Uncalibrated Vision for 3-D Underwater Applications

K. Plakas, E. Trucco

Computer Vision Group and Ocean Systems Laboratory
Dept. of Computing and Electrical Engineering
Heriot-Watt University, UK
{costas,mtc}@cee.hw.ac.uk

A. Fusiello

Dip. di Matematica e Informatica
Università di Udine, ITALY
fusiello@dimi.uniud.it

Abstract— **This paper presents an uncalibrated vision system recovering 3-D reconstructions of underwater scenes using no camera calibration at all. The only information available is a set of point correspondences plus some a-priori knowledge on the scene structure. The system includes three main components: (1) a *robust tracker*, which targets optimal features for video tracking, and implements a completely automatic mechanism for rejecting unstable image features; (2) a module computing a *projective reconstruction* of the scene from the set of features declared reliable by the tracker; (3) a module recovering the target *Euclidean reconstruction* from the projective one, employing a-priori knowledge of the distances between five identifiable scene points. We sketch the design of each component and report experiments with our working implementation on real data sets, collected by cameras immersed in our laboratory tank.**

I. Introduction

Uncalibrated vision refers to the extraction of useful 3-D information about a scene from images acquired by an unspecified moving camera, for which the intrinsic and the extrinsic parameters are unknown. This is obviously an interesting issue for vision systems operating underwater, where calibration can be lost due to collisions or other unwanted events.

It is well known that when the intrinsic parameters of the moving camera are known it is possible to recover the Euclidean structure of the scene up to a similarity tranformation. When dealing with uncalibrated cameras, reconstruction is possible up to an unknown projective transformation, unless additional information on the scene – such as the reciprocal position of at least 5 points or 3 pairs of parallel lines – are known, thereby allowing to upgrade the reconstruction to Euclidean. This makes the method suitable for application with structured scenes like ducts, manifolds, or benthic stations.

A. Notation and Camera Model

Vectors and matrices are represented in boldface matrices in uppercase), while geometric entities such as points and lines in Times Roman. For vectors $\mathbf{u}$ and $\mathbf{v}$, their inner product is represented as $\mathbf{u}^T\mathbf{v}$.

The camera model considered is the well known *pinhole model.* The camera performs a perspective projection of an object point M onto a pixel m in the retinal plane through the optical center C. The optical axis is defined as the line going through the optical center C and perpendicular to the image plane.

II. The Feature Tracker

The tracker is based upon the previous work of Tomasi-Kanade [1] and Tomasi-Shi [2]. The Shi-Tomasi-Kanade tracker uses a SSD matching approach to feature tracking utilising an affine model for frame-to-frame dependencies. This system classified a tracked feature as reliable or unreliable according to the residual of the match between the associated image region in the first and subsequent frames; if the residual exceeded a user-defined threshold, the feature was rejected.

This work has been extended by introducing an *automatic* scheme for rejecting spurious features. In order to do that job, a simple, efficient, model-free outlier rejection rule, called X84, is employed [3].

A. Feature detection and Tracking

Let us consider an image sequence $I(\mathbf{x}, t)$, with $\mathbf{x} = [u, v]^T$, the coordinates of an image point. Provided that the sequence is sampled at a sufficiently high frequency, we can assume that small image regions are displaced without any noticeable change in their intensity:

$$I(\mathbf{x}, t) = I(\delta(\mathbf{x}), t + \tau), \qquad (1)$$

where $\delta(\dot{})$ is a *translational motion field* specifying the *warping* that is applied to image points between frames. This motion field can be written as: $\delta(\mathbf{x}) = \mathbf{x} + \mathbf{d}$, where $\mathbf{d}$ is a displacement vector.

The motion parameter $\mathbf{d}$, can now be estimated by minimizing the residual

$$\epsilon = \sum_{W} [I(\mathbf{x} + \mathbf{d}, t + \tau) - I(\mathbf{x}, t)]^2. \qquad (2)$$

