ELSEVIER

Signal Processing 80 (2000) 1–21

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

Multicamera motion estimation for high-accuracy 3D reconstruction

F. Pedersini, P. Pigazzini, A. Sarti*, S. Tubaro

Dipartimento di Elettronica e Informazione – Politecnico di Milano, Piazza L. Da Vinci, 32-20133 Milano, Italy

Received 18 May 1998; received in revised form 10 May 1999

Abstract

In this article, we approach the problem of accurately estimating the motion of a multi-ocular camera system from the analysis of point-like features extracted from the acquired images. We propose a method based on optimal 3D data fusion, and compare it with an approach based on projective constraints. We also assess the critical problem of the accurate feature localization on the image planes. The performance of the proposed solutions is here evaluated through simulation tests and through experiments conducted on real scenes using three standard TV-resolution CCD cameras. © 2000 Elsevier Science B.V. All rights reserved.

Zusammenfassung

Dans cet article, nous approchons le problème de l'estimation précise du mouvement d'un système de caméras multioculaires à partir de l'analyse de caractéristiques de types ponctuelles extraites des images acquises. Nous proposons une méthode basée sur une fusion de données 3D optimale, et nous la comparons avec une approache basée sur des contraintes projectives. Nous évaluons aussi le problème critique de la localisation précise des caractéristiques dans les plans d'images. La performance des solutions proposées est ici évaluée au travers de tests de simulation et par des expériences conduites sur des scènes réelles en utilisant trois cameras CCD de résolution TV standard. © 2000 Elsevier Science B.V. All rights reserved.

Résumé

In diesem Artikel gehen wir das Problem einer genauen Schätzung der Bewegung eines multiokularen Kamerasystems aus der Analyse eines punktuellen Merkmales eines Bildes an. Wir schlagen eine Methode vor, die auf optimaler 3D Datenfusion basiert und vergleichen sie mit einer auf Projektions-Nebenbedingungen basierenden Vorgehensweise. Wir behandeln ebenfalls das kritische Problem der genauen Merkmal-Lokalisation aus Bildebenen. Die Leistungsfähigkeit der vorgeschlagenen Lösungen wird hier durch Simulationen und Experimenten mit realen Szenen unter Benutzung dreier Standard-CCD-Kameras mit TV-Auflösung bewertet. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Motion estimation; 3D reconstruction; Data fusion

* Corresponding author. Tel.: 392 2399 3647; fax: 392 2399 3413.
E-mail address: sarti@elet.polimi.it (A. Sarti)

PII: S0165-1684(99)00108-5

Nomenclature

$\Re^n$	set of all vectors of dimension n
$\Re^{n\times m}$	set of all $n\times m$ matrices
$\mathscr{P}^n$	projective space of dimension n
$\boldsymbol{P}_i \in \Re^{4\times 3}$	projection matrix from $\mathscr{P}^3$ to $\mathscr{P}^2$ (rank-3 matrix), associated to the ith camera
	$(\boldsymbol{t}\times)\,\boldsymbol{s}$ vector product in matrix form, where $(\boldsymbol{t}\times) = \left(\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}\times\right) = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$
$\boldsymbol{0}_n \in \Re^n$	vector with zero components
$\boldsymbol{1}_n \in \Re^n$	vector with unit components
$\boldsymbol{I}_n \in \Re^{n\times n}$	identity matrix
$\boldsymbol{x} \in \mathscr{P}^3$	object point expressed in projective world coordinates
$\boldsymbol{x}_i \in \mathscr{P}^3$	object point expressed in camera-i coordinates
$\boldsymbol{u}_i \in \mathscr{P}^2$	projective image coordinates, as seen by the ith camera
$\boldsymbol{p}_i \in \Re^3$	object point expressed in camera-i coordinates
$\mathbf{diag}(\lambda_1, \lambda_2, \lambda_3) \in \Re^{3\times 3}$	diagonal matrix whose diagonal elements are λ_1, λ_2 and λ_3, respectively.

1. Introduction

The accuracy of 3D surveys based on digital image analysis is an important issue for a number of application areas. This need, however, often contrasts with requirements of low-cost and flexible procedures of acquisition and reconstruction. In order to achieve accurate 3D modeling results with a modest investment, multi-camera acquisition systems based on *off-the-shelf* low-resolution digital cameras represent a reasonable choice. In fact, a number of fairly simple, accurate and reliable camera calibration techniques are currently available [1,2], which can be made robust against parameter drifts through parameter tracking strategies [3]. Furthermore, the literature on calibrated–stereo is rich with methods and results that are suitable for high-accuracy applications [4–7].

A multi-camera system usually enables a reliable reconstruction of just a portion of the object's surfaces. In fact, occlusions or self-occlusions often prevent parts of the imaged surfaces from being simultaneously viewed by all cameras. In addition, in order to obtain a high-accuracy reconstruction of the viewed surfaces, the distance from the object must be comparable with the average distance between cameras, therefore one multi-camera view it is most often not sufficient for reconstructing the whole imaged scene. One way to produce full-3D models, proposed by Kanade et al. [8], is to synchronously acquire a large number of views of the scene using a calibrated set of many cameras that sorround the volume to be reconstructed. A *multi-baseline stereo* (MBS) technique [9] is used to extract the 3D structure from the collected images. One major advantage of this solution is that it enables the reconstruction of dynamic scenes. When the scene is static, however, the complexity of the acquisition system can be greatly reduced. In this case, in fact, it is sufficient to acquire a series of multi-ocular views, each of which will originate a partial 3D reconstruction of the object. Indeed, the resulting 3D data will have to be fused together into a global full-3D model, which raises the problem of how to determine position and orientation of the camera system for each acquisition (egomotion).

In order to reliably estimate the motion of the camera system from the available views, we need to use some feature localization strategy that guarantees high accuracy (at subpixel level) on the image plane. The feature localization accuracy is of outmost importance for accurate camera motion estimation. Therefore the problem of how to maximize it will be here treated in depth for viewer-independent point-like features such as corners and vertices.

We will then propose and evaluate a 'Euclidean' ($\mathscr{R}^3$) method for estimating the motion of a multi-ocular acquisition system from the analysis of point-correspondences in the object space. Using the image features extracted by any subpixel corner detector such as the one discussed in this article, the method generates one 3D data cluster for each one of the available multi-ocular views. A robust 'cluster-to-cluster' 3D matching procedure (based on rigidity constraints) is used for performing feature matching in the object space. The motion of the camera system is finally estimated as the one that best merges such data in the object space. We will compare this solution with a simple projective ($\mathscr{P}^2$) method based on the essential constraint [10–13]. This method uses the same image features of the $\mathscr{R}^3$ technique, but is based on a robust matching procedure based on projective constraints and invariants [14]. The $\mathscr{P}^2$ method is here adapted to our trinocular camera system in such a way to estimate the whole (non-scaled) camera motion.

Although the techniques discussed in this article are valid for an arbitrary number of cameras, in what follows we will focus on trinocular acquisition systems. In order to test our 3D approach to ego-motion estimation on the field, we performed some experiments on real scenes by implementing a low-cost high-accuracy system, based on a calibrated set of three standard TV-resolution CCD cameras, which is able to reconstruct a 3D scene through the fusion of several partial reconstructions.

In order to make the article as self-contained as possible, we included in Section 2 a brief summary of facts from basic projective geometry that are extensively used throughout the article. The feature extraction method that we used in the article is presented in Section 3. A description of the two motion estimation methods is presented in Section 4. The acquisition setup and an evaluation of the accuracy of the 3D reconstruction are illustrated in Section 5. Section 6 contains conclusive considerations and proposals on future developments.

2. Preliminaries

In this section we provide a description of the adopted camera model and we will briefly summarize some basic ideas from projective geometry [4,15–17] that will be used in the following sections.

2.1. Camera model

The camera description adopted in this article is basically a pinhole model whose image plane is nonlinearly stretched in order to take the geometric distortion of the optics into account.

Let $\boldsymbol{R}$ be the rotation matrix that describes the orientation of the camera frame in world coordinates, $\boldsymbol{o} \in \Re^3$ the euclidean world coordinates of the origin of the camera frame and f the camera's focal length. The relationship between projective image coordinates $\boldsymbol{u} \in \mathscr{P}^2$ and projective object coordinates $\boldsymbol{x} \in \mathscr{P}^3$ is of the form $\boldsymbol{u} = \boldsymbol{P}\boldsymbol{x}$, whose rank-3 *projection matrix* $\boldsymbol{P} \in \Re^{3\times 4}$ is given by $\boldsymbol{P} = \boldsymbol{T}_{per}\boldsymbol{T}_{wc}$, where

$$\boldsymbol{T}_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}, \quad \boldsymbol{T}_{wc} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R}\boldsymbol{o} \\ 0\,0\,0 & 1 \end{bmatrix},$$

therefore

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{r}_1 & -\boldsymbol{r}_1\boldsymbol{o} \\ \boldsymbol{r}_2 & -\boldsymbol{r}_2\boldsymbol{o} \\ \frac{1}{f}\boldsymbol{r}_3 & -\frac{1}{f}\boldsymbol{r}_3\boldsymbol{o} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$ are the rows of the rotation matrix $\boldsymbol{R}$.

The metric image coordinates $\bar{\boldsymbol{u}} = [u_1,\bar{u}_2,1]^{\mathrm{T}}$, as referred to the camera frame, are obtained by normalizing the homogeneous coordinates $\boldsymbol{u} = [u_1, u_2, u_3]^{\mathrm{T}}$ while, the image coordinates can be expressed in *pixel* through a further 2D translation combined with a scale change, which can be embedded in the projection matrix as follows:

$$\boldsymbol{P}_{ic} = \boldsymbol{T}_{ic}\boldsymbol{P}, \qquad \boldsymbol{T}_{ic} = \begin{bmatrix} \frac{1}{d_1} & 0 & -t_1 \\ 0 & \frac{1}{d_2} & -t_2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where the scale factors d_1 and d_2 represent the horizontal and vertical size of the pixel, respectively, while t_1 and t_2 specify the horizontal and vertical offsets of the principal point, i.e. of the intersection between image plane and optical axis.

The above camera model is based on the assumption that the lens can be well-described by an ideal perspective projection. In fact, lens distortion causes the acquisition system to deviate from ideality and is often well described by a polynomial model [18]. In order to accurately model lens distortion both of its radial and tangential components should be considered [19]. Radial distortion causes the image coordinates to be radially shifted from the principal point, while tangential distortion accounts for the component that is perpendicular to the radial direction. In some situations the tangential component can be assumed as negligible with respect to the radial one [20]. In this case, the undistorted coordinates can be simply expressed as a truncated power series of the form

$$r_{\mathrm{u}} = r_{\mathrm{d}}(1 + k_3 r_{\mathrm{d}}^2 + k_5 r_{\mathrm{d}}^4 + \cdots), \tag{3}$$

where r_{d} and r_{u} are the distances from the principal point of the *distorted* and *undistorted* image points, respectively. Usually, only the first one or two coefficients of Eq. (3) are considered, depending on the application [21]. An alternative way of specifying the camera model consists of adopting a single (larger) projection matrix that maps a vector containing the object coordinates (and, in some cases, mixed products of coordinates up to a certain order) onto the final image coordinates [22].

Throughout the article we will assume that the image coordinates that we work with are all undistorted, which means that the image coordinates must be previously mapped onto undistorted ones through Eq. (3).

