mathbf{V}$ can benefit from point trajectories that are longer than three frames: If, for example, a four-frame point trajectory is available for images I_1, I_2, I_3 and I_4, the constraints generated by the triplet I_1, I_3 and I_4 can be combined with those arising from I_2, I_3 and I_4. This variant of chaining from multiple triplets can be carried out by maintaining a small moving window of past frames.

Input: *3 successive frames I_1, I_2, I_3 and the homography* $\mathbf{U}$ *induced by a plane* Π *between I_1, I_2*

Output: *The homography* $\mathbf{V}$ *induced by* Π *between I_2, I_3*

1. *Extract and match point features across I_1, I_2 and I_3*
2. *Normalize the coordinates of matched points in each frame*
3. *Estimate the fundamental matrices between frames I_1, I_2 and I_2, I_3 along with the epipoles*
4. *Use Eq. (6) with frames I_1, I_2 to compute the κ_i and then use I_2 and I_3 to form the DLT constraints*
5. *Compute the primitive homographies of frames I_2 and I_3 by employing the corresponding epipolar geometry*
6. *Use the LMedS robust estimator to identify the estimates of λ_i that give rise to an estimate of* $\mathbf{V}$ *supported by the majority of the DLT constraints of Eq. (9)*
7. *Reestimate the λ_i using all DLT constraints corresponding to point inliers identified in step 6*
8. *Starting with the estimate of step 6, refine the λ_i and thus* $\mathbf{V}$*, using Levenberg-Marquardt non-linear minimization of Eq. (10) over the LMedS inliers in step 6*

Fig. 1. An overview of the proposed homography chaining operation; see text for details.

4 Camera Tracking

The homographies of a tracked plane that are estimated as described in the preceding section, can serve as the input to batch, plane-based reconstruction methods such as [12,13]. In this section, the successive plane homographies serve as the basis of a faster, more straightforward incremental reconstruction scheme. The symbols $\mathbf{H}_{i,j}$ and $\mathbf{e}_{i,j}$ will be used to denote respectively the tracked plane homography and the epipole in I_j for the image pair I_i and I_j. Let $\mathbf{H}_{1,2}$ be the homography induced by some plane between the first two images; section 5 discusses plane selection in more detail. Assume also that using the method outlined in section 3, plane homography $\mathbf{H}_{2,3}$ has been

estimated from the matching triplets among images I_1, I_2 and I_3. Recalling that these homographies are, by computation, scale compatible with the corresponding epipoles, Eq. (3) yields the image projections of a 3D point $\mathbf{X}$ as $\mathbf{x} \simeq \mathbf{H}_{2,1}\mathbf{x}^{'} + \kappa\mathbf{e}_{2,1}$ and $\mathbf{x}^{''} \simeq \mathbf{H}_{2,3}\mathbf{x}^{'} + \kappa\mathbf{e}_{2,3}$, implying that $\mathbf{X} \simeq \left[\mathbf{x}^{'T},\ \kappa\right]^T$. Therefore, a set of consistent (i.e. defined up to the same projective transformation) projective camera matrices in canonical form for the three views is given by [4,18]

$$\mathbf{P}_1 = [\mathbf{H}_{2,1} \mid \mathbf{e}_{2,1}]\,, \quad \mathbf{P}_2 = [\mathbf{I} \mid \mathbf{0}]\,, \quad \mathbf{P}_3 = [\mathbf{H}_{2,3} \mid \mathbf{e}_{2,3}]\,, \tag{11}$$

where $\mathbf{I}$ denotes the 3×3 identity matrix. Since it is customary to have the world's coordinate frame aligned with the initial camera location, application of an appropriate 3D projective mapping can transform Eqs.(11) so that $\mathbf{P}_1$ becomes equal to $[\mathbf{I} \mid \mathbf{0}]$. Indeed, right multiplication of the camera matrix $[\mathbf{A} \mid \mathbf{b}]$ by the 4×4 matrix $\mathbf{M}$ given by

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\,\mathbf{b} \\ \mathbf{0}^T & 1 \end{array}\right], \tag{12}$$

makes the former equal to $[\mathbf{I} \mid \mathbf{0}]$. Therefore, to make $\mathbf{P}_1$ equal to $[\mathbf{I} \mid \mathbf{0}]$, the projection matrices in Eq. (11) should be right multiplied by the matrix given by Eq.(12) for $\mathbf{A} = \mathbf{H}_{2,1}$ and $\mathbf{b} = \mathbf{e}_{2,1}$, which, taking into account that $\mathbf{H}_{j,i}^{-1}\ \mathbf{e}_{j,i} = \mathbf{e}_{i,j}$ and $\mathbf{H}_{2,3}\ \mathbf{H}_{1,2} = \mathbf{H}_{1,3}$, yields after some algebraic manipulation

$$\mathbf{P}_1 = [\mathbf{I} \mid \mathbf{0}]\,, \quad \mathbf{P}_2 = [\mathbf{H}_{1,2} \mid \mathbf{e}_{1,2}]\,, \quad \mathbf{P}_3 = [\mathbf{H}_{1,3} \mid \mathbf{e}_{1,3}]\,. \tag{13}$$

Suppose now that by employing the plane tracker for the image triplet I_2, I_3 and I_4, the homography $\mathbf{H}_{3,4}$ induced by the tracked plane has been estimated. If $\mathbf{P}_3$ were equal to $[\mathbf{I} \mid \mathbf{0}]$, a projection matrix for I_4 consistent with the projection matrices of the previous three images would simply be $[\mathbf{H}_{3,4} \mid \mathbf{e}_{3,4}]$. Here, the former should be right multiplied by the matrix given by Eq.(12) for $\mathbf{A} = \mathbf{H}_{1,3}$ and $\mathbf{b} = \mathbf{e}_{1,3}$, to account for the fact that the employed coordinate system coincides with that of I_1. Thus, $\mathbf{P}_4$ is equal to $[\mathbf{H}_{3,4}\ \mathbf{H}_{1,3} \mid \mathbf{H}_{3,4}\ \mathbf{e}_{1,3} + \mathbf{e}_{3,4}]$, which in turn is simplified to $[\mathbf{H}_{1,4} \mid \mathbf{e}_{1,4}]$. Clearly, the procedure for obtaining $\mathbf{P}_4$ just described, can be generalized to incorporate the projection matrix $\mathbf{P}_n$ corresponding to any image I_n with $n > 4$. Thus, $\mathbf{P}_n$ is given by a recursive formula involving homographies and epipoles defined in successive images, namely

$$\mathbf{P}_n = \mathbf{H}_{n-1,n}\,\mathbf{P}_{n-1}\ +\ [\mathbf{0} \mid \mathbf{e}_{n-1,n}]\,, \quad \text{with}\ \ \mathbf{P}_{n-1} = [\mathbf{H}_{1,n-1} \mid \mathbf{e}_{1,n-1}]\,. \tag{14}$$

The camera projection matrices recovered with the aid of the homographies induced by a tracked plane are defined in a projective coordinate frame. Hence, the camera intrinsic calibration parameters are necessary for upgrading those projective camera matrices to Euclidean. In order to increase stability and, at the same time, relieve the camera tracker from the computational burden associated with their estimation, the camera intrinsics are assumed here to be constant and known, either as a result of a self-calibration algorithm or of an off-line, grid based calibration method [31]. The matrix of the intrinsic calibration parameters has the following well-known form [4]:

$$\mathbf{K} = \begin{bmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

The parameters α_u and α_v correspond to the focal distances in pixels along the axes of the image, θ is the angle between the two image axes, $-\alpha_u \cot\theta$ is the camera skew and (u_0, v_0) are the coordinates of the image principal point. Given $\mathbf{K}$, a projective camera matrix can be upgraded to Euclidean by right multiplication with the 4×4 matrix defined as

$$\mathbf{H}_{euc} = \left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \\ -\mathbf{p}^T\,\mathbf{K} & 1 \end{array}\right], \tag{16}$$

where $\mathbf{p}$ is such that the coordinates of the plane at infinity in the projective reconstruction are given by $\left[\mathbf{p}^T, 1\right]^T$. Following this, the 3D translation and rotation corresponding to the Euclidean camera matrix can be estimated with RQ decomposition [4].

Hitherto, knowledge of the plane homographies has permitted the direct recovery of a set of consistent projective camera matrices, without the need for 3D structure estimation and resectioning. A sparse representation of 3D scene structure in the form of a point cloud can be built-up incrementally as new image triplets become available. More specifically, when the camera matrices for a new image triplet have been estimated, the 3D coordinates of points that became visible in the new triplet can be recovered with the aid of a triangulation algorithm [32]. The main problem that needs to be addressed by all triangulation algorithms is the fact that errors in the estimates of camera matrices as well as mislocalized image corners, result in making skew the back-projected 3D lines defined by the camera optical centers and the corresponding image projections. This work employs a triangulation method that exploits the knowledge of the camera intrinsic parameters to express two back-projected 3D lines in an Euclidean coordinate frame in which the notion of

length becomes meaningful. Then, a 3D point is reconstructed as the midpoint of the minimal length straight line segment whose endpoints lie on the skew back-projected lines [33]. Since an image triplet gives rise to three different image pairs, the reconstructed point is taken here to be the median of the three 3D points reconstructed from the triplets' image pairs. To avoid reconstructing points arising from triplets $(\mathbf{x},\ \mathbf{x}^{'},\ \mathbf{x}^{''})$ involving spurious matches, the projection matrices of Eq. (11) are used to compute the corresponding trifocal tensor using a closed form formula. More specifically, given the canonical projection matrices of Eq. (13), the corresponding trifocal tensor in matrix notation is made up from the set of matrices $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$, where

$$\mathbf{T}_k = \mathbf{H}_{1,2}^k \, \mathbf{e}_{1,3}^T \ - \ \mathbf{e}_{1,2} \, {\mathbf{H}_{1,3}^k}^T, \quad k = 1 \ldots 3 \tag{17}$$

and $\mathbf{H}_{i,j}^k$ denotes the k-th column of matrix $\mathbf{H}_{i,j}$ [4]. Then, using point matches $\mathbf{x},\ \mathbf{x}^{'}$ from the first two views along with the tensor computed with Eq. (17), permits the elimination of point triplets whose transferred third image point lies at a distance further than a certain threshold from the point $\mathbf{x}^{''}$. 3D points are often visible in more than three successive frames which give rise to more than one image triplets. Each time a 3D point that has been reconstructed from one or more previous triplets is seen in a new triplet, an estimate of its Euclidean coordinates is recovered from the new triplet. Then, this estimate is used to refine the existing one through recursive median filtering [34].