Substituting the first-order Taylor expansion of $I(\mathbf{x}+\mathbf{d}, t+\tau)$ into (2), and ensuring that the derivatives with respect to $\mathbf{d}$ are zero, we obtain the linear system

$$\mathbf{G}\mathbf{d} = \mathbf{e}, \tag{3}$$

where

$$\mathbf{G} = \sum_{\mathcal{W}} \left[\begin{array}{cc} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{array} \right], \quad \mathbf{e} = -\tau \sum_{\mathcal{W}} I_t \left[I_u \; I_v \right]^T,$$

with $I_u = \partial I/\partial u$, $I_v = \partial I/\partial v$ and $I_t = \partial I/\partial t$. Equation (3) is solved for $\mathbf{d}$ using a Newton-Raphson iterative scheme. In this framework, a feature can be reliably tracked if a numerically stable solution to (3) can be found, which requires that $\mathbf{G}$ is well conditioned and its entries above the noise level. This poses the requirement that the smallest eigenvalue of $\mathbf{G}$ is sufficiently large, since, in practice, the largest eigenvalue is bounded by the maximum allowable pixel value. Therefore, if λ_1 and λ_2 are the eigenvalues of $\mathbf{G}$, a feature is accepted if $\min(\lambda_1, \lambda_2) > \lambda$, where λ is a user defined threshold.

B. Robust Outlier Rejection

To monitor the quality of the features, the tracker checks the residuals between the first and the current frame: high residuals indicate bad features which must be rejected. Following [2], we adopt the affine model, as a pure translational model would not work well with long sequences: too many good features are likely to undergo significant rotation, scaling or shearing, and would be incorrectly discarded. The *affine motion field* can be written as

$$\delta(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{d}, \tag{4}$$

where $\mathbf{d}$ is the displacement between frames, and $\mathbf{A}$ is a 2×2 matrix accounting for affine warping.

The residuals (computed according to the affine model over a $N \times N$ window W) when comparing good features can be seen as samples coming from a Gaussian distribution:

$$\epsilon \simeq \eta(N^2, 2N^2).$$

When the two regions over which the residual is computed are a bad match (i.e. they are not warped by an affine transformation), the residual is not a sample from the normal distribution of good features. Hence, our outlier rejection becomes a problem of estimating the mean and variance of the corrupted Gaussian distribution.

To increase robustness, X84, a simple but effective model-free rejection rule is employed. X84 achieves robustness by employing median and median deviation instead of the usual mean and standard deviation. This rule rejects features whose residuals are more than k *Median Absolute Deviations* (MADs) away from the median:

$$MAD = med_i\{|\epsilon_i - med_j \epsilon_j|\}$$

where ϵ are the computed tracking residuals. A value of $k = 5.2$ is always adequate in practice since it corresponds to about 3.5 standard deviations, a range containing more than 99% of a Gaussian distribution. X84 has a breakdown point of 50%, meaning that any majority in the data can overrule any minority.

III. The Projective Reconstruction Module

There are two steps in obtaining the projective reconstruction of the scene, from the set of point correspondences provided by the tracker.

First the *Fundamental Matrix* [4] relating two views must be determined. The fundamental matrix expresses the *epipolar constraint* and captures all the essential information about the epipolar geometry between the two views and allows us to proceed with the projective reconstruction.

There are various different ways in the literature to estimate the fundamental matrix. We have chosen to implement a simple linear estimation of the matrix [4], aided by the modifications proposed by Hartley [5]. Although this method is not as accurate as the best of its nonlinear relations [4], it is fast and sufficiently accurate for our purposes.

The second step is to do the projective reconstruction itself, utilising the fundamental matrix that has been calculated in the first step. The method chosen, follows closely the one proposed by Faugeras [6], with minor modifications.

A. Fundamental Matrix Estimation

We can calculate the fundamental matrix by expressing the epipolar constraint as follows: For any given point m in the first retina, the projective representation $\mathbf{l}'_m$ of its epipolar line in the second retina can be written as:

$$\mathbf{l}'_m = \mathbf{F}\mathbf{m}$$

But the point m', corresponding to m, is by definition a point of l'_m. One can therefore see that:

$$\mathbf{m}'^T \mathbf{F}\mathbf{m} = 0 \tag{5}$$

The 3×3 matrix $\mathbf{F}$ is the fundamental matrix.

Starting with (5), and writing $\mathbf{m} = (u, v, 1)^T$ and $\mathbf{m}' = (u', v', 1)^T$, we see that each point match gives rise to a set of linear equations in the entries of $\mathbf{F}$:

$$\begin{gathered} uu'f_{11} + uv'f_{21} + uf_{31} + vu'f_{12} + vv'f_{22} \\ +vf_{32} + u'f_{13} + v'f_{23} + f_{33} = 0 \end{gathered}$$

so each row of the equation matrix can be represented by the vector $(uu', uv', u, vu', vv', v, u', v', 1)$. Eight point matches are therefore sufficient to solve for the matrix $\mathbf{F}$ and the method is therefore known as the *8-point algorithm.* If more than eight matches are available, then a linear least-squares minimisation problem must be solved.