2.2. Some facts from epipolar geometry

Two projective views of a point in object space, are bound to comply with the so-called ‘epipolar’ (or ‘essential’) constraint, according to which the two lines that connect the object point with the optical centers of the two projective cameras are coplanar (see Fig. 1). The plane identified by the object point and the two optical centers is called *epipolar plane*. The intersection between the epipolar plane and an image plane is called *epipolar line*, and represents the projective view of the other optical ray. The intersection between an image plane and the line that connects the two optical centers is called *epipole*: all epipolar lines meet at that point, as all epipolar planes have the line that connects the optical centers in common.

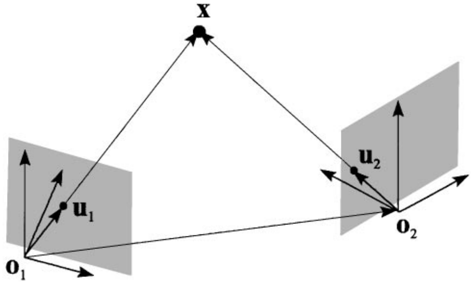

Fig. 1. Essential constraint: the lines of sight of two corresponding points are bound to be coplanar.

Let $\boldsymbol{u}_1 = \boldsymbol{P}_1\boldsymbol{x} \in \mathscr{P}^2$ and $\boldsymbol{u}_2 = \boldsymbol{P}_2\boldsymbol{x} \in \mathscr{P}^2$ be the projective coordinates of an object point (with projective world-coordinates $\boldsymbol{x} \in \mathscr{P}^3$), as seen by to two cameras with projection matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively. Let $\boldsymbol{R}$ and $\boldsymbol{t}$ be the rotation matrix and the translation vector, respectively, that describe the change of camera frame $\boldsymbol{p}_2 = \boldsymbol{R}\boldsymbol{p}_1 + \boldsymbol{t}$, where $\boldsymbol{p}_1 \in \Re^3$ and $\boldsymbol{p}_2 \in \Re^3$ are the camera-coordinates of the object point. The coplanarity of the two lines of sight (lines that connect the object point with the optical centers of the cameras) can be expressed in terms of the orthogonality between $\boldsymbol{p}_2$ and the normal to the plane formed by $\boldsymbol{t}$ and $\boldsymbol{p}_1$. In matrix notation we have $\boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{E}\boldsymbol{p}_1 = 0$, where $\boldsymbol{E} = \boldsymbol{T}\boldsymbol{R} = (\boldsymbol{t}\times)\boldsymbol{R} \in \Re^{3\times 3}$ is called *essential matrix*. Since the projective coordinates $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are proportional to the camera coordinates $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, respectively, the essential constraint can be equivalently rewritten as $\boldsymbol{u}_2^{\mathrm{T}}\boldsymbol{E}\boldsymbol{u}_1 = 0$. Furthermore, if the camera frames are chosen in such a way for the projections to result in canonical form,[1] then this equation can be rewritten as $\bar{\boldsymbol{u}}_2^{\mathrm{T}}\boldsymbol{E}\bar{\boldsymbol{u}}_1 = 0$, where $\bar{\boldsymbol{u}}_1$ and $\bar{\boldsymbol{u}}_2$ are obtained by scaling $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ in such a way that their third component is equal to one. If the projections that describe the cameras are not in

[1] z axis passing through the origin of the image plane and perpendicular to it, unit focal distance.

canonical form, then they can always be made canonical through appropriate homographies (invertible linear projective transformations) $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$

$$\bar{\boldsymbol{u}}_2^{\mathrm{T}} \boldsymbol{Q} \bar{\boldsymbol{u}}_1 = 0, \qquad \boldsymbol{Q} = \boldsymbol{H}_2^{\mathrm{T}} \boldsymbol{E} \boldsymbol{H}_1. \tag{4}$$

Notice that Eq. (4) provides us with a closed-form equation for the epipolar line on either one of the two cameras, associated to a point on the other camera. For example, by letting $\boldsymbol{w} = \boldsymbol{Q} \upsilon_1$, Eq. (4) becomes

$$\boldsymbol{w}^{\mathrm{T}} \upsilon_2 = 0, \tag{5}$$

which is the equation of the epipolar line on camera 2, corresponding to the image point υ_1.

3. Feature extraction

Feature detection and localization is a crucial step in the global reconstruction chain [23], particularly when camera motion estimation is involved. When dealing with features that have been artificially added to the scene, the most common choice is to adopt some advanced template matching process [24]. As we are interested in detecting natural and viewer-invariant scene features, we developed a method for accurately localizing vertices, i.e. crossings between edges. Two classes of solutions are available in the literature for extracting this type of image features: the former works on luminance edges by looking for the edge points of maximum curvature; the latter works on grey-levels by analyzing the gradient and/or the curvature of the luminance profile surface (see [25] for a review). What we propose in this section is an improved version of this last approach.

One advantage of grey-level corner detectors over edge-based methods is that they are able to localize features with sub-pixel accuracy even when highly localized. An edge-based approach, in fact, will produce accurate results only when the segments to be intersected are fairly long. Furthermore, gray-level methods are able to localize corners that correspond to complex intersections of edges. In fact, besides single corners (V-type), they can localize multiple coinciding corners (T-type or Y-type).

It is well-known that, if a luminance transition is modeled as a smoothed step edge, its luminance profile about a vertex point will exhibit a zero Laplacian, no matter how the edges meet at that point [26]. Furthermore, at the same vertex point the Baudet operator

$$\mathrm{DET} = \det \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} = I_{xx} I_{yy} - I_{xy}^2 \tag{6}$$

will exhibit a relative maximum in all directions. An interesting property of the Baudet operator is that the maximum about the vertex will move along a line that intersects the vertex point as we blur the image with a progressively zero-phase low-pass filter. This property can be used together with the one about the Laplacian in order to locate the vertex with super-resolution accuracy. More specifically, we can search for the zero-crossing of the Laplacian along the line of the maxima of the DET.

In [25] the image is initially filtered with a low-pass gaussian anisotropic 2D filter f_1, which is chosen in order to obtain an accurate edge detection in the presence of noise [27]. This 2D filter is separable into two 1D gaussian filterings (horizontal and vertical) with the same standard deviation σ_1. At this point, the elliptical maxima (those that exceed a prefixed threshold) of the DET operator are detected on the filtered image. A second gaussian filter f_2, with std. deviation $\sigma_2 < \sigma_1$, is then applied to the original image and the elliptical maxima of the DET are searched for on the new image in the proximity of the previously determined maxima. As a final step, corners and vertices are determined as the zero crossings of the Laplacian along the lines that connect each pair of maxima (those determined at the two different resolutions). Notice that this corner location can have sub-pixel accuracy.

In order to evaluate the performance of the above corner detector, we conducted some tests on typical real images. The algorithm turned out to extract a significant number of feature points only when the std. deviations σ_1 and σ_2 of the impulse responses were kept modest, which resulted in a modest sub-pixel localization performance.

We found that this limitation can be avoided by extracting four different DET maxima per corner, each corresponding to a progressively more filtered version of the input image. As the four maxima are only approximately collinear, the line on which to search for the zero crossing of the Laplacian is determined through linear regression. The four DET maxima locations are determined with sub-pixel accuracy by interpolating a quadric surface over a regular grid in the neighborhood of the integer (coarsely quantized) location of the maximum. Due to the high level of activity of the DET in the proximity of a vertex, the most reasonable size for the adopted grid turned out to be 3 × 3, as a wider region would not improve the results. At this point we can search for the zero-crossing of the Laplacian of the luminance profile along the determined line. Also in this case we use a polynomial approximation of the Laplacian's profile in order to evaluate the sub-pixel location of the zero-crossing. It is important to point out that, in order to reduce the impact of the noise, the Laplacian operator (as in [25]) is not applied to the original image but on a filtered version of it (using the gaussian impulse response with the smallest scale factor).

In order to further improve the localization's accuracy of our corner detector, we also exploited the fact that the images acquired with an analog CCD camera connected to a frame grabber exhibit different horizontal and vertical spectral extensions. In Fig. 2 we can see the typical luminance profiles associated to a vertical and a horizontal sharp edge as imaged by one of our cameras (we used black and white SONY XC77CE CCD cameras with standard TV resolution).

As we can see, the horizontal luminance profile (vertical edge) turns out to be smoother than the vertical luminance profile (horizontal edge) because the video scan lines are filtered by an analog signal amplifier [28]. This anisotropy can significantly reduce the accuracy with which corner and vertex locations are determined. One way to overcome this problem is to adopt different filtering scales in the horizontal and vertical directions. In order to confirm this fact, we conducted a series of experiments on images that were acquired by a well calibrated camera system based on three of the above CCD TV cameras. After performing feature detection on the 3 available images, a matching process was used for determining correspondences between vertices/corners on different views. Using the calibration information, the feature location on one image can be estimated from the feature location of corresponding points on the other two views. Using the difference between this estimate and the detected location of the feature point as a measure of the quality of the vertex/corner extractor, we found that the best choice of the filtering scales was: $\sigma_{h1} = 1.8$ pel, $\sigma_{h2} = 1.5$ pel, $\sigma_{h3} = 1.2$ pel, $\sigma_{h4} = 0.9$ pel, $\sigma_{v1} = 2.1$ pel, $\sigma_{v2} = 1.8$ pel, $\sigma_{v3} = 1.5$ pel, $\sigma_{v4} = 1.2$ pel, where the subscripts *denote* scale indices and filtering direction. As we can see, the scales of the vertical filter are all approximately 0.3 pel wider than the horizontal ones. The difference between the feature location estimated from the other two images (using the calibration information) and the location determined by the corner extractor turned out to be about 0.15–0.2 pel.

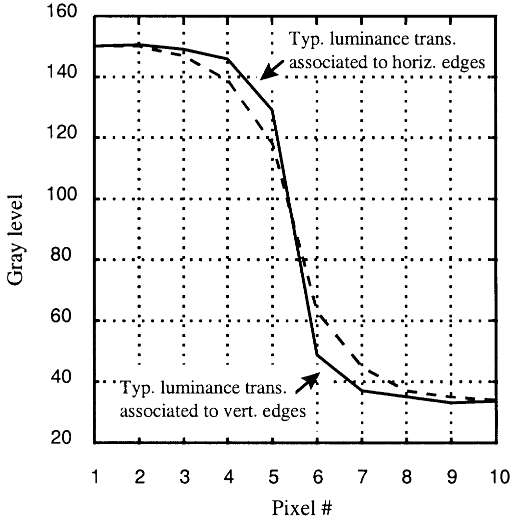

Fig. 2. Typical luminance profiles associated to horizontal and vertical edges.

Fig. 3 shows all the information exploited by the corner extractor, while Fig. 4 shows an example of corner detection.

4. Egomotion estimation

Since our sources of information are the acquired images, it is quite natural to think of motion

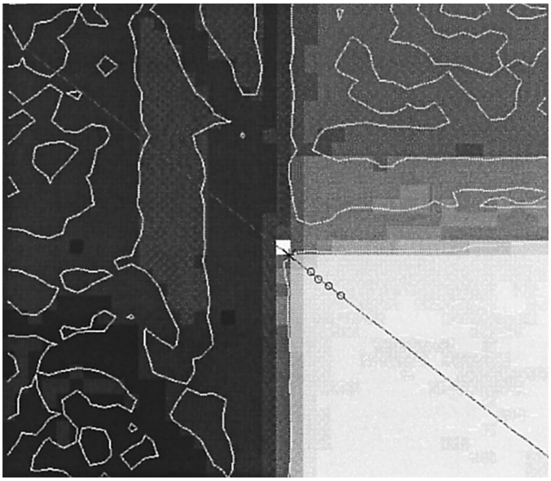

Fig. 3. Zoom-in of the image in a neighborhood of a corner. Circular dots denote the detected sub-pixel positions of DET maxima. The closed curves represent the zero-crossings of the Laplacian. The white square denotes the pixel location of the corner, while the asterisk denotes its sub-pixel location, as detected by the algorithm.

