Vision-Based Camera Motion Recovery for Augmented Reality

Manolis I. A. Lourakis and Antonis A. Argyros
Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
Vassilika Vouton, P.O. Box 1385, GR 711 10
Heraklion, Crete, GREECE
{lourakis, argyros}@ics.forth.gr   http://www.ics.forth.gr/cvrl/

Abstract

We address the problem of tracking the 3D position and orientation of a camera, using the images it acquires while moving freely in unmodeled, arbitrary environments. This task has a broad spectrum of useful applications in domains such as augmented reality and video post production. Most of the existing methods for vision-based camera tracking are designed to operate in a batch, off-line mode, assuming that the whole video sequence to be tracked is available before tracking commences. Typically, such methods operate non-causally, processing video frames backwards and forwards in time as they see fit. Furthermore, they resort to optimization in very high dimensional spaces, a process that is computationally intensive. For these reasons, batch methods are inapplicable to tracking in on-line, time-critical applications such as video see-through augmented reality. This paper puts forward a novel feature-based approach for camera tracking. The proposed approach operates on images continuously as they are acquired, has realistic computational requirements and does not require modifications of the environment. Sample experimental results demonstrating the feasibility of the approach on video images are also provided.

1 Introduction

Camera matchmoving is an application involving synthesis of real scenes and artificial objects, in which the goal is to insert computer-generated graphical 3D objects into live-action footage depicting unmodeled, arbitrary scenes. Graphical objects should be inserted in a way so that they appear to move as if they were a part of the real scene. Seamless, convincing insertion of graphical objects calls for accurate 3D camera motion tracking (i.e. pose estimation), stable enough over extended sequences so as to avoid the problems of jitter and drift in the location and appearance of objects with respect to the real scene. Additionally, the placement of the objects with respect to the real scene often requires the availability of some 3D geometry information; for instance, accurate 3D reconstruction of a few guiding control points is in most cases sufficient. Matchmoving finds several important applications in augmented reality as well as the creation of special effects in the post-production industry [26]. To provide the versatility required by such applications, very demanding camera tracking requirements, both in terms of accuracy and speed, are imposed [3].

Optical and electromechanical camera tracking are technologies that have successfully proven themselves in applications such as virtual studio TV shooting [27]. Nevertheless, apart from suffering from range limitations, such technologies call for special modifications of the environment 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, computer vision techniques provide an attractive alternative to optical and electromechanical tracking methods for recovering camera motion. Tracking the pose of a camera using the images it acquires while moving freely in unmodeled, arbitrary environments, is a very challenging problem in visual motion analysis. 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 [8]. Before briefly reviewing a few representative ones, it is pointed out that our requirement for operation in unprepared environments exclude methods such as [14] that rely upon the presence of fiducial markers or special calibration objects in the environment.

Methods that avoid making any assumptions regarding the environment exploit geometric constraints that arise from the automatic extraction and matching of appropriate 2D image features such as corner points. Corners are simply points of localized image structure, formed at the boundaries of different brightness image regions. 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 [7, 5]. Such methods process image data in a batch mode and usually are non-causal, employing both past and future frames for deducing the camera motion corresponding to the current frame. Commercially available camera tracking software products such as `boujou`, `MatchMover`, `3D-equalizer` and `PFTrack` also fall into this category. Albeit accurate, batch techniques share the limitation of being computationally demanding due to the use of global bundle adjustment, which involves the solution of large, non-linear optimization problems [25]. This, plus the requirement of operating on the whole sequence at once, makes batch methods inappropriate for use in time-critical applications. Methods operating in a continuous mode, in which images are processed incrementally as acquired, constitute the second class of camera tracking techniques [4, 24, 2]. Typically, such methods are causal, relying only on past frames for estimating the camera motion for the current one.