From all the point matches we obtain a system of linear equations

$$\mathbf{A}\mathbf{f} = 0 \tag{6}$$

where $\mathbf{f} = (f_{11}, \ldots, f_{33})$ is a 9-dimensional vector containing the unknown entries of the fundamental matrix, and $\mathbf{A}$ is the equation matrix. The fact that $\mathbf{F}$ and hence vector $\mathbf{f}$ are defined only up to a scale factor, requires us to impose an additional constraint on the solution in order to avoid the trivial solution $\mathbf{f} = 0$. This additional constraint can be, for instance, that the norm of $\mathbf{f}$ should be one, $\| \mathbf{f} \| = 1$.

In the presence of noise, another point must be taken into account. Ordinarily, and assuming the existence of a non-zero solution, the matrix $\mathbf{A}$ must be of rank 8. But in the presence of noise, $\mathbf{A}$ will not be rank deficient, it will have rank 9. In this case we will not be able to find a non-trivial solution to (6), but we shall seek instead for a least-squares solution. We shall seek the vector $\mathbf{f}$ that minimises $\| \mathbf{AF} \|$, subject to the constraint $\| \mathbf{f} \| = \mathbf{f}^T\mathbf{f} = 1$. It is well known that this vector is the unit eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue of $\mathbf{A}$.

Simply deriving the fundamental matrix in this way though does not guarantee an important property, namely that the matrix is singular and in fact of rank 2. It is therefore necessary to enforce that, and we can achieve that by correcting the matrix $\mathbf{F}$ found as the solution of (5). Matrix $\mathbf{F}$ will be replaced by a matrix $\mathbf{F}'$ that minimises the *Frobenius Norm* $\|\mathbf{F} - \mathbf{F}'\|$, subject to the condition $\det \mathbf{F}' = 0$.

Still more modifications have to be made, as this simple derivation of the fundamental matrix has been shown to be very sensitive to noise. It has been demonstrated though [5], that by taking into account specific numerical considerations about the condition number of the linear system being solved, the performance of the eight-point algorithm can be improved considerably.

A simple transformation (translation and scaling) of the point coordinates is enough to dramatically improve the stability of the solution, without adding any significant complexity. This transformation involves translating all the data points so that their centroid is at the origin, and isotropically scaling the points so that the average distance of a point from the origin is equal to $\sqrt{2}$.

The translation and scaling steps can be combined into a single transformation $\mathbf{T}$, so the algorithm for determining the fundamental matrix would now read:

- Transform the image coordinates according to $\hat{\mathbf{u}}_i = \mathbf{Tu}_i$and $\hat{\mathbf{u}}'_i = \mathbf{Tu}'_i$.
- Find the fundamental matrix $\hat{\mathbf{F}}$ corresponding to the matches $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$, and enforce the singularity constraint.
- Set $\mathbf{F} = \mathbf{T}^{-T}\hat{\mathbf{F}}\mathbf{T}$.

B. Projective Reconstruction

The 3D projective reconstruction of the scene contains four steps. First the projection matrix is defined, then the location of the epipoles is determined, followed by the location of the optical centers, and finally the projective reconstruction of the points is effected.

B.1 Estimation of the projection matrix

The perspective projection of a four-dimensional scene point $\mathbf{M}$ to a three-dimensional image point $\mathbf{m}$ (all quantities are represented in homogenous coordinates), can be represented by a 3×4 matrix $\mathbf{P}$, such that:

$$\mathbf{m} = \mathbf{PM} \tag{7}$$

and accordingly:

$$\mathbf{m}' = \mathbf{P}'\mathbf{M} \tag{8}$$

for the second camera.

The reconstruction problem can be cast in the following way: given the set of pixel correspondences $\{\mathbf{m}_i,\ \mathbf{m}'_i\}$, find the camera matrices $\{\mathbf{P},\ \mathbf{P}'\}$ and the scene structure $\{\mathbf{M}_i\}$ such that (7) and (8) hold for each i.