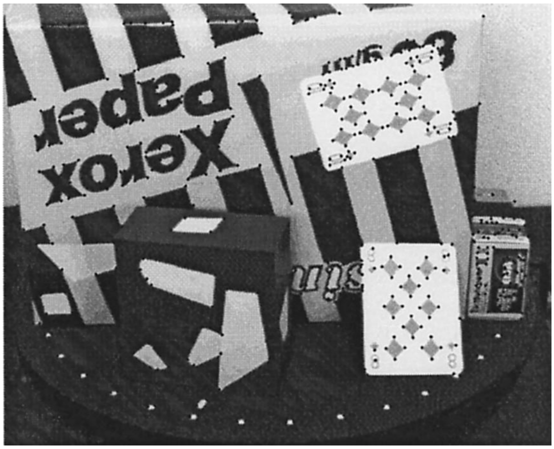

Fig. 4. Example of corner extraction. Dots denote the detected corners.

estimation as a *projective* $(\mathscr{P}^2)$ problem. The literature, in fact, is rich with methods that exploit projective constraints which could be used for a wide variety of applications and data. A number of methods have been developed with the general goal of jointly estimating the structure of the 3D scene and the motion of a camera from a sequence of its views [17,29–31]. As for the egomotion estimation problem, there are methods based on the optical flow [32] or on the tracking of characteristic points [33,34], which are suitable for video acquisitions or sequences of views characterized by a small motion between consecutive frames. One classical projective approach to egomotion estimation, developed for monocular acquisition systems, is that based on the *essential constraint* [35]. Other solutions that are able to deal with significantly large rigid motions of the camera system can be found in [10–13]. Such methods can be quite easily adapted to a multi-camera configuration and, using the information on the camera parameters, it is possible to determine the exact motion as opposed to a scaled one.

One limitation that can be recognized in projective methods is that they do not take full advantage of the multi-ocular geometry. In fact, as each calibrated multi-view originates a partial set of 3D data, we can estimate the motion of the camera system by first performing 3D feature matching and then searching for the rigid motion that best merges the 3D features that are in common between different partial reconstructions. By doing so, we exploit rigidity constraints and transform the egomotion estimation into a *euclidean* $(\mathscr{R}^3)$ problem [36]. One advantage of this solution is to keep the 3D reconstruction phase separate from the 3D data fusion process.

Let us assume that a number of point-like features have been extracted and accurately localized on the available images. In this section we will assess the problem of how to use such information for the purpose of estimating the motion of a multi-ocular camera system. We will consider and compare two solutions: the former is a fairly simple *projective* $(\mathscr{P}^2)$ approach based on the application of the epipolar constraint to corresponding image points, while the latter is a novel *euclidean* $(\mathscr{R}^3)$ solution based on the application of the rigidity constraint to back-projected object points.

4.1. Projective approach

The projective approach to monocular camera motion estimation considered in this section is well-known in the literature and quite common to all non-calibrated reconstruction applications. This method (see Section 2.2) is based on the application of the epipolar constraint [10–12] and, given a number of correspondences between two monocular views, it allows to determine the rigid motion between the two viewpoints up to a scale factor. In this section we illustrate a simple application of this projective approach to multi-ocular camera systems, which allows us to estimate the full (unscaled) egomotion. This solution will be later used as a term of comparison for the Euclidean approach that will be illustrated in Section 4.2.

As a first step, we need to perform point detection and localization with high accuracy, followed by point matching. If, in order to extract the camera motion, some artificial marks have been added to the scene, then we can extract their corresponding image location by means of some advanced template matching algorithm. If such *fiducial* marks are not available, then we can use a corner detector such as that of Section 3. Feature matching is performed through a RANSAC (*RANdom SAmple Consensus*) approach [14], in which a set of candidate matches, generated through a correlation-based approach, is narrowed down to a small set of safe matches by exploiting projective constraints and invariants (e.g. epipolar and trifocal constraints) [30,31].

The literature is rich with methods for estimating the epipolar geometry of the camera system (projective reconstruction) from a number of feature correspondences (see, for example [37]). As the camera system is assumed as calibrated, this operation consists of the sole estimation of the essential matrix. From the essential matrix it is quite straightforward to determine the rotation matrix and the normalized translation vector associated to the rigid motion between views (see, for example, Appendix A).

Let us assume that the acquisition system is made of M cameras. Let $\boldsymbol{p}_i^{\mathrm{b}}$ and $\boldsymbol{p}_i^{\mathrm{a}}$ be the coordinates of a 3D point relative to the ith camera frame *before* and *after* the motion, respectively. Using, for example, the monocular egomotion estimation algorithm of Appendix A, we can compute the rigid motion (up to a scale factor) from each one of the cameras before motion, to each one of the cameras after the motion; and find M^2 pairs of rotation matrices and M^2 (scaled) translation vectors. Since we know the calibration parameters of the camera system and, in particular, the relative position and orientation of the cameras, the information on the M^2 scaled motions can be merged and used synergically. In fact, it is possible to refine the estimates of the scaled rigid motions through a process of data averaging and, at the same time, we can determine the magnitude of the translation as well.

In order to be able to combine the information coming from the M^2 monocular motion estimates, we need to adopt a common reference frame, which can be, for example, that of camera one. In order to express in camera-1 coordinates the rigid motion from the ith camera ($i \neq 1$) and the jth camera ($j \neq 1$), already estimated through the monocular technique, we combine the relationships (obtained through calibration) between cameras frames

$$\boldsymbol{p}_i^{\mathrm{b}} = \boldsymbol{R}_{1i}\boldsymbol{p}_1^{\mathrm{b}} + \boldsymbol{t}_{1i},$$

$$\boldsymbol{p}_j^{\mathrm{a}} = \boldsymbol{R}_{1j}\boldsymbol{p}_1^{\mathrm{a}} + \boldsymbol{t}_{1j},$$

with the estimated motion from any camera frame *before* the motion to any camera frame *after* the motion (obtained through monocular estimation)

$$\boldsymbol{p}_j^{\mathrm{a}} = \boldsymbol{R}_{ij}^{\mathrm{ba}}\boldsymbol{p}_i^{\mathrm{b}} + \gamma_{ij}\bar{\boldsymbol{t}}_{ij}^{\mathrm{ba}},$$

where γ_{ij} is a scale factor that accounts for the fact that only the direction $\bar{\boldsymbol{t}}_{ij}^{\mathrm{ba}}$ of the translation $\boldsymbol{t}_{ij}^{\mathrm{ba}}$ can be estimated through a monocular technique. For any pair of cameras before and after the motion, we obtain a different estimate of the motion of the reference camera

$$\boldsymbol{p}_1^{\mathrm{a}} = \boldsymbol{R}_1^{(ij)}\boldsymbol{p}_1^{\mathrm{b}} + \mu_{ij}\bar{\boldsymbol{t}}_1^{(ij)},$$

where

$$\boldsymbol{R}_1^{(ij)} = \boldsymbol{R}_{1i}^{\mathrm{T}}\boldsymbol{R}_{ij}^{\mathrm{ba}}\boldsymbol{R}_{1j},$$

$$\bar{\boldsymbol{t}}_1^{(ij)} = \boldsymbol{R}_{1i}^{\mathrm{T}}(\boldsymbol{R}_{ij}^{\mathrm{ba}}\boldsymbol{t}_{1j} - \boldsymbol{t}_{1i} + \gamma_{ij}\bar{\boldsymbol{t}}_{ij}^{\mathrm{ba}}). \qquad (7)$$

Through this procedure we obtain M^2 different estimates of the rotation matrix $\boldsymbol{R}_1^{(ij)}$, $i,j = 1,\dots,M$, which can be averaged together in order to obtain a more accurate estimate $\boldsymbol{R}_1$ of the global rotational motion. In order to do so, we proceed by averaging the rotation vectors associated to the available instances of $\boldsymbol{R}_1^{(i,j)}$, so that the whole operation is performed in a vector space

$$\boldsymbol{R}_1 = \exp\left(\frac{1}{M^2}\sum_{i,j=1}^{M} \log \boldsymbol{R}_1^{(i,j)}\right),$$

where the computation of the (principal value of the) matrix logarithm of the rotation matrices $\boldsymbol{R}_1^{(i,j)}$ and the exponentiation of the resulting skew-symmetric average can be performed by means of the Cayley's formula [38], and the Rodrigues formula [39], respectively. Although averaging rotation matrices is correctly done in exponential coordinates, it is also possible to directly average Euler angles. In fact, the results in this case do not significantly differ from those obtained as shown above.

The above procedure is useful for improving the accuracy of the rotation's estimate but does not help the estimation of the translation vector. In fact, since we do not know the scale factors γ_{ij}, Eq. (7) cannot be used. We thus need to determine the magnitude of the translation, which is chosen as the distance between the origin of the reference camera frame before the motion, and the origin of the same camera frame after the motion. In order to do so, we consider the direction of translation from the center $\boldsymbol{o}_i^{\mathrm{b}}$ of each one of the camera frames before the motion and the center $\boldsymbol{o}_j^{\mathrm{a}}$ of any reference frame after the motion. Ideally, the lines characterized by such directions are bound to meet at $\boldsymbol{o}_j^{\mathrm{a}}$, which allows us to determine the magnitude of the translation, as shown in Fig. 5. In practice, however, the lines do not meet at one point due to the noise and the unavoidable feature localization errors. It is thus reasonable to determine the position of $\boldsymbol{o}_j^{\mathrm{a}}$ as the point at minimum distance from the above lines. By performing this operation for $j = 1, \dots, n$, we end up with M different estimates of the translation's magnitude, which can be averaged for accuracy improvement.

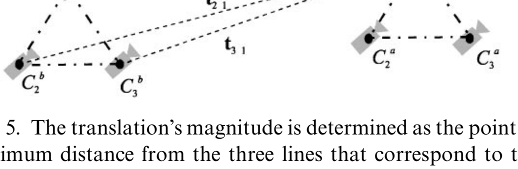

Fig. 5. The translation's magnitude is determined as the point of minimum distance from the three lines that correspond to the directions of translation from any camera before the motion to one specific camera after the motion.

It is important to notice that the accuracy of the translation's magnitude estimation depends on how far apart the two multi-view acquisition are, compared with the distance between cameras.

4.2. Euclidean approach

As already said above, the euclidean approach to camera motion estimation consists of searching for the rigid motion that best merges the 3D data shared by different partial reconstructions. In order to do so, we need to define an index of the *quality* of the merging process as a function of the 6 motion parameters $(\boldsymbol{\alpha},\boldsymbol{t})$, where $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\alpha_3]$ is the vector of the Euler angles and $\boldsymbol{t} = [t_1, t_2, t_3]$ is the vector of the translations. If we assume that all pairs of points of the two partial reconstructions have been correctly matched, a reasonable choice for this *merging cost* is the mean square distance between corresponding points

$$\mathscr{C}_{3D}(\boldsymbol{\alpha},\boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n} \delta_i^2, \tag{8}$$

δ_i being the distance between the ith pair of corresponding points in common between the two partial reconstructions.