In this paper, a novel feature-based approach to camera tracking is presented. 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. The tracked plane is not required to be physically present in the scene; it can be a virtual one. 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 a segmentation of the tracked plane from the scene. 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 and used for refining the latter through local bundle adjustment. Intended for use in close to on-line applications, the proposed method is designed to operate in a continuous mode. The proposed method follows a strategy similar to [2]. However, it is based on much simpler constraints whose derivation is shorter and does not involve neither the trifocal tensor nor tensorial notation. 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 [21], is a theoretically minimal one. Compared to [2] and [24] 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 briefly 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. An extended version of this paper can be found in [15].

2 Notation and Background

In the following, vectors and arrays appear in boldface and are represented using projective (homogeneous) coordinates. The symbol $\simeq$ denotes equality of vectors up to an arbitrary scale factor. 3D points are written in uppercase, while their image projections in lowercase (e.g. $\mathbf{X}$ and $\mathbf{x}$).

A well-known constraint for a pair of perspective views of a rigid scene is the *epipolar constraint*. This constraint states that for each point in one of the images, the corresponding point in the other image must lie on a straight line. Assuming that no calibration information is available, the epipolar constraint is expressed mathematically by a 3×3 singular matrix, known as the *fundamental matrix* and denoted by $\mathbf{F}$. Denoting by O and $O^{'}$ the centers of projection corresponding to the two perspective views, the points of intersection of the line $\overline{OO^{'}}$ with the first and second image planes are the *epipoles*, depending on relative translational motion only and being denoted by $\mathbf{e}$ and $\mathbf{e}^{'}$, respectively. For example, epipole $\mathbf{e}^{'}$ corresponds to the uncalibrated translational component of the camera motion from the first to the second image. Given $\mathbf{F}$, the epipoles can be recovered by finding the kernels of $\mathbf{F}$ and $\mathbf{F}^T$. Another important concept in projective geometry is the *plane homography* $\mathbf{H}$, a nonsingular 3×3 matrix which relates two uncalibrated retinal images of a 3D plane. More specifically, if $\mathbf{x}$ is the projection in one view of a point on the plane and $\mathbf{x}^{'}$ is the corresponding projection in a second view, then the two projections are related by the linear projective transformation $\mathbf{x}^{'} \simeq \mathbf{H}\mathbf{x}$. For more detailed treatments of the application of projective geometry to computer vision, the interested reader is referred to [13].

As shown in [21], 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 [1] 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

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

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 ($\mathbf{H}_1$, $\mathbf{H}_2$ 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

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

for some scalars λ_i.

Next, a result due to Shashua and Navab [22] 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) dictates 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 term 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 [22]. 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 $\kappa = (\mathbf{H}\mathbf{x} \times \mathbf{x}^{'})^T (\mathbf{x}^{'} \times \mathbf{e}^{'}) \ / \ ||\mathbf{x}^{'} \times \mathbf{e}^{'}||^2$.

3 3D Plane Tracking

Point $\mathbf{X}_0$ plays a special role in the derivation of Eq. (3). Specifically, 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 defined by Eq. (3). 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.

Suppose now 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_1, I_2) 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 next section.

3.1 Chaining Homographies Among Consecutive Frames

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.

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 $\mathbf{U}^{-1}$, Eq. (3) takes the following form for all point matches in those two images

$$\mathbf{x}_i \simeq \mathbf{U}^{-1}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}. \tag{4}$$

Solving for κ_i yields the former as

$$\kappa_i = \frac{(\mathbf{U}^{-1}\mathbf{x}_i^{'} \times \mathbf{x}_i)^T (\mathbf{x}_i \times \mathbf{e})}{||\mathbf{x}_i \times \mathbf{e}||^2}. \tag{5}$$

Regarding point matches in frames I_2 and I_3, Eq. (3) gives

$$\mathbf{x}_i^{''} \simeq \mathbf{V}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}^{''}, \tag{6}$$

where the κ_i are given by Eq. (5). In order for Eq. (6) to hold for those κ_i, the scale of $\mathbf{V}$ in it has to be compatible with that of the estimated $\mathbf{e}^{''}$. For this reason, $\mathbf{V}$ in Eq. (6) is no longer a homogeneous 3×3 matrix but rather an ordinary, inhomogeneous one. Equation (6) 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 [1]

$$\begin{array}{rcl} \mathbf{v}_3^T \mathbf{x}_i^{'} x_i^{''} - \mathbf{v}_1^T \mathbf{x}_i^{'} & = & \kappa_i e_x^{''} - \kappa_i e_z^{''} x_i^{''} \\ \mathbf{v}_3^T \mathbf{x}_i^{'} y_i^{''} - \mathbf{v}_2^T \mathbf{x}_i^{'} & = & \kappa_i e_y^{''} - \kappa_i e_z^{''} y_i^{''} . \end{array} \qquad (7)$$

