# Error propagation for 2D–to–3D matching with application to underwater navigation

W.J. Christmas, J. Kittler and M. Petrou
Vision, Speech and Signal Processing Group
Department of Electronic and Electrical Engineering
University of Surrey

email: w.christmas,j.kittler,m.petrou@ee.surrey.ac.uk
http://www.ee.surrey.ac.uk/

**Abstract**

When a Bayesian approach is used for feature matching, the method relies on the ability to specify reasonable error distributions for the input data. Although in practice the precise form of the distribution is not needed, the method becomes more reliable the more closely the true distribution shape is followed.

In this paper we show how error covariances can be propagated through a relatively complex set of calculations to generate an improved set of covariance matrices for the matching algorithm. We demonstrate the technique with an application in which 2D features are derived from a perspective projection of a 3D CAD model consisting of cylinders. The errors created by uncertainty in the projection parameters are propagated through the projection process to create the covariance matrices for the relations between the feature pairs required by the matching algorithm.

## 1 Introduction

This paper describes an extension to the work in [1], in which the authors describe a feature labelling algorithm based on probabilistic reasoning. The method was used to identify the correspondence between features extracted from a 2-dimensional image and those from a corresponding 2-dimensional model. This work was extended in [2] to show how estimates of the feature measurement errors can be naturally incorporated into the labelling process. In this paper we extend this approach to show how errors from a common source further back in a complex processing chain can be propagated through into the matching process. We use this technique to establish correspondences between features from a 2D image and components from a 3D CAD model, in order to determine the pose of the camera

BMVC 1996 doi:10.5244/C.10.7

with respect to some 3D object that the CAD model describes, for the case in which some (inexact) estimate of the camera pose is available.

The method described in [1] uses relations between pairs of features in both image and model to encapsulate the structural arrangement of the features. By using pairs of features, correspondences can be found in situations in which the unknown transformation between image and model is Euclidean. The complexity of this algorithm is then $O(L^2)$, where $L$ is the total number of labellings considered. In the worst case, $L$ is the product of the image and model feature set sizes. Ideally we would like to extend this method to find correspondences directly between 2D image features and 3D model components. Assuming that such a correspondence exists, then in order to generate a complete set of quantities that are invariant to the projection of the model onto the image, it would be necessary to use higher-order combinations of features. Given that the complexity of the existing 2D-to-2D method can already be problematic for sets of more than a few dozen features, it seemed worthwhile to explore other approaches.

In many practical applications there is some inaccurate initial estimate available of the pose of the camera relative to some 3D object for which a model exists, for example when the camera is moving relative to the object. This initial pose estimate is used to project the model onto the image plane, so that a 2D-to-2D match can be made using the existing labelling algorithm. The labelling results are then used to compute a more accurate camera pose. Initial attempts to do this, using the unmodified algorithm of [1], were partially successful. The difficulty was that the correct covariance matrices for the individual distributions of the feature relations varied considerably, depending in particular on the orientation of the relevant model features relative to the camera. In practice compromise values for these covariances had to be found by trial and error.

To circumvent these problems, in this paper we describe a method in which the error variances of the camera pose estimate are propagated right through to the generation of the feature relations, and hence to the compatibility coefficients (see [1]) of the labelling process. This permits the generation of individual covariance matrices for the relation distributions that accurately reflect the influence of the camera pose errors (modified by the projection process). There still remains the problem of estimating variances for the camera pose parameters. However the method proves to be relatively insensitive to these, because it is the relative size of the relation distributions that is important rather than their absolute size.

In the next section we briefly summarise the matching algorithm, after which we describe the error propagation method. In the following section we describe an application in which the matching algorithm is used as a component of a larger process. Here the task is to determine the pose of a camera in relation to a physical model of an oil rig jacket composed of cylinders. We then examine the sensitivity of the matching process to the variance estimates of the camera pose parameters.