Prior to merging partial reconstructions, we need to find the correspondences between 3D features that pertain different multi-ocular views. Unlike the projective method, which is based on image feature matching, the euclidean method is based on the determination of 3D data correspondences. Consequently, data matching can now rely on much stronger constrains (rigidity) and invariants (inner and vector products), which makes the matching process more robust.

Our 3D point matching strategy is based on a robust triplet-by-triplet approach. We select a random triplet of non-collinear 3D points in the first data set and search, in the second data set, for a triplet whose distances match those of the reference one. For each of such triplets, we determine the rigid motion associated to it, we apply it to all other the points of the data set and we count the number of matches that we obtain. We then determine a limited number of candidate rigid motions through a clustering process applied to the determined solutions. The final match is chosen as the one whose rigid motion maximizes a quality index based on the count of matches and on the intra-cluster scattering.

Now that the correspondences have been determined, we can proceed with the motion estimation of the camera system. Although the global minimum of the merging cost (8) can be determined through any multivariate minimization process, some precautions are to be taken in order to prevent the process from converging to a relative minimum instead. The behavior of the merging cost function, in fact, is quite regular only in the close proximity of the global minimum. This fact is shown in Fig. 6, where the merging cost of the multi-ocular method is plotted as a function of two of the three translational components in the case when the other motion parameters are correct (a) and in the case when the other parameters are changed of just 0.5% (b). As we can see, such a small deviation from correctness causes the merging cost to behave quite wildly in the proximity of the global minimum. In order to overcome this difficulty, we need to make sure that algorithm will start from a point that is already in the close proximity of the global minimum, so that no relative minima will be encountered during the iterations.

The presence of relative minima is to be attributed to the fact that the merging cost (8) is a highly nonlinear function of the Euler angles, which represent a minimal parametrization of the rotation. If we express the merging cost as a function of a rotation matrix (which is an overparametrized representation of the rigid rotation), then instead of a problem of nonlinear minimization, we end up with a problem of constrained linear minimization (the matrix is bound to be orthogonal). As a matter of fact, the mean square distance is a quadratic function of the nine components of the rotation matrix, therefore it only has one global minimum. On the other hand, due to the limited accuracy of the data, this minimum is not generally an orthogonal matrix, therefore using this approach would require us to *project* this matrix onto the space of rotation matrices, i.e. to find its *closest* orthogonal matrix. This reasoning leads to a simple method for determining a good starting point for the nonlinear minimization algorithm, which consists of projecting the *linear* minimum onto the manifold of the rotation matrices. This approach can be summarized in the following two steps:

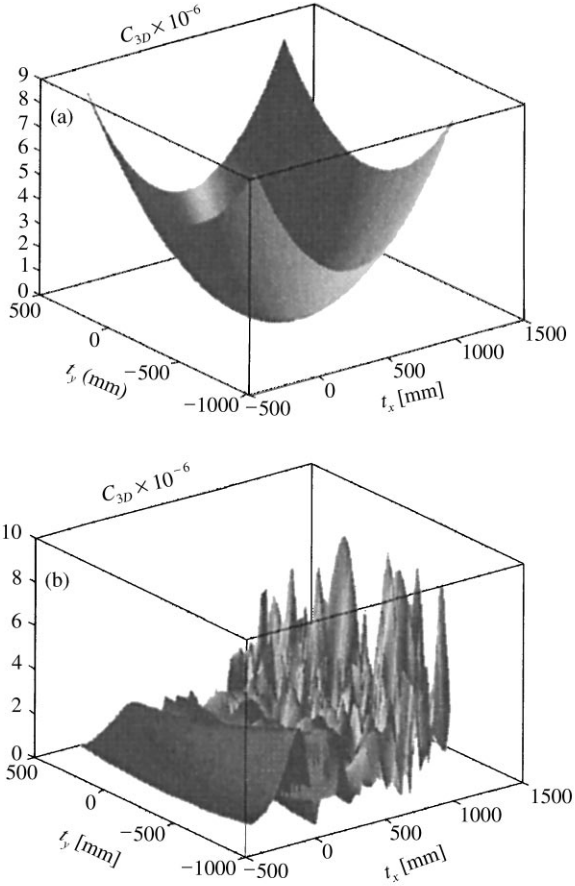

Fig. 6. Behavior of the merging cost of the multi-ocular method in the proximity of the global minimum. The merging cost is plotted as a function of two of the three translation components in the case when the other motion parameters are correct (a) and in the case when the other parameters are changed of 0.5% (b).

Linear estimation — determinate a rough approximation of the global motion

1. • find the 9 + 3 parameters $(\boldsymbol{R},\boldsymbol{t})$ that minimize the merging cost (8),

- *project* the result onto the set of rigid motions;

Nonlinear estimation – refine the estimate of the global motion

1. ● find the 6 parameters $(\boldsymbol{\alpha},\boldsymbol{t})$ that minimize the merging cost (8) starting from the solution of Step 1.

4.2.1. Linear estimation

Let us consider two sets of 3D data, associated to two multi-views acquired before and after the camera motion. Given a pair of corresponding 3D points (one per data set), the *merging error* δ_i between them is defined as the distance between such points. Let $\boldsymbol{p}^{\mathrm{b}}(i)$ be the coordinates of the ith point relative to the reference camera frame before the motion, and let $\boldsymbol{p}^{\mathrm{a}}(i)$ be the coordinates of the corresponding point relative to the reference camera frame after the motion. If $(\boldsymbol{R},\boldsymbol{t})$ is the camera motion to be estimated, then the merging error for this pair of 3D points can be easily written as

$$\delta_i(\boldsymbol{R},\boldsymbol{t}) = \|\boldsymbol{R}\boldsymbol{p}^{\mathrm{b}}(i) + \boldsymbol{t} - \boldsymbol{p}^{\mathrm{a}}(i)\|.$$

With N pairs of corresponding points the global merging cost becomes

$$\begin{aligned}\mathscr{C}_{3D}(\boldsymbol{R},\boldsymbol{t}) &= \sum_{i=1}^{N} \delta_i^2(\boldsymbol{R},\boldsymbol{t}) \\ &= \sum_{i=1}^{N} (\boldsymbol{R}\boldsymbol{p}^{\mathrm{b}}(i) + \boldsymbol{t} - \boldsymbol{p}^{\mathrm{a}}(i))^{\mathrm{T}}(\boldsymbol{R}\boldsymbol{p}^{\mathrm{b}}(i) + \boldsymbol{t} - \boldsymbol{p}^{\mathrm{a}}(i)). \quad (9)\end{aligned}$$

This merging cost is a quadratic function of the $9+3$ components of $(\boldsymbol{R},\boldsymbol{t})$. Therefore the global minimum of this function can be easily determined through the solution of a linear system of 12 equations in 12 unknowns of the form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad (10)$$

whose solution $\boldsymbol{x}$, in the ideal case of perfect 3D data, contains the elements of the components of the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$. In general, due to the fact that we are dealing with real data, the solution of the linear system of equations (10) is not a rigid motion and, in particular, the resulting matrix $\hat{\boldsymbol{R}}$ is not orthogonal. We thus need to *project* $\hat{\boldsymbol{R}}$ onto the set SO(3) of the rotation matrices, which can be done by selecting the rotation matrix $\boldsymbol{R}$ that minimizes the Frobenius norm of $\hat{\boldsymbol{R}} - \boldsymbol{R}$. One way to do so is to look for the Euler angles $\boldsymbol{\alpha}$ that minimize

$$\mathscr{D}(\boldsymbol{\alpha}) = \sum_{i=1}^{3}\sum_{j=1}^{3} (\hat{R}(i,j) - R_{\boldsymbol{\alpha}}(i,j))^2,$$

where $R_{\boldsymbol{\alpha}}(i,j)$ is the (i,j)th component of the rotation matrix, written as a function of the Euler angles $\boldsymbol{\alpha} = [\alpha_1,\alpha_2,\alpha_3]^{\mathrm{T}}$ through Eq. (B.4). This minimization can be performed, for example, through a downhill simplex method [40].

An alternative method for computing the projection of $\hat{\boldsymbol{R}}$ onto SO(3) is to first compute $\hat{\boldsymbol{V}} = \log \hat{\boldsymbol{R}}$, then project it onto the 3D vector space so(3) of all skew-symmetric matrices. The final rotation matrix is then computed through exponentiation of the result, by means of the Rodrigues' formula (B.2).

4.2.2. Nonlinear estimation

If we express the rotation matrix as a function of the Euler angles, the global merging cost

$$\mathscr{C}_{3\mathrm{D}}(\boldsymbol{\alpha},\boldsymbol{t}) = \sum_{i=1}^{m} (\boldsymbol{R}(\boldsymbol{\alpha})\boldsymbol{p}^{\mathrm{b}}(i) + \boldsymbol{t} - \boldsymbol{p}^{\mathrm{a}}(i))^{\mathrm{T}}(\boldsymbol{R}(\boldsymbol{\alpha})\boldsymbol{p}^{\mathrm{b}}(i) + \boldsymbol{t} - \boldsymbol{p}^{\mathrm{a}}(i)) \quad (11)$$

becomes a nonlinear function of the six motion parameters $(\boldsymbol{\alpha},\boldsymbol{t})$. However, since we now have a rather accurate first estimate of the rigid motion, computed through linear minimization, we can trust a nonlinear minimization algorithm (such as the downhill simplex method) to converge to the global minimum when starting from that estimate. The global minimum that we find represents our best estimate of the rigid motion between the two multi-views that originated the two partial reconstructions.

Notice that the linear system (10), used for accurately initializing the minimization process, has 12 unknowns and 12 equations. Therefore it can only be solved when at least 4 pairs of corresponding 3D points are available. On the other hand, we know that a rigid motion is fully specified by six parameters. Therefore, in order to completely determine the motion parameters, only three noncollinear 3D points are strictly necessary. When only three matches are available, however, it is always possible to minimize their merging cost through a geometrical approach.

4.3. Comparative analysis

One major reason to prefer a Euclidean approach to a projective one for an accurate estimation of the camera motion is the reliability of the feature matching process. In both the proposed methods, correspondences can be determined through some robust procedure that compares clusters of a set of (2D or 3D) points. In this last step, however, the reliability of the comparisons between clusters of points severely depends on the constraints and invariants that are being exploited. For example, the projective approach can obviously rely only on projective invariants and constraints (essential, trifocal, etc.), while the correspondences between 3D features are based on Euclidean invariants (distance) and constraints (rigidity). Nonetheless, the comparative performance evaluation conducted in this section assumes that all the determined correspondences in the projective case are correct, and only focuses on the egomotion estimation algorithm. The fact that the 3D matching process is virtually exempt from errors makes the evaluation of the performance improvement (from a projective method to a 3D approach) more conservative.

In order to conduct a thorough comparative analysis of the performance of the two methods, it would be desirable to study the error propagation through the various stages of the motion estimation technique in both the projective and the Euclidean cases. Unfortunately, this sort of analysis is quite complex to conduct, especially in the projective case. For this reason, in order to compare the accuracy and the convergence properties of the two egomotion estimation methods introduced in this section, we performed a series of experiments with synthetic data, relative to a trinocular acquisition system. We generated a cluster of N points, uniformly scattered in a volume of $300\times 300\times 100$ mm, at a distance of about 1 m from the acquisition system. The cameras were assumed as being attached to a rigid frame, at the vertices of a triangle whose sides were about 60 cm, with quite strongly converging axes; and their focal length was assumed as equal to 16 mm. After having applied a rigid motion to the cluster of points, we determined their perspective projection onto the three image planes and, finally, added to the image coordinates of the projected points a Gaussian noise with independent components which had zero mean and a standard deviation of $\sigma_x = \sigma_y = 2.75$ μm (corresponding to 1/4 of a pixel). This noise was added in order to model the feature localization error. With this data, we estimated the rigid motion by using the $\mathscr{P}^2$ approach and the $\mathscr{R}^3$ method proposed in this section. We determined the rigid motion for many different numbers N of points. More precisely, we computed 200 instances of the rigid motion for each choice of N.

