Computer Vision and Image Understanding **79,** 66–91 (2000)
doi:10.1006/cviu.2000.0848, available online at http://www.idealibrary.com on IDEAL®

Underwater Video Mosaics as Visual Navigation Maps

Nuno Gracias and José Santos-Victor[1]

Instituto Superior Técnico and Instituto de Sistemas e Robótica, Av. Rovisco Pais, Torre Norte 7.26, 1049-001 Lisbon, Portugal
E-mail: ngracias@isr.ist.utl.pt, jasv@isr.ist.utl.pt

Received March 19, 1999; accepted February 4, 2000

This paper presents a set of algorithms for the creation of underwater mosaics and illustrates their use as visual maps for underwater vehicle navigation. First, we describe the automatic creation of video mosaics, which deals with the problem of image motion estimation in a robust and automatic way. The motion estimation is based on a initial matching of corresponding areas over pairs of images, followed by the use of a robust matching technique, which can cope with a high percentage of incorrect matches. Several motion models, established under the projective geometry framework, allow for the creation of high quality mosaics where no assumptions are made about the camera motion. Several tests were run on underwater image sequences, testifying to the good performance of the implemented matching and registration methods. Next, we deal with the issue of determining the 3D position and orientation of a vehicle from new views of a previously created mosaic. The problem of pose estimation is tackled, using the available information on the camera intrinsic parameters. This information ranges from the full knowledge to the case where they are estimated using a self-calibration technique based on the analysis of an image sequence captured under pure rotation. The performance of the 3D positioning algorithms is evaluated using images for which accurate ground truth is available.

Key Words: video mosaics; underwater visual navigation; pose estimation.

1. INTRODUCTION

In the past few years, computer vision has increasingly been used as a sensing modality for underwater vehicles used in tasks where accurate measures at short range are needed

[1]The work described in this paper has been supported by the Portuguese Foundation for Science and Technology PRAXIS XXI BD/13772/97 and Esprit-LTR Proj. 30185, NARVAL.

1077-3142/00 $35.00

[28]. Previous applications include seabed reconstruction, pipeline inspection [4], and object tracking [21].

A considerable amount of research interest has been directed toward providing autonomy to underwater vehicles using vision, namely in self-location and motion estimation. Automatic station-keeping, involving motion estimation in order to maintain a fixed position, has been implemented for remotely operated vehicles with two [20] and three [21] degrees of freedom. A method for 3D motion estimation and mosaic construction was proposed by Xu *et al.* [31] and tested on a floating platform. Based on underwater image formation models using surface irradiance and light attenuation, Yu *et al.* [32] proposed estimation methods for motion recovery and surface orientation.

Ocean floor exploration constitutes an important application area for video mosaicking, in such operations as site exploration, wreckage visualization, and navigation. Due to the underwater limited visual range, the registration of close range images is often the only solution for obtaining large visual areas of the floor. This limitation has motivated research on automatic mosaic creation for underwater applications over the past few years. In [18] a setup was proposed for creating mosaics by taking images at locations whose coordinates are known with high precision. Image merging can thus be performed without image analysis, because the frame-to-frame motion parameters can be computed directly from the camera positions. Marks *et al.* [22] have developed a system for ocean floor mosaic creation in real time. In their work, a four-parameter semirigid motion model is used, and small rotation and zooming on the image frames are assumed. This allows for fast processing algorithms, but restricts the scope of applications to the case of images taken by a camera whose retinal plane is closely parallel to the ocean floor. A common difficulty in underwater mosaicking arises from the presence of 3D occlusions caused by seabed irregularities. Strategies for dealing with such occlusions are discussed by Tiwari in [29]. Another difficulty comes from the propagation of image alignment errors which as been addressed by Fleischer *et al.* in [10] and [11].

The use of mosaics as tools to provide visual maps for navigation has been explored by Zheng *et al.* [34], in the context of land robotics and route recognition. In their work, a visual memory of the motion of a mobile robot is created in the form of panoramic mosaics that are later used for robot positioning. However, the visual representations are used solely for navigational purposes and the panoramic views created do not correspond to geometrically and visually correct mosaics.

The work described in this paper addresses the issues of mosaic creation and vehicle self-location using the mosaics as visual maps. The approach for automatic creation of video mosaics is based on image motion estimation in a robust and automatic way. The motion estimation starts from an initial matching of corresponding areas over pairs of images, followed by the use of a robust matching selection technique, which can cope with an high percentage of wrong matches. Several motion models, established under the projective geometry framework, are then used to allow the creation of high quality mosaics where no assumptions are made on the camera motion. This is an improvement over traditional approaches for underwater mosaicking, often relying on the camera to be facing the seafloor, so that the image plane is approximately parallel to the floor plane.

Next, we deal with the issue of determining the 3D position and orientation of a vehicle from a new view of a previously created mosaic. The problem of pose estimation is tackled, using the available information on the camera intrinsic parameters. This information ranges from full knowledge to the case where they are estimated using a self-calibration technique

based on the analysis of an image sequence captured under pure rotation. The presented techniques are suitable for autonomous underwater vehicle navigation near a flat oceanic floor, where a planar map is an accurate representation of the environment.

A possible application scenario for the methods described in this paper would be on underwater archeological site exploration. An autonomous underwater vehicle (AUV) equipped with a camera is assigned to map the area of a newly discovered wreckage. As an initial goal, the vehicle will thoroughly cover the area and build a general view of the site. Later, the vehicle might be instructed to explore a specific region of interest. By using other types of sensor modality, such as long-baseline acoustic ranging, which provide a rough estimate of its position, it will be able to coarsely locate itself with respect to the previously constructed mosaic. Then, using the onboard camera, it registers the current image with the mosaic to obtain a finer estimate of its position and make its way to the desired location. Another application scenario might be in marine geological surveys, for studying the evolution of oceanic geological activity by checking bubble sources using periodic inspections by an AUV.

The paper is organized as follows. Section 2 describes some of the geometric foundations and models required for the methods used later. These include the used camera model, the recovery of the intrinsic parameter matrix from the projection matrix, planar transformations, and self-calibration from a rotating camera. Mosaic creation methods are presented in Section 3, with mosaicking results from underwater video footage. Section 4 is devoted to the registration of new views on a previously constructed mosaic and to the problem of estimating the camera pose. Trajectory recovery results are presented for an image sequence for which ground truth is available. Finally, Section 5 summarizes and draws some conclusions on the performance and applicability of the methods.

2. GEOMETRIC BACKGROUND

2.1. Camera Model

The camera model used in this work is the standard pinhole model, under which the camera performs a linear projective mapping from the projective space $\mathbb{P}^3$ to the projective plane $\mathbb{P}^2$. For a 3D point with coordinates (x, y, z) and its corresponding 2D projection (u, v), the camera mapping that can be expressed, in the general form, as

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} fk_u & fk_\theta & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} {}^C_W R & {}^C_W \mathbf{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{1}$$

where f is the focal distance, k_u and k_v are scaling factors, (u_0, v_0) is the location of the principal point in the new image referential, and λ is an unknown scale factor. The parameter k_θ accounts for the skew between the image axes and, for most CCD cameras, can be considered to be zero on applications not relying on highly accurate calibration.

The (3×3) matrix K is the intrinsic parameter matrix. The extrinsic parameters are represented by the (3×3) rotation matrix ${}^C_W R$ and by the (3×1) vector ${}^C_W \mathbf{t}$, containing the coordinates of the origin of the world frame expressed in camera frame coordinates.

Combining both intrinsic and extrinsic parameters, the overall camera mapping can be represented by a (3×4) perspective projection matrix $P = K[{}^{C}_{W}R \;\; {}^{C}_{W}\mathbf{t}]$.

The task of camera calibration can be accomplished, for this model, by a simple linear least-squares minimization. If we assume that a set of 3D points is available with their image projections, then each point will impose two linear constraints on the elements of P. For a set of n points, a homogeneous system can be created in the form $H \cdot \mathbf{p}_l = 0$, where H is a $(2n \times 12)$-data matrix containing the coordinates of the 3D points and projections, and $\mathbf{p}_l$ is a column vector containing all the 12 elements of P.

If six or more 3D points are on a general configuration, and their projections are known with sufficiently high accuracy, then H will have exactly rank 11. By a general configuration we mean that no four of the points are coplanar, nor do they all lie on a twisted cubic as described by Faugeras in [7], although this latter situation is very unlikely to occur in practice. The vector $\mathbf{p}_l$ is the null space of H, thus defined up to scale. To avoid the trivial solution $\mathbf{p}_l = 0$, one has to impose an additional constraint on P, usually $\|\mathbf{p}_l\| = 1$. Furthermore, real applications are prone to inaccuracies on the measurements of point locations, and H will not be rank deficient. In order to find a least-squares solution for this equation, we can formulate the classical minimization problem

$$\min_{\mathbf{p}_l} \|H \cdot \mathbf{p}_l\| \text{ constrained to } \|\mathbf{p}_l\| = 1. \tag{2}$$

By the use of the Lagrange multipliers it can be easily shown that the solution to this problem is the eigenvector associated with the smallest singular value of H. A suitable algorithm for finding the eigenvector is singular value decomposition (SVD) [26].

2.1.1. Estimation of the K matrix using the camera projection matrix. The intrinsic parameter matrix K can be obtained from P by means of the QR factorization [13, 19]. For the case of a 3×3 matrix A, there is a unitary 3×3 matrix Q and a upper triangular 3×3 matrix R, such that $A = QR$. For a nonsingular A, the R matrix can be chosen to have all positive diagonal entries. It can be shown that, in this case, the factorization is unique [19]. An algorithm for the QR factorization is described in [26], and a number of implementations exist in commonly used mathematical packages.