## 2 Summary of the matching algorithm

We can very briefly summarise the algorithm described in [1] as follows. There are $N$ image features, or objects, $\{\mathcal{O}_i,\ i \in 1 \ldots N\}$. There are $M+1$ model features,

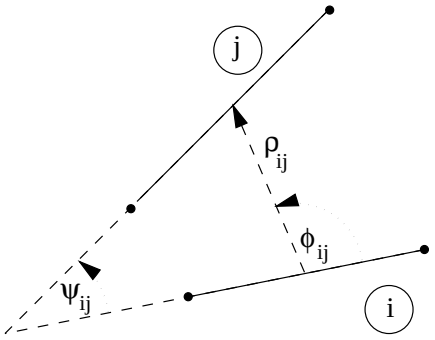



Figure 1: The three types of relation that were used

or labels, $\{\omega_\alpha,\ \alpha \in 0 \ldots M\}$. The label $\omega_0$ denotes a null label that is used when no physical label is appropriate. A labelling of $\mathcal{O}_i$ with $\omega_\alpha$ is denoted by $\mathcal{O}_i \leftarrow \omega_\alpha$.

Feature measurements that can be directly compared between image and model are *attributes* of the features; the set of attributes for image feature $\mathcal{O}_i$ is denoted by $\mathbf{a}_i$, and that for model feature $\omega_\alpha$ by $\mathbf{A}_\alpha$. Since the attributes are in general not sufficient to specify the geometric structure of the feature sets, *relations* between pairs of features are also used. The set of relations between the image features $\mathcal{O}_i$ and $\mathcal{O}_j$ is denoted by $\mathbf{r}_{ij}$, and that between model features $\omega_\alpha$ and $\omega_\beta$ by $\mathbf{R}_{\alpha\beta}$. Three types of relations are used; these are illustrated as $\rho$, $\phi$ and $\psi$ in Fig 1.

The "correct" label for an object $\mathcal{O}_i$ is ascertained by evaluating the labelling probabilities conditional on the data, $\Pr(\mathcal{O}_i \leftarrow \omega_\alpha | \mathbf{a}, \mathbf{r})$, of each label $\omega_\alpha$ for that object. The conditional labelling probabilities are expanded using Bayes's rule together with some simplifying assumptions. The resulting expression is then used in a relaxation process that iteratively updates these probabilities to generate a more consistent labelling. Assuming a set of initial probabilities $\Pr^{(1)}$ are provided, this relaxation process can be expressed as:

Iterate from $n = 1$ {

$\forall i \quad \forall \alpha \quad \{$

$$Q^{(n)}(\mathcal{O}_i \leftarrow \omega_\alpha) = p(\mathbf{a}_i \mid \mathcal{O}_i \leftarrow \omega_\alpha) \prod_{\forall j \neq i} \sum_{\forall \beta} \Pr^{(n)}(\mathcal{O}_j \leftarrow \omega_\beta)\ p(\mathbf{r}_{ij} \mid \mathcal{O}_i \leftarrow \omega_\alpha, \mathcal{O}_j \leftarrow \omega_\beta)$$

$$\Pr^{(n+1)}(\mathcal{O}_i \leftarrow \omega_\alpha) = \frac{\Pr^{(n)}(\mathcal{O}_i \leftarrow \omega_\alpha)\ Q^{(n)}(\mathcal{O}_i \leftarrow \omega_\alpha)}{\sum_{\forall \lambda} \Pr^{(n)}(\mathcal{O}_i \leftarrow \omega_\lambda)\ Q^{(n)}(\mathcal{O}_i \leftarrow \omega_\lambda)}$$

}

}

On convergence, the largest labelling probability for each object is selected.