From the results of the simulation, we computed the relative standard deviation of the estimation errors associated to each one of the motion parameters. More specifically, denoting with $\boldsymbol{m}$ the vector of the motion parameters, with m_k its kth component, and with σ_k the standard deviation of its estimation error, we computed the relative error (in percent) associated to the kth parameter of the rigid motion as $e_k = 100\,\sigma_k/m_k$.

With reference to Figs. 7a and b, we immediately notice that the Euclidean method results as being much more robust against noise than the projective approach. In fact, with less than a dozen points the relative error associated to the $\mathscr{P}^2$ method results as being from 6 to 10 times larger than that of the multi-ocular method. As N increases, the ratio between the relative error in the $\mathscr{P}^2$ case and that of the $\mathscr{R}^3$ case decreases until it settles to a value that lies between 2 and 3. A further advantage of the Euclidean method with respect to the projective approach lies on the fact that it works with as little as three points, while the 2D approach requires at least 5 points, although it can be considered as reliable only when using at least 8 points (see Section A).

The simulation described above, when applied with different noise levels, provides us with a rather precise indication of the error that we can expect to have when using a certain number of corresponding points in the estimation of the motion between

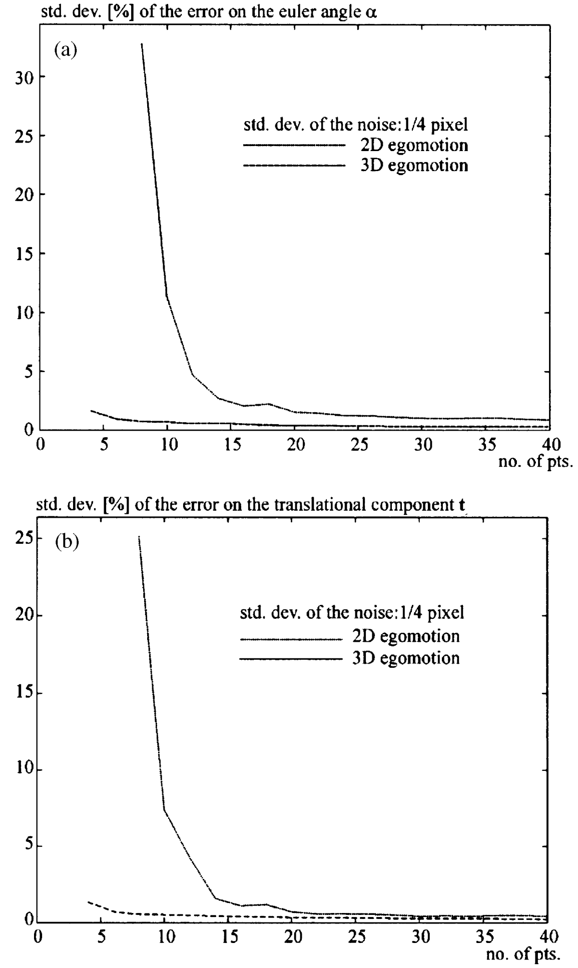

Fig. 7. Accuracy of the estimation of a translation component (a) and of a rotation angle (b), plotted as a function of the number of matched pairs of points. The curves concern the projective case and the Euclidean case.

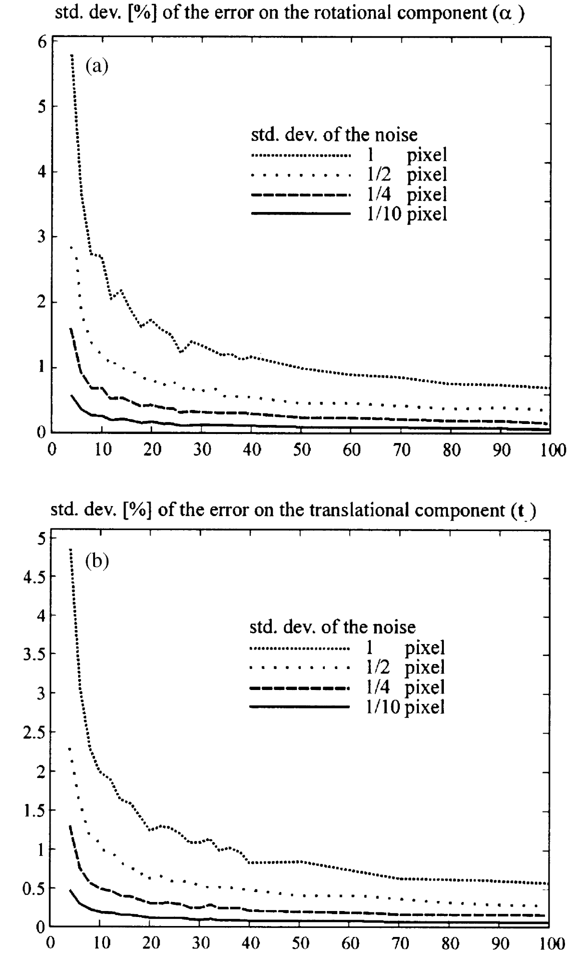

Fig. 8. Estimation error of a rotation angle (a) and of a translational component (b) as a function of the number of matched pairs of points, as the noise power varies.

two different 3D views. As we can see from Figs. 8a and b, which plot the results for 4 different noise levels, the behavior of all four curves approximately follow the inverse of the square root of the number of points n. From Fig. 9, which plots the motion estimation error as the noise power on the image plane increases, we also notice that the error on the estimated motion parameters is approximately proportional to the noise intensity that the data are affected by.

As already said above, both methods proposed in this article are based on a nonlinear multivariate minimization process, whose ability to converge to a global minimum relies on a careful choice of the starting point. In order to have an idea of the behavior of the cost functions in the proximity of the global minimum (see Sections 4.1 and 4.2), both the 3D and the projective cost functions are plotted in Figs. 10 and 11, for the same rigid motion, the same noise configuration (isotropic Gaussian noise with zero mean and 1/4 pixel of standard deviation) and the same number of points ($n = 10$). For obvious graphical reasons, Figs. 10a and b have been obtained by holding the Euler angle relative to the z axis, while varying the other two Euler angles;

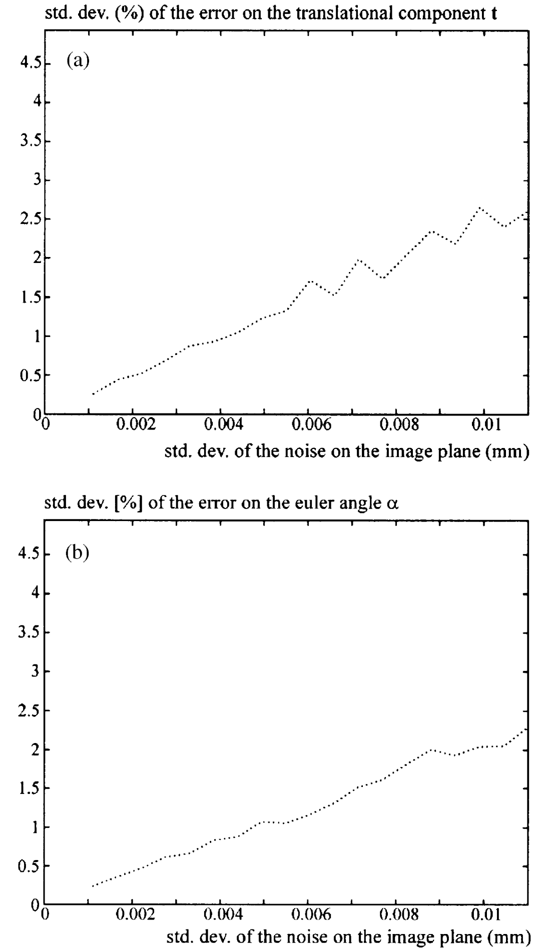

Fig. 9. Estimation error of a rotation angle (a) and of a translational component (b) as a function of the noise power on the image plane.

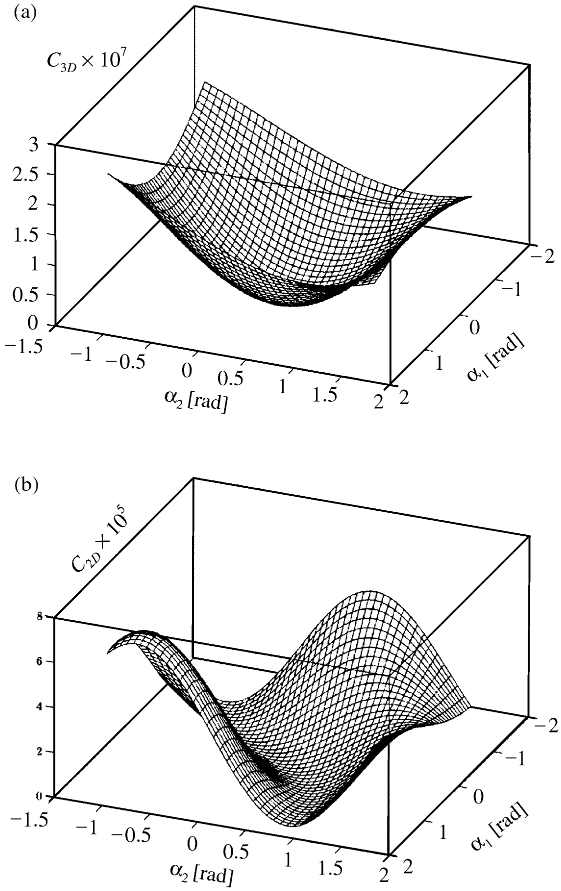

Fig. 10. Cost function for a cluster of 10 points as a function of two Euler angles: (a) multi-ocular estimations, (b) extended monocular estimation. Notice that the scales differ of more than one order of magnitude.

similarly, in Figs. 11c and d the cost function is plotted as a function of the x and y components of the translation vector. Although the two cost functions are of a different nature and, therefore, cannot be directly confronted, some qualitative conclusions can still be drawn. The same multi-variate optimization process is, in fact, applied to both cost functions in the same conditions and as a function of the same parameters. Considering that, for graphical reasons, the two plots have different scales, we notice immediately that the Euclidean merging cost decreases much more rapidly toward the global minimum than the projective cost function. In fact, the slope of the Euclidean cost function at the border of the plotted region is about one order of magnitude larger than that of the projective case. It is thus reasonable to expect the $\mathscr{R}^3$ method to largely outperform the $\mathscr{P}^2$ method in terms of convergence speed, and this is confirmed by the simulation experiments.