For the case of K matrix estimation, we are interested in the dual of the QR factorization, in the form $A = R'Q'$, where R' and Q' have the same structure as that of R and Q. The K matrix can be recovered from the RQ factorization of the first three columns of P. The RQ factorization for 3×3 nonsingular matrices can be computed from the QR counterpart, thus allowing the use of existing implementations of the QR algorithm.

Let $A^T E = QR$ be the QR factorization of $A^T E$, where A is a 3×3 nonsingular matrix and

$$E = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

By means of algebraic manipulation, it can be seen that, $A = R'Q'$, where $R' = ER^T E$ is upper triangular and $Q' = EQ^T$ is unitary.

2.2. Planar Transformations

As we are interested in registering scenes with planar content, we will now focus on 2D projective transformations whose importance is emphasized by the fact that they can be

used as models for image motion with an enormously vast field of application in computer vision. It can be easily shown [14, 24] that two different views of the same planar scene in 3D space are related by a collineation in $I\!P^2$, represented by a (3×3) matrix defined up to scale and establishing a one-to-one relation between corresponding points over two images. Thus, for a pair of image points of the same 3-D point of a planar scene with homogeneous coordinates $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{u}}'$, the collineation T_{2D} relating $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{u}}'_i$ will impose $\tilde{\mathbf{u}}' \doteq T_{2D}\tilde{\mathbf{u}}$, where the symbol $\doteq$ denotes equality up to scale. A collineation in $I\!P^2$ is also commonly referred to as a planar transformation.

The computation of a planar transformation requires at least four pairs of corresponding points. If we have more than four correspondences, a least-squares minimization can then be accomplished in a manner similar to the one outlined above for the camera calibration. Let T_{2D} be the collineation relating two image planes from which we have a set of n correspondences such that $\tilde{\mathbf{u}}'_i \doteq T_{2D}\tilde{\mathbf{u}}_i$, for $i = 1, \ldots, n$. For each pair we will have two linear constraints on the elements of T_{2D}. A homogeneous system of equations $H \cdot \mathbf{t}_l = 0$ can thus be written, where $\mathbf{t}_l$ is the column vector containing the elements of T_{2D} rowwise, and H is a $(2n \times 9)$ matrix. The system can now be solved by means of SVD, after the additional constraint of unit norm for $\mathbf{t}_l$, i.e., $\|\mathbf{t}_l\| = 1$, is imposed.

As it is defined up to scale, the most general collineation in $I\!P^2$ has eight independent parameters. If additional information is available on the camera setup, such as camera motion constraints, then the coordinate transformation $\tilde{\mathbf{u}}'_i \doteq T_{2D}\tilde{\mathbf{u}}_i$ might not need the eight independent parameters of the general case to accurately describe the image motion. As an example we can point out the case where the camera is just panning, thus inducing a simple sideways image translation. If we know beforehand which is the simplest model that can explain the data equally well, then there will be no reason for using the most general. Table 1 illustrates some of the commonly used restricted models.

2.3. Self-Calibration from a Rotating Camera

An alternative method can be devised for estimation of the K matrix, for the case where a sequence of images is available, taken by a camera with constant intrinsic parameters, and undergoing pure rotation. This method does not require any knowledge of the scene

TABLE 1
Some of the Possible Motion Models Used for Image Merging, Ordered by the Number of Free Parameters *p*

Image model	Matrix form	p	Domain
Translation and zoom	$T_{2D} = \begin{bmatrix} t_1 & 0 & t_2 \\ 0 & t_1 & t_3 \\ 0 & 0 & t_4 \end{bmatrix}$	3	Image plane is parallel to the planar scene. No rotation but with variable focal length or distance to the scene.
"Semirigid"	$T_{2D} = \begin{bmatrix} t_1 & t_2 & t_3 \\ -t_2 & t_1 & t_4 \\ 0 & 0 & t_5 \end{bmatrix}$	4	Same as above but with rotation and scaling along the image axes.
Affine transformation	$T_{2D} = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & t_7 \end{bmatrix}$	6	Distant scene subtending a small field of view.

structure, nor the rotation of the camera frame between images. Therefore, it is specially suited to applications where the camera can be rotated around its optical center, and online calibration is required. An example of such is in underwater robotics, for vehicles equipped with pan and tilt camera heads.

Although the problem of camera calibration has been an active research topic in computer vision from its early days, only in the past few years has the issue of self-calibration been addressed in the literature. The theory behind this method was first presented by Hartley in [16], and more in-depth in [17], for the case of constant intrinsic parameters. Recent work by Agapito *et al.* [6] has extended the method to deal with cameras for which the intrinsic parameters are allowed to vary.

For the case of stationary cameras (where no translation is allowed), the world coordinate frame can usefully be chosen to match the first camera frame. Therefore the projection matrices for cameras may be written in the form $P_i = K[R_i \;\; \mathbf{0}]$, where $R_1 = I_3$ is the (3×3) identity matrix. The projection of a 3D world point $\tilde{\mathbf{M}} = [x \;\; y \;\; z \;\; 1]^T$ results in an image point $\tilde{\mathbf{m}} \doteq K[R_i \;\; \mathbf{0}][x \;\; y \;\; z \;\; 1]^T$, which is independent of the last element of $\tilde{\mathbf{M}}$. By dropping the last element of $\tilde{\mathbf{M}}$, the projection equations can written in the form $\tilde{\mathbf{m}} \doteq K R_i [x \;\; y \;\; z]^T$, where the reduced projection matrix $\bar{P}_i = K R_i$ performs a 2D projective mapping, similar to the planar transformations described above. Considering the case where the same 3D point $\mathbf{M} = [x \;\; y \;\; z]^T$ is projected on two different images using the projection matrices $\bar{P}_i = K R_i$ and $\bar{P}_j = K R_j$, we will have $\tilde{\mathbf{m}}_i \doteq K R_i \mathbf{M}$ and $\tilde{\mathbf{m}}_j \doteq K R_j \mathbf{M}$, from which the following relation between $\tilde{\mathbf{m}}_i$ and $\tilde{\mathbf{m}}_j$ can be written

$$\tilde{\mathbf{m}}_i \doteq K R_i R_j^{-1} K^{-1} \tilde{\mathbf{m}}_j = K R_{j,i} K^{-1} \tilde{\mathbf{m}}_j .$$

This equation represents a 2D homography $T_{j,i} = K R_{j,i} K^{-1}$ that maps corresponding points in two views, taken by a rotating camera. It can be computed directly from image measurements and depends only on the intrinsic parameter matrix and on the camera rotation $R_{j,i}$ between the two images. As noted in [17], $T_{j,i}$ is only meaningfully defined up to scale, but taking into account the fact that the product $K R_{j,i} K^{-1}$ has a unit determinant, the exact equality $T_{j,i} = K R_{ji} K^{-1}$ will hold if $T_{j,i}$ is scaled by an appropriate factor.

The problem remains on how to recover K and a set of rotation matrices from homographies computed from image correspondences. By using the rotation matrix property $R_{j,i} = R_{j,i}^{-T}$ and rewriting $T_{j,i} = K R_{j,i} K^{-1}$ as $R_{j,i} = K^{-1} T_{j,i} K$, the following equations can be written

$$\begin{aligned} K^{-1} T_{j,i} K &= K^T T_{j,i}^{-T} K^{-T} \\ T_{j,i} K K^T &= K K^T T_{j,i} . \end{aligned} \tag{3}$$

A linear system of equations can thus be constructed on the elements of the symmetrical matrix $C = K K^T$. For a set of $n > 1$ homographies, C can be estimated linearly by writing the system of equations in the form $H \cdot \mathbf{c}_l = 0$, where H is a $(9n \times 6)$ matrix and $\mathbf{c}_l$ is a column vector containing the independent entries of C. This is the same type of minimization problem as the one in Section 2.2 and can be solved using the SVD.

The recovery of K can be achieved if C is positive-definite, by means of the Choleski decomposition [13], and is unique if K is assumed to have positive-diagonal entries. For noise-free data, C is positive-definite by construction, and for noisy data it might not be

so. However, as reported in [17], the cases where C was found to be not positive-definite happened only for gross errors in the matching process.

The knowledge of some of the intrinsic parameters can easily be incorporated into this self-calibration scheme with great improvement on the accuracy of the overall estimation. Namely, for the most widely used CCD video cameras and image acquisition boards, the skew between the image axes is quite often negligible. On the other hand, a rough estimate of the principal point is the central image point location, or it can be more precisely determined from radial distortion analysis [30].

For the case of zero skew and known location of the principal point, the least-squares method can still be used with straightforward modifications. Let $K = UA$ be a decomposition for a zero skew intrinsic parameter matrix, so that

$$U = \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} fk_u & 0 & 0 \\ 0 & fk_v & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

From the equation $T_{j,i} = K R_{j,i} K^{-1}$ it follows that $T_{j,i} = UAR_{j,i}A^{-1}U^{-1}$ and $U^{-1}T_{j,i}U = AR_{j,i}A^{-1}$. By conducting the same algebraic manipulation as in (3), a similar system of equations can be constructed for the elements of the diagonal matrix $D = AA^T$. The above considerations hold for the minimization method, with the H matrix being of size $(9n \times 3)$. The recovery of A is possible if the elements of D are positive, which is true for noise-free data but might not be in the presence of large matching errors.

3. MOSAIC CREATION

We will now deal with the problem of creating mosaics from a sequence of video images. The creation of video mosaics is accomplished in two stages: registration and rendering. During the registration stage, we estimate the parameters of point correspondence between frames, then fit individual frames to a global model of the sequence. The rendering stage deals with the creation of a single mosaic, by applying a temporal operator over the registered and aligned images.

3.1. Feature Selection

The work presented here evolves from the analysis of point projections and their correspondence between image frames. In order to improve the correspondence finding, a number of points are selected corresponding to image corners or highly textured patches. The selection of image points is based on a simplified version of the well-known corner detector proposed by Harris [15]. This detector finds corners in step edges by using only first-order image derivative approximations. Further details on the implemented detector are presented in [14].