By trading some speed for increased accuracy in camera tracking, structure information can be used in a local resectioning framework for evenly distributing the camera tracking error among consecutive images belonging to the same sliding time window. Assume that a narrow window of the image points matched among the W most recent frames numbered $n - W + 1 \ldots n$ is maintained, with n being the most recent frame. Also, suppose that M Euclidean 3D points $\mathbf{X}_j,\ j = 1 \ldots M$ are visible in some of these W frames and let $\mathbf{R}_i$ and $\mathbf{t}_i$ be the estimates of camera orientation and position for frame i, $i = n - W + 1 \ldots n$. Then, each Euclidean projection matrix $\mathbf{P}_i$ is equal to $\mathbf{K}\,[\mathbf{R}_i \mid \mathbf{t}_i]$. Local resectioning amounts to refining all motion parameters $\mathbf{R}_i$ and $\mathbf{t}_i$ so that the sum of squared image distances between reprojected and detected, actual image points is minimized, namely

$$\min_{\mathbf{R}_i, \mathbf{t}_i} \sum_{i=\max(n-W+1,\ 1)}^{n} \ \sum_j \ d(\mathbf{P}_i \, \mathbf{X}_j,\ \mathbf{x}_j^i)^2 \quad \text{with} \quad \mathbf{P}_i = \mathbf{K}\,[\mathbf{R}_i \mid \mathbf{t}_i], \tag{18}$$

where $d(\mathbf{x},\ \mathbf{y})$ denotes the Euclidean distance between the inhomogeneous image points represented by $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{x}_j^i$ is the detected projection of point j in image i, $i \in \{n - W + 1, \ldots, n\}$. Rotation matrices $\mathbf{R}_i$ are parametrized using quaternions. The gauge issue [14] (i.e. the choice of the coordinate system for expressing the reconstruction) is resolved by fixing the projection matrix of

the first frame equal to $\mathbf{K}\,[\mathbf{I}_{3\times 3} \mid \mathbf{0}]$ and not altering it during the minimization. To keep the computational overhead of the minimization low, observe that Eq. (18) is minimized with respect to the 3D motion only and not with respect to the 3D structure as well, as is the case with bundle adjustment. The minimization of Eq. (18) is performed with the aid of a non-linear least squares algorithm [35] that is initialized with the motion parameters computed directly from the estimates of the camera matrices obtained with the aid of the tracked plane homographies. Finally, since the projective camera matrix for I_n will be needed for determining the camera motion of subsequent frames, it is recomputed as the product of the Euclidean camera matrix amounting to the refined camera motion by the inverse of matrix $\mathbf{H}_{euc}$ from Eq. (16). The complete algorithm for camera tracking using the homographies of a tracked plane is listed in pseudocode in Fig. 2.

5 Using a Quasi-Metric Virtual Plane

As explained in section 4, camera tracking requires a 3D plane to be tracked over an image sequence. Although planes abound in man-made environments and fully automatic methods exist for detecting them [36], it would be preferable if the proposed method did not rely on the assumption of a physical 3D plane being present in the scene. To achieve this, recall that in order for plane tracking to commence, the homography induced by the tracked plane between the first two frames of the sequence must be available. Apart from this requirement, however, no other information regarding the plane must be supplied. The tracked plane can actually be a virtual one, i.e. not corresponding to a physical 3D plane present in the scene. All that is needed is that the plane's homography is compatible with the underlying epipolar geometry. The rest of this section describes how can such a virtual plane be selected.

Let $\mathbf{x}_i,\ \mathbf{x}_i^{'},\quad i = 1,\ldots,N$ be a set of matching point pairs in the first two frames. The virtual plane can be chosen so that it closely approximates the set of available point matches. In other words, the virtual plane is situated "in-between" the 3D space points giving rise to the set of available point matches. Assuming that the epipolar geometry corresponding to the two images has been estimated, we therefore seek the planar homography $\mathbf{H}$ for which the contribution of the parallax term in Eq. (3) is as little as possible. As it has been explained in section 2, any planar homography defined between two images can be expressed as a linear combination of the four primitive homographies of Eq. (1). The sought $\mathbf{H}$ is thus computed from the coefficients $\mu_j,\quad j = 1,\ldots,4$ minimizing

$$\sum_{j=1}^{4}(\mu_j\,\mathbf{H}_j)\,\mathbf{x}_i \simeq \mathbf{x}_i^{'},\quad i = 1,\ldots,N \tag{19}$$

Input: *Incrementally acquired successive frames* I_1, I_2, ... *and the camera intrinsic parameters matrix* $\mathbf{K}$

Output: *The Euclidean camera motion* $\mathbf{R}_n, \mathbf{t}_n$ *corresponding to each frame*

---

1. *Initialization: Select a plane in* I_1 *and* I_2 *and estimate the homography* $\mathbf{H}_{1,2}$ *(see section 5)*
2. *For each image* I_n, $n > 2$ *do*

   2.a Use the algorithm of Fig. 1 on triplet $\tau = (I_{n-2}, I_{n-1}, I_n)$ *to track the plane between images* I_{n-1} *and* I_n, *thus estimating* $\mathbf{H}_{n-1,n}$ *and* $\mathbf{e}_{n-1,n}$

   2.b if n=3 then

   Use Eqs. (13) to compute a canonical set of projection matrices for images I_1, I_2 *and* I_3

   else

   $\mathbf{P}_n := [\mathbf{H}_{n-1,n}\ \mathbf{H}_{1,n-1} \mid \mathbf{H}_{n-1,n}\ \mathbf{e}_{1,n-1} + \mathbf{e}_{n-1,n}] \equiv [\mathbf{H}_{1,n} \mid \mathbf{e}_{1,n}]$,

   where $[\mathbf{H}_{1,n-1} \mid \mathbf{e}_{1,n-1}]$ *is the projection matrix for* I_{n-1}

   endif

   2.c Upgrade $\mathbf{P}_n$ *to Euclidean using the intrinsics* $\mathbf{K}$ *and Eq. (16)*

   2.d Recover $\mathbf{R}_n, \mathbf{t}_n$ *using the RQ decomposition of the Euclidean* $\mathbf{P}_n$

   2.e Use $\mathbf{P}_{n-2}$, $\mathbf{P}_{n-1}$ *and* $\mathbf{P}_n$ *to reconstruct 3D points such that triplet* τ *is the first one to contain their projections in all its images*

   2.f Use $\mathbf{P}_{n-2}$, $\mathbf{P}_{n-1}$ *and* $\mathbf{P}_n$ *to refine previously reconstructed 3D points whose projections appear in* τ *and at least one earlier triplet*

   2.g Refine $\mathbf{R}_n, \mathbf{t}_n$ *using Eq. (18)*

   2.h Recompute the projective $\mathbf{P}_n$ *using* $\mathbf{K}$ *and the refined* $\mathbf{R}_n, \mathbf{t}_n$

   2.i Output $\mathbf{R}_n, \mathbf{t}_n$

   endfor

Fig. 2. Camera tracking based on tracked plane homographies; see section 4 for details.

Each of the available point matches provides two independent linear constraints for the μ_j, therefore $N \geq 2$ matches yield an overdetermined system from which the μ_j can be estimated using robust least squares. The LMedS

robust estimator is again employed to find the set of μ_j corresponding to the homography minimizing Eq. (19) for at least 70% of the available matches; the estimated μ_j are then refined by applying least squares to the constraints corresponding to the LMedS inliers. The plane computed in this manner is referred to as "quasi-metric" in [24] and gives rise to a projective reconstruction of space that is characterized by a small amount of projective distortion.

6 Implementation and Experimental Results

In the following, subsection 6.1 provides some details regarding the practical implementation of the proposed camera tracking technique. Since the performance of plane tracking is crucial for the overall performance of camera tracking, subsection 6.2 presents experimental results demonstrating its effectiveness. Quantitative and qualitative experimental validation results regarding the complete camera tracking system are reported in subsections 6.3 and 6.4 respectively.