Notice that Eqs. (7) do not require that the employed point matches have been identified as lying on the plane. 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 a virtual (i.e. not physically present in the scene) plane. Since $\mathbf{v}_j^T \mathbf{x}_i^{'} = \mathbf{x}_i^{'T} \mathbf{v}_j$, Eqs. (7) can be written in matrix form as

$$\begin{bmatrix} -\mathbf{x}_i^{'T} & \mathbf{0}^T & \mathbf{x}_i^{'T} x_i^{''} \\ \mathbf{0}^T & -\mathbf{x}_i^{'T} & \mathbf{x}_i^{'T} y_i^{''} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \kappa_i e_x^{''} - \kappa_i e_z^{''} x_i^{''} \\ \kappa_i e_y^{''} - \kappa_i e_z^{''} y_i^{''} \end{bmatrix} . \qquad (8)$$

Thus, each triplet of corresponding points provides two equations in the elements of $\mathbf{V}$. Concatenating the equations arising from five triplet correspondences, a matrix equation of the form $\mathbf{Mv} = \mathbf{b}$ 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 linear system with 9 unknowns that may be solved using LU decomposition. In the case that more than five triplet matches are available, Eq. (8) gives rise to an over-constrained system from which $\mathbf{V}$ can be estimated in a least squares manner with the aid of SVD.

According to the terminology of [13], 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 [10] 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 is expressed by a 3×3 linear transformation $\mathbf{L}_i$. 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. (5) 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) [20] 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.

Since the DLT constraints minimize an algebraic error term with no physical meaning, the estimate computed by LMedS is refined by 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

$$\sum_i \left(d(\mathbf{x}_i^{''},\ \mathbf{V}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}^{''})^2 \ + \right. \\ \left. d(\mathbf{x}_i^{'},\ \mathbf{V}^{-1}\mathbf{x}_i^{''} - \frac{||\mathbf{x}_i^{''}||}{||\mathbf{V}\mathbf{x}_i^{'} + \kappa_i \mathbf{e}^{''}||} \kappa_i \mathbf{V}^{-1} \mathbf{e}^{''})^2 \right) \qquad (9)$$

with respect to $\mathbf{V}$. This criterion involves the mean 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 [18], initialized with the least squares estimate 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.

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

[1] 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.

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.

3.2 Reducing the DOFs of Plane Tracking

In the following, the basic method of the previous section will be refined, aiming to derive a model involving 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 4, spanned by the 4 primitive homographies of Eq. (1). Knowledge of those homographies allows any other homography $\mathbf{H}$ to be expressed as a linear combination encompassing 4 scalars λ_i (see Eq. (2)). 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. (7) depend on four rather than nine parameters. Therefore, 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, both the linear and the non-linear estimation processes that have been described above are performed with four rather than nine unknowns. This reduction in the dimensionality of the problem is of utmost importance since fewer degrees of freedom entail less computation time for the homography (particularly for the non-linear refinement) as well as more accurate estimates. It has been found experimentally that the execution time for plane tracking using the formulation involving λ_i is by an order of magnitude shorter than that required when estimating $\mathbf{V}$ directly. Space considerations prevent us from deriving here the exact form of Eqs. (7) and Eq. (8) after introducing the coefficients λ_i.