Another important aspect to consider in the comparison between the two methods is the computational load. Both the projective approach and the Euclidean method require feature extraction to be carried out, therefore this block can be ignored in the comparison. Point matching in the projective case, compared with 3D point matching, is quite

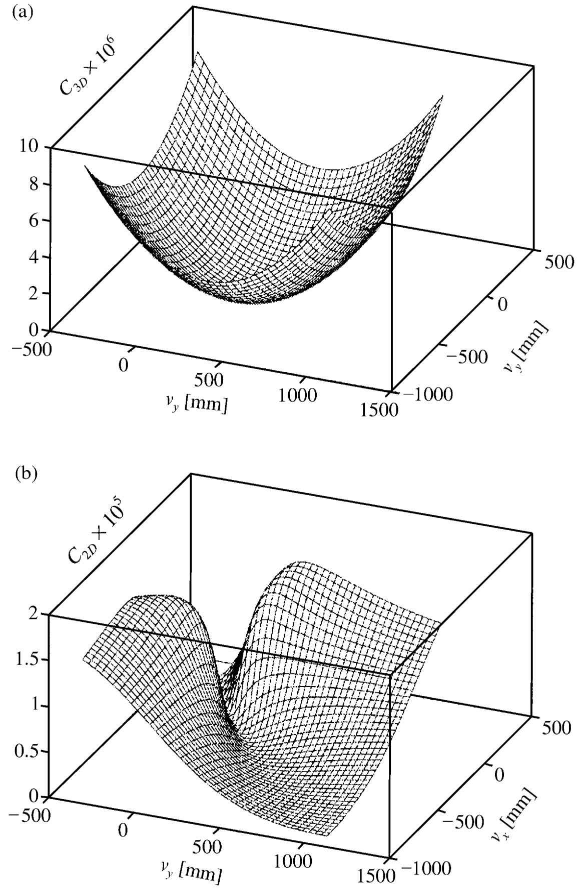

Fig. 11. Cost function for a cluster of 10 points as a function of two translation components: (a) multi-ocular estimations and (b) extended monocular estimation. Notice that the scales differ of more than one order of magnitude.

a heavy task to perform as it requires to apply a correlation-based matching procedure, followed by a series of tests to be performed on a large number of combinations of points. Furthermore, while 3D point matching is performed only once for each pair of multi-views, in the projective case matches must be found between 9 pairs of views. The same considerations apply for the motion estimation procedures. In practice, even with an equal number of matched points (which means better performance for the 3D method), the projective motion estimation requires a computational load that is approximately two orders of magnitude heavier than in the 3D case.

5. Examples of application on real data

The analysis of the two motion estimation methods presented in Section 4 confirms that fully exploiting the calibrated multi-ocular geometry results in a substantial improvement of the accuracy and of the convergence properties of the motion estimator. The experiments that we performed on a real multi-ocular acquisitions are thus limited to the Euclidean approach.

In order to quantify the accuracy and test the convergence of the Euclidean technique of Section 4.2, some experiments of data merging were performed with an acquisition system based on three standard TV-resolution CCD cameras mounted on a rigid frame at the vertices of a triangle (whose sides were approximately 600 mm) and strongly converging axes. We conducted a series of tests on a number of objects. For each one of them we acquired a sequence of triplets of views. In order to speed up the acquisition, the objects were placed on a low-cost turntable while the acquisition system was still. However, no motion constraints were considered for the estimation process.

For each sequence of views, we considered two types of experiments, depending on whether the artificial targets (black circles) printed on the rotating support; or natural point-like features (corners) extracted from the images; were used for computing the motion. In the former case, an accurate template-matching technique was used for estimating the subpixel location of the projection of the centers of the circles on the image planes. In the latter case, we used the corner extraction approach described in Section 3.

The fact that the cameras were placed at the vertices of a triangle guaranteed favorable conditions for the exploitation of the epipolar geometry and for an accurate detection, matching and back-projection of the 3D features. With this acquisition setup and our camera calibration procedures [1], the 3D localization's accuracy of back-projected data turned out to be of about 100–200 ppm [41].

The first object that we used for our experiments was a toy elephant ('elephant' sequence, see top of Fig. 12) made of Lego blocks (approximate size: 120 × 70 × 100 mm). We acquired a sequence of 36 triplets, taken approximately every 10 degrees of

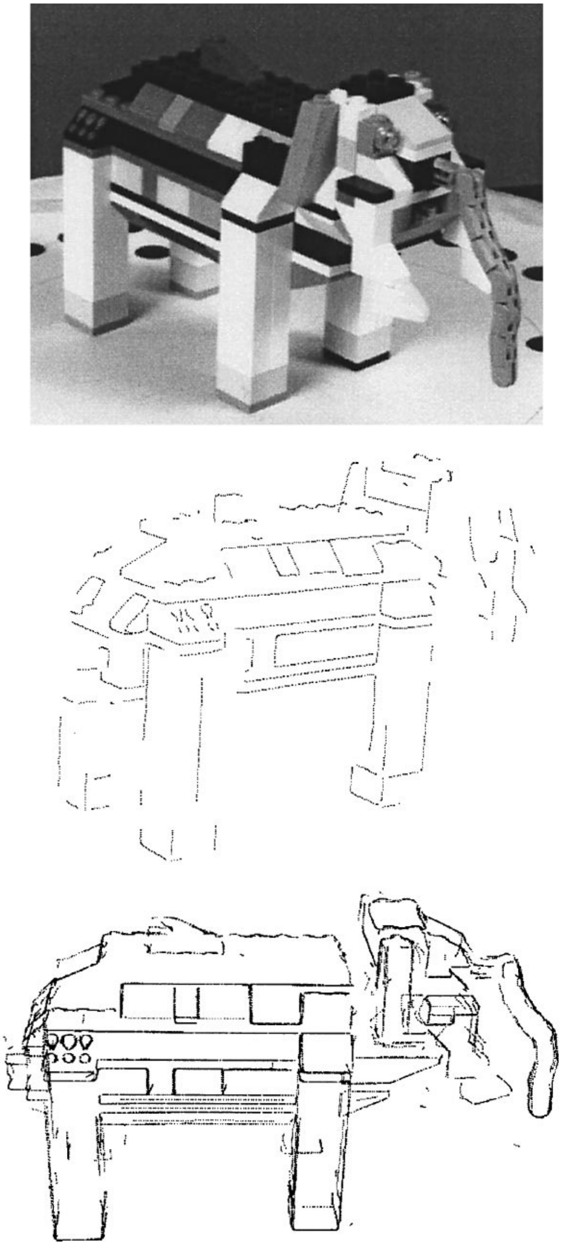

Fig. 12. Example of 3D data fusion using the Euclidean approach. From top to bottom: one of the original views of the object; one of the partial 3D reconstructions obtained through edge matching from a triplet of views; and a view of the result of 3D edge merging using the motion information estimated with the Euclidean approach. Only front partial 3D data is here merged for reasons of rendering intelligibility.

rotation of the turntable. We estimated the viewpoint positions and orientations through the multi-ocular technique proposed in Section 4.2 and, in order to evaluate their accuracy, we chained together all the rigid motions, the last motion being between the last position and the first one. As the final position was expected to correspond to the first one, this process allowed us to measure the *closing error*, i.e. the magnitude of the residual motion after completing the full sequence of motions. We did this using both the artificial targets (circles printed on the turntable's plane) and the corners that are naturally present on the available views. As the stickers were regularly scattered on a circle around the object, in order to avoid alias in their matching between different partial reconstructions, a second series of circles was irregularly scattered at a smaller distance from the turntable's rotation axis. Thanks to the accuracy of our target localization method and the optimal relative positioning of the markers in the images, the global closing error (after chaining 36 motion estimates) turned out to be as low as 0.3 mm. As for the motion estimation based on natural features, the global closing error on a chain of 36 motions turned out to be about 0.8 mm. This accuracy reduction was, indeed, expected as the targets are optimally positioned in the scene and their localization accuracy is generally better than that of the corners. In fact, although the corner detection on the 'elephant' sequence turned out to be very accurate, the shape of the object was such that 3D features used for motion detection were not optimally scattered in the scene (features extracted from a front or a rear view of the object turn out to be very close to each other). If we compare the accuracy of the motion detection from natural features with that performed on artificial features, we find that the motion estimation error changes with the viewing angle; and that the maximum error is, in this case, five times larger than the minimum error.

The object that we used in the second experiment was a toy train engine ('train' sequence, see top of Fig. 13), with an approximate size of 300 × 100 × 150 mm. The sequence was made of 12 triplets of views taken approximately every 30 degrees of rotation of the turntable. The accuracy of the motion estimation based on the artificial targets

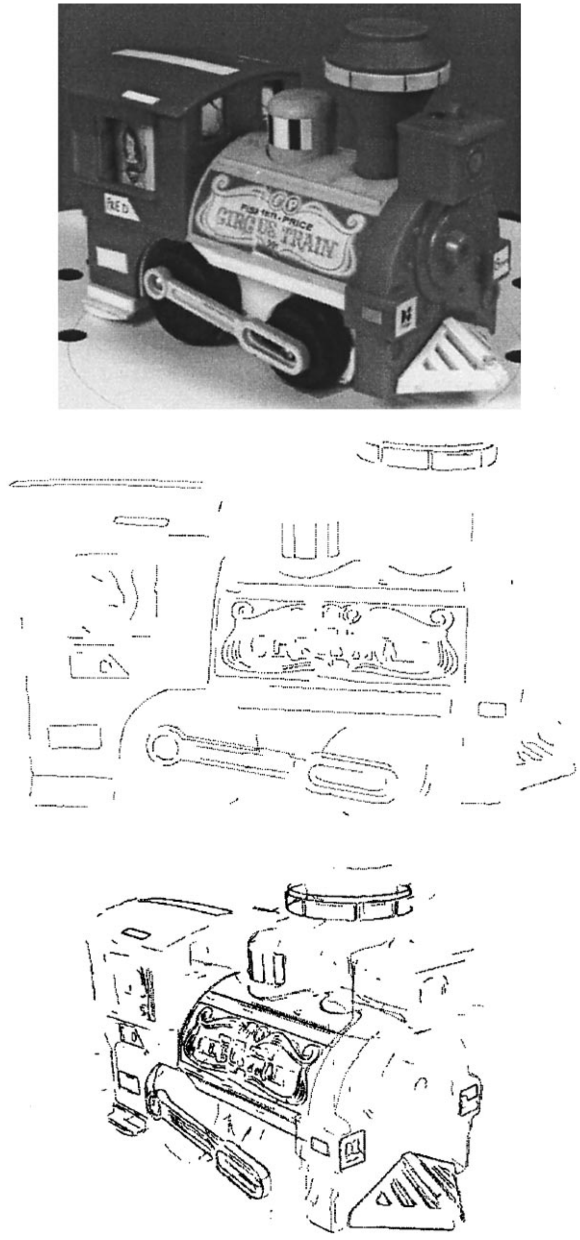

Fig. 13. A second example of 3D data fusion using the Euclidean approach. From top to bottom: one of the original views of the object; one of the partial 3D reconstructions obtained through edge matching from a triplet of views; and a view of the result of 3D edge merging using the motion information estimated with the Euclidean approach. Only front partial 3D data is here merged for reasons of rendering intelligibility.

printed on the turntable's surface was approximately the same as in the previous case. Motion estimation from natural features, however, turned out to be less accurate than before (1.8 mm of closing error). In fact, the intrinsic accuracy of the corner localization on the available images cannot be very high, as such features correspond in 3D to points that are mildly viewer-dependent (nonsharp edges, non-optimal illumination conditions, etc.). Furthermore, also in this case the detected corners tend to cluster together.

The above tests have been conducted on 'circular' sequences of motions that take the acquisition back to the initial position. Although this is not an a priori assumption of the acquisition system, this choice is quite convenient for a performance evaluation of the motion estimation technique, as it allows us to compute the closing error. As a matter of fact the motion estimation accuracy can be further improved through a constrained motion estimation strategy in which the global closing error is forced to be zero.