The important quantities as far as the present discussion is concerned are the relation p.d.f.s, $p(\mathbf{r}_{ij} \mid \mathcal{O}_i \leftarrow \omega_\alpha, \mathcal{O}_j \leftarrow \omega_\beta)$ (and, to a lesser extent, the attribute

p.d.f.s $p(\mathbf{a}_i \mid \mathcal{O}_i \leftarrow \omega_\alpha)$). In practice we consider them to be normally distributed, although the error analysis in the following section does not require this assumption. Thus the relation p.d.f. can be written as:

$$p(\mathbf{r}_{ij} \mid \mathcal{O}_i \leftarrow \omega_\alpha, \mathcal{O}_j \leftarrow \omega_\beta) = \mathcal{N}_{\mathbf{r}_{ij}}(\mathbf{R}_{\alpha\beta}, \Sigma^{\mathbf{r}})$$

*i.e.* a normal distribution for the random vector $\mathbf{r}_{ij}$, with mean $\mathbf{R}_{\alpha\beta}$ and covariance matrix $\Sigma^{\mathbf{r}}$. It is the evaluation of this covariance matrix that is particularly of interest; we discuss how it might be done in the following section.

# 3 Error propagation

Consider a set of measurements, each of which is subject to some error, that form a vector of random variables $\mathbf{x}$. The errors of the components may not be independent; we therefore define a covariance matrix $\Sigma^{\mathbf{x}}$ for the measurements. Now consider a second set of random variables forming a vector $\mathbf{y}$, that are a (vector) function of the first set:

$$\mathbf{y} = \mathbf{f}(\mathbf{x})$$

We can define a similar covariance matrix $\Sigma^{\mathbf{y}}$ for the vector $\mathbf{y}$. If the function $\mathbf{f}$ is everywhere differentiable and the covariances $\Sigma^{\mathbf{x}}$ are sufficiently small that $\mathbf{f}$ may be considered locally linear, then the covariances are related by:

$$\Sigma^{\mathbf{y}} = J^{\mathbf{xy}T} \Sigma^{\mathbf{x}} J^{\mathbf{xy}}$$

where $J^{\mathbf{xy}}$ is the Jacobian matrix whose element in the $k$th row and $l$th column is given by:

$$J^{\mathbf{xy}}{}_{kl} = \frac{\partial f_l}{\partial x_k}$$

Several such functions can be concatenated — thus for a third set of values, $\mathbf{z} = \mathbf{g}(\mathbf{y})$, we could write:

$$\begin{aligned} \Sigma^{\mathbf{z}} &= J^{\mathbf{yz}T} \Sigma^{\mathbf{y}} J^{\mathbf{yz}} \\ &= (J^{\mathbf{xy}} J^{\mathbf{yz}})^T \Sigma^{\mathbf{x}} J^{\mathbf{xy}} J^{\mathbf{yz}} \end{aligned}$$

We can apply the above technique to the evaluation of the relation covariance matrices discussed in the previous section. The ultimate source of the errors in the relations is in the initial estimate of the camera pose parameters. There are two stages to the generation of the relations and their covariances: the projection process, and the generation of the model relations from the projected measurements. In the first stage, the set of initial pose parameters $\mathbf{p}$ are used to project the CAD components onto the image plane. If a CAD component is denoted by some set of (known) parameters $\mathcal{X}$, and a corresponding projected 2D model feature by a set of measurements $\mathbf{X}$, the projection of the component can be represented by a vector function $\mathbf{f}$:

$$\mathbf{X} = \mathbf{f}(\mathcal{X}, \mathbf{p})$$

Since the pose parameters $\mathbf{p}$ contain errors, they are regarded here as a set of random variables, while the CAD measurements $\mathcal{X}$ are assumed to be correct. Thus there is associated with the parameters a covariance matrix $\Sigma^{\mathbf{p}}$; consequently the projected measurements $\mathbf{X}$ have associated covariances $\Sigma^{\mathbf{X}}$, and $J^{\mathbf{pX}}$ is the corresponding Jacobian matrix:

$$\Sigma^{\mathbf{X}} = {J^{\mathbf{pX}}}^T \Sigma^{\mathbf{p}} J^{\mathbf{pX}}$$

Note that $J^{\mathbf{pX}}$ (and hence $\Sigma^{\mathbf{X}}$) is also a function of $\mathcal{X}$, and will therefore vary from feature to feature.