4 Camera Tracking

In this section, $\mathbf{H}_{i,j}$ 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. Assume also that $\mathbf{H}_{1,2}$ has been supplied and, using the method outlined in section 3, the 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 consistent set of projective camera matrices in canonical form for the three views is given by [13, 2]:

$$\mathbf{P}_1 = [\mathbf{H}_{2,1} \mid \mathbf{e}_{2,1}]\,,\ \ \mathbf{P}_2 = [\mathbf{I} \mid \mathbf{0}]\,,\ \ \mathbf{P}_3 = [\mathbf{H}_{2,3} \mid \mathbf{e}_{2,3}]\,, \quad (10)$$

where $\mathbf{I}$ denotes the 3×3 identity matrix. Since it is customary to express the camera matrices relative to the first image I_1, application of an appropriate 3D projective mapping can transform Eqs.(10) 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

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\,\mathbf{b} \\ \mathbf{0}^T & 1 \end{array}\right] \quad (11)$$

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. (10) should be right multiplied by the matrix given by Eq.(11) 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

$$\mathbf{P}_1 = [\mathbf{I} \mid \mathbf{0}]\,,\ \ \mathbf{P}_2 = [\mathbf{H}_{1,2} \mid \mathbf{e}_{1,2}]\,,\ \ \mathbf{P}_3 = [\mathbf{H}_{1,3} \mid \mathbf{e}_{1,3}]\,. \quad (12)$$

Suppose now that by employing the plane tracker for the image triplet I_2, I_3 and I_4, the homography $\mathbf{H}_{34}$ 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.(11) 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 $\mathbf{P}_i$ corresponding to any image I_i with $i > 4$. 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.

In order to relieve the camera tracker from the computational overhead associated with their estimation, the camera intrinsic calibration parameters 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 [17]. Given the 3×3 camera intrinsic calibration matrix $\mathbf{K}$, a projective camera matrix can be upgraded to Euclidean by right multiplication with the 4×4 matrix defined as

$$\left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \\ -\mathbf{p}^T\,\mathbf{K} & 1 \end{array}\right], \quad (13)$$

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 [13].

A rough 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 [12]. The main problem that needs to be addressed by all triangulation algorithms is the fact that the 3D lines defined by the camera optical centers and the corresponding image projections are skew, due to mislocalized image corners and errors in the estimates of camera matrices. In this work, 3D structure recovery is achieved by exploiting the knowledge of the camera intrinsic parameters in order to express two back-projected 3D lines in an Euclidean coordinate frame. 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 [9]. Since an image triplet gives rise to three different image pairs, the reconstructed point is taken here to be the centroid 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. (10) are used to compute the corresponding trifocal tensor using a closed form formula [11]. Then, point matches $\mathbf{x},\ \mathbf{x}^{'}$ from the first two views are used together with the computed tensor to predict the position of the corresponding point $\mathbf{x}^{''}$ in the third view. Point triplets for which the distance between the actual and predicted third view projections exceeds a certain threshold, are eliminated from further consideration.

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

$$\min_{\mathbf{R}_i, \mathbf{t}_i} \sum_{i=1}^{W} \sum_{j} d(\mathbf{P}_i\, \mathbf{X}_j,\ \mathbf{x}_j^i)^2 \quad \text{with} \quad \mathbf{P}_i = \mathbf{K}\, [\mathbf{R}_i \mid \mathbf{t}_i], \tag{14}$$

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, $i = 1 \ldots W$. Rotation matrices $\mathbf{R}_i$ are parametrized using an axis-angle representation; a parametrization based on quaternions is also possible. To keep the computational overhead of the minimization low, observe that Eq. (14) is minimized with respect to the 3D motion only and not the 3D structure. The minimization of Eq. (14) is performed with the aid of a non-linear least squares algorithm [6] that is initialized with the motion parameters computed directly from the tracked plane homography. Finally, since the projective camera matrix for I_i 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 the matrix in Eq. (13).