The extracted features will be matched over two images and used for motion estimation. Since motion estimation is more noise-sensitive to location errors when the features are close to each other, it is convenient to select features not just on the "amount of texture," but also using some interfeature distance criterion. Bearing this in mind, the implemented algorithm selects the features by finding the peaks of the "texture" image and excluding the

subsequent selection on a circular neighborhood. This process is repeated iteratively up to the point where no peaks above a defined threshold can be found.

3.2. Matching

The first step toward the estimation of the image registration parameters consists of finding point correspondences between images. This is referred to as the matching problem, which is considered a challenging task due to its difficulty. Contributing factors to this difficulty include the lack of image texture, object occlusion, and acquisition noise, which are frequent in real imaging applications. Several matching algorithms have been proposed over the past two decades, usually based on correlation techniques or dynamic programming. For a comparative analysis of stereo matching algorithms dealing with pair of images, refer to [2].

In this work, a correlation-based matching procedure was implemented. It takes a list of features selected from the first image I_1 and tries to find the best match for each, over a second image I_2. The cost criterion that drives the search on the second image is known in the literature as the sum of squared differences (SSD) [1]. For a given feature $\mathbf{f}_i = (u_i, v_i)$, it is defined as

$$SSD(x, y) = \sum_{(u,v)\in W_i} [I_1(u, v) - I_2(u - x, v - y)]^2,$$

where W_i is an image patch around $\mathbf{f}_i$.

The assumption of large overlap of image contents between the two frames can be used to significantly reduce the computational burden of the matching. This is achieved by limiting the search area in I_2. In order to compute the appropriate limits, the two images are cross-correlated and a global displacement vector $\mathbf{d}_G$ is obtained. By applying a threshold to the cross-correlation image, we can estimate a bounding box around $\mathbf{d}_G$ that can be loosely interpreted as a confidence area for the global displacement. Then, for a given feature $\mathbf{f}_i$ the search area on I_2 is constrained to the rectangular area by the size of the bounding box and centered on $\mathbf{f}_i + \mathbf{d}_G$. Figure 1 illustrates the procedure.

3.3. Robust Motion Parameter Estimation

In this section we will describe a procedure for the estimation of the motion parameters for a sequence of images. The images are processed as shown in the diagram of Fig. 2. For each image I_k, a set of features is extracted and matched directly on the following image I_{k+1}, as described in the previous sections. The result of the matching process is two lists of

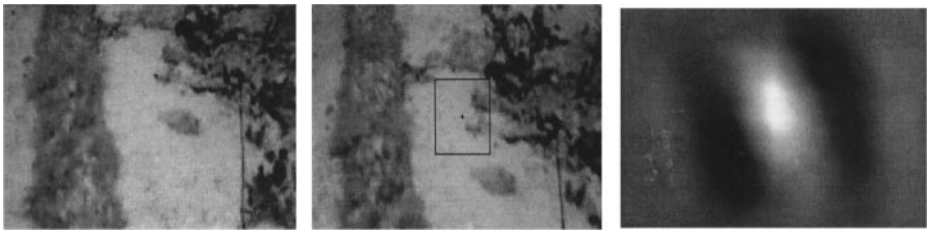

FIG. 1. Search area selection: Image I_1 (left) with selected feature, search area on I_2 (center), and cross-correlation image (right).

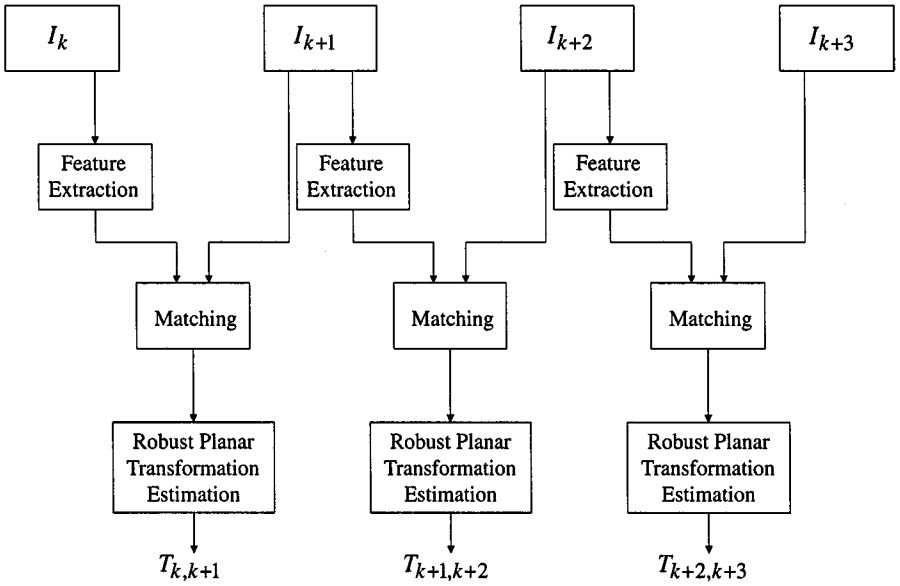

FIG. 2. Block diagram of the sequence of operations on the images I_k for the motion parameter estimation. The output is the set of planar transformation matrices $T_{k,k+1}$.

coordinates of corresponding points. Due to the error prone nature of the matching process, it is likely that a number of point correspondences will not relate to the same 3D point. For this reason, this subsection is devoted to the robust estimation of the motion parameters taking into account the existence of mismatches.

The first step in image registration is to find the motion parameters for the image motion, between consecutive frames. In this work, no automatic selection for the motion model is performed. The most appropriate model is assumed to be known. In the following sections, the most general planar transformation model (performing a collineation between planes) will be considered.

Let ${}^{(k)}\mathbf{u}$ be a point on frame k, and let ${}^{(k+1)}\mathbf{u}$ be its correspondence on frame $k+1$. If $T_{k,k+1}$ is the planar transformation matrix relating the frames k and $k+1$, then the point coordinates relate by ${}^{(k)}\tilde{\mathbf{u}} \doteq T_{k,k+1}\,{}^{(k+1)}\tilde{\mathbf{u}}$. A robust estimation method is required for the estimation of $T_{k,k+1}$. For this, a random sampling algorithm was used, using the following minimization criterion. Let ${}^{(k)}\mathbf{u}_i$ be the location of the ith feature extracted from image I_k, and matched with ${}^{(k+1)}\mathbf{u}$ on image I_{k+1}. The criterion to be minimized is the median of the sum of the square distances,

$$\underset{i}{\text{med}} \left(d^2\left({}^{(k)}\mathbf{u}_i,\, T_{k,k+1}\quad {}^{(k+1)}\mathbf{u}_i\right) + d^2\left({}^{(k+1)}\mathbf{u}_i,\, T^{-1}_{k,k+1}\quad {}^{(k)}\mathbf{u}\right)\right), \tag{4}$$

where $d(\cdot,\cdot)$ stands for the image point-to-point Euclidean distance.

The random sampling algorithm is a simple two-step variant of least-median-of-squares (LMedS), referred to as MEDSERE. It exhibits a similar breakdown point [23], but requires less random sampling in order to achieve the same degree of outlier rejection.

The MEDSERE algorithm comprises two phases of random sampling LMedS. After the first phase, the data set is reduced by selecting the best data points in the sense of the chosen cost function. Next, the reduced data undergoes another random sampling LMedS phase. For the computation of a homography the algorithm is illustrated by the following

operations:

1. Randomly sample the complete set of matched points S_{total} for a set of p pairs.
2. Estimate the $T_{k,k+1}$ matrix and compute the median of the sum of the point distance squares, for S_{total},

$$\underset{i}{\text{med}} \left(d^2 \left({}^{(k)}\mathbf{u}_i, T_{k,k+1} \quad {}^{(k+1)}\mathbf{u}_i \right) + d^2 \left({}^{(k+1)}\mathbf{u}_i, T_{k,k+1}^{-1} \quad {}^{(k)}\mathbf{u} \right) \right),$$

where $d(\cdot, \cdot)$ is the orthogonal distance. If the median is below a given threshold m_T, return $T_{k,k+1}$ and exit.

3. Repeat 1 and 2 for a specified number of samples m_1.
4. Select the $T_{k,k+1}$ matrix for which the minimal median was found, and sort the matched points by their sum of the point distance squares, using $T_{k,k+1}$.
5. Create the set S_{best} with the elements of S_{total} whose distances are below the median.
6. Repeat 1 and 2 on S_{best} for a m_2 number of samples.
7. Select the minimal median matrix found.
8. For this matrix select the matched points whose average distance,

$$\frac{1}{2} \left(d \left({}^{(k)}\mathbf{u}_i, T_{k,k+1} \quad {}^{(k+1)}\mathbf{u}_i \right) + d \left({}^{(k+1)}\mathbf{u}_i, T_{k,k+1}^{-1} \quad {}^{(k)}\mathbf{u} \right) \right), \tag{5}$$

is less than or equal to a specified distance threshold d_T.

9. Compute and return the final $T_{k,k+1}$ using simple least-squares with all the selected matched points above.

The required parameters are the number of samplings on each part m_1 and m_2, the median threshold, and the distance threshold. Since the first two directly determine the number of operations, they can be defined by processing time constraints.

As it was emphasized before, the use of a robust matching selection procedure is essential for the accurate estimation of the image motion parameters. However, accurate results similar to the ones presented in this paper could be obtained by other widely used random sampling algorithms, such as standard LMedS and RANSAC [9]. In the context of computer vision, an in-depth comparison of LMedS and RANSAC can be found in [23], while results on the use of the MEDSERE algorithm for mosaicking and fundamental matrix estimation are presented in [14].