In the second stage, from each pair of projected features, $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ say, the measurements are used to generate a set of relations $\mathbf{R}_{\alpha\beta}$ between the features. Thus we need the covariance matrix $\Sigma^{\mathbf{X}_\alpha \mathbf{X}_\beta}$ for a *pair* of projected features:

$$\Sigma^{\mathbf{X}_\alpha \mathbf{X}_\beta} = \left[ J^{\mathbf{pX}_\alpha} \mid J^{\mathbf{pX}_\beta} \right]^T \Sigma^{\mathbf{p}} \left[ J^{\mathbf{pX}_\alpha} \mid J^{\mathbf{pX}_\beta} \right]$$

where the notation $[\mathbf{a}|\mathbf{b}]$ denotes the (horizontal) concatenation of the matrices $\mathbf{a}$ and $\mathbf{b}$ into a single matrix. We can then calculate $\Sigma^{\mathbf{r}}$, the covariances for the relations:

$$\begin{aligned} \Sigma^{\mathbf{r}} &= {J^{\mathbf{Xr}}}^T \Sigma^{\mathbf{X}_\alpha \mathbf{X}_\beta} J^{\mathbf{Xr}} \\ &= \left( \left[ J^{\mathbf{pX}_\alpha} \mid J^{\mathbf{pX}_\beta} \right] J^{\mathbf{Xr}} \right)^T \Sigma^{\mathbf{p}} \left[ J^{\mathbf{pX}_\alpha} \mid J^{\mathbf{pX}_\beta} \right] J^{\mathbf{Xr}} \end{aligned}$$

where $J^{\mathbf{Xr}}$ is the Jacobian matrix relating the relation errors to the errors in the pair of features $\omega_\alpha$ and $\omega_\beta$.

A similar (but simpler) analysis can be made to derive covariance matrices for the attributes.

The alert reader will have noticed that, in the previous section, we regarded the image relations as the random variables, with means given by the corresponding model relations; on the other hand the analysis of this section treats the model relations as if *they* were the random variables. This apparent discrepancy can be resolved by assuming that we are working in the space of the true projection rather than that of the erroneous projection.

If the errors due to the extraction of the edge line segments from the image are significant, a separate covariance matrix can be composed to reflect this, following the method described in [2]; the two matrices are then combined by summing them.

# 4 An application

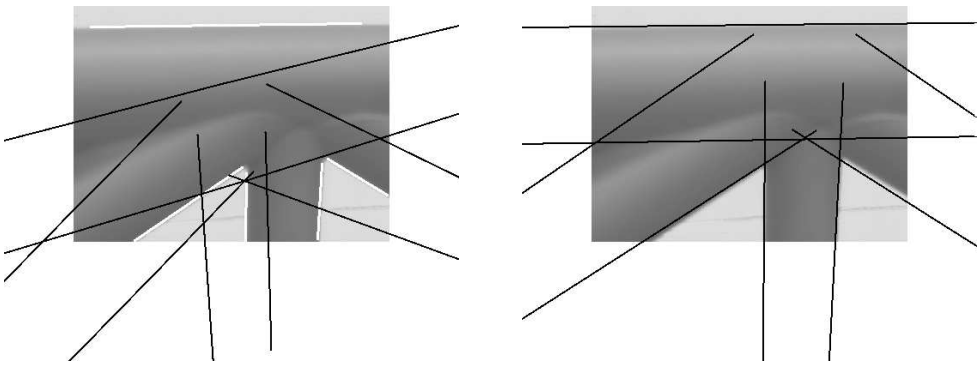
To demonstrate the method, we used the matcher in an application in which the CAD model was of part of an oil rig jacket. The assumption is that, in real life, a camera is mounted on an autonomous submarine, the objective being to find where the submarine is in relation to the oil rig (*i.e.* in effect to find the camera pose). In practice we made use of a physical scale model of the oil rig, in a laboratory.