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 [16], 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 any plane is adequate for camera tracking as long as it can be tracked along the whole sequence. In order for plane tracking to commence, the plane homography induced among 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 $\mathbf{x}_i,\ \mathbf{x}_i^{'},\ \ i = 1, \ldots, N$ be a set of matching point pairs in the first two frames. The virtual plane can be chosen so that it approximates the set of available point matches as much as possible. 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. It has been explained in section 2 that any planar homography defined between two images can be expressed as the linear combination of the four primitive homographies of Eq. (1). The sought $\mathbf{H}$ is thus computed from the coefficients $\mu_j,\ \ j = 1, \ldots, 4$ minimizing

$$\sum_{j=1}^{4} (\mu_j\ \mathbf{H}_j)\ \mathbf{x}_i \simeq \mathbf{x}_i^{'}, \quad i = 1, \ldots, N \tag{15}$$

Each of the available point matches provides two independent linear constraints for the μ_j, therefore N 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 μ_j corresponding to the homography minimizing Eq. (15) 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 [1] and gives rise to a projective reconstruction of space that is characterized by a small amount of projective distortion.

6 Implementation and Experimental Results

A prototype of the proposed camera tracking method has been implemented in C. The point features that are required as input have been extracted and matched automatically with the aid of the KLT corner tracker [23]. Using the resulting matches, the epipoles were computed by applying SVD on the fundamental matrices estimated using an implementation of [28].

The current implementation of the plane tracker performs chaining based on constraints arising from three frames at a time. Possible camera lens distortions (e.g. radial distortion) are neglected. The intrinsic camera parameters (i.e. focal length, aspect ratio, principal point and skew), were determined by using the auto-calibration method described in [17]. 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. 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. W in Eq. (14) is set to 5 and the minimization is carried out using the NL2SOL algorithm [6], 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 [18], NL2SOL was found in this case to converge faster while producing results of similar accuracy. The NLSCON non-linear least squares routine [19] has also been evaluated and yielded results slightly worse than those of DN2G. The jacobians of Eq. (9) and Eq. (14) that are necessary for the non-linear minimizations are computed analytically with the aid of MAPLE's symbolic differentiation facilities.

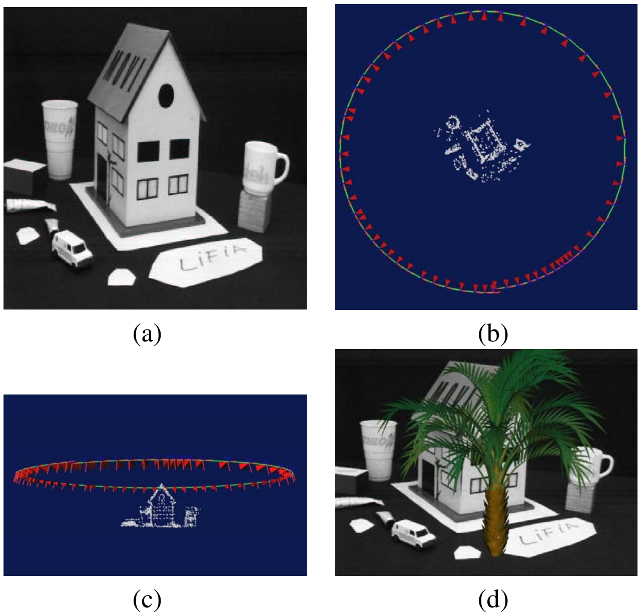

Figure 1. (a) the first frame from the "house" sequence (courtesy of the INRIA MOVI Group), (b) and (c) top and side views of the 3D reconstruction and the camera trajectory and (d) first frame of the original sequence augmented with a virtual pine tree.