6.1 Implementation Issues

A prototype of the proposed camera tracking method has been implemented in C. Linear algebra numerical operations were carried out using LAPACK [37]. Corresponding image points that are required as input are determined between pairs of consecutive images as follows. First, the Harris interest operator [38] is employed to extract corner features with subpixel accuracy from each image. Then, the similarity of corners located within a maximum disparity search window is assessed using the zero-mean normalized cross-correlation of image templates and a set of preliminary matches is established by solving an assignment problem on a flow graph [39]. Preliminary matches are used to robustly estimate the fundamental matrix with the seven point algorithm [28]. Following this, final corner matches are determined by employing guided matching based on the estimated epipolar geometry [28]. Eventually, the median flow filter algorithm of [38] is employed to cope with the fact that guided matching can produce mismatches that accidentally agree with the epipolar geometry. This algorithm has proven to be very effective in removing many of the outliers contained in the set of pairwise matches. The image triplet matches are drawn from the two underlying sets of successive pair matches. Epipoles are computed by finding the kernels of the estimated fundamental matrices. A technique that directly derives the epipoles from point matches [40] has also been evaluated and was found to produce similar results. The computational overhead incurred by the LMedS estimator can be reduced by noting that the computation of the median of the squared residuals that is needed at each

iteration can be attained without resorting to sorting them. Instead, an algorithm that finds the k-th largest out of n numbers can be employed [41]. This algorithm has a time complexity of $O(n)$, which is lower than the $O(n \log n)$ complexity of the best serial sorting algorithm.

At this point, it should be noted that the estimation of epipolar geometry in a video sequence whose consecutive frames are very close together is ill-conditioned. Therefore, the previously outlined corner matching algorithm implicitly assumes that the baseline of the images to be matched is not negligible. Additionally, it is well known that such a baseline permits more accurate reconstruction. On the other hand, the baseline cannot be increased too much, since in that case the number of corners that can be successfully matched between images becomes very small. Thus, when the image sequence to be tracked is characterized by small interframe motion, one has to select appropriate *keyframes*, i.e. frames in the image sequence that are sufficiently apart in time from each other, so that sufficient translational camera motion exists between them and their effective baseline is suitable. Despite that automatic methods such as [42,43] exist for determining keyframes, in this work we have chosen to time subsample the image sequence by a factor of k, i.e. use every k-th frame for computing camera motion. Note that keyframing is necessary only when applying the proposed method to prerecorded, video-rate benchmark sequences. When performing on-line tracking, the latency of the camera tracker provides enough time for the camera to translate sufficiently between images. In the case that the camera motion needs to be estimated for frames among the keyframes, this can be achieved using a pose estimation procedure based on resectioning using the 3D structure computed from keyframes [44,11].

The current implementation of the plane tracker performs chaining based on constraints arising from only three frames at a time. Possible camera lens imperfections (e.g. radial distortion) are neglected. The intrinsic camera parameters in Eq. (15) were determined by using the auto-calibration method described in [31]. This method exploits constraints arising from a simplified version of the Kruppa equations that is derived with the aid of SVD of pairwise fundamental matrices [45]. Throughout all experiments, the plane homography between the first two images that is necessary for bootstrapping plane tracking was determined as described in section 5. The triangulation technique described in section 4 was found experimentally to perform better compared to the three view triangulation method of Avidan and Shashua [24], which involves a modification of the corners' image coordinates that ensures that the trifocal constraints are satisfied exactly. The size of the sliding window W in Eq. (18) is set to 7 frames and the minimization is carried out using the NL2SOL algorithm [35], as implemented by the DN2G routine in the PORT3 library from Bell Labs. Compared to the Levenberg-Marquardt algorithm as implemented by the LMDER routine [30], NL2SOL was found in this case to

converge faster while producing results of similar accuracy. The NLSCON non-linear least squares routine [46] has also been evaluated and yielded results slightly worse than those of DN2G. The jacobians of Eq. (10) and Eq. (18) that are necessary for the non-linear minimizations have been computed analytically with the aid of MAPLE's symbolic differentiation facilities. Analytical differentiation was preferred over numerical one owing to its better performance and convergence characteristics.

6.2 Plane Tracking Experiments

This section presents results from three experiments that demonstrate the performance of plane tracking, which constitutes a key component of the proposed method. To aid in the visual interpretation of results, 3D planes that are physically present in the scene have been employed in all experiments. The spatial extend of the employed planes has been defined manually using a polyline in the first frame. Following this, the plane homography between the first two images that is necessary for bootstrapping plane tracking (i.e. $\mathbf{U}$ in section 3.1), is estimated from the point matches lying within the specified polyline. Alternatively, plane tracking could have been bootstrapped by applying to the first pair of images an automatic plane detection algorithm such as [36].

The first experiment was performed on the well-known "basement" indoors image sequence, two frames of which (namely 0 and 8) are shown in Figs. 3(a) and (b). This sequence consists of 11 512×512 frames acquired by a camera mounted on a mobile robot as it approached the scene while smoothly turning left. The plane corresponding to the right corridor wall was tracked from frame 0 to frame 8 using the proposed method. Then, by employing the estimated homography, the right wall from frame 0 was warped towards frame 8. Fig. 3(c) shows the warped wall stitched with frame 8. A second plane, namely the one corresponding to the floor, was also tracked between frames 0 and 8. Fig. 3(d) shows the result of warping the floor plane from frame 0 towards frame 8 and stitching them together. As it is clear from the results, the accuracy of the homographies estimated using the proposed method is satisfactory in both cases. Excluding the time required to detect and match corners between successive frames, the average running times for tracking the wall and floor planes in the whole sequence were respectively 50 and 53 ms per frame on an Intel P4@2.5 GHz laptop.

In order to quantitatively evaluate the performance of plane tracking, the floor plane was tracked from frame 0 to frame 10 and then back to frame 0, reversing the order of intermediate frames. This effectively simulates a camera trajectory that is closed, i.e. ends at the location where it started. Composing the pairwise

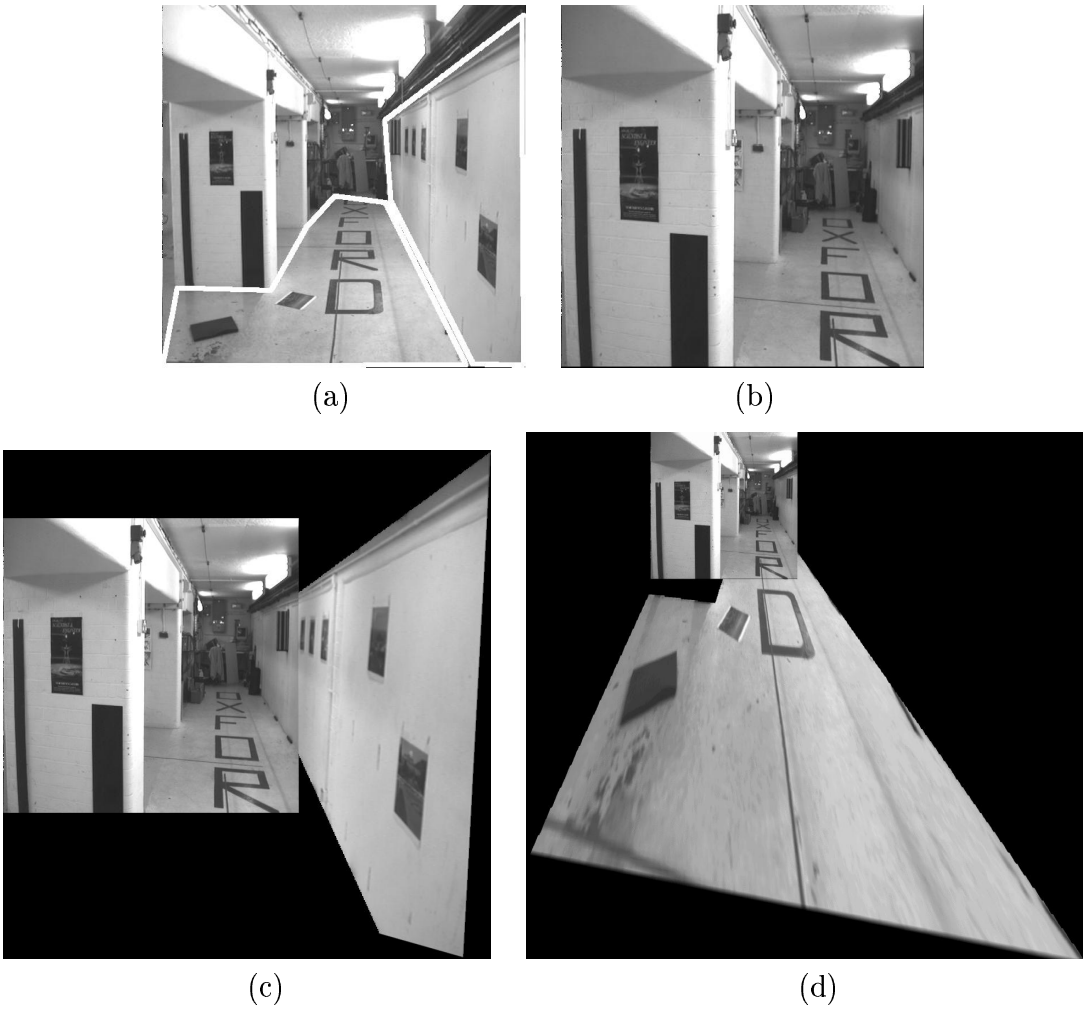


Fig. 3. (a), (b) two views of a basement (courtesy of the Oxford Visual Geometry Group). The two polylines in (a) delineate the planar regions tracked over the whole sequence. (c) right wall warped and stitched with (b), (d) floor warped towards and stitched with (b); see text for explanation.

homographies estimated by the plane tracker, the floor's homography from the first frame through the last and back to the first can be estimated. Ideally, this homography should be equal to the identity matrix. In practice, the deviation in the position of floor points transferred using this homography from their actual locations in the first frame, indicates the accuracy of plane tracking. The root mean square (RMS) error corresponding to the 91 transferred floor points was found to be 19.3 pixels, corresponding to an average RMS error of 0.91 pixels for each of the 21 frames involved in tracking. However, since certain floor points correspond to mismatches or poorly localized corners, a more appropriate error measure is given by the root median square (RMedS) error, which was found to be equal to 8.47 pixels or on average 0.40 pixels per tracked frame.