3.4. Global Registration

After the frame-to-frame motion parameters are estimated, these parameters are cascaded to form a global model. The global model takes the form of a global registration, where all frames are mapped into a common, arbitrarily chosen, reference frame. Let $T_{\text{Ref},1}$ be the transformation matrix relating the chosen reference frame and the first image frame. The global registration is defined by the set of transformation matrices $\{T_{\text{Ref},k} : k = 1 \ldots N\}$, where for $2 \leq k \leq N$,

$$T_{\text{Ref},k} = T_{\text{Ref},1} \prod_{i=1}^{k-1} T_{i,i+1}.$$

In this work, the chosen reference is computed using some points with known metric coordinates. Let us assume that a set of four or more points have known positions on a 2D world coordinate frame, and that those points have identifiable projections on the first image of the

sequence. The homography $T_{\text{World},1}$ relating the 2D world frame and the first image frame can, therefore, be computed using the linear method outlined in Section 2.2. If the mosaic reference frame is chosen to be coincident with the world frame, i.e., $T_{\text{Ref},1} = T_{\text{World},1}$, then the coordinates on any image frame are straightforwardly related to the world coordinate system.

3.5. Rendering

After global registration, the following step consists of merging the images. On ovelapping regions there are more multiple contributions for a single point on the output image, and some method has to be established in order to determine the unique intensity value that will be used. The contributions for the same output point can be thought of as lying on a line which is parallel to the time axis, in a space–time continuum of the globally aligned images. Therefore, the referred method operates on the time domain, thus called a *temporal operator*. Some of the commonly used methods are the use-first, use-last, mean, and median. The first two use only a single value from the contributions vector, respectively the first and the last entries of the timely ordered vector. The mean operator takes the average over all the point contributions and is effective in removing temporal noise inherent in video. Finally, the median operator also removes temporal noise but is particularly effective in removing transient data, such as fast moving objects whose intensity patterns are stationary for less than half the frames. It is therefore adequate for underwater sequences of the seabed, where moving fish or algae are captured.

3.6. Mosaicking Results

The ocean floor mosaics presented were created from a number of video sequences where no information, other than the images themselves and the most suitable motion model, was used.

An example of a seabed mosaic is given in Fig. 3. It was composed of 101 frames, registered under the semirigid model and rendered with the median operator. The original sequence was obtained by a manually controlled underwater vehicle and depicts a human-made construction. No information was provided about the camera motion, which is composed of translation and rotation and zoom out. The captured scene is not planar nor

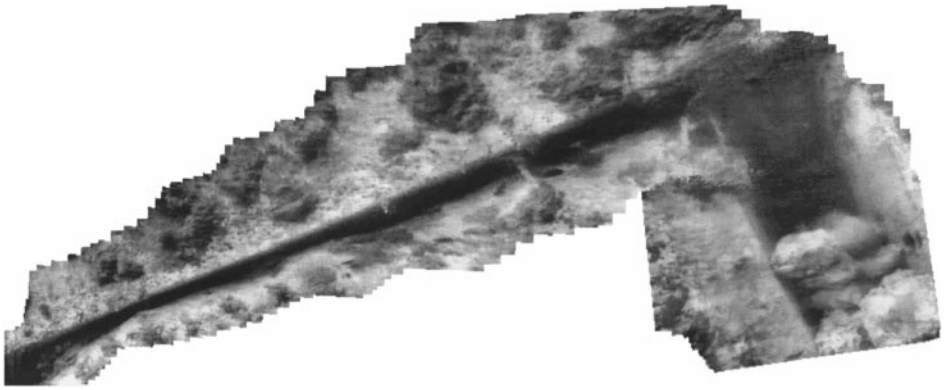

FIG. 3. Seabed mosaic example. The images were registered using the semirigid motion model and rendered using the median operator.

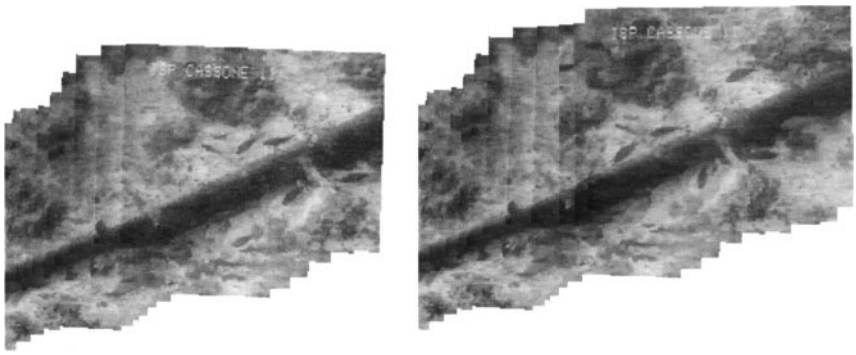

FIG. 4. Example of mosaic creation where the static scene assumption is violated by the presence of moving fish.

static. The camera is moving along a seabed fracture with some rocks inside. In the fracture area, there are noticeable depth variations as opposed to the almost planar surrounding seabed. Even so, the seabed is mostly covered with algae and weeds, which provide good features for the matching process but violate the underlying planar scene assumption. Another assumption violation is caused by some moving fish. Figure 4 shows two submosaics in which the motion of the fish can be clearly noticed. Although constructed from the same sequence, these submosaics were rendered using the use-last temporal operator.

Figure 5 presents two views of a mosaic from a sequence of images captured by a surface-driven ROV, on a pipe inspection task. In this example the perspective distortion effects are

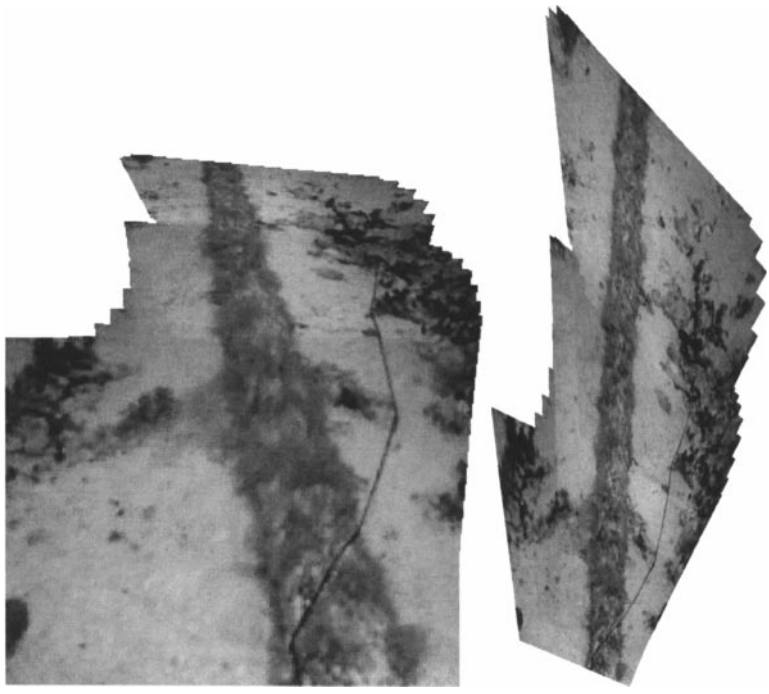

FIG. 5. Underwater pipe mosaic example. For the image registration, the full planar transformation model was used. The images were registered with the use-last operator (left). A useful reference frame can be chosen in order to have a better perception of the seafloor (right).

noticeable, since the image plane of the camera is distinctly not parallel to the seafloor. The most suitable motion model is, therefore, the full planar transformation. The left image was created using the first frame of the sequence as the reference frame. For the right image, a reference frame was chosen as to make the contour lines of the pipeline approximately parallel, yielding a top view of the floor.

4. POSE ESTIMATION

For an underwater navigation application, we are interested in using video mosaicking as a tool to provide visual navigation maps. Having such maps referenced to a world coordinate system will enable a camera-equipped autonomous vehicle (in an unknown position and orientation) to locate itself once it has found the correct mapping from the mosaic to the image frame. Some methods will be presented in this section for the pose estimation. Trajectory recovery is illustrated based on the assumption of constant intrinsic parameters.

Over the years, the problem of camera pose estimation has been thoroughly addressed in the computer vision literature. It is a central problem in photogrammetry applications where it is also referred to as the space resection problem. Due to its widespread use in aerial photography and cartography, there have been many methods developed for both minimum data requirements and redundant data. A review of photogrammetry methods and of a proof of uniqueness of the solution for the case of coplanar points is presented by Yuan [33]. For recent progress in linear methods in pose estimation the reader is referred to [27] and [3]. The pose estimation algorithm presented in this section decomposes an image-to-mosaic homography matrix in order to find the rotation matrix and displacement vector relating the camera frame to a world frame (extrinsic parameters). In this sense, it relates to the work of Ganapathy [12], where the extrinsic parameters are recovered directly from a camera projection matrix.

The use of image homographies induced by a plane in the scene has been explored by Faugeras and Lustman [8], for robot navigation tasks. They have shown how the homographies could be used to directly recover the camera rotation and translation up to scale, assuming the camera has been intrinsically calibrated beforehand. The main difference between their approach and the work presented here lies in the fact that we use the same world plane for inducing all the interimage homographies. This allows the trajectory reconstruction to be based on the analysis of the image-to-world homographies rather than on the interimage homographies. The main advantage is that small errors on the homography estimation do not tend to accumulate, as will be illustrated later in this section.

4.1. Algorithm for New View Registration on the Mosaic

We will now deal with the problem of finding the homographies relating a sequence of images to a previously constructed mosaic. In this, we will explicitly take advantage of the timely order nature of the image sequence to reduce the computational burden of finding correspondences on the mosaic. By assuming that adjacent image frames correspond to a small camera translation and rotation, we have a large image overlap. Thus, the feature matching search can be restricted to a neighboring area of the location predicted by the last image-to-mosaic homography.

One of the main difficulties with this procedure is due to the fact that feature matching using correlation produces poor results in the presence of significant image warping between

the image features and the mosaic. However, this condition can be greatly alleviated by using an estimate of the image-to-mosaic homography, $T_{M,i}$, and the performing feature warping before correlation.