The result of the global merging of the back-projected 3D edges obtained from a number of triplets (not all of them only for reasons of visual intelligibility) is shown in Figs. 12 and 13. As we can see, homologous 3D edges bundle up quite accurately (no visible differences can be detected between using artificial targets or image corners). In fact, in the case of motion detection with artificial targets, the maximum thickness of the bundles of homologous edges results as being smaller than 0.5 mm, which corresponds to a relative precision of about 200 ppm. This result, which is in agreement with the expected accuracy of the 3D reconstruction, confirms the quality of the camera motion estimation.

6. Conclusions

In this article we presented our global approach to 3D modeling through an accurate merging of partial reconstructions obtained with a calibrated multi-camera system. We assessed the problem of how to maximize the accuracy of natural feature detection in order to obtain the best quality in the final reconstruction. We analyzed and compared two motion estimation methods in which no a priori information and no constraints on the motion of the acquisition system are assumed available. The former solution is a simple adaptation to the multi-camera case of a classical projective approach to motion estimation (well-known for monocular acquisition systems), while the latter is a Euclidean method that we specifically developed for calibrated multi-ocular camera systems. We evaluated the impact of the a priori knowledge of the camera geometry on the accuracy of the motion estimation and on the convergence properties of the algorithm. In both the projective and the Euclidean cases we solved the point correspondence problem using a robust method based on the RANdom SAmple Consensus approach.

In order to implement and test the two methods proposed in this article, we implemented a low-cost high-accuracy *full*-3D reconstruction system based on a calibrated set of three standard TV-resolution CCD cameras. Extensive tests of this camera setup with several multi-view sequences of real scenes, proved the reliability of the Euclidean approach. The proposed method was proven to be capable of highly-accurate results even when using low-cost imaging devices (standard TV-resolution cameras connected to a commercial frame-grabber).

Further extensions of this work are being made in order to estimate the camera motion from 3D data of various nature, such as 3D points, curves [42] and texture.

Appendix A. A projective monocular approach

Let $\boldsymbol{u}^{\mathrm{b}}(i)$ and $\boldsymbol{u}^{\mathrm{a}}(i)$ be the camera coordinates of the ith point, $i = 1,\dots,N$, of two views. For each one of the matched pairs of points we can write the essential constraint as follows:

$$(\boldsymbol{u}^{\mathrm{b}}(i))^{\mathrm{T}}\boldsymbol{E}\boldsymbol{u}^{\mathrm{a}}(i) = 0,$$

$$\boldsymbol{u}^{\mathrm{b}}(i) = \begin{bmatrix} u^{\mathrm{b}}(i) \\ v^{\mathrm{b}}(i) \\ f^{\mathrm{b}} \end{bmatrix}, \boldsymbol{x}^{\mathrm{a}}(i) = \begin{bmatrix} u^{\mathrm{a}}(i) \\ v^{\mathrm{a}}(i) \\ f^{\mathrm{a}} \end{bmatrix}, \quad i = 1,\dots,N. \tag{A.1}$$

This set of N equations can be rewritten in a more compact matrix form as

$$\boldsymbol{C}\boldsymbol{e} = \boldsymbol{0}_n, \tag{A.2}$$

where

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{u}^{\mathrm{b}}(1)^{\mathrm{T}}u^{\mathrm{a}}(1) & \boldsymbol{u}^{\mathrm{b}}(1)^{\mathrm{T}}v^{\mathrm{a}}(1) & \boldsymbol{u}^{\mathrm{b}}(1)^{\mathrm{T}}f^{\mathrm{a}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{u}^{\mathrm{b}}(n)^{\mathrm{T}}u^{\mathrm{a}}(n) & \boldsymbol{u}^{\mathrm{b}}(n)^{\mathrm{T}}v^{\mathrm{a}}(n) & \boldsymbol{u}^{\mathrm{b}}(n)^{\mathrm{T}}f^{\mathrm{a}} \end{bmatrix},$$

$$\boldsymbol{e} = \begin{bmatrix} e_{11} \\ e_{21} \\ \vdots \\ e_{33} \end{bmatrix}.$$

Due to the limited resolution of the images and the inevitable localization errors, the elements of a generic essential matrix $\boldsymbol{E}$ will not exactly satisfy Eqs. (A.1). The rigid motion that best complies with the essential constraint is thus the one that minimizes some norm of the vector of the residuals $\boldsymbol{\varepsilon} = [\varepsilon_1,\varepsilon_2,\dots,\varepsilon_n]^{\mathrm{T}} = \boldsymbol{C}\boldsymbol{e}$. Adopting a mean square norm, the cost function that measures 'how well' the essential constraint is satisfied becomes

$$\mathscr{C}_{2D}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = \sum_{i=1}^{n} \varepsilon_i^2, \tag{A.3}$$

where the Euler angles $\boldsymbol{\alpha}$ and the spherical coordinates $\boldsymbol{\beta} = [\beta_1,\beta_2]^{\mathrm{T}}$ that characterize the direction of translation represent a minimal parametrization of the essential matrix $\boldsymbol{E}$. In conclusion, the projective approach to motion estimation consists of the minimization of the cost function (A.3) in the 5-dimensional motion parameter space $(\boldsymbol{\alpha},\boldsymbol{\beta})$.

The cost function (A.3) is highly nonlinear in the parameters $(\boldsymbol{\alpha},\boldsymbol{\beta})$, therefore its minimization must be carefully dealt with in order to prevent the algorithm from settling for a relative minimum rather than converging to the global one. Several strategies for finding the global minimum are possible. The approach we adopt in this article consists of starting the minimization algorithm from a point which is close enough to the global minimum of the cost function. Once in the proximity of the global minimum, we can perform a global nonlinear minimization by means of a downhill simplex algorithm [40].

As often seen in the literature, a good approximation of the global minimum can be found by solving Eq. (A.2) with respect to a scaled version of $\boldsymbol{e}$, while ignoring the rigidity constraints that make of $\boldsymbol{E}$ an essential matrix. In order to do so, we can rewrite the homogeneous matrix Eq. (A.2) as

$$\boldsymbol{A}\bar{\boldsymbol{e}} = \boldsymbol{b}, \tag{A.4}$$

where $\boldsymbol{e}' = [e_{11}, e_{21}, e_{31}, \dots, e_{32}]^{\mathrm{T}}$, $\boldsymbol{b} = f^{\mathrm{b}} f^{\mathrm{a}} \boldsymbol{1}_n$, and $\boldsymbol{A}$ is obtained from $\boldsymbol{C}$ by eliminating its last column. This corresponds to searching for a scaled version of the actual essential matrix ($\boldsymbol{e}$ contains elements of $\boldsymbol{E}$ that are scaled in such a way to have $e_{33} = 1$). The fact that we can only estimate $\boldsymbol{E}$ up to a scale factor, justifies the fact that monocular motion estimation does not allow us to determine the magnitude of the translational component [35]. In fact, we know that $\boldsymbol{E}{=}(\boldsymbol{t}\times)\boldsymbol{R}$, where $\boldsymbol{R}$ is bound to be a rotation matrix, therefore it cannot be scaled. If we have $N = 8$ noncollinear points for each image, then $\boldsymbol{A}$ is invertible and we have $\bar{\boldsymbol{e}} = \boldsymbol{A}^{-1}\boldsymbol{b}$. If, on the other hand, the number of available corresponding pairs is $N > 8$, then the system of equations is overdetermined, and we have to determine the least-squares solution $\bar{\boldsymbol{e}} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$.

Due to the limited resolution and the point localization error, the matrix $\hat{\boldsymbol{E}}$ that best complies with the constraints $\boldsymbol{A}\bar{\boldsymbol{e}} = \boldsymbol{b}$, cannot generally be written as a product of a skew-symmetric matrix and a rotation matrix, therefore it cannot exactly describe a rigid motion. We thus need to determine the essential matrix that 'lies the closest' to a given matrix $\hat{\boldsymbol{E}}$. This 'projection' onto the set of the essential matrices can be performed by exploiting the fact that an essential matrix $\boldsymbol{E}$ is characterized by its having singular values λ_1, λ_2 and λ_3 that satisfy the constraints $\lambda_1 = \lambda_2$ and $\lambda_3 = 0$.

Let $\hat{\boldsymbol{E}} = \boldsymbol{U}\,\mathbf{diag}(\lambda_1,\lambda_2,\lambda_3)\boldsymbol{V}^{\mathrm{T}}$ be the singular value decomposition of $\hat{\boldsymbol{E}}$. The essential matrix $\boldsymbol{E}$ that lies the closest to $\hat{\boldsymbol{E}}$ (in Frobenius norm) can be readily computed [11,12] as $\boldsymbol{E} = \boldsymbol{U}\,\mathbf{diag}(1,1,0)\boldsymbol{V}^{\mathrm{T}}$, which can be readily decomposed into the product $\boldsymbol{E} = (\boldsymbol{t}\times)\,\boldsymbol{R}$, through the relationships

$$(\bar{\boldsymbol{t}}\times) = \boldsymbol{U}\,\mathbf{diag}(1,1,0)\boldsymbol{R}_z\boldsymbol{U}^{\mathrm{T}}, \quad \boldsymbol{R} = \boldsymbol{U}\boldsymbol{R}_z^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}},$$

where

$$\boldsymbol{R}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Of course, while $\boldsymbol{R}$ represents the estimate of the rotation undergone by the camera between the two acquisitions, $\bar{\boldsymbol{t}}$ represents only a scaled version of the direction of translation's estimate. The normalized translation vector can also be written in explicit form as

$$\bar{\boldsymbol{t}} = \begin{bmatrix} U_{21}U_{32} - U_{22}U_{31} \\ U_{12}U_{31} - U_{11}U_{32} \\ U_{11}U_{22} - U_{21}U_{12} \end{bmatrix}.$$

We can easily verify that det($\boldsymbol{U}$) $= \boldsymbol{u}_3^{\mathrm{T}}\bar{\boldsymbol{t}}$, $\boldsymbol{u}_3$ being the third column of $\boldsymbol{U}$. Since both $\boldsymbol{u}_3^{\mathrm{T}}$ and $\bar{\boldsymbol{t}}$ have unit norm, like their scalar product det($\boldsymbol{U}$), they are bound to have the same direction. We can thus conclude that direction is given by the third column of $\boldsymbol{U}$.

From the rotation matrix $\boldsymbol{R}$ we now compute the Euler angles $\boldsymbol{\alpha}$, while from the normalized translation vector $\bar{\boldsymbol{t}}$ we can compute the spherical coordinates $\boldsymbol{\beta}$ of the direction of translation. Such five parameters describe (up to a scale factor) the rigid motion that lies the closest (in Frobenius sense) to the linear estimate $\hat{\boldsymbol{E}}$ of the essential matrix, and provide a good starting point for the global nonlinear minimization algorithm.

The global nonlinear minimization of the cost function is performed in a space of 5 unknowns. Since each homologous pair of points corresponds to an equation, we need $n \geqslant 5$ pairs of homologous points. However, choosing less than 8 matched pairs of points would not allow us to find a good initial point through linear minimization. In conclusion, in order for the estimation algorithm to be reliable, at least 8 homologous pairs of points are required.