The second experiment employs another well-known image sequence, the first

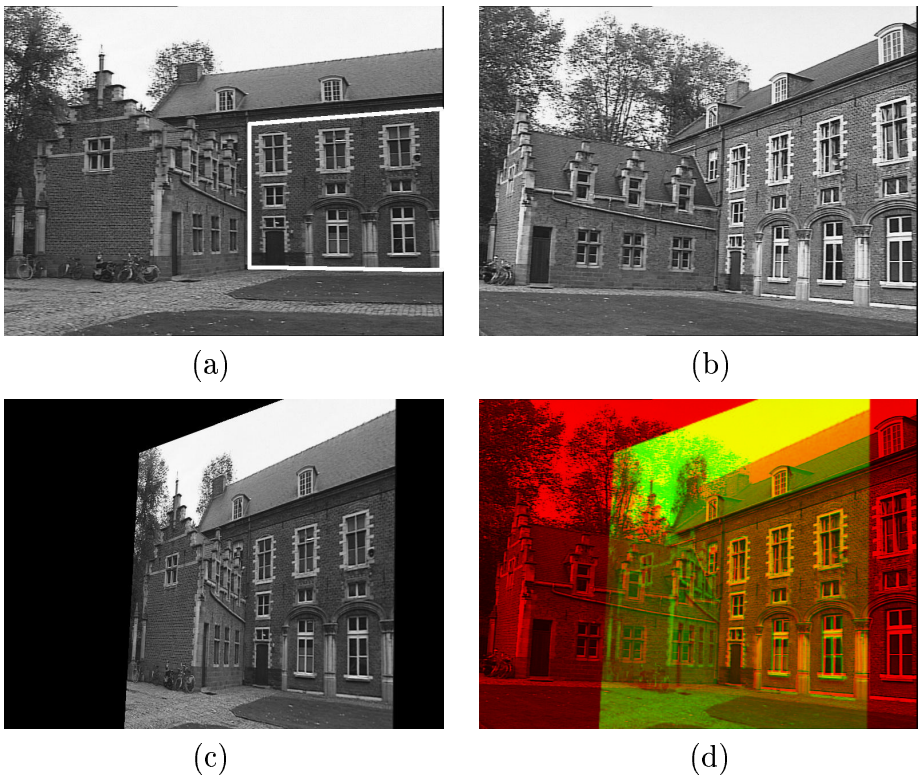


Fig. 4. (a), (b) first and last images from the Arenberg castle sequence (courtesy of the University of Leuven VISICS Group), (c) first image warped towards the second using the estimated homography; notice the distortion due to parallax on the left part of the image and (d) warped image in (c) superimposed on (b).

and last frames of which are shown in Figs. 4(a) and (b). The sequence depicts the Arenberg castle in Belgium and consists of 22 768 × 576 frames acquired with a handheld camera. Using the proposed method and the image frames between those in Figs. 4(a) and (b), the 3D plane defined by the rightmost wall (see Fig. 4(a)) was tracked throughout the sequence. Fig. 4(c) illustrates the result of warping the first frame towards the last using the estimated homography. To aid in the evaluation of this result, Fig. 4(d) shows it superimposed on Fig. 4(b), using different color channels for each image. As can be clearly seen, image warping according to the estimated homography successfully registers the plane's image in Fig. 4(a) with that in Fig. 4(b). In this case, the average running time for plane tracking was 66 ms per frame. The plane of the right wall was again tracked from the first to the last frame and back (for a total of 43 frames) and the RMS and RMedS errors in this case were 31.7 and 18.3 pixels, amounting to average errors of 0.73 and 0.43 pixels per frame respectively.

The third experiment employs a sequence depicting the remains of a roman tavern (*thermopolium*) in ancient Pompeii. This sequence is quite shaky, due to the fact that it was shot with a camcorder as the operator walked approaching the tavern. It consists of 80 720 × 576 frames, the first and last of which are

(a) (b)

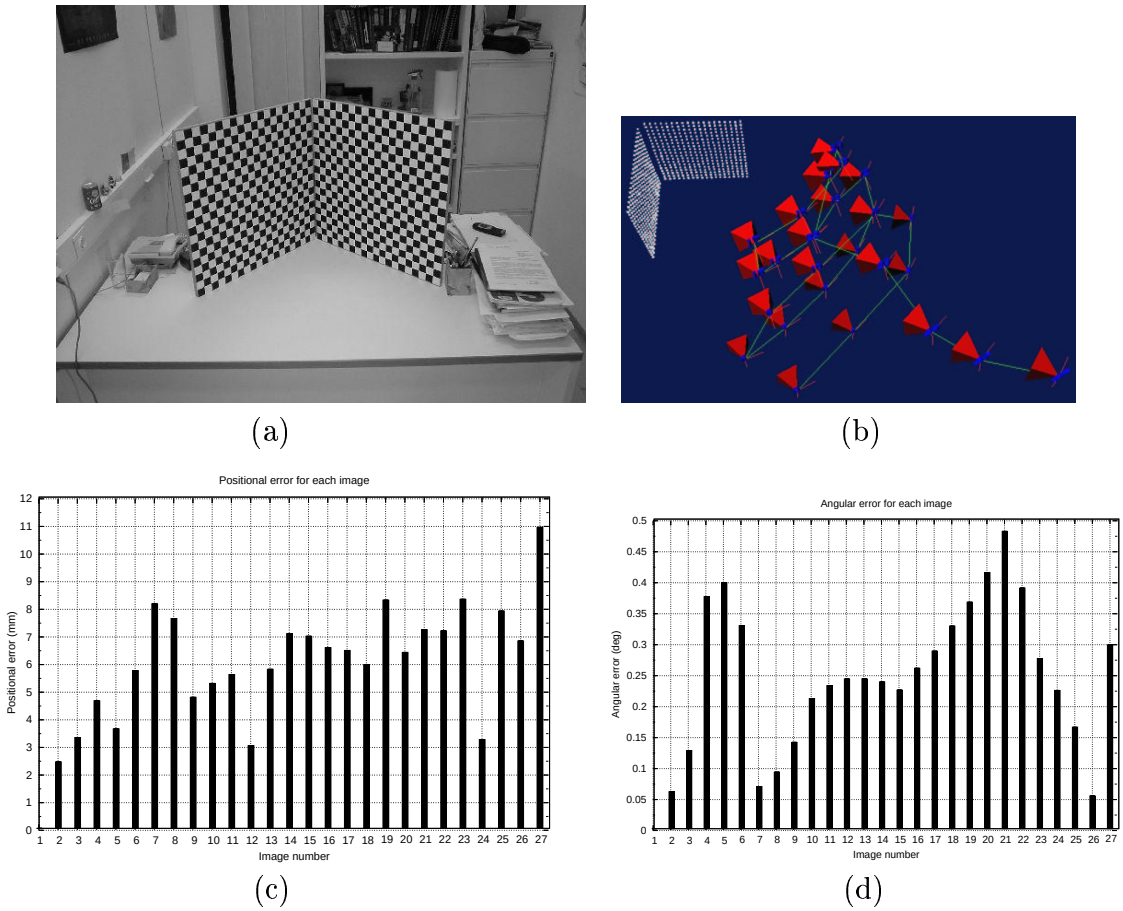
(c)

Fig. 5. (a), (b) first and last frames from a clip imaging the remains of a Pompeian tavern, (c) fountain front face warped and stitched with (b).

shown in Figs. 5(a) and (b) respectively. In this experiment, keyframes were determined by time subsampling the original sequence by four and the plane defined by the front face of the foreground fountain was tracked up to the last frame. Fig. 5(c) shows the warped plane from the first plane stitched with the last. The average running time for plane tracking was 52 ms per frame. The RMS and RMedS errors computed after tracking the plane of the fountain front face back and forth (39 frames total), were 50.3 and 34.1 pixels respectively, corresponding to average errors of 1.28 and 0.87 pixels per frame respectively.

6.3 Quantitative Camera Tracking Experiments

In order to quantitatively assess the accuracy of the proposed camera tracking method, it is necessary to employ image sequences for which the camera motion can be accurately determined with independent means. This motion can then be compared with the one estimated by the proposed method. In this work, we have chosen to base our comparison on the motion estimates obtained through two different approaches, namely grid-based calibration and 3D reconstruction with the aid of a state of the art structure and motion estimation system.



