Improving model shape acquisition by incorporating geometric constraints

N. Werghi, R. B. Fisher, A. Ashbrook, C. Robertson
Department of Artificial Intelligence, Edinburgh University

Abstract

While the problem of model fitting for 3-dimensional range data has been addressed with some success, the problem of increasing the accuracy of the whole fit still remains. This paper describes a technique of global shape improvement based upon feature position and shape constraints. These constraints may be globally applied or inferred from general engineering principles. This paper describe a general, incremental, framework whereby constraints can be added and integrated in the model reconstruction process, resulting on optimal trade-off between minimization of the shape fitting error and the constraint's tolerances.

1 Introduction

There has been a recent flurry of effort on reconstructing 3D geometric models of objects from single [2, 5, 6] or multiple [3, 10, 9, 11] range images, in part motivated by improved range sensors, and in part by demand for geometric models in the CAD and Virtual Reality (VR) application areas. Mainly, these reconstructions are of objects with smooth, free-form surfaces. Oddly enough, in this case, curved surface objects are easier to work with, as: 1) the variety of surface geometry provides many more features for multiple dataset registration, and 2) the tolerances needed for most curved surface applications are not high. Or, conversely, one could say that objects with developable surfaces are harder to reconstruct accurately, because: 1) the developable surfaces (*e.g.* including standard engineering surfaces produced by simple machining – planes, cylinders and cones) allow translations of the surfaces from different observations relative to each other that still satisfy distance constraints (*i.e.* two views of a planar surface that slide in the same infinite plane relative to each other), and 2) developable surfaces tend to have shape tolerances that are much higher than that achievable by standard range sensors because these surfaces are commonly used to mate parts together, whereas smooth freeform or spline surfaces tend to have shape tolerances comparable to typical range sensor data.

Further, even if all of the data were from a single view, thus avoiding multiple dataset registration errors, reconstruction must still cope with errors from miscalibration across the full sensor field of view.

This paper describes a technique of global shape improvement based on feature position and shape constraints. The constraints might be either interactively supplied by a user, or inferred by a knowledge-based system reasoning from general engineering principles.

The types of constraints exploited here are of these families:

1. a set of features have a fixed orientation relationship (*e.g.* a set of surfaces or edges that meet at a specified angle or are parallel) and

2. a set of features have a fixed separation (*e.g.* the distance between a pair of parallel lines or planes).

These are typical engineering relationships, and, in particular, are the sorts of properties that fix relationships between part-mating features.

The key to the approach is to parameterize the features in a way that allows constraints to be expressed as a function of the shape parameters, and to then apply an optimization procedure that searches for parameter vectors that satisfy the constraints while simultaneously optimizing the surface fit to the range data.

2 Background

The integration of geometric constraints into the shape fitting process has been treated for wire frame model construction by Porrill [8]. Wire frame models were constructed from stereo-data. The model features were given statistical distributions and geometric constraints between features produced dependencies in the distributions. The model adjustment process maximized the *a posteriori* probability of the models. Since the models were based on wire frames, the constraints were related to lines. Four types of constraints were considered: orthogonality, intersection, equality and connection by a small rigid motion. The optimal feature parameters were estimated using an extended Kalman filter. At each iteration, constraints are linearized in the neighborhood of the current estimate, and then used to correct the measurement. Porrill's approach is nice since it takes advantage of the recursive linear estimation of Kalman filtering, however it assumes a Gaussian distribution which may not always the case. Moreover, the method, guarantees the satisfaction of the constraints only to linearized first order. Additional iterations at each estimation step are needed if one would like to obtain more accuracy. This last condition has been taken into account in the work of De Geeter and al [4] by defining a “Smoothly Constrained Kalman Filter”. The key of their approach is to replace a nonlinear constraint by a set of linear constraints applied iteratively and updated by new measurements in order to reduce the linearization error.

3 Problem Definition

Given sets of 3D measurement points representing surfaces belonging to a certain object, we want to estimate the different surface parameters taking into account the geometric constraints between these surfaces.