The implemented algorithm requires an estimate $\hat{T}_{M,1}$ of the first image-to-mosaic homography $T_{M,1}$, which need not be too accurate. The procedure can be described by the following steps:

1. Select features from current image i and perform feature warping using $\hat{T}_{M,i}$.
2. Match each feature to the mosaic, over a neighboring area around the position predicted by $\hat{T}_{M,i}$. Use robust matching selection to compute $T_{M,i}$.
3. If the number of matched pairs used for computing $T_{M,i}$ is greater than (or equal to) a given acceptable minimum number n_m, then go to 7. Otherwise, if i is the first image, then stop with error condition. If not, continue to 4.
4. Compute the $T_{i-1,i}$ by matching the unwarped feature with the previous image. If the number of matched pairs used for $T_{i-1,i}$ is below n_m, then stop with error condition.
5. Update $\hat{T}_{M,i}$ by taking $\hat{T}_{M,i} = \hat{T}_{M,i} \cdot T_{i-1,i}$. Repeat 4 and recompute $T_{M,i}$.
6. If the number of matched pairs used for $T_{M,i}$ is less than n_m, then assume $T_{M,i}$ to be the composition of the last correctly computed image–mosaic homography with the image to previous image homography; i.e., $T_{M,i} = T_{M,i-1} \cdot T_{i-1,i}$.
7. If i is not the last image, then use $T_{M,i}$ as an estimate for the computation of the next image-to-mosaic homography, i.e., $\hat{T}_{M,i+1} = T_{M,i}$. Select the next image and go to the beginning.

For each image, the algorithm tries to find a reliable image-to-mosaic homography. Reliability is insured by a specified minimum number of correct matches. If it fails to find it, the algorithm uses the homography with the previous image to compute it. The advantage of registering each frame directly on the mosaic (as opposed to computing the registration by sequentially cascading the homographies $T_{i-1,i}$ between previous images) is due to the fact that small estimation errors on $T_{i-1,i}$ are not accumulated. This condition is apparent in the results in Section 4.4.3.

Once the image–mosaic homographies have been computed and the mosaic is referenced to a world frame, the camera pose can be estimated with respect to the world frame. Four methods have been implemented for this, which require different amounts of information about the intrinsic parameters of the camera. They are:

- Completely known intrinsic parameter matrix.
- Known principal point and skewing.
- Unknown intrinsic parameter matrix, but estimated using self-calibration from rotating scenes.
- Unknown intrinsic parameter matrix, estimated using self-calibration from rotating scenes with additional knowledge on the principal point and skew.

The last two methods are extentions of the first. Under these two, the self-calibration scheme described in Section 2.3 is initially used to estimate the K matrix. Afterward, the first method is used to recover the trajectory. In practical setups, this implies an additional maneuver using a pan-and-tilt head mounted on the vehicle, or the possibility of the vehicle rotating, maintaining the camera optical center approximately at the same position.

4.2. Known Intrinsic Parameter Matrix

Throughout the rest of this section we will consider a 3D world frame to be such that all the points in the planar scene have a null $\vec{z}$ coordinate.

Let us now assume that the intrinsic parameter matrix K has been estimated beforehand. Let $\tilde{\mathbf{u}} \doteq P\tilde{\mathbf{X}}$ be the projection of a 3D point $\tilde{\mathbf{X}}$ of a planar scene, where $\doteq$ denotes equality up to a scale factor. From Eq. (1), the projection matrix P can be expressed as $P \doteq K[I_3 \;\; \mathbf{o}]\,{}^{C}_{W}G$, where I_3 is the 3×3 identity matrix, $\mathbf{o}$ is the 3×1 null vector, and ${}^{C}_{W}G$ represents the 3D rigid transformation

$$ {}^{C}_{W}G = \begin{bmatrix} {}^{C}_{W}R & {}^{C}_{W}\mathbf{t} \\ 0\,0\,0 & 1 \end{bmatrix}. $$

The planar points $\tilde{\mathbf{x}}$ of the 2D world frame relate with the 3D frame, by

$$ \tilde{\mathbf{X}} \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}} = J\tilde{\mathbf{x}}. \tag{6} $$

The projection of a planar world point is therefore $\tilde{\mathbf{u}} \doteq PJ\tilde{\mathbf{x}}$. It can be seen that the matrix product PJ implements the homography between the scene plane and the image plane; i.e., $PJ \doteq T_{\text{image,World}}$.

Considering the components of ${}^{C}_{W}G$,

$$ {}^{C}_{W}G = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

one can write

$$ \begin{aligned} T_{\text{image,World}} &\doteq PJ \\ T_{\text{image,World}} &\doteq K[I_3 \quad \mathbf{o}]\,{}^{C}_{W}GJ \\ T_{\text{image,World}} &\doteq KL, \end{aligned} \tag{7} $$

where

$$ L = \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}. $$

The L matrix comprises the first two columns of the camera-to-world rotation matrix ${}^{C}_{W}R$, and the position of the 3D world referential in the 3D camera frame, ${}^{C}_{W}\mathbf{t}$. It can be computed up to scale, from $L \doteq K^{-1}T_{\text{image,World}}$, since $T_{\text{image,World}}$ is also only defined up to scale.

In order to recover the ${}^{C}_{W}R$ matrix, one has first to estimate the unknown scale factor λ. This factor affects the matrix L, therefore scaling the first two columns of ${}^{C}_{W}R$. Let $\mathbf{r}_1$ and $\mathbf{r}_2$ be the first two columns of L. It is easy to see that the absolute value of λ is given by the

norm of $\mathbf{r}_1$ (or $\mathbf{r}_2$), due to the fact that the columns of a rotation matrix have unit norm. In this work we have used the following formula for $\|\lambda\|$,

$$\|\lambda\| = \frac{\text{norm}\,\mathbf{r}_1 + \text{norm}\,\mathbf{r}_2}{2}.$$

A simple way of recovering the rotation matrix ${}^{C}_{W}R$ is, first, to scale $\mathbf{r}_1$ and $\mathbf{r}_2$ by the two symmetric solutions for λ and, secondly, to compute their cross-product. This cross-product is, in fact, the same for both solutions. Since $T_{\text{image,World}}$ is estimated from noisy image measurements, it is to expect $\mathbf{r}_1$ and $\mathbf{r}_2$ to be nonorthogonal. In this work, the orthogonality condition was enforced by computing the two orthogonal unit vectors that have the same bisectrix and stand on the same plane as $\mathbf{r}_1$ and $\mathbf{r}_2$.

Let ${}^{C}_{W}R_1$ and ${}^{C}_{W}R_2$ be the two candidates for ${}^{C}_{W}R$, corresponding respectively to the scaling by $+\|\lambda\|$ and $-\|\lambda\|$. The corresponding optical center locations are given by

$${}^{W}_{C}\mathbf{t}_1 = -\frac{1}{\lambda}\,{}^{C}_{W}R_1^T \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad \text{and} \quad {}^{W}_{C}\mathbf{t}_2 = \frac{1}{\lambda}\,{}^{C}_{W}R_2^T \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}.$$

It can easily be seen that ${}^{C}_{W}R_1$ and ${}^{C}_{W}R_2$ relate by

$${}^{C}_{W}R_1 = {}^{C}_{W}R_2 \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

which represents a rotation of 180° around the $\vec{z}$ axis. Therefore the locations of the optical centers differ by the last coordinate which is symmetric. Both solutions for ${}^{C}_{W}R$ and ${}^{W}_{C}\mathbf{t}$ are coherent with $T_{\text{image,World}}$, and therefore valid. In the application of this work, we are only interested in the positive $\vec{z}$ axis solution for ${}^{W}_{C}\mathbf{t}$, which corresponds to the camera being above the plane of the floor.

4.3. Known Principal Point and Skewing

An alternative method for estimating the camera pose can be devised if only the principal point location and the skewing ratio fk_θ / fk_v are known, instead of the full K matrix. Let us decompose K as the product of a upper triangular matrix U with ones on the diagonal by a diagonal matrix A, so that

$$K \doteq UA = \begin{bmatrix} 1 & \frac{fk_\theta}{fk_v} & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} fk_u & 0 & 0 \\ 0 & fk_v & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Since U is invertible, one can extend Eq. (7) to

$$U^{-1}T_{\text{image,World}} \doteq AL. \tag{8}$$

The left side of Eq. (8) can be computed from image measurements. As we are interested in estimating the unknown intrinsic parameters in the A matrix, we will start by explicitly including an unknown scale factor λ, in order to remove the equality up to scale. Let

$M = U^{-1} T_{\text{image,World}}$. Equation (8) can thus be written as $A(\lambda L) = M$. By considering the first two columns of this equality, where the unknown scale factor λ has been multiplied to the elements of L, the following system of equations can be written,

$$\begin{bmatrix} fk_u & 0 & 0 \\ 0 & fk_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda \cdot l_{11} & \lambda \cdot l_{12} \\ \lambda \cdot l_{21} & \lambda \cdot l_{22} \\ \lambda \cdot l_{31} & \lambda \cdot l_{32} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{bmatrix}.$$

By imposing the additional conditions of equal norm and vector orthogonality on $[\lambda \cdot l_{11} \ \lambda \cdot l_{21} \ \lambda \cdot l_{31}]^T$ and $[\lambda \cdot l_{12} \ \lambda \cdot l_{22} \ \lambda \cdot l_{32}]^T$, a system of equations on fk_u and fk_v can be written in the form of

$$\begin{bmatrix} m_{11} \cdot m_{12} & m_{21} \cdot m_{22} \\ m_{11}^2 - m_{12}^2 & m_{21}^2 - m_{22}^2 \end{bmatrix} \begin{bmatrix} \frac{1}{fk_u^2} \\ \frac{1}{fk_v^2} \end{bmatrix} = - \begin{bmatrix} m_{31} \cdot m_{32} \\ m_{31}^2 - m_{32}^2 \end{bmatrix}.$$

After estimating fk_u and fk_v the pose can be recovered using the method described in Section 4.2.