(a) (b) (c) (d)

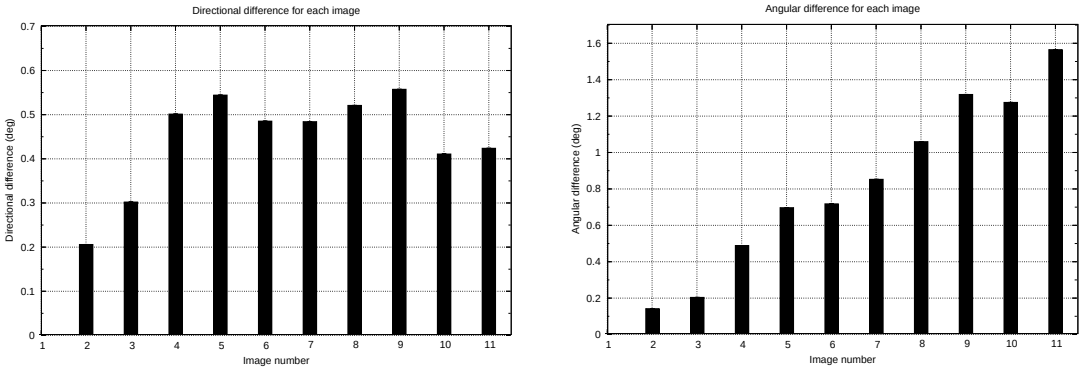
Fig. 6. (a) a frame of a sequence depicting a calibration object consisting of two planes, (b) view of the 3D reconstruction and the camera trajectory, (c) and (d) the positional and angular errors (in millimeters and degrees resp.) of the camera trajectory computed by the proposed method. The 3D camera locations are indicated in (b) with red pyramids whose apexes are located on the camera optical centers; the green curve connecting the optical centers corresponds to the recovered camera trajectory whereas the white dots illustrate the reconstructed 3D points cloud.

More specifically, the first quantitative experiment employs a 27 frames, 1280× 960 sequence of a calibration object consisting of two planes as shown in Fig. 6(a). Using Bouguet's matlab calibration toolkit [47], the extrinsic calibration (i.e. camera pose) parameters were determined for each image. The pose of every image was then computed relative to that of the first one, thus effectively aligning the employed world coordinate system with the latter. Following this, the proposed method was applied to the image corners that were extracted during calibration as the intersections of line segments fitted to the chessboard pattern. Fig. 6(b) illustrates the corresponding VRML 3D model that was recovered. For each image, the camera positional and angular errors were computed by comparing the camera positions and orientations obtained from calibration with those obtained from the proposed method. The overall scene scale missing from our reconstruction was inferred from knowledge of

the physical dimensions of the squares in the calibration pattern. Denoting by $(\mathbf{R}_c,\ \mathbf{t}_c)$ the camera pose obtained from calibration and by $(\hat{\mathbf{R}},\ \hat{\mathbf{t}})$ the pose estimated by the proposed method, the positional error was computed as $||\mathbf{t}_c - \hat{\mathbf{t}}||$. The angular error was computed as

$$\arccos(\frac{1}{2}\left[\text{trace}(\mathbf{R}_c^{-1}\,\hat{\mathbf{R}}) - 1\right]), \tag{20}$$

and corresponds to the amount of rotation about a unit vector that transfers $\mathbf{R}_c$ to $\hat{\mathbf{R}}$. Figures 6(c) and (d) show plots of the positional and angular errors, measured in millimeters and degrees, respectively. Clearly, the camera poses computed by the proposed method are very accurate, with the positional and rotational errors remaining below 11mm and 0.5 degrees.



(a) (b)

Fig. 7. The positional (a) and angular (b) differences in degrees between the camera trajectory computed for the basement sequence by the proposed method and that corresponding to VGG's refined reconstruction.

The second experiment compares the results obtained by applying the proposed method to the basement sequence of Fig. 3 against those produced on the same sequence by a well established tensor-based batch approach, namely Oxford's Visual Geometry Group (VGG) structure and motion recovery system [11,48]. Note that the forward translational motion of this particular sequence results in the angle between the triangulating 3D lines being small which, in turn, hinders accurate structure recovery. The comparison was performed on the basis of the Euclidean reconstruction data for the basement sequence that are publicly provided by VGG [3] in the form of projection matrices, 3D points and corresponding image projections. Application of RQ decomposition to VGG's projection matrices revealed that the camera intrinsic calibration parameters pertaining to the reconstruction varied slightly from frame to frame and, most importantly, had considerable non-zero skew. Since our camera calibration method assumes zero skew intrinsic parameters that

[3] See http://www.robots.ox.ac.uk/$\tilde{\text{v}}$gg/data.html

remain fixed throughout the whole sequence, a fair evaluation called for modifying VGG's reconstruction prior to the comparison, as follows.

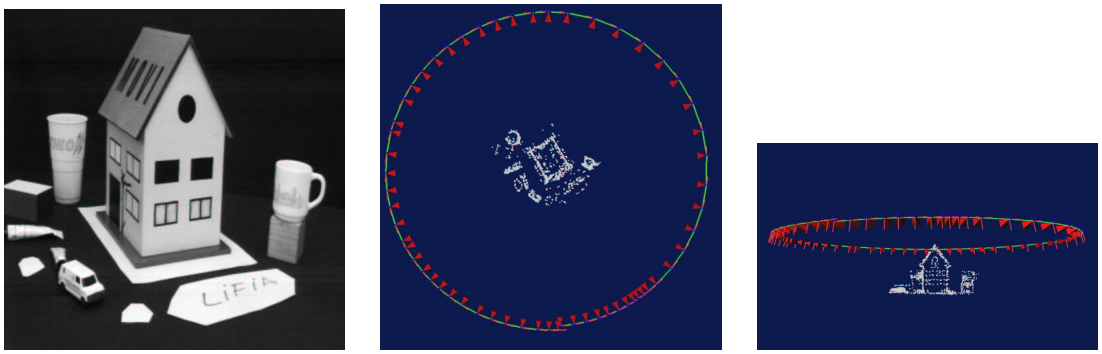
First, the non-zero skew was eliminated by transforming each frame's 2D coordinates through multiplication with a matrix

$$\mathbf{B} = \begin{bmatrix} 1 & \alpha_u/\alpha_v \cos\theta & 0 \\ 0 & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

where α_u, α_v and θ are as defined for Eq. (15). Note that this is equivalent to left multiplying each projection matrix $\mathbf{P}_i = \mathbf{K}_i \, [\mathbf{R}_i \mid \mathbf{t}_i]$ by $\mathbf{B}$, which yields a zero skew intrinsic calibration matrix $\mathbf{BA}$ and leaves the motion parameters $\mathbf{R}_i$ and $\mathbf{t}_i$ unchanged. Following the unskewing of image coordinates, the intrinsic calibration parameters of all frames were fixed to those corresponding to the first frame after unskewing. As expected, assuming fixed intrinsic parameters for all frames resulted in the existing reconstruction being less accurate in terms of the average reprojection error per frame: The latter indeed increased to 34.9 pixels. To account for the new, fixed intrinsics, VGG's reconstruction served as the starting point for a Euclidean bundle adjustment processing step. This step simultaneously refined the 3D structure and camera motion parameters while keeping all images intrinsic parameters constant and known [49]. Bundle adjustment resulted in a small average reprojection error of 0.45 pixels per frame, which indicated that the refined reconstruction was consistent with the constraint of all images having identical intrinsic calibration parameters. Since camera translation is recovered up to an unknown scale factor which is different for the two camera trajectories to be compared, we have chosen to treat translation vectors as directions and measure their difference with their angular separation. The difference between two rotations was again measured using Eq. (20). Figures 7(a) and (b) show graphs of the positional and angular difference between the camera trajectory estimated by our method and that corresponding to VGG's refined reconstruction. Inspection of both graphs indicates that the two trajectories are very similar.

6.4 Using Camera Tracking for Scene Augmentation

As it must be evident from section 6.3, rigorous quantitative evaluation of the performance of camera tracking for an arbitrary image sequence is difficult due to the fact that ground truth for the camera motion is usually unavailable. Therefore, in the case of sequences for which the true camera motion is not precisely known, we have chosen to indirectly evaluate camera tracking



(a) (b) (c)

Fig. 8. (a) the first frame from the "house" sequence (courtesy of the INRIA MOVI Group), (b) and (c) top and side views of the 3D reconstruction and the camera trajectory. Note that the camera trajectory is very close to being a full circle.

from the quality of sequences resulting from matchmoving, that is augmenting the original ones with artificial 3D objects. Accurately estimated camera trajectories should result in the artificial objects exhibiting low drift and jitter in the augmented sequences. To achieve augmentation, the estimated camera trajectories were exported to the 3DSMax graphics package [50] using MaxScript and then the augmented sequences were generated with the aid of 3DSMax's rendering engine that used the original sequence as a background. The initial alignment of the world coordinate systems employed by the camera tracker and 3DSMax was achieved interactively, by manually rotating and translating them until they lined up. The placement of the artificial graphical objects into the scene was guided by the structure information also provided by the camera tracker. Representative results from five of the conducted experiments are given in this section. Sample augmented sequences can be found at `http://www.ics.forth.gr/~lourakis/camtrack/`.

The first experiment was performed on the well known "MOVI house" image sequence, consisting of 119 512×512 frames acquired by a fixed camera as a model house on a turntable made a full revolution around its vertical axis. This is equivalent to the camera making a complete circular orbit around the house. Apart from giving rise to a geometrically regular camera trajectory, this sequence is also characterized by smooth interframe motion. The first frame of the sequence is shown in Fig. 8(a), while Figs. 8(b)-(c) illustrate different views of the VRML 3D model recovered using the proposed method on keyframes corresponding to odd numbered frames (59 in total). As can be seen from Fig. 8(b), the estimated trajectory is very close to being a circle. Figure 9 shows snapshots of the original sequence after augmenting it with the addition of a palm tree. As can be verified from the accompanying videos, the augmenting object has been convincingly merged with the original sequence.