Without further restrictions we will – in general – obtain a projective reconstruction [6], that differs from the true one by an arbitrary projective transformation. Indeed, if $\mathbf{P}$ and $\mathbf{M}$ satisfies (7), also $\mathbf{PT}$ and $\mathbf{T}^{-1}\mathbf{M}$ satisfies (7) for any 4×4 nonsingular matrix $\mathbf{T}$.

Hence, one can obtain a projective reconstruction of the scene – thereby fixing the matrix $\mathbf{T}$ – by choosing a set of (any) five point correspondences between the two images, and regarding the five 3D points that give rise to these correspondences as the *standard projective basis* in projective space.Accordingly the first four points of each set of points in each image form the standard projective basis for that image.

With this choise of coordinate systems, the projection matrix can be written simply as [6]:

$$\mathbf{P} = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}$$

where $[\alpha,\ \beta,\ \gamma]$ is the fifth of the five-point set in each image. So $\mathbf{P}$ is written now a function of just one projective parameter x, which will be determined once the epipoles and optical centers have been defined.

B.2 Epipole Determination

The left and right null-spaces of the fundamental matrix $\mathbf{F}$ are generated by the vectors representing (in homogenous coordinates) the two epipoles in the two images. Since the fundamental matrix $\mathbf{F}$ has been determined, the epipoles can now be very easily determined by solving the systems $\mathbf{Fe} = 0$ and $\mathbf{F^T e}' = 0$, for the left and right epipoles respectively.

B.3 Optical Center Estimation

With all the above knowledge, we can now compute the coordinates of the optical centers $\mathbf{C}$ and $\mathbf{C}'$ and calculate the projective parameter x of the projection matrix. Observe that the projection matrix $\mathbf{P}$ projects all world points onto the image, except for the optical center $\mathbf{C}$ and all the points belonging to the focal plane. Therefore:

$$\mathbf{PC} = 0 \tag{9}$$

All that is left now is to determine x and x'. Since we know the epipoles, x and x' can be obtained from: $\mathbf{PC}' = \sigma\mathbf{e}$ and $\mathbf{P}'\mathbf{C} = \sigma'\mathbf{e}'$ with $\sigma \neq 0$ and $\sigma' \neq 0$.

B.4 Relative Reconstruction of Points

We are now in a position, given a correspondence $(\mathbf{m}, \mathbf{m}')$ to reconstruct the three dimensional point $\mathbf{M}$ in the coordinate system defined by the five points. The way to do this reconstruction is to intersect the lines that go through $\mathbf{m}$ and $\mathbf{C}$, and $\mathbf{m}'$ and $\mathbf{C}'$ respectively. These lines are:

$$\lambda\mathbf{C} + \mu\left[C_x m_x, C_y m_y, C_z m_z, 0\right]^T$$

and

$$\lambda'\mathbf{C}' + \mu'\left[C'_x m'_x, C'_y m'_y, C'_z m'_z, 0\right]^T$$

with $\lambda, \mu \in R$ and not both 0, and $\lambda', \mu' \in R$ and not both 0. Intersecting the lines, the three-dimensional point $\mathbf{M}$ can be determined as:

$$\mathbf{M} = -\mathbf{C}\left[\mu + \lambda m_x, \mu + \lambda m_y, \mu + \lambda m_z, \mu\right]^T$$

IV. The Euclidean Reconstruction Module

It is relatively easy to obtain a projective reconstruction. However we want to obtain an *Euclidean* reconstruction, a very special one that differs from the true reconstruction by a similarity transformation.

Once the Euclidean coordinates of five points in 3D-space are known, it is very straightforward to upgrade the projective reconstruction to Euclidean. All that is needed is to find the 4×4 matrix that relates the five Euclidean points with their projective equivalents, and apply this matrix to the rest of the reconstruction.

V. Experimental Results

In this section we report experiments with real data collected in our laboratory tank. A video sequence of objects in the tank is collected without any knowledge of the camera characteristics and while the camera motion remains completely arbitrary and unknown. Examples of frames from this sequence can be seen in Figure 1 (for reasons of space, only one experiment is reported).

This video sequence is fed to the tracker which in turn provides the projective reconstruction module with correspondences. The detected and tracked features can be seen in Figure 2. Once the projective reconstruction is effected, we need to supply the algorithm with the Euclidean 3D coordinates of five points, in order to be able to move from the projective to the Euclidean reconstruction.