3.1 Surface Parametrization

Consider $S_1, ., .S_N$ the set of surfaces and $\vec{p_1}, ., .\vec{p_N}$ the set of parameter vectors related to them. Each vector $\vec{p_i}$ has to minimize a given error criterion J_i associated with the surface S_i. A reasonable criterion is the least squared error one. So let's consider the following objective function composed of the sum of error criterions

$$J = J_1 + J_2 + ...J_N \tag{1}$$

By considering the implicit equation representation of surfaces, a surface S_i is represented by:

$$\vec{h_i}^T \vec{p_i} = 0 \tag{2}$$

where $\vec{h_i}$ is the measurement vector. Note that any polynomially describable surface can be presented in this scheme, as each component in $\vec{h_i}$ can be of the form $(x^\alpha y^\beta z^\gamma)$ for some (α, β, γ).

Given m_i measurements, the least squares criterion related to this equation is

$$J_i = \sum_{l=1}^{m_i} (\vec{h_i^l}^T \vec{p_i})^2 = \vec{p_i}^T H_i \vec{p_i} \tag{3}$$

where $H_i = \sum_{l=1}^{m_i} (\vec{h_i^l} \vec{h_i^l}^T)$ represents the sample covariance matrix of the surface S_i. (We assume that the assignment of measurements to surfaces is known.) The objective function (1) can then be written as :

$$J = \sum_{i=1}^{N} \vec{p_i}^T H_i \vec{p_i} \tag{4}$$

By concatenating all the vectors $\vec{p_i}^T$ into one vector $\vec{p}^T = [\vec{p_1}^T, \vec{p_2}^T, ., ., ., \vec{p_N}^T]$ equation (4) can be written as

$$J = \vec{p}^T \mathcal{H}_o \vec{p}, \qquad \mathcal{H}_o = \begin{bmatrix} H_1 & (0) & . & (0) \\ (0) & H_2 & . & (0) \\ (0) & . & . & (0) \\ (0) & . & (0) & H_N \end{bmatrix} \tag{5}$$

3.2 Constraint Representation

The constraints can be classified into two main categories, constraints on the surface parameter vectors and constraints on both data and parameters. A constraint belonging to the second category would be rather considered as an observation equation since it involves measurement. By a reasoning similar to that in Section (3.1) such kind of constraints can be put into the following form

$$C(\vec{p}) = \vec{p}^T \mathcal{H}_c \vec{p} + \vec{h}_c \vec{p} + K_c \tag{6}$$

where $\mathcal{H}_c$, $\vec{h}_c$ and K_c are respectively a matrix, a vector and constant, all depending on the data.

The constraints which involves only the vector parameters can be represented by the set of implicit equations

$$C_k(\vec{p}) = 0, \quad k = 1..K \tag{7}$$

An example of how these equations are instantiated is given in Section 4.2.

3.3 Optimization of shape satisfying the constraints

The parameter vector $\vec{p}$ has to minimize the objective function (5) subject to the constraints (6) and (7) imposed by the model. So, the problem that we are dealing with is a constrained optimization problem to which an optimal solution may be provided by minimizing the following energy function:

$$E = \vec{p}^T(\mathcal{H}_o + \mathcal{H}_c)\vec{p} + \vec{h}_c^T\vec{p} + K_c + \sum_{k=1}^{K} \lambda_k C_k(\vec{p}), \quad \lambda_k \geq 0 \tag{8}$$

known in the literature as the Lagrangian function. the above function contains two components, the least squares function:

$$F(\vec{p}) = \vec{p}^T(\mathcal{H}_o + \mathcal{H}_c)\vec{p} + \vec{h}_c^T\vec{p} + K_c \tag{9}$$

and the constraint function:

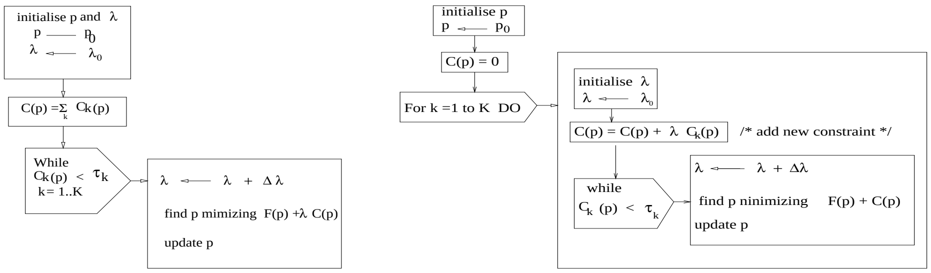
$$C(\vec{p}) = \sum_{k=1}^{K} \lambda_k C_k(\vec{p}) \tag{10}$$

The method of solving this problem depends on the nature of the objective function (convex or not), the type of the constraints (linear or not) and whether the constraints could be merged together in order to reduce the number of parameters and eventually combined with the least square objective function.