Appendix B. Some notes on rigid motions

Let $\boldsymbol{p}^{(1)} \in \Re^3$ and $\boldsymbol{p}^{(2)} \in \Re^3$ be the camera coordinates of a point before and after the camera undergoes a rigid motion, respectively. The transformation that describes this rigid motion is described by $\boldsymbol{p}^{(2)} = \boldsymbol{R}\boldsymbol{p}^{(1)} + \boldsymbol{t}$, where $\boldsymbol{R} \in \mathscr{R}^{3\times 3}$ and $\boldsymbol{t} \in \mathscr{R}^3$ are the rotation matrix and the translation vector, respectively. This description of the rigid motion is based on 12 parameters, which are bound to satisfy the constraints $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \boldsymbol{I}_{3\times 3}$ and $\det(\boldsymbol{R}) = 1$. In alternative, a rigid motion can be described by six unconstrained parameters by using the rotation vector $\upsilon = [\upsilon_1, \upsilon_2, \upsilon_3]^{\mathrm{T}}$, whose orientation specifies the direction of the axis of rotation and whose magnitude $\|\upsilon\|$ represents the angle of rotation in radians. A rotation of $\|\upsilon\|$ radians about the υ axis maps the point $\boldsymbol{p}^{(1)}$ onto the point $\boldsymbol{p}^{(2)} = \boldsymbol{R}\boldsymbol{p}^{(1)}$, where $\boldsymbol{R}$ is obtained through matrix exponentiation

$$\boldsymbol{R} = e^{(\upsilon\times)} = \boldsymbol{I} + (\upsilon\times) + \frac{(\upsilon\times)^2}{2!} + \frac{(\upsilon\times)^3}{3!} + \cdots \tag{B.1}$$

$\boldsymbol{I}$ being the identity matrix and $(\upsilon\times)$ being a skew-symmetric 3×3 matrix associated to the vector υ.

The fact that the exponential of a skew-symmetric matrix is a rotation matrix can be easily checked. Less straightforward is the fact that *any* rotation matrix can be written as the exponential of some skew-symmetric matrix. This last result is known as Euler's theorem, which states that any rigid rotation can be thought of as a rotation about a fixed axis. In order to efficiently compute the exponential of a skew-symmetric matrix, we can use the Rodrigues formula [39],

$$\boldsymbol{R} = \boldsymbol{I} + \frac{(\upsilon\times)}{\|\upsilon\|}\sin\|\upsilon\| + \frac{(\upsilon\times)^2}{\|\upsilon\|^2}(1 - \cos\|\upsilon\|). \tag{B.2}$$

En efficient formula for computing (the principal value of) the logarithm of a rotation matrix exists as well and is known as Cayley's formula [38]

$$V = \log \boldsymbol{R} = (\boldsymbol{R} - \boldsymbol{I})(\boldsymbol{R} + \boldsymbol{I})^{-1}. \tag{B.3}$$

The exponential coordinates are said to be the canonical representation of a rotation. Such a representation, however, is not the only one possible. In alternative, we can think of a generic rotation $\boldsymbol{R}$, as composed of an ordered sequence of three elementary rotations about the three axes $\upsilon_1 = [1,0,0]^{\mathrm{T}}$, $\upsilon_2 = [0,1,0]^{\mathrm{T}}$ and $\upsilon_3 = [0,0,1]^{\mathrm{T}}$ of the reference frame. An elementary rotation of α_1 radians about υ_1 is given by the rotation matrix $\boldsymbol{R}_1(\alpha_1) = e^{\alpha_1 \upsilon_1 \times}$. Similarly, the elementary rotations about the other two axes are expressed as $\boldsymbol{R}_2(\alpha_2) = e^{\alpha_2 \upsilon_2 \times}$ and $\boldsymbol{R}_3(\alpha_3) = e^{\alpha_3 \upsilon_3 \times}$. Any rotation $\boldsymbol{R}$ can thus be easily obtained as an ordered sequence of the above three elementary rotations

$$\boldsymbol{R}(\boldsymbol{\alpha}) = e^{\alpha_1(\upsilon_1\times)}e^{\alpha_2(\upsilon_2\times)}e^{\alpha_3(\upsilon_3\times)}, \tag{B.4}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]^{\mathrm{T}}$ is the vector of the Euler angles. A matrix of the above form is indeed a rotation matrix, as it is a product of rotation matrices. Notice, however, that

$$e^{\alpha_1(\upsilon_1\times)}e^{\alpha_2(\upsilon_2\times)}e^{\alpha_3(\upsilon_3\times)} \neq e^{\alpha_1(\upsilon_1\times) + \alpha_2(\upsilon_2\times) + \alpha_3(\upsilon_3\times)}.$$

The Euler angles can be expressed as a function of the components of $\boldsymbol{R}$ as follows:

$$\begin{aligned} \cos\alpha_3 &= R_{11}/k & \sin\alpha_3 &= R_{12}/k, \\ \cos\alpha_2 &= k & \sin\alpha_2 &= -R_{13}, \\ \cos\alpha_1 &= R_{33}/k & \sin\alpha_1 &= R_{23}/k, \end{aligned}$$

where R_{ij} be the (i,j) element of $\boldsymbol{R}$, and $k = \sqrt{R_{11}^2 + R_{12}^2} = \cos\alpha_2$, which is assumed to be nonzero.

References

[1] F. Pedersini, A. Sarti, S. Tubaro, Accurate and low-cost calibration and self-calibration of multi-camera acquisition systems, EURASIP Signal Process. 77 (3) (1999).

[2] F. Pedersini, A. Sarti, S. Tubaro, Multicamera systems: calibration and applications, IEEE Signal Process. Mag. (special issue on Stereo and 3D Imaging) (June 1999).

[3] F. Pedersini, A. Sarti, S. Tubaro, Accurate feature detection and matching for the tracking of calibration parameters in multi-camera acquisition systems, IEEE International Conference on Image Processing, ICIP-98, Chicago, Illinois, USA, Vol. 2, October 4–7, 1998, pp. 598–602.

[4] N. Ayache, Artificial Vision for Mobile Robots, MIT Press, Cambridge, MA, 1990.

[5] N. Ayache, F. Lustman, Trinocular stereovision for robotics, IEEE Trans. Pattern Anal Mach. Intell. 13 (1) (January 1991) 73–85.

[6] Y. Otha, T. Kanade, Stereo by intra- and inter-scanline search using dynamic programming, IEEE Trans. Pattern Anal. Mach. Intell. 7 (2) (1985) 139–154.

[7] P. Pigazzini, F. Pedersini, A. Sarti, S. Tubaro, 3D area matching with arbitrary multiview geometry, EURASIP Signal Process: Image Comm. (special issue on 3D Video Technol.) 14 (1-2) (1998) 71–94.

[8] T. Kanade, P. Rander, P.J. Narayanan, Virtualized reality: constructing virtual worlds from real scenes, IEEE Multimedia 4 (1) (January–March 1997) 34–47.

[9] Sing Bing Kang, J.A. Webb, C.L. Zitnick, T. Kanade, A multibaseline stereo system with active illumination and real-time image acquisition, IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995, pp. 88–93.

[10] T.S. Huang, O.D. Faugeras, Some properties of the E matrix in two-view motion estimation, IEEE Trans PAMI 2(12) (December 1989).

[11] C. Braccini, G. Gambardella, A. Grattarola, Processing 2-D views for 3-D shape and motion reconstruction, Adv. Image Process. and Pattern Recog. 1986.

[12] C. Braccini, G. Gambardella, A. Grattarola, S. Zappatore, Motion estimation of rigid bodies: effects of the rigidity constraints, Signal Process. III: Theor. Appl. (1986).

[13] Z. Zhang, A new multistage approach to motion and structure estimation: from essential parameters to euclidean motion via fundamental matrix, INRIA, Technical Report No. 2910, June 1996.

[14] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography, Comm. Ass. Comp. Mach. 24 (1981) 381–395.

[15] J.L. Mundy, A. Zisserman, Projective geometry for machine vision, in: J.L. Mundy, A. Zisserman (Eds.), Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA, 1992.

[16] Q.-T. Luong, T. Vieville, Canonical representations for the geometries of multiple projective views, Comput. Vision and Image Understanding 64 (2) (September 1996) 193–229.

[17] O. Faugeras, L. Robert, What can two images tell us about a third one?, Internat. J. Comput. Vision 18 (1996) 5–19.

[18] M. Born, E. Wolf, Principles of Optics, Pergamon Press, Oxford, 1959.

[19] J. Weng, P. Cohen, M. Herniou, Camera calibration with distortion model and accuracy evaluation, IEEE Trans. PAMI 14 (10) (October 1992) 965–980.

[20] L. Levi, Applied Optics - A Guide to Optical System Design, Vol. I, Wiley, New York, 1968.

[21] R. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE J. Robot. Automat. 3 (4) (August 1987) 323–344.

[22] G.Q. Wei, S. De Ma, Implicit and explicit camera calibration: theory and experiments, IEEE Trans PAMI 16 (5) (May 1994) 469–480.

[23] 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, INRIA, Report No. RR2273, 1994.

[24] H.A. Beyer, Geometric and radiometric analysis of a CCD-camera based photogrammetric close-range system, Ph.D. Thesis, No. 51, Institut für Geodäsie und Photogrammetrie, ETH, Zürich, May 1992.

[25] R. Deriche, G. Giraudon, A computational approach for corner and vertex detection, Internat. J. Comput. Vision 10 (2) (1993) 101–124.

[26] E. DeMicheli, B. Caprile, P. Ottonello, V. Torre, Localization and noise in edge detection, IEEE Trans. PAMI 11 (October 1989) 1106–1117.

[27] J. Canny, A computational approach to edge detection, IEEE Trans. PAMI 8 (6) (November 1996) 679–698.

[28] D. Barbe, Imaging devices using the charge-coupled concept, Proc. IEEE 63 (1) January 1975.

[29] O. Faugeras, Stratification of three dimensional vision: projective, affine and metric representations, J. Opt. Soc. Amer. 12 (3) 465–484.

[30] L. Van Gool, A. Zisserman, Automatic 3D model building from video sequences, Eur. Trans. Telecomm. 8 (4) (July–August 1997) 369–378.

[31] P.A. Beardsley, P.H.S. Torr, A. Zisserman, 3D model acquisition from extended image sequences, Proceedings of the fourth European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 1065, Cambridge, 1996, pp. 683–695.

[32] C. Fermuller, Y. Aloimonos, Qualitative egomotion, Internat. J. Comput. Vision 15 (1) (July 1995).

[33] T.H. Wu, R. Chellappa, Q. Zheng, Experiments on estimating egomotion and structure parameters using long monocular image sequences, Internat. J. Comput. Vision 15(3) (July 1995).

[34] Q. Zheng, R. Chellappa, Automatic feature point extraction and tracking in image sequences for arbitrary camera motion, Internat. J. Comput. Vision 15 (2) (July 1995).

[35] H. Longuet-Higgins, A computer algorithm for reconstruction of a scene from two proiection, Nature (1981) 133–135.

[36] F. Pedersini, A. Sarti, S. Tubaro, 3D motion estimation of a trinocular system for a full-3D object reconstruction, IEEE International Conference on Image Processing, Lausanne, Switzerland, September 1996.

[37] Q.T. Luong, O. Faugeras, The Fundamental matrix: theory, algorithms, and stability analysis, Internat. J. Comput. Vision 17 (1) (1996) 43–76.

[38] J.M. McCarthy, Introduction to Theoretical Kinematics, MIT Press, Cambridge, MA, 1990.

[39] R. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotics Manipulation, CRC Press, Boca Raton, 1994.

[40] J. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.

[41] F. Pedersini, A. Sarti, S. Tubaro, A multi-view trinocular system for automatic 3D object modeling and rendering. XVIII International Congress for Photogrammetry and Remote Sensing, ISPRS-96, Vienna, Austria, 1996.

[42] F. Pedersini, A. Sarti, S. Tubaro, Egomotion estimation of a multicamera system through line correspondence, IEEE International Conference on Image Processing, ICIP-97, Santa Barbara, CA, USA, October 26–29, 1997.