Although this method can be used independently for each frame of the image sequence, higher accuracy for the pose recovery can be achieved by using more than one frame in the estimation of fk_u and fk_v. For the experiments conducted in this paper, we have used an iterative least-squares method [5], which is suitable for the online processing of image sequences.

4.4. Pose Estimation Results

4.4.1. Original sequence. In order to evaluate the performance of the pose estimation algorithms, accurate ground truth is required. For this reason we have used the mosaic of Fig. 5 and synthesized new views according to a specified camera matrix and trajectory. These images are then used to retrieve the camera and position parameters. The mosaic was set to cover an area of 6 by 14.5 m. The sequence comprises 40 images of 320×240 pixels taken by a camera on a moving vehicle combining 3D motion and rotation. The camera is pointing downward with a tilt angle of approximately 150° with respect to the horizontal. The used intrinsic parameters matrix K accounts for a skewless camera with the following intrinsics,

$$K = \begin{bmatrix} 480 & 0 & 160 \\ 0 & 480 & 120 \\ 0 & 0 & 1 \end{bmatrix}.$$

In order to simulate the vehicle drift induced by water currents, perturbations have been added to the nominal forward motion of 0.23 m/frame and to the nominal height above seafloor of 3 m. The perturbations account for periodic drifts of around 0.4 m in position and 15° in orientation. The combined movement of the camera is depicted on Fig. 6, where the camera is represented with the optical axis, for each frame.

The resulting synthetic sequence is fairly realistic-looking. Frames 9 and 18 are presented in Fig. 7. Although it does not take into account the effects of nonuniform lighting and barrel distortion, usually present in underwater imagery, this sequence can still be a good approximation of real images taken in shallow waters under daylight illumination.

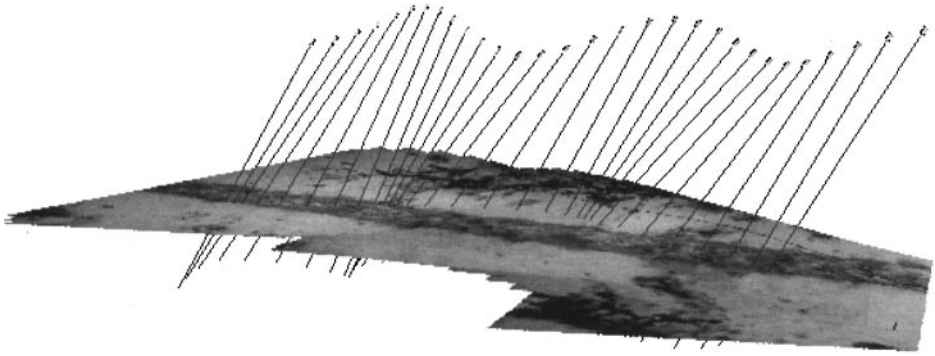

FIG. 6. 3D view of the camera positions and corresponding optical axes used for generating the sequence with available ground truth.

4.4.2. Camera self-calibration. In order to use the self-calibration method, an additional set of 20 images was produced, in which the camera undergoes pure rotation. The optical center remained fixed at 4 m above the sea bottom, while the camera faced down, and rotated around the 3 axes (pan, tilt, and yaw). For each axis, the angle range is $\pm 5^{\circ}$. The sequence of angular positions used for pan and tilt are plotted in Fig. 8, while a 3D view of the cameras optical axes over the mosaic is presented in Fig. 9. The intrinsic parameters matrix K used for this sequence was the same as the one used for the other sequence.

The estimation of the homographies between adjacent images constitutes the starting point for the self-calibration procedures. The homographies were computed using the algorithms described above for the mosaic creation. For the sequence of 20 images, 19 homographies were estimated using 6 to 70 matched pairs of points, within a 0.5 pixel distance threshold. Using the self-calibration methods for unknown intrinsics and known principal point and skew, the recovered matrices were, respectively,

$$K_{\text{rec}} = \begin{bmatrix} 499.8 & -10.1 & 159.2 \\ 0 & 482.9 & 82.8 \\ 0 & 0 & 1.00 \end{bmatrix} \quad \text{and} \quad K_{\text{rec}PP} = \begin{bmatrix} 484.38 & 0 & 160.00 \\ 0 & 487.33 & 120.00 \\ 0 & 0 & 1.00 \end{bmatrix}.$$

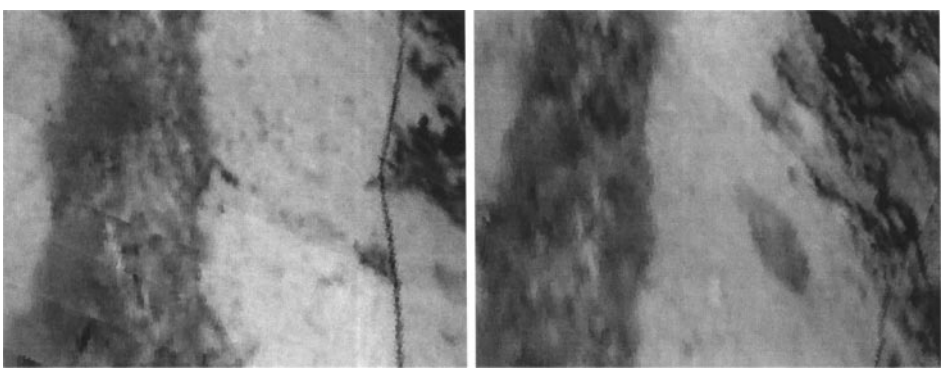

FIG. 7. Two frames of the generated image sequence used for positioning evaluation with ground truth.

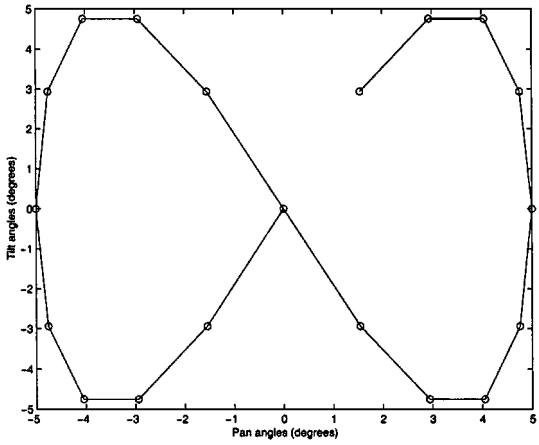

FIG. 8. Pan-and-tilt angles used for generating the sequence containing pure camera rotation.

4.4.3. Camera trajectory recovery. The performance of the pose estimation was experimentally evaluated by testing the camera path reconstruction using the available ground-truth data. The test results presented assume constant intrinsic parameters in time and differ on the amount of intrinsic parameter information used. With decreasing required calibration information, the conducted tests were the following:

Exp1—Trajectory recovery with known K matrix

Exp2—Trajectory recovery with known principal point and zero skew

Exp3—Trajectory recovery with self-calibration with known principal point and zero skew

Exp4—Trajectory recovery with self-calibration for all intrinsic parameters.

The synthetic images from the sequence containing camera translation were registered directly on the mosaic, using the algorithm described in Section 4.1. The algorithm was run with a specified acceptable minimum of 8 matched pairs per homography. In each frame it

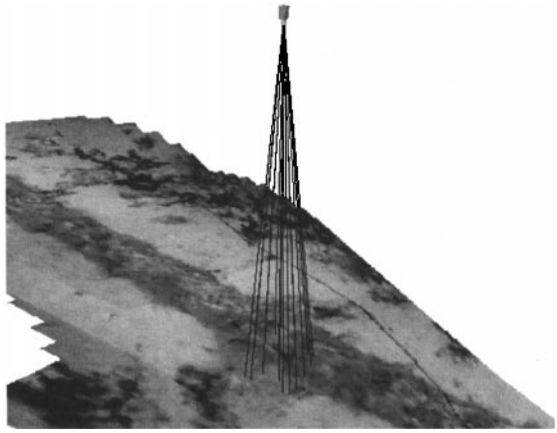

FIG. 9. 3D view of the camera positions and corresponding optical axes used for generating the sequence containing pure camera rotation.

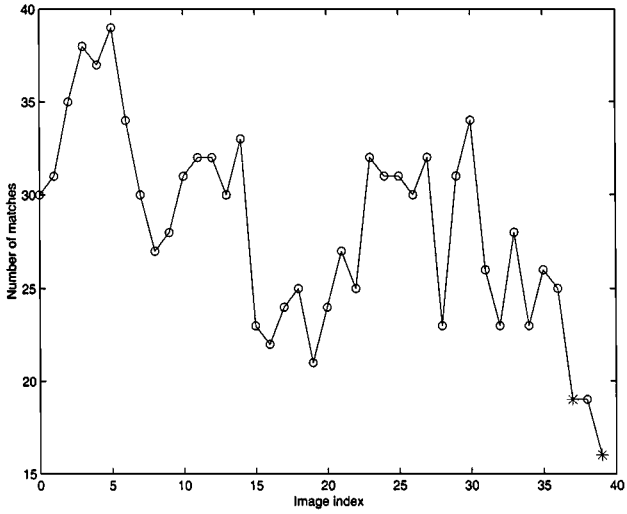

FIG. 10. Number of correctly matched pairs of points used for the final computation of the each of the image-to-mosaic homographies. The circular marks refer to successful matching on the first attempt, while the star-shaped ones refer to sucessful matching on the second attempt.

was able to find between 16 and 39 pairs, as displayed in Fig. 10. For the set of 40 images, 2 homographies were computed with matched pairs from a second attempt, while the other 38 were computed at the first attempt.