The objective function is convex since it is quadratic and the matrix $\mathcal{H}_o$ is positive definite (since each matrix H_i is positive definite). The same propriety can be satisfied by $\mathcal{H}_c$ as well.

So the problem can be said to be a convex optimization problem if the constraints $C_k(\vec{p})$ are also convex functions. On the other hand, the existence of an optimal solution necessitates that both the least squares function and the constraint function are differentiable. A detailed analysis of the convexity and the optimality conditions is available in [7].

In some particular cases it is possible to get a closed form solution of (8). This depends of the characteristics of the constraint functions and whether it is possible to combine them efficiently with the objective function. But generally, it is not trivial to develop a closed form solution especially when the constraints are nonlinear and their number is high. In such case, an algorithmic approach could be of great help taking into account the increasing capabilities of computing. The main idea was to develop a search optimization scheme for determining the best set $(\vec{p}, \lambda_1, ., ., \lambda_k)$. Moreover, we have been seeing whether it is possible that one can get the solution which satisfies a desired tolerance. So the objective is to determine the vector $\vec{p}$ which satisfies the constraints to the desired accuracy and which fits the data to a reasonable degree.



(a) (b)

Figure 1: (a): *optim*1 - batch constraint optimization algorithm. (b): *optim*2 - sequential constraint introduction optimization algorithm.

To solve the optimization problem, we have simplified the full problem slightly. As first step, we have given an equal weight to each constraint, so a single λ is considered for all the constraints:

$$E = \vec{p}^T(\mathcal{H}_o + \mathcal{H}_c)\vec{p} + \vec{h}_c^T\vec{p} + K_c + \lambda \sum_{k=1}^{K} C_k(\vec{p}), \quad \lambda \geq 0 \tag{11}$$

The problem now is how to find $\vec{p}$ that minimizes (11). Because (11) may be a non-convex problem (thus having local minima), we solve the problem in a style similar to the GNC method [1]. That is we start with a parameter vector $\vec{p}^{[0]}$ that satisfies the least squares constraints, and attempt to find a nearby vector $\vec{p}^{[1]}$ that minimizes (11) for small λ (in which the constraints are weakly expressed). Then iteratively, we increase λ slightly and solve for a new optimal parameter $\vec{p}^{[n+1]}$ using the previous $\vec{p}^{[n]}$. While we cannot (so far) guarantee that we converge to an optimal value, at least so far we have not seen cases of suboptimal solutions.

At each iteration n the algorithm increases λ by a certain amount, and a new $\vec{p}^{[n]}$ is found such that the optimization function is minimized by means of the standard Levenberg-Marquardt algorithm. The parameter vector $\vec{p}^n$ is then updated to the new estimate $\vec{p}^{[n+1]}$ which become the initial estimate at the next value of λ. The algorithm stops when the constraints are satisfied to the desired degree, or when the parameter vector remains stable for a certain number of iterations. The above algorithm is illustrated in Figure 1a.

The initialization of the parameter vector is crucial to guarantee the convergence of the algorithm to the desired solution. For this reason the initial vector should be taken as the one which best fitted the set of data. This vector can be obtained by estimating each surface's parameter vector separately and then concatenating the vectors into a single one. Naturally the option of minimizing the least squares function $F(\vec{p})$ alone has to be avoided since it leads to the trivial null vector solution. On another hand, the initial value λ has to be large enough in order to avoid first the above trivial solution and second to give the constraints a certain weight. A convenient value for the initial λ is

$$\lambda_0 = \frac{F(\vec{p}^{[0]})}{C(\vec{p}^{0[]})} \tag{12}$$

where $\vec{p}^{[0]}$ is the initial parameter estimation.

Another option of the algorithm consists of adding the constraints incrementally. At each step a new constraint is added to the constraint function $C(\vec{p})$ and then the optimal value of $\vec{p}$ is found according to the scheme shown in Figure 1b. For each new added constraint $C_k(\vec{p})$, λ_k is initialized at λ_0, whereas $\vec{p}$ is kept at its current value.