The second experiment employs the "cooks" sequence, also resulting from a

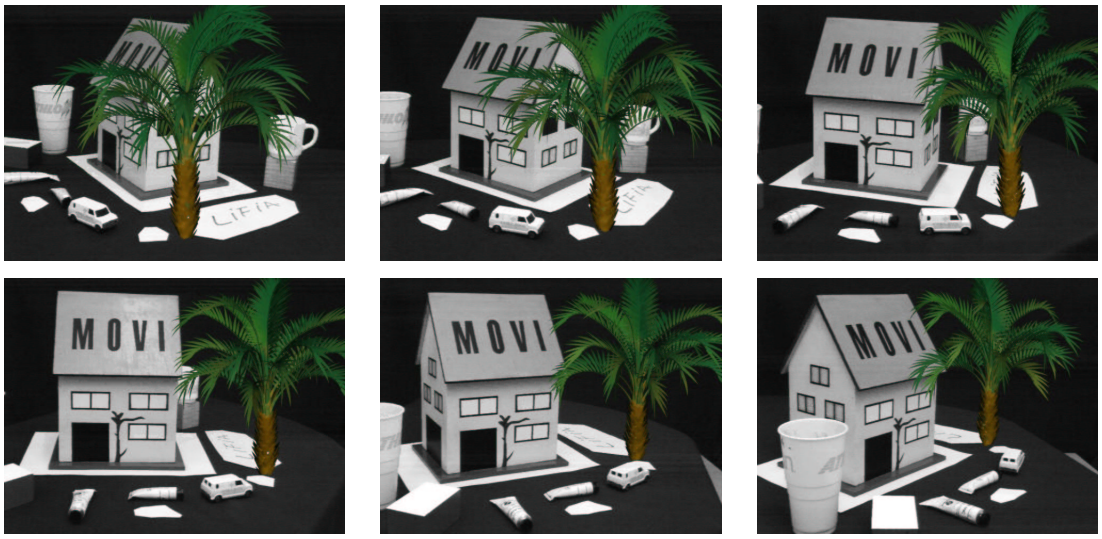


Fig. 9. Snapshots of the augmented "house" sequence corresponding to frames 0, 4, 8, 12, 16 and 20 respectively.

geometrically simple camera movement and whose first and last frames are shown in Figs. 10 (a) and (b). This sequence is recorded using Digital Air's TimeTrack camera[4], that is made up of an array of lenses that simultaneously photograph a scene from different viewpoints. If the entire set of images recorded by all lenses at a specific time instant is played back as a sequence, a viewer has the impression of a camera that appears to move relative to a subject which appears stopped in time. The particular camera model used for shooting the "cooks" sequence had 80 lenses configured in a 66 degree arc spanning 3m with a 2.6m radius and produced images of dimension 720×480. In a sense, a TimeTrack camera provides sequences that are analogous to those using a fixed camera and a turntable, without, however, the limitations of the latter technique. Different views of the VRML 3D model recovered by the proposed method using as keyframes every third frame of the original sequence (27 frames in total) are shown in Figs. 10(c) and (d). The recovered camera trajectory is indeed a circular arc. Figure 11 illustrates the result of augmenting the original sequence by placing an artificial kettle on top of the stove.

The third experiment refers to the 22 frame Arenberg castle sequence. This sequence is of dimensions 768×576 and is characterized by considerable lighting variations, relatively large interframe translational motion and epipoles being located outside the images. Fig. 12 shows several frames of the original sequence augmented with a helicopter. A top view of the VRML 3D model that was recovered, showing also the the camera locations and trajectory, is illustrated in Fig. 15(a). Evidently, the right angles between walls have been correctly recovered.

[4] See http://www.virtualcamera.com for more details.

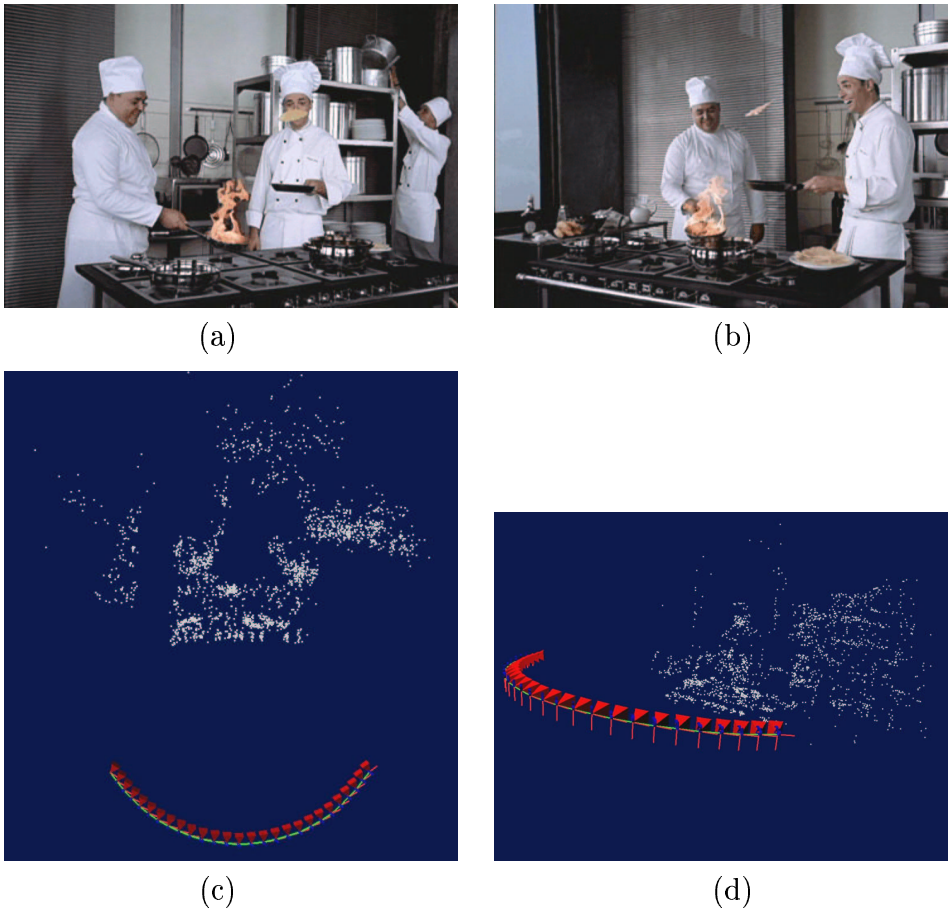


Fig. 10. (a), (b) the first and last frames of the "cooks" sequence (courtesy of Dayton Taylor/Digital Air Inc.), (c) and (d) top and side views of the 3D reconstruction and the camera trajectory. Observe that the recovered trajectory is indeed a circular arc.

The fourth experiment employs another well-known outdoors sequence, namely the one showing the ruins of the Agora in the Sagalassos archaeological site. The imaged scene contains two dominant planes, relative to which the camera moves laterally. This sequence has been shot using a camcorder and consists of 126 720×576 frames with very small interframe motion. Using time subsampling by a factor of five, 26 keyframes were determined. Fig. 13 shows snapshots after augmenting the original sequence with a virtual Roman standing on the wall. A side view of the recovered VRML 3D model is also illustrated in Fig. 15(b).

The fifth and last experiment is based on the 46 frame "desk" sequence, sparsely sampled in time and depicting the top of an office desk. This sequence is comprised of 640×480 images that were acquired while tracking a firewire webcam undergoing complex motion: At the beginning, the camera moves laterally with a right to left direction, then moves laterally from left to right, then right to left again, then moves away from the desk, then it starts

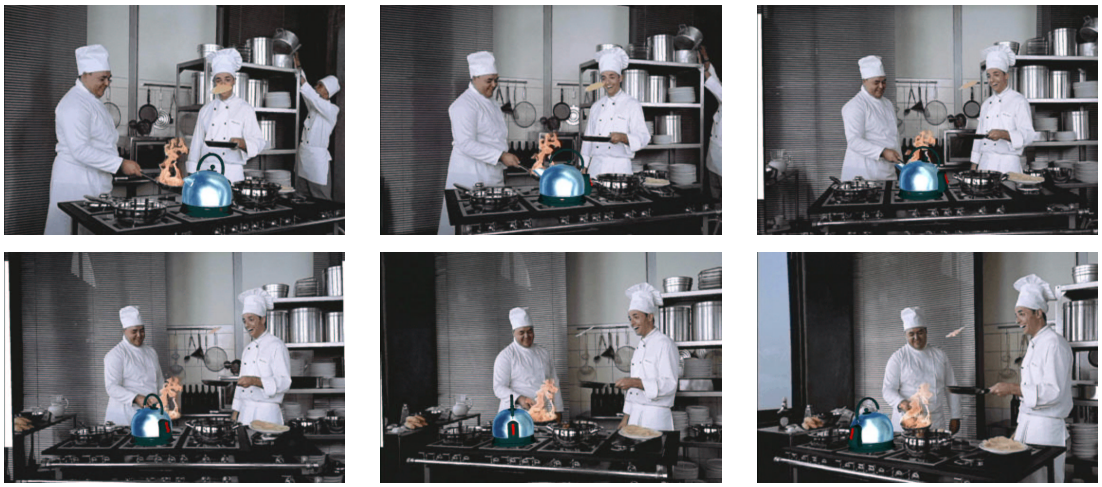


Fig. 11. Snapshots of the augmented "cooks" sequence corresponding to frames 0, 5, 10, 15, 20 and 26 respectively.