For the trajectory recovery with known principal point and zero skew (Exp2), an iterative least-squares estimation method was implemented for the estimation of the two unknown intrinsic parameters. When processing a sequence of images, this method uses the current frame-to-mosaic homography, with all the past homographies.

Statistics for the reconstruction errors are presented in Table 2. The position errors were measured by taking the Euclidean distance between the ground-truth position and the estimated position. As for the orientation, the error was measured by computing the angle between the true and the estimated camera frame orientations. For each image and method, the position errors are plotted in Fig. 11.

In these results, the lowest position and orientation errors correspond to the trajectory recovery with known K matrix (Exp1). This is not surprising, as this method uses the

TABLE 2

Trajectory Recovery Results for Known K Matrix, Known Principal Point, Self-Calibration with Known Principal Point, and Full Self-Calibration—Average, Maximum, and Standard Deviation of Position and Angular Errors

	Position errors (meters)			Angular errors (degrees)		
Method	Avg.	Maximum	Std.dev.	Avg.	Maximum	Std.dev.
Known K (Exp1)	0.031	0.159	0.031	0.610	2.932	0.525
Known PP (Exp2)	0.045	0.159	0.030	0.636	2.978	0.509
Self-calib. with PP (Exp3)	0.061	0.163	0.025	0.690	2.675	0.463
Full self-calib. (Exp4)	0.258	0.366	0.036	1.678	2.754	0.421

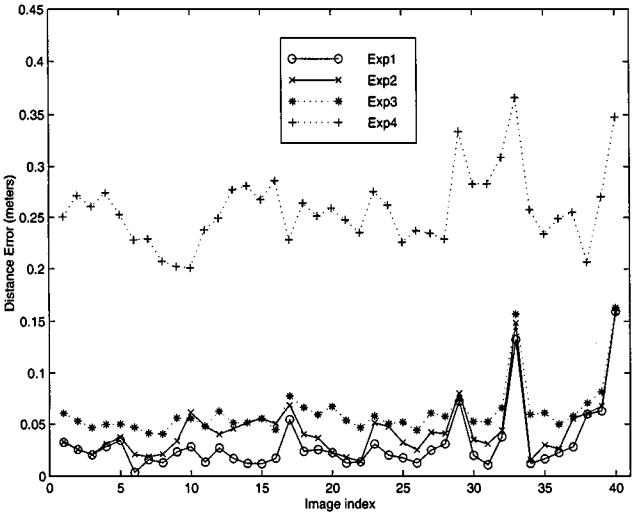

FIG. 11. Trajectory position errors for experiments 1 to 4.

most prior information. For Exp2, a 50% increase on the average error is observed. With self-calibration for known principal point and zero skew (Exp3), the average position error is approximately two times larger than the one for Exp1. However the worst case error is 15 cm, which can be considered small when compared to the distance to the seafloor of 3 m (5%). The average error is 6.1 cm which accounts for slightly more than 2% of the distance to the seafloor. When using unconstrained self-calibration (Exp4), position errors of around 25 cm are observed.

An additional experiment was conducted in order to compare the following image registration schemes:

Exp1—Image-to-mosaic homographies computed by direct mosaic registration

Exp5—Image-to-mosaic homographies computed by cascading interimages homographies.

The first scheme refers to the use of the algorithm of Section 4.1 with the same setup as in (Exp1). In the second, the true camera position and orientation is used for computing the first image-to-mosaic homography $T_{M,1}$. The subsequent homographies are calculated by

$$T_{M,i} = T_{M,1} \cdot \prod_{k=2}^{i} T_{k-1,k} \quad i > 1,$$

where $T_{k-1,k}$ are the interimage homographies and the matrix product is computed by right-multiplying for each increment of the index k. The set of $T_{k-1,k}$ was estimated from the same sequence of images, and the number of used matched points varied from 10 to 76 pairs, as displayed in Fig. 12, with an average of 60.

Table 3 and Fig. 13 present, respectively, the observed position and orientation error statistics, and the plot of the position errors for each frame. It can be seen that the second scheme produces less accurate results, due to the fact that small errors, inherent to the interimage homography estimation, are accumulated. This phenomenon is in many ways comparable to the positioning errors arising from the use of dead-reckoning during navigation.

TABLE 3

Error Statistics for the Direct Mosaic Registration Scheme (Same as in Exp1) and for the Interimage Registration Scheme—The Values of the Average, Maximum and Standard Deviation Are Shown for the Position and Angular Errors

	Position errors (meters)			Angular errors (degrees)		
Method	Mean	Maximum	Std.dev.	Mean	Maximum	Std.dev.
Direct reg.	0.031	0.159	0.031	0.610	2.932	0.525
Interimage reg.	0.252	0.437	0.151	2.667	5.528	1.313

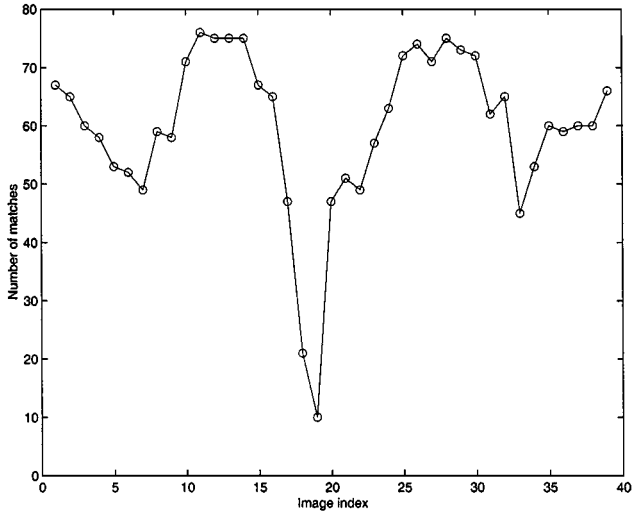

FIG. 12. Number of correctly matched pairs used for the final computation of the each homography between consecutive frames.

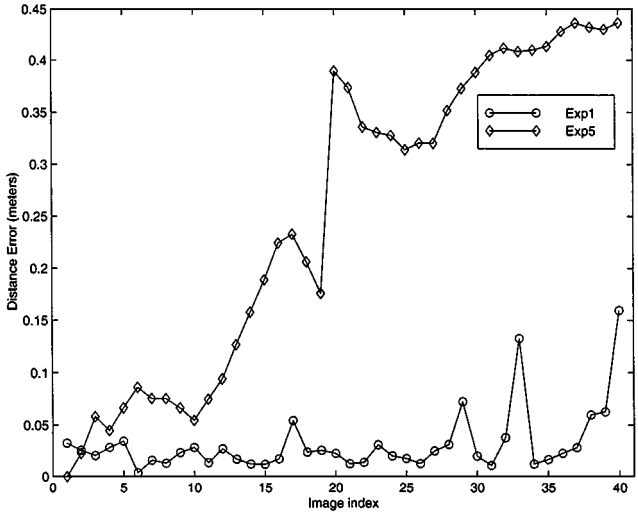

FIG. 13. Trajectory position errors for experiments 1 and 5. The error accumulation resulting from the interimage homography estimation is apparent for Exp5.

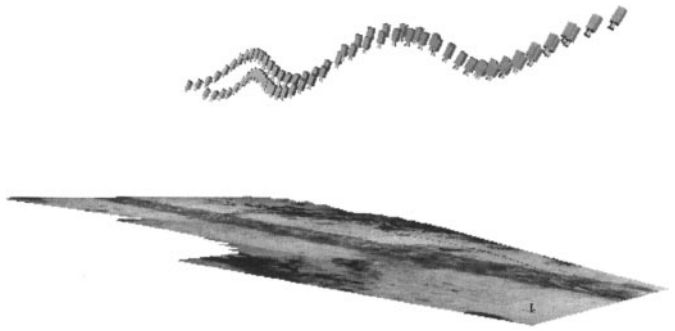

FIG. 14. 3D representation of the camera trajectories for the ground truth (upper set) and the result of Exp5 (lower set).

The drift induced by the error accumulation is apparent in Fig. 14, where a 3D view of the correct and estimated trajectories is given.

A final experiment was conducted for evaluating the quality of reconstruction using self-calibration when the principal point location is known or estimated beforehand with a different method. For the set of synthetic images described above, the self-calibration algorithm with defined principal point was used, with the location of the principal point varying from (0, 0) to (320, 240) in equally spaced intervals of 1 pixel. For each location, the average error on camera position and orientation was computed. Contour plots showing the results for average errors are given in Figs. 15 and 16.

The lowest errors correspond to the true principal point at (160, 120), at the center of the graphs. This case reverts to the third experiment (Exp3) described above, for which the average position and orientation errors were 0.061 m and 0.690°. The method used in

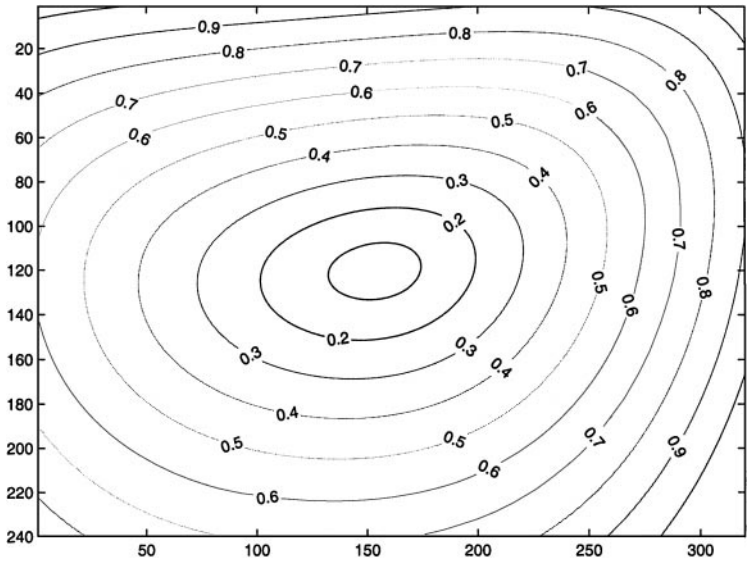

FIG. 15. Results for the trajectory recontruction procedure using self-calibration from rotating camera with known principal point. This contour plot contains the average position error (in meters) as a function of the assumed principal point location.