4 Case of Polyhedral Objects

Polyhedral objects involves the two types of constraints mentioned in the introduction. They are represented in this case by fixed angles between the planes' normals and the fixed distances between parallel planes.

4.1 Planes with a fixed orientation relationship

A plane surface can be represented by this following equation:

$$p_x x + p_y y + p_z z + d = 0; \tag{13}$$

where $[p_x, p_y, p_z]^T$ is unit normal vector to the plane and d is the distance to the origin. For each plane surface we consider a local frame centered on a point belonging to the plane (in practice this point is taken as the center of gravity of the measurement points), so the plane equation can be written as

$$p_x x + p_y y + p_z z = 0 \tag{14}$$

Let's consider N planes, where the angles between planes' normals are known. The orientation relationship between the different planes are defined by the following constraints:

$$(\vec{p_i}^T \vec{p_j} - cos(\alpha_k))^2 = 0, \;\; i,j \in [1..N], \quad i > j, \;\; k \in [1..K = (N-1) \times N/2] \tag{15}$$

Each plane normal has also to satisfy the unity constraint

$$C_i(\vec{p}) = (||\vec{p_i}||^2 - 1)^2 = 0, \;\; i \in [1..N] \tag{16}$$

The constraint functions are squared in order to have convex functions. The constraints (15) and (16) can be written under a matrix formulation:

$$\begin{aligned} U_i(\vec{p}) = (\vec{p}^T \mathcal{U}_i \vec{p} - 1)^2 &= 0, \;\; i \in [1..N] && (17)\\ A_k(\vec{p}) = (\vec{p}^T \mathcal{A}_k \vec{p} - 2cos(\alpha_k))^2 &= 0, \;\; k \in [1..K = (N-1) \times N/2] && (18) \end{aligned}$$

where $\vec{p}^T = [p_1^T, ., ., p_N^T]$, U_i and $\mathcal{A}_k$ are $N \times N$ block matrices defined by:

$$\mathcal{U}_i = \begin{bmatrix} (0) & (0) & . & (0) \\ (0) & (I_3)_{ii} & . & (0) \\ (0) & . & (0) & (0) \\ (0) & . & (0) & (0) \end{bmatrix}, \;\; \mathcal{A}_k = \begin{bmatrix} (0) & (0) & . & . \\ (0) & . & (I_3)_{ij} & (0) \\ (0) & (I_3)_{ji} & (0) & (0) \\ . & . & (0) & (0) \end{bmatrix}$$

and I_3 is the 3×3 identity matrix.

4.2 Parallel planes with a given separation

Consider without loss of generality two parallel planes S_p and S_q containing respectively N_p and N_q points and separated by the algebraic distance d_{pq}. Since the two planes have a common orientation a single normal can be associated with them. Each pair of points (M_i^p, M_j^q), $M_i^p \in S_p$, $M_j^q \in S_q$ has to satisfy the following equation:

$$(M_i^p M_j^q)^T \vec{p} - d_{pq} = 0 \tag{19}$$

by considering all the planes' points, the normal $\vec{p}$ has to minimize the above least squares criterion:

$$C(\vec{p}) = \sum_{i,j}^{N_p,N_q} ((M_i^p M_j^q)^T \vec{p} - d_{pq})^2 = \vec{p}^T H_{pq} \vec{p} - 2 d_{pq} \vec{h}_{pq}^T \vec{p} + N_{pq} d_{pq}^2 \tag{20}$$

where $$H_{pq} = \sum_{i,j}^{N_p,N_q} (M_i^p M_j^q)(M_i^p M_j^q)^T, \quad \vec{h}_{pq}^T = \sum_{i,j}^{N_p,N_q} M_i^p M_j^q, \quad N_{pq} = N_p N_q$$

5 Experiments

A series of experiments on synthetic and real data have been carried out to check the behavior and the convergence of the algorithm. Two representative samples will be shown here, the first concerned a real tetrahedron, the second a synthetic step model object. The algorithm *optim*1 was applied in the first case. For the second object both algorithms *optim*1 and *optim*2 were applied in order to compare their performances. The behavior of the algorithms are checked through the unit constraint error, the angle constraint error, the normal orientation error (the angle between the exact surface normal and the constrained one), the least square function and the constraint function, all mapped as function of λ.