Fig. 12. The "castle" sequence (courtesy of the University of Leuven VISICS Group). Snapshots of the augmented sequence corresponding to frames 0, 4, 8, 12, 16 and 20.

approaching the desk while moving upwards and finally ends its trajectory by moving downwards vertically. Due to its motion pattern, this sequence is characterized by large changes in the field of view (for example, the book on the left moves out of the field of view in frame 6 and reappears in frame 10) that make corner matching difficult. Figs. 15(d) and (e) show a side and a top view respectively of the VRML model corresponding to the recovered camera trajectory and 3D structure. Camera tracking results were used to augment the original sequence with a virtual spider climbing on the book's front cover. Selected snapshots from the augmented sequence are illustrated in Fig. 14.

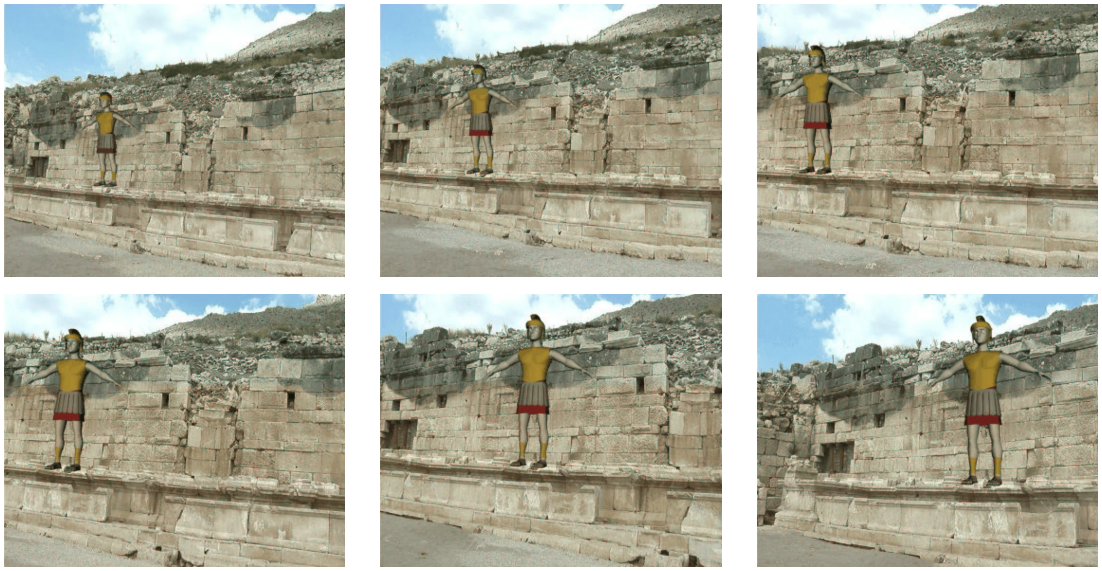


Fig. 13. The "sagalassos" sequence (courtesy of the University of Leuven VISICS Group). Snapshots of the augmented sequence corresponding to frames 0, 25, 50, 75, 100 and 125 respectively.

Fig. 14. The "desk" sequence depicting an office desk. Snapshots of the augmented sequence corresponding to frames 0, 4, 15, 29, 35 and 40 respectively.

Table 1 summarizes some performance statistics gathered from the application of the proposed technique on the five test sequences. Specifically, the first two columns correspond to the mean and standard deviation of the total running time per frame, the next two to the mean and standard deviation of the time required for corner matching per frame, the fifth column to the average number of matched corners per image triplet and the last column to the average number of matched corners per image pair. Execution times were measured on an Intel P4@2.5 GHz laptop. As can be clearly seen in the first

and third columns of Table 1, about 80% of the execution time is spent for detecting and matching image corners. The refinement of Eq. (18) accounts for most of the remaining 20% of the cycle time. A second observation that can be made based on the tabulated data is that matching a few hundred image corners across each triplet suffices for camera tracking.

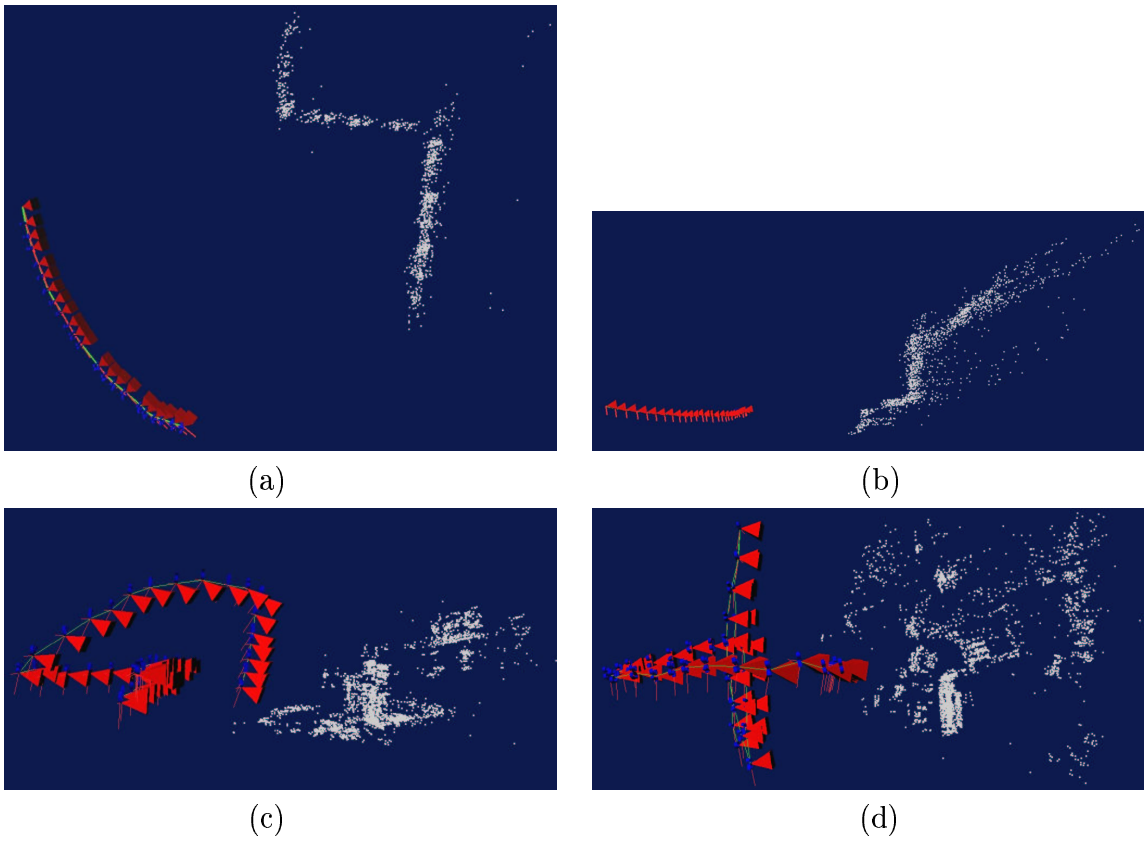


Fig. 15. Sample views of different VRML 3D reconstructions corresponding to (a) "castle" top, (b) "sagalassos" side, (c) and (d) "desk" side and top.

	Cycle time (ms)		**Matching time (ms)**		**#Matches**	
Sequence	mean	sdev	mean	sdev	triplet	pair
"movi"	485.05	95.54	397.89	32.38	127.96	197.70
"cooks"	946.89	184.81	748.79	18.93	354.05	486.49
"castle"	1173.2	268.88	938.95	15.91	373.65	483.43
"sagalassos"	987.062	261.95	750.41	29.60	331.31	481.94
"desk"	681.20	130.33	536.69	59.24	211.23	330.22

Table 1
Execution statistics for the experiments: Mean and standard deviation of the total running time per frame, mean and standard deviation of the matching time required per frame, average number of matched corners per triplet, average number of matched corners per pair.

7 Conclusions

This paper has presented a method for automatic camera tracking across an image sequence acquired without modifying the imaged environment. The method is based on tracking a virtual 3D plane, a task involving the estimation of a quadruple of plane parameters that is achieved using a combination of linear and non-linear optimization techniques operating on sets of corner matches. Knowledge of the homographies induced by the same 3D plane across the whole sequence permits the direct recovery of the camera projection matrices and thus of the Euclidean camera 3D motion, which is later refined through a local resectioning process. The proposed method is causal, tolerant to missing and erroneous data and has reasonable computational requirements, permitting an efficient implementation on commodity hardware. Although not statistically optimal in the MLE sense, the tracking results obtained with the proposed method are of very satisfactory accuracy for various types of imaged scenes and camera motions. Experimental results have also emphasized the implications of corner matching, both in terms of accuracy of the tracking results and of contribution to the total execution time.

Future work will focus on directions aiming to further increase the performance and robustness of the camera tracker. Considering that the task of corner detection and matching is characterized by data parallel computations (e.g. correlations) and that its execution dominates the cycle time, substantial speedups could be gained by adopting a SIMD computation model, implemented either on current high-performance programmable graphics chips [51] or using extended instruction sets such as MMX and SSE. The preceding discussion in section 4 has assumed that the same virtual plane is being tracked throughout the entire image sequence. This assumption might cause problems when tracking extended sequences for which the tracked virtual plane moves far away from the currently visible scene 3D points. To remedy this, a new virtual plane can occasionally be selected as described in section 5 and then tracking can be handed off from the employed plane to the new one. An issue that has not been addressed in this paper concerns the detection and handling of degenerate cases in which the imaged scene is entirely planar or the underlying camera motion is purely rotational. Since in such cases the fundamental matrix cannot be estimated uniquely, camera motion recovery should be based on resectioning rather than on homography chaining. Some work related to the detection of degenerate cases is reported by Pollefeys et al [52]. Another issue pertains to incorporating drift counter-measures that prevent the former from accumulating over long sequences [53].