Rigorous performance evaluation of camera tracking for an image sequence is difficult, due to the fact that ground truth for the camera motion is usually unavailable. For this reason, we have chosen to indirectly evaluate camera tracking from the sequences resulting from augmenting the original ones with artificial 3D objects. To achieve this, the estimated camera trajectories were exported to the 3DSMax graphics package 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 positioning of the artificial graphical objects into the scene was guided by the structure information also provided by the camera tracker. Sample augmented sequences can be found at http://www.ics.forth.gr/cvrl/demos.html. Due to space limitations, results from just one experiment are included here; additional results can be found at the above URL as well as in [15]. The reported experiment was performed on the well known "MOVI house" image sequence, consisting of 119 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. The first frame of the sequence is shown in Fig. 1(a), Figs. 1(b)-(c) illustrate different views of the VRML 3D model recovered using the proposed method on odd numbered frames and Fig. 1(d) shows a frame of the original sequence augmented with a virtual pine tree. In Figs. 1(b)-(c), the 3D camera locations are indicated 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. As can be seen from them, the estimated trajectory is very close to being a full circle. The average running time of the proposed tracking method for each image frame was 317 ms on an Intel P4@1.8 GHz running Linux. Most of this time is spent in the bundle adjustment of Eq. (14) and does not include the time required for matching between 200 and 330 points between successive frames. The aforementioned cycle time is expected to decrease considerably by employing an implementation of bundle adjustment which explicitly takes into account the sparseness of the matrices involved in the minimization of Eq. (14). For comparison, note that the time required by existing commercial products such as 2D3's `boujou` for batch camera tracking on such sequences is in the order of several minutes for the whole sequence.

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 bundle adjustment process. The proposed method is causal and has reasonable computational requirements, permitting an efficient implementation on commodity hardware. Although not statistically optimal in the MLE sense, the results of the proposed method are of very satisfactory accuracy for various types of image sequences.

Acknowledgements

This work was partially supported by the European Union projects IST-2001-34545 LifePlus and IST-2001-32184 ActIPret.

References

[1] S. Avidan and A. Shashua. Tensor Embedding of the Fundamental Matrix. In *Proc. of post-ECCV SMILE'98*, volume Springer LNCS 1506, pages 47–62, 1998.

[2] S. Avidan and A. Shashua. Threading Fundamental Matrices. *IEEE Trans. on PAMI*, 23(1):73–77, Jan. 2001.

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

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

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

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

[7] A. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *Proceedings of ECCV'98*, pages 311–326, 1998.

[8] A. Fusiello. Uncalibrated Euclidean Reconstruction: A Review. *IVC*, 18(6-7):555–563, 2000.

[9] R. Goldman. Intersection of Two Lines in Three-Space. In A. Glassner, editor, *Graphics Gems I*, page 304. Academic Press, San Diego, 1990.

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

[11] R. Hartley. Lines and Points in Three Views and the Trifocal Tensor. *IJCV*, 22(2):125–140, 1997.

[12] R. Hartley and P. Sturm. Triangulation. *CVIU*, 68(2):146–157, 1997.

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

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

[15] M. Lourakis and A. Argyros. Efficient 3D Camera Matchmoving Using Markerless, Segmentation-Free Plane Tracking. Technical Report 324, ICS/FORTH, Sep. 2003. Available at ftp://ftp.ics.forth.gr/tech-reports/2003.

[16] M. Lourakis, A. Argyros, and S. Orphanoudakis. Detecting Planes In An Uncalibrated Image Pair. In *Proc. of BMVC'02*, volume 2, pages 587–596, 2002.

[17] M. Lourakis and R. Deriche. Camera Self-Calibration Using the Singular Value Decomposition of the Fundamental Matrix. In *Proceedings of ACCV'00*, pages 403–408, 2000. Detailed version in INRIA RR-3748.

[18] J. Moré, B. Garbow, and K. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, Aug. 1980.

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

[20] P. Rousseeuw. Least Median of Squares Regression. *Journal of the American Statistics Association*, 79:871–880, 1984.

[21] A. Shashua and S. Avidan. The Rank-4 Constraint in Multiple View Geometry. In *Proc. of ECCV'96*, volume 2, pages 196–206, 1996.

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

[23] J. Shi and C. Tomasi. Good Features to Track. In *Proceedings of CVPR'94*, pages 593–600, 1994. Free implementation available at `http://vision.stanford.edu/~birch/klt/`.

[24] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless Tracking using Planar Structures in the Scene. In *Proc. of Int'l Symposium on Augmented Reality*, 2000.

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

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

[27] Y. Xirouhakis, A. Drosopoulos, and A. Delopoulos. Efficient Optical Camera Tracking in Virtual Sets. *IEEE Trans. on IP*, 10(4):609–622, Apr. 2001.

[28] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *AI Journal*, 78:87–119, 1995. Detailed version in INRIA RR-2273.