5.1 The tetrahedron

Figure 2a (left) shows a tetrahedron with three faces visible, This object involves three constraints represented by the three angles 90^o, 90^o and 120^o between the three surface normals, as well as the unit vector constraints, The energy function is:

$$E(\vec{p}) = \vec{p}^T \mathcal{H} \vec{p} + \lambda(\sum_{k=1}^{3} A_k(\vec{p}) + \sum_{l=1}^{3} U_l(\vec{p})) \tag{21}$$

The data was acquired with a 3D triangulation range sensor. All constraints were applied simultaneously according to algorithm *optim*1. The results are the average of 100 trials, with the initial vector $\vec{p}^{[0]}$ corrupted by a uniform deviation of scale 5%.

The angle constraint errors (Fig. 2b) are decreasing linearly at a logarithmic scale. Both constraints are highly satisfied for large value of λ. One can observe that increasing λ by factor of 10 leads nearly to an accuracy improvement factor of 10 in the constraint. It is seen also that the least squares function converges to a stable value, whereas the constraint function decreases to zero at the end of the estimation (Fig. 2c)

5.2 The stepmodel object

This object contains sets of parallel planes. The prototype objects is composed of seven faces. We have studied the case when five faces are visible (Fig. 3a). For this view we have assigned a single normal for each set of parallel planes. By this way three normals $\vec{p_1}, \vec{p_2}, \vec{p_5}$ are associated respectively to surfaces (S_1, S_4), (S_2, S_3), and S_5. Besides the three angle constraints (orthogonality of each two vectors) and the three unit constraints, this object involves as well two distance constraints related to the fixed distances between (S_1, S_4) and (S_2, S_3). The surfaces' points

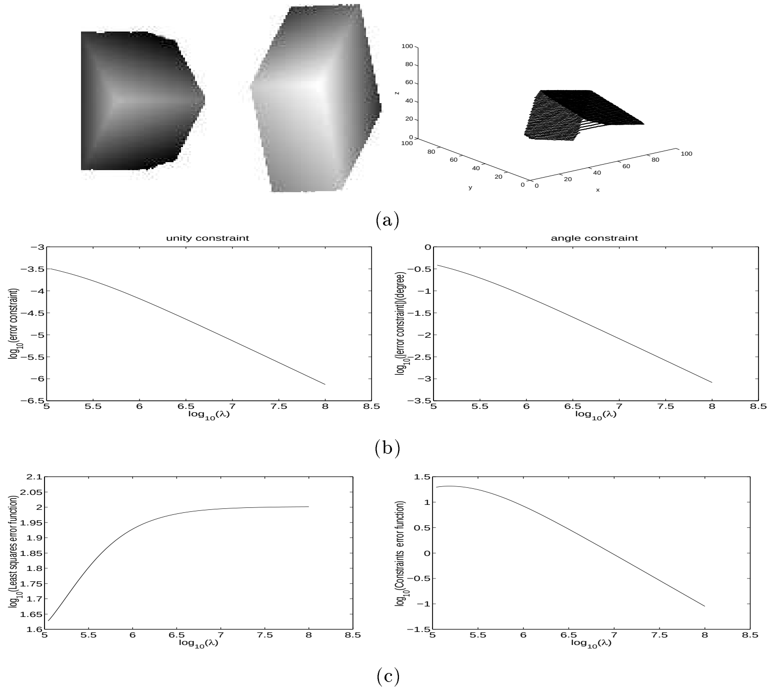


Figure 2: (a) Acquired and segmented real data. (b) decrease of the unit vector and the angle constraint error functions with respect to λ. (c) variation of the LS function and the constraint function with respect to λ.

have been corrupted with a Gaussian noise of $2mm$ variance. Using equations (5),(6) and (20) the least squares function is:

$$F(\vec{p}) = \vec{p}^T \mathcal{H} \vec{p} - 2\vec{h}^T \vec{p} + K, \tag{22}$$

$$\mathcal{H} = \begin{bmatrix} H_1 + H_4 + H_{14} & 0 & 0 \\ 0 & H_2 + H_3 + H_{23} & 0 \\ 0 & 0 & H_5 \end{bmatrix}, \quad \begin{array}{l} \vec{h}^T = [d_{14}\vec{h}_{14}^T, d_{23}\vec{h}_{23}^T, 0, 0, 0], \\ K = N_{14}d_{14}^2 + N_{23}d_{23}^2 \end{array}$$

The first series of tests have been carried out with the algorithm $optim1$ in which all the constraints are applied simultaneously. Some results are shown in Fig. 3(b,c).