The components of the model are cylinders, represented by the 3D positions of their endpoints together with the cylinder radius. The corresponding projected feature is a directed line segment representing an edge of the cylinder, represented

in turn by the 2D endpoint positions in the image plane. The tolerances of the CAD model were not given, and so the model itself was assumed to be accurate.

The image depicts a node of the structure, *i.e.* a point on the structure where several cylinders meet. A rough estimate of the camera pose for the image was given, together with a very approximate estimate of the likely magnitude of the pose errors.

The results of the experiment are shown in Fig. 2. The cylinder edges are extracted from the image and fitted to straight line segments, shown in white in Fig. 2(a). The projection of the CAD model is shown superimposed (in black) on the image, using the initial (erroneous) camera pose parameters. Because of the mismatch in image and model segment lengths, they are both chopped into smaller segments, of length roughly corresponding to the projected cylinder diameter (which in turn can be estimated from the initial pose parameters). The matcher is then used to find correspondences between the image and projected model segments. From the match, a more accurate pose is computed, using an iterative algorithm based on [5, 6]. The CAD model is then re-projected onto the image (Fig. 2(b)). A flow diagram for the entire pose estimation process is shown in Fig. 3.



(a) initial projection (b) final projection

Figure 2: Example of pose estimation

Another example, using a different node on the structure, is shown in Figs. 4. This node is on a corner of the structure, and several of the cylinders are hidden by the main upright member, as can be seen in the projected model overlay. The image contains some additional clutter (some pipes mounted on the laboratory wall); however this clutter, although detected by the edge finder, was rejected by the matcher, so that the correct final pose was still obtained.

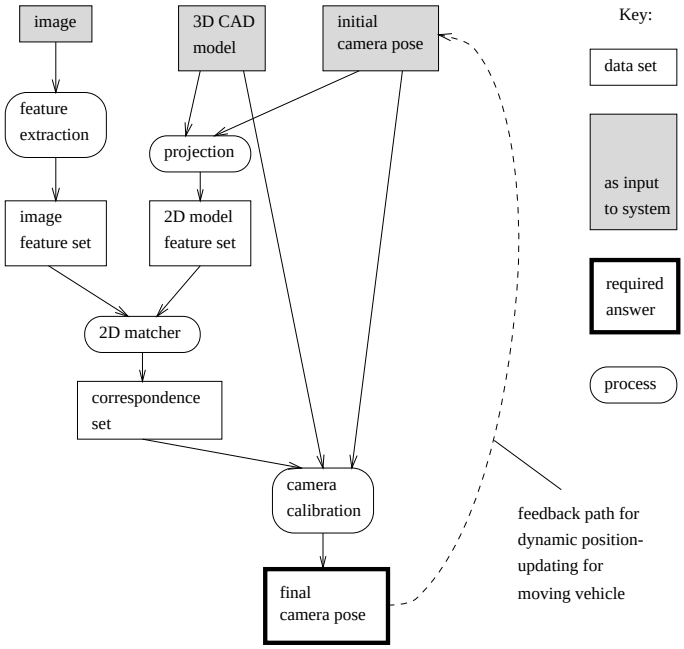
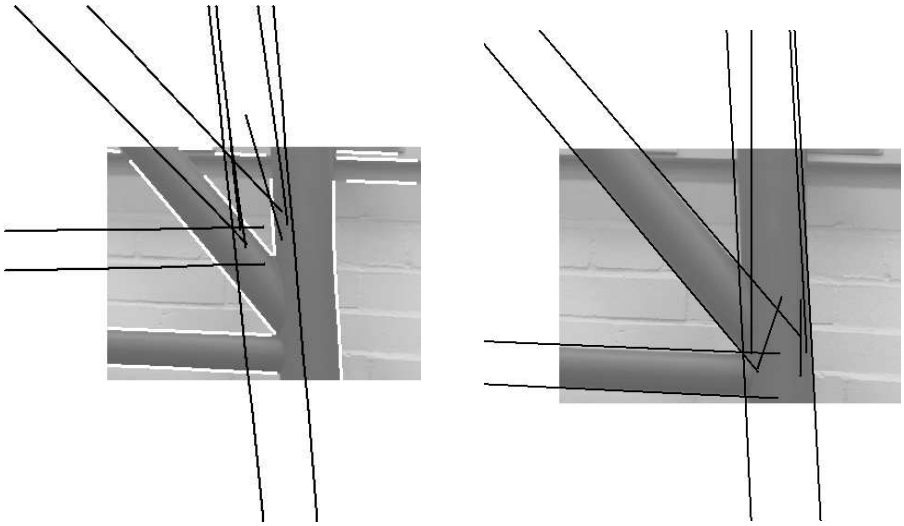