The coordinates of these five points do not need to be known with any great accuracy. Indeed, in the reported case, they were extracted by having knowledge of the various objects dimensions and then trying to extract the coordinates of the five points by aproximating the positioning of one object with respect to the other, based on optical observations from the video sequence only. The reconstructed features can be seen in Figure 3.

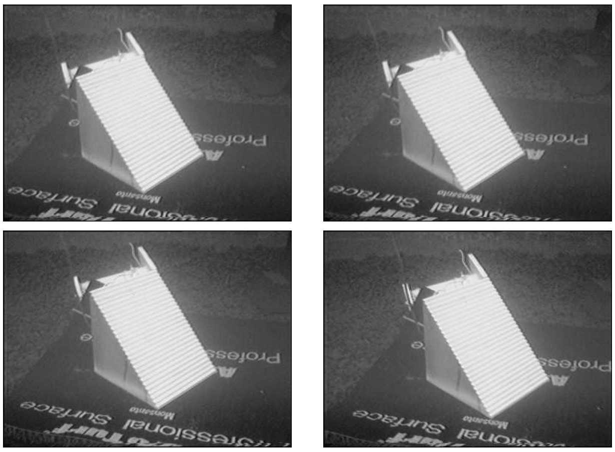
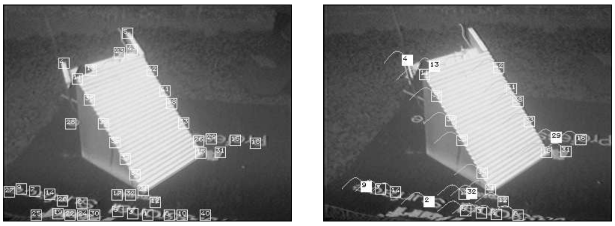


Fig. 1: Some frames in the image sequence.



(α) (β)

Fig. 2: (α) First image in the sequence with detected features superimposed. (β) Last image in the sequence with tracked features superimposed. Outliers in solid white squares.

As can be clearly seen from Figure 3, although the camera caracteristics and motion were unknown and the a-priori information available at best vague, the algorithm manages to reconstruct the tracked features of Figure 2(β) with acceptable accuracy.

VI. Conclusions

This paper has presented a method for 3-D shape reconstruction from uncalibrated underwater video sequences. Approximate knowledge of a few distances between scene points is necessary, which makes the method suitable for estimating the shape of man-made objects (typically, installations). The reconstruction is efficient and experimentation indicates that accuracies are suitable for ROV visual servoing in typical tasks like station keeping. We are currently working on a dynamic extension of this work suitable for integration in a complete ROV visual servoing system.

Acknowledgements

This work was supported by the TMR-MAST programme of the EC, under contract MAS3-CT97-5038.

References

[1] C. Tomasi and T. Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA, April 1991.

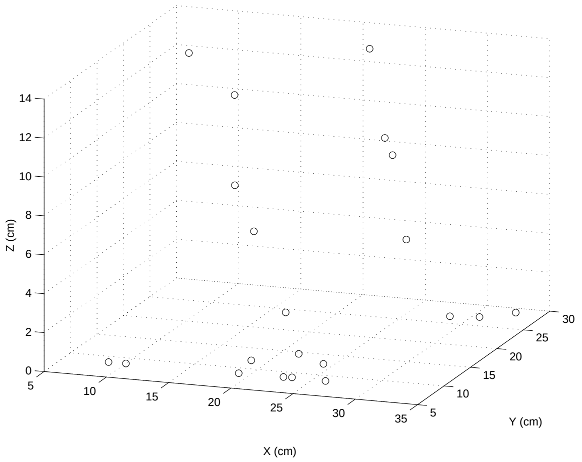


Fig. 3: 3D reconstruction of the tracked features in Fig. 2(β).

[2] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[3] T. Tommasini, A. Fusiello, V. Roberto, and E. Trucco, "Robust feature tracking in underwater video sequences," in *These Proceedings*, 1998.

[4] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, vol. 17, pp. 43–75, 1996.

[5] R. I. Hartley, "In defence of the 8-point algorithm," in *Proceedings of the IEEE International Conference on Computer Vision*, 1995.

[6] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," in *European Conference on Computer Vision*, 1992, pp. 563–578.