References

[1] G. Welch, E. Foxlin, Motion Tracking: No Silver Bullet, but a Respectable Arsenal, IEEE Computer Graphics and Applications 22 (6) (2002) 24–38.

[2] R. Azuma, Tracking Requirements for Augmented Reality, CACM 36 (7) (1993) 50–51.

[3] G. Thomas, J. Jin, T. Niblett, C. Urquhart, A Versatile Camera Position Measurement System for Virtual Reality TV Production, in: Proc. of Int'l Broadcasting Convention (IBC'97), 1997, pp. 284–289.

[4] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[5] K. Kutulakos, J. Vallino, Calibration-Free Augmented Reality, IEEE Trans. on VCG 4 (1) (1998) 1–20.

[6] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, M. Tuceryan, Real-time Vision-Based Camera Tracking for Augmented Reality Applications, in: Proceedings of VRST'97, 1997, pp. 87–94.

[7] M. Lourakis, Egomotion Estimation Using Quadruples of Collinear Image Points, in: Proc. of ECCV'00, Vol. 2, 2000, pp. 834–848.

[8] T. Viéville, O. Faugeras, Q. Luong, Motion of Points and Lines in the Uncalibrated Case, IJCV 17 (1) (1996) 7–41.

[9] P. Sturm, B. Triggs, A Factorization Based Algorithm for multi-Image Projective Structure and Motion, in: Proc. of ECCV'96, Vol. 1065, 1996, pp. 709–720.

[10] K. Cornelis, M. Pollefeys, M. Vergauwen, F. Verbiest, L. V. Gool, Tracking Based Structure and Motion Recovery for Augmented Video Productions, in: Proceedings of VRST'01, 2001, pp. 17–24.

[11] A. Fitzgibbon, A. Zisserman, Automatic Camera Recovery for Closed or Open Image Sequences, in: Proceedings of ECCV'98, 1998, pp. 311–326.

[12] C. Rother, S. Carlsson, Linear Multi View Reconstruction and Camera Recovery Using a Reference Plane, IJCV 49 (2/3) (2002) 117–141.

[13] R. Kaucic, R. Hartley, N. Dano, Plane-based Projective Reconstruction, in: Proc. of ICCV'01, Vol. I, 2001, pp. 420–427.

[14] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle Adjustment – A Modern Synthesis, in: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, 1999, pp. 298–372.

[15] P. Beardsley, A. Zisserman, D. Murray, Sequential Updating of Projective and Affine Structure From Motion, IJCV 23 (1997) 235–259.

[16] G. Simon, A. Fitzgibbon, A. Zisserman, Markerless Tracking using Planar Structures in the Scene, in: Proc. of Int'l Symposium on Augmented Reality, 2000, pp. 120–128.

[17] P. McLauchlan, A Batch/Recursive Algorithm for 3D Scene Reconstruction, in: Proc. of CVPR'00, Vol. 2, 2000, pp. 738–743.

[18] S. Avidan, A. Shashua, Threading Fundamental Matrices, IEEE Trans. on PAMI 23 (1) (2001) 73–77.

[19] S. Prince, K. Xu, A. Cheok, Augmented Reality Camera Tracking with Homographies, IEEE Computer Graphics and Applications 22 (6) (2002) 39–45.

[20] D. Nistér, An Efficient Solution to the Five-Point Relative Pose Problem, IEEE Trans. on PAMI 26 (6) (2004) 756–777.

[21] A. Davison, Real-Time Simultaneous Localisation and Mapping with a Single Camera, in: Proc. of ICCV'03, 2003, pp. 1403–1410.

[22] P. Torr, A. Zisserman, Robust Parameterization and Computation of the Trifocal Tensor, IVC 15 (8) (1997) 591–607.

[23] A. Shashua, S. Avidan, The Rank-4 Constraint in Multiple View Geometry, in: Proc. of ECCV'96, Vol. 2, 1996, pp. 196–206.

[24] S. Avidan, A. Shashua, Tensor Embedding of the Fundamental Matrix, in: Proc. of post-ECCV SMILE'98, Vol. Springer LNCS 1506, 1998, pp. 47–62.

[25] A. Shashua, N. Navab, Relative Affine Structure: Canonical Model for 3D from 2D Geometry and Applications, IEEE Trans. on PAMI 18 (9) (1996) 873–883.

[26] R. Hartley, In Defense of the 8-Point Algorithm, IEEE Trans. on PAMI 19 (6) (1997) 580–593.

[27] P. Rousseeuw, Least Median of Squares Regression, Journal of the American Statistics Association 79 (1984) 871–880.

[28] 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, AI Journal 78 (1995) 87–119, detailed version in INRIA RR-2273.

[29] M. Fischler, R. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, CACM 24 (1981) 381–395.

[30] J. Moré, B. Garbow, K. Hillstrom, User guide for MINPACK-1, Tech. Rep. ANL-80-74, Argonne National Laboratory (Aug. 1980).

[31] M. Lourakis, R. Deriche, Camera Self-Calibration Using the Singular Value Decomposition of the Fundamental Matrix, in: Proceedings of ACCV'00, 2000, pp. 403–408, detailed version in INRIA RR-3748.

[32] R. Hartley, P. Sturm, Triangulation, CVIU 68 (2) (1997) 146–157.

[33] R. Goldman, Intersection of Two Lines in Three-Space, in: A. Glassner (Ed.), Graphics Gems I, Academic Press, San Diego, 1990, p. 304.

[34] S.-J. Ko, Y. Lee, A. Fam, Efficient Implementation of One-Dimensional Recursive Median Filters, IEEE Trans. on Circuits and Systems 37 (11) (1990) 1447–1450.

[35] J. Dennis, D. Gay, R. Welsch, An Adaptive Nonlinear Least-Squares Algorithm, ACM Trans. on Math. Software 7 (3) (1981) 348–368.

[36] M. Lourakis, A. Argyros, S. Orphanoudakis, Detecting Planes In An Uncalibrated Image Pair, in: Proc. of BMVC'02, Vol. 2, 2002, pp. 587–596.

[37] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, SIAM, 1999.

[38] P. Smith, D. Sinclair, R. Cipolla, K. Wood, Effective Corner Matching, in: Proc. of BMVC'98, Vol. 2, 1998, pp. 545–556.

[39] M. Lourakis, A. Argyros, K. Marias, A Graph-Based Approach to Corner Matching Using Mutual Information as a Local Similarity Measure, in: Proc. of ICPR'04, Vol. 2, 2004, pp. 827–830.

[40] B. Boufama, R. Mohr, A Stable and Accurate Algorithm for Computing Epipolar Geometry, IJPRAI 12 (6) (1998) 817–840.

[41] W. Press, S. Teukolsky, A. Vetterling, B. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, New York, 1992.

[42] D. Nistér, Frame Decimation for Structure and Motion, in: 3D Structure from Images - SMILE 2000, Vol. LNCS 2018, 2000, pp. 17–34.

[43] T. Thormaehlen, H. Broszio, A. Weissenfeld, Keyframe Selection for Camera Motion and Structure Estimation from Multiple Views, in: Proc. of ECCV'04, Vol. 1, 2004, pp. 523–535.

[44] C.-P. Lu, G. Hager, E. Mjolsness, Fast and Globally Convergent Pose Estimation from Video Images, IEEE Trans. on PAMI 22 (6) (2000) 610–622.

[45] R. Hartley, Kruppa's Equations Derived from the Fundamental Matrix, IEEE Trans. on PAMI 19 (2) (1997) 133–135.

[46] U. Nowak, L. Weimann, A Family of Newton Codes for Systems of Highly Nonlinear Equations, Tech. Rep. 91-10, Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB), available at http://www.zib.de/ (Dec. 1991).

[47] J.-Y. Bouguet, Camera Calibration Toolbox for Matlab, [web page] `http://www.vision.caltech.edu/bouguetj/calib_doc/`, [Accessed on 1 Dec. 2004.] (2004).

[48] A. Fitzgibbon, A. Zisserman, Automatic 3D Model Acquisition and Generation of New Images from Video Sequences, in: Proc. of EUSIPCO '98, 1998, pp. 1261–1269.

[49] M. Lourakis, A. Argyros, The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm, Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, available from `http://www.ics.forth.gr/~lourakis/sba` (Aug. 2004).

[50] Discreet (Autodesk Inc.), 3ds max, [web page] http://www.discreet.com/3dsmax, [Accessed on 8 Jul. 2004] (2004).

[51] J. Fung, S. Mann, Computer Vision Signal Processing on Graphics Processing Units, in: Proc. of ICASSP'04, 2004, pp. 93–96.

[52] M. Pollefeys, F. Verbiest, L. V. Gool, Surviving Dominant Planes in Uncalibrated Structure and Motion Recovery, in: Proc. of ECCV'00, Vol. 2, 2002, pp. 837–851.

[53] K. Cornelis, F. Verbiest, L. V. Gool, Drift Detection and Removal for Sequential Structure from Motion Algorithms, IEEE Trans. on PAMI 26 (10) (2004) 1249–1259.