Figure 3: Flowchart for pose estimation process



(a) initial projection (b) final projection

Figure 4: Additional example using corner node

# 5 Parameter sensitivity

The only parameters required by the matcher that are not specified by the theory are the components of the covariance matrix for the pose parameter errors. There are 6 pose parameters — 3 angles for the orientation and 3 position parameters. We assumed that the errors are all independent; thus 6 variances have to be specified, using some prior knowledge about the likely accuracy of the initial camera pose estimate.

We tested the sensitivity of the matching algorithm to the values of these variances. We used the image in Fig. 2 for these tests, with the same initial pose estimate. There were 6 line segment features extracted from the scene, and 20 from the projected model (longer segments being subdivided, as discussed earlier). We assumed that the line segments were fitted to the image edges with a standard deviation of 1 pixel.

In order to reduce the dimensionality of the parameter space that is to be investigated, the orientation errors were all assumed to have the same standard deviation, $\sigma_\theta$; similarly the position errors have a single standard deviation $\sigma_p$. Note that, to give an indication of the scale of $\sigma_p$, the cylinders in Fig. 2 have diameters in the range 33–48 mm. The results, for various combinations of these two values, are shown in Table 1, in which:

- $\surd$ A tick indicates that all image features were correctly matched, leading to a good fit of the re-projected model onto the image (as in Fig 2(b)).
- ? A question mark indicates that one or more of the image features were labelled with the null (*i.e.* "don't know") label. The pose estimator fitted the remaining segments well, but the pose estimate itself was in some cases less reliable.
- × A cross indicates that one or more features were incorrectly matched, thus supplying the pose estimator with incorrect information, causing the wrong pose to be found. This could in principle be detected by measuring the magnitude of the residual error of the pose estimation. In each example of this last category, the lower edge of the left-hand diagonal member was incorrectly labelled as the lower edge of the horizontal member.

| $\sigma_\theta$ (°) | $\leq 0.2$ | 0.5 | 1 | 2 | $\geq 5$ |
|---|---|---|---|---|---|
| $\sigma_p$ (mm) $\leq 1$ | ? | ? | ? | ? | ? |
| 2 | ? | $\surd$ | ? | ? | ? |
| 5 | $\surd$ | $\surd$ | $\surd$ | ? | ? |
| 10 | $\surd$ | $\surd$ | $\surd$ | ? | ? |
| 20 | $\surd$ | $\surd$ | $\surd$ | × | ? |
| 50 | × | × | × | $\surd$ | ? |
| $\geq 100$ | ? | ? | ? | ? | ? |