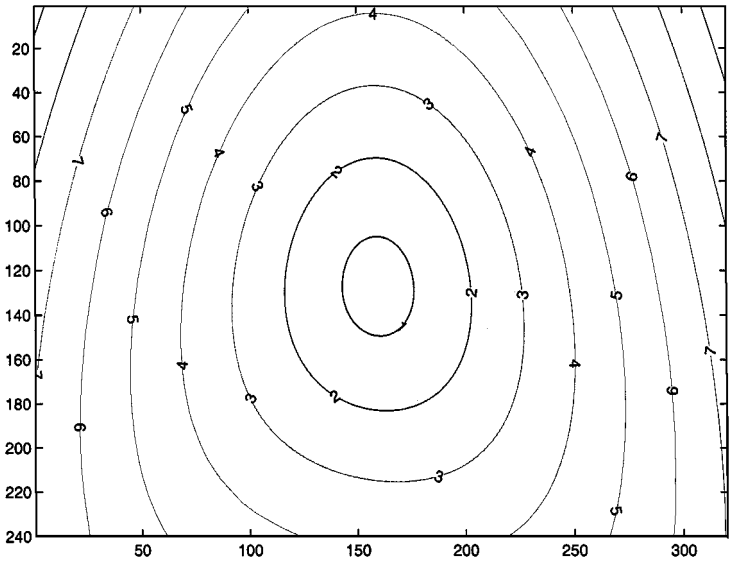

FIG. 16. Results for the trajectory recontruction procedure using self-calibration from rotating scamera with known principal point. This contour plot contains the average orientation error (in degrees) as a function of the assumed principal point location.

that experiment outperforms the unknown intrinsics self-calibration method (Exp4), for a considerably large area around the principal point.

From this, a conclusion with practical implications for the setups where self-calibration is feasible can be drawn. Even if the principal point location is not known precisely, it might be more advantageous to use a rough estimate and perform self-calibration for known principal points than to do it for unconstrained intrinsics. For the case presented here, better results on position error are achieved even if the assumed principal point location is within a 35-pixel neighborhood of the correct value.

5. CONCLUSIONS

We have presented an approach for the automatic creation of underwater video mosaics and illustrated their use as visual reference maps for subsequent vehicle localization. Key issues for the mosaicking process are the robust selection of correspondences and the use of geometric models capable of registering any view of a planar scene. Presented mosaics illustrated the good performance of the implemented matching and registration methods. Even with notorious violations of the assumed model, the algorithm is still able to find the image motion parameters and to create a mosaic with small misalignments to the human eye.

Methods for pose estimation were presented, which allow the estimation of the 3D position and orientation of a vehicle from a view of a previously created mosaic. The performance was evaluated using images corresponding to known camera motion that served as the ground truth.

An emphasis was put on using several degrees of available information on the camera intrinsic parameters, including self-calibration. The possibility of calibrating a camera

online can be of practical importance for a number of visually guided tasks, especially if the camera parameters are subject to change slowly in time. This paper illustrated how relevant information for the pose estimation process can be obtained by the analysis of rotation images. These images are easier to acquire than having to resort to calibration grids. Also, we have shown that the knowledge of the principal point can easily be incorporated in the self-calibration scheme with benefits on the accuracy. This is true even if its location is not precisely known.

By automatically creating visual representations of the seafloor and using them for navigation, the methods in this paper provide an important capability for the autonomous operation of submersibles.

REFERENCES

1. P. Anandan, A computational framework and an algorithm for the measurement of visual motion, *Internat. J. Comput. Vision* **2**(4), 1989, 283–310.
2. A. Arsénio and J. Marques, Performance analysis and characterization of matching algorithms. in *Proc. of the International Symposium on Intelligent Robotic Systems, Stockholm, Sweden, July 1997*.
3. J. Batista, H. Araújo, and A. Almeida, Iterative multi–step explicit camera calibration, in *Proc. of the 6th International Conference on Computer Vision, Bombay, India, January 1998*.
4. A. Branca, E. Stella, and A. Distante, Autonomous navigation of underwater vehicles, in *Proc. of the IEEE OCEANS'98, Nice, France, September 1998*.
5. C. Brown, *Tutorial on Filtering, Restoration and State Estimation*, Technical Report 534, Department of Computer Science, University of Rochester, June 1995.
6. L. de Agapito, E. Hayman, and I. Reid, Self-calibration of a rotating camera with varying intrinsic parameters, in *Proc. of the Ninth British Machine Vision Conference BMVC98, Southampton, UK, September 1998*.
7. O. Faugeras, *Three Dimensional Computer Vision*, MIT Press, Cambridge, MA, 1993.
8. O. Faugeras and F. Lustman, Motion and structure from motion in a piecewise planar environment, *Internat. J. Pattern Recog. Artif. Intell.* **2**(3), 1988, 485–508.
9. M. Fischler and R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* **6**(24), 1981, 381–395.
10. S. Fleischer, R. Marks, S. Rock, and M. Lee, Improved real-time video mosaicking of the ocean floor, in *Proc. of the IEEE OCEANS 95 Conference, California, October 1995*, pp. 1935–1944.
11. S. Fleischer and S. Rock, Experimental validation of a real-time vision sensor and navigation system for intelligent underwater vehicles, in *Proc. of the 1998 International Conference on Intelligent Vehicles, Stuttgart, Germany, October 1998*.
12. S. Ganapathy, Decomposition of transformation matrices for robot vision, in *Proc. 1st IEEE Conf. Robotics, IEEE, 1984*, pp. 130–139.
13. G. Golub and C. van Loan, *Matrix Computations*, John Hopkins Press, Baltimore, 1989.
14. N. Gracias, *Application of Robust Estimation to Computer Vision: Video Mosaics and 3-D Reconstruction*, Master's thesis, Lisbon, Portugal, April 1998, *available at* http://www.isr.ist.utl.pt/labs/vislab/thesis.
15. C. Harris, Determination of ego-motion from matched points, in *Proceedings, Alvey Conference, Cambridge, UK, 1987*.
16. R. Hartley, Self-calibration from multiple views with a rotating camera, in *Proc. of the 3rd. European Conference on Computer Vision, Stockholm, Sweden, May 1994*, Vol. I, pp. 471–478, Springer–Verlag, Berlin/New York.
17. R. Hartley, Self-calibration from stationary cameras, *Internat. J. Comput. Vision* **22**(1), 1997, 5–23.
18. R. Haywood, Acquisition of a micro scale photographic survey using an autonomous submersible, in *Proc. of the OCEANS 86 Conference, New York, 1986*.
19. R. Horn and C. Johnson, *Matrix Analysis*, Cambridge Univ. Press, Cambridge, UK, 1985.

20. K. Leabourne, S. Rock, S. Fleischer, and R. Burton, Station keeping of an ROV using vision technology, in *Proc. of the Oceans '97 Conference, Halifax, Canada, October 1997*, pp. 634–40.
21. R. Marks, S. Rock, and M. Lee, Using visual sensing for control of an underwater robotic vehicle, in *Proceedings of IARP Second Workshop on Mobile Robots for Subsea Environments, Monterey, California, May 1994*.
22. R. Marks, S. Rock, and M. Lee, Real-time video mosaicking of the ocean floor, *IEEE J. Ocean. Engg.* **20**(3), 1995, 229–241.
23. P. Meer, D. Mintz, A. Rosenfeld, and D. Kim, Robust regression methods for computer vision: A review, *Internat. J. Comput. Vision* **6**(1), 1991, 59–70.
24. R. Mohr and B. Triggs, Projective geometry for image analysis, tutorial given at International Symposium of Photogrammetry and Remote Sensing, Vienna, Austria, July 1996.
25. S. Negahdaripour, X. Xu, and A. Khamene, Applications of direct 3D motion estimation for underwater machine vision systems, in *Proc. of the IEEE OCEANS'98, Nice, France, September 1998*.
26. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge Univ. Press, Cambridge, UK, 1988.
27. L. Quan and Z. Lan, Linear $n \geq 4$-point pose determination, in *Proc. of the 6th International Conference on Computer Vision, Bombay, India, January 1998*.
28. J. Santos-Victor and J. Sentieiro, The role of vision for underwater vehicles, in *Proc. of the 1994 Symposium on Autonomous Underwater Vehicle Technology, Cambridge, MA, July 1994*, pp. 28–35.
29. S. Tiwari, Mosaicking of the ocean floor in the presence of three-dimensional occlusions in visual and side-scan sonar images, in *Proc. of the 1996 Symposium on Autonomous Underwater Vehicle Technology, Monterey, CA, June 1996*, pp. 308–314.
30. R. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses, *IEEE J. Robot. Automat.* **RA-3**(4), 1987, 323–344.
31. X. Xu and S. Negahdaripour, Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images, in *Proc. of the Oceans '97 Conference, Halifax, Canada, October 1997*, Vol. 2, pp. 1412–1417.
32. C. Yu and S. Negahdaripour, Underwater experiments for orientation and motion recovery from video images, in *Proceedings of 1993 IEEE International Conference on Robotics and Automation, Atlanta, GA, May 1993*, Vol. 2, pp. 93–98.
33. J. Yuan, A general photogrammetric method for determining object position and orientation, *IEEE Trans. Robot. Automat.* **5**(2), 1989, 14–23.
34. J. Zheng and S. Tsuji, Panoramic representation for route recognition by a mobile robot, *Internat. J. Comput. Vision* **9**(1), 1992, 55–76.