Table 1: Sensitivity of matcher to pose error standard deviations

The interpretation that we drew from these results is as follows. A relation p.d.f. value, or "compatibility coefficient" (*c.f.* [3, 4]), represents the degree of compatibility between a pair of image features and a pair of model features. Because of the assumption that the error propagation is linear, the width of the relation p.d.f. will be directly proportional to the pose error standard deviations. If these standard deviations are set too small, resulting in a very narrow relation p.d.f., then mismatches between image and model relations will often cause the p.d.f. to be evaluated somewhere far along its tail, resulting in a very low value for the compatibility coefficient. Since p.d.f.s involving the null label have a constant value, the null label will then be preferred. If on the other hand the pose error standard deviations are set too high, the relation p.d.f. will be very wide, and with correspondingly low amplitude. In this case, two things can happen. Because the p.d.f.s are so wide they will overlap to a large extent, so that support for a labelling can easily be given by the wrong relation. Also, because of the reduced p.d.f. amplitude, even though the p.d.f. is likely to be evaluated near its centre, its amplitude can still be less than that of the null p.d.f., giving preferential support to the null label.

We see however that both parameters can be varied over about an order of magnitude, while still obtaining good results. This is in marked contrast to a previous version of this application that used a constant, diagonal covariance matrix (as recommended in [1]); in that case a suitable combination of covariance values was found with difficulty, by trial and error.

In the example used here there were relatively few image features, so that if one or two were null-labelled this sometimes caused a noticeable degradation in the performance of the following pose estimation stage. In many applications there will be rather more image features, in which case a few null labellings will matter rather less. Incorrect labellings, on the other hand, are likely to create a more serious problem. Thus it would seem preferable to underestimate rather than overestimate the pose error standard deviations.

# 6 Conclusions

In [1] a method for matching 2D geometrical features was described, using probabilistic relaxation. The method hinges on the evaluation of probability density functions for relations between pairs of features. However it was not clear how the form of the density functions should be determined, and in particular the method assumed that the same form could be used for all relations. In [2] the authors used an improved error model to show the deficiencies of this assumption. However the latter method still assumed that all the feature measurement errors were independent. In this paper we indicate how to proceed in cases for which the feature measurement errors are not independent, but are in turn a result of some common source of errors — which in the application discussed here are the errors in the camera pose estimate.

We demonstrated the technique on an application that simulates an underwater navigation task in the vicinity of an oil rig. A 3D CAD model of the oil rig was used consisting of a set of cylinders, together with images of a scale model of the oil rig. The CAD model was projected onto the images using some initial erroneous

estimate of the camera pose. The projected model was then matched with the image features, and the results used to compute more accurately the camera pose. Estimates of the likely magnitude of the pose errors (in the form of their variances) were propagated through the projection process to create more reliable covariance matrices for the feature relations in the matching algorithm. Experiments showed that the method was tolerant to a reasonably wide spread of pose error estimates, and that intuitively reasonable values could be used for these estimates.

In order to apply the method to a real underwater navigation task, further study of the feature extraction process would be needed, since the visibility would be much worse than in the laboratory. There are two different lighting conditions that have to be considered. When operating near the surface in conditions of relatively good visibility, without artificial lighting, the oil rig will appear as a silhouette. In this case the method described is appropriate, although the signal-to-noise ratio will be much worse. When artificial lighting is used, typically mounted on the vehicle near the camera, the situation is altered. The actual outline of the cylinder is unlikely to be easily visible; instead, a reflection of the vehicle lights in the cylinders will be picked up, leading to an overestimate of the distance of the vehicle from the structure.

## References

[1] W.J. Christmas, J. Kittler, and M. Petrou. Matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):749–764, August 1995.

[2] W.J. Christmas, J. Kittler, and M. Petrou. Modelling compatibility coefficient distributions for probabilistic feature-labelling schemes. In D. Pycock, editor, *Proceedings of the Sixth British Machine Vision Conference*, volume 2, pages 603–612, Birmingham, U.K, 1995.

[3] R.A. Hummel and S.W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(3):267–286, May 1983.

[4] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems, Man, and Cybernetics*, 6:420–433, June 1976.

[5] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3:323–344, 1987.

[6] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5:345–358, 1989.