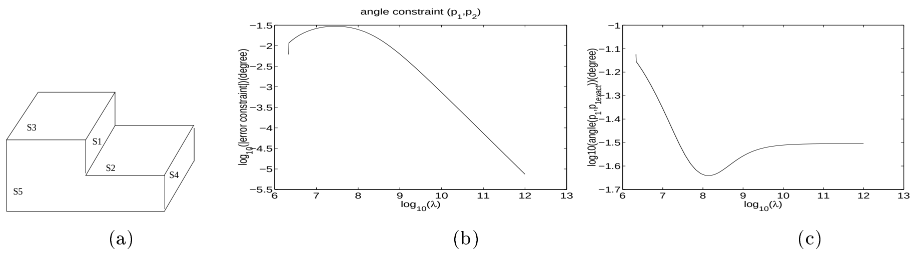


Figure 3: (a) the stepmodel object. (b) decrease of the constraint error function related to one plane normal. (c) orientation error related to one surface normal in function of λ.

In the second series of experiments, the algorithm $optim2$ was applied. According to this algorithm, the constraint function changes each time a new constraint

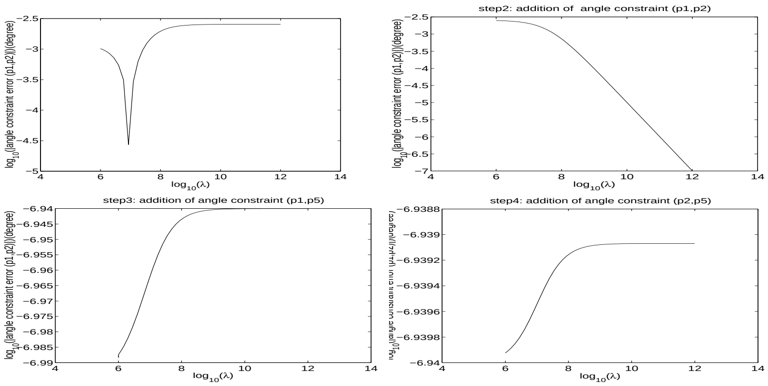


Figure 4: Variation of the angle constraint error related to $(\vec{p_1}, \vec{p_2})$ all along the four steps of the algorithm *optim*2.

is added. Normally the incremental process contains six steps, however since the unit constraints are used mainly to avoid the null solution there is no need to apply them incrementally, instead they are inferred at once simultaneously in a single step. Thus the algorithm will comprises four steps, in the first the unit constraints are considered, afterwards the three angle constraints are inferred one by one. Some results are illustrated in Fig. 4 and Fig. 5.

Results similar to the tetrahedron case were obtained in both algorithms for the unit constraint, the angle constraint, the least squares function and the constraint function. Comparison of Fig. 3b and Fig. 4 shows that the angle constraint is well satisfied in the two algorithms.

This synthetic example allows the comparison of the estimated surfaces' normals to the actual ones. Fig. 3c and Fig. 5 shows that the estimated vectors in each of the two algorithms are very close to the actual ones, however we observe that the normal orientation error is reduced by more than 100 in the second algorithm. This fact shows that the estimated solution moves toward the actual one, and it is almost completely reached. So we can say the optimization technique satisfies the constraints while improving the localization to a high degree.

6 Discussion and conclusion

The experiments presented in the previous section show that the incremental representation of constraints and parameter optimization search does produce shape fitting that satisfies the constraints with low error. The experiments also show that the least-square error grows as the constraints are applied; however, what is important is reconstructing shapes that satisfy the given constraints, while also binding the remaining unconstrainted shape parameters using the range information. The magnitude of the actual least-square error, even relative to the least-square error of the unconstrained fit, is unimportant relative to the constraint satisfaction. The amount of change in position of the constrained surfaces relative to the original position is similarly very irrelevant.

The option of adding the constraints incrementally has also been investigated. We have chosen to start from the previous optimal position when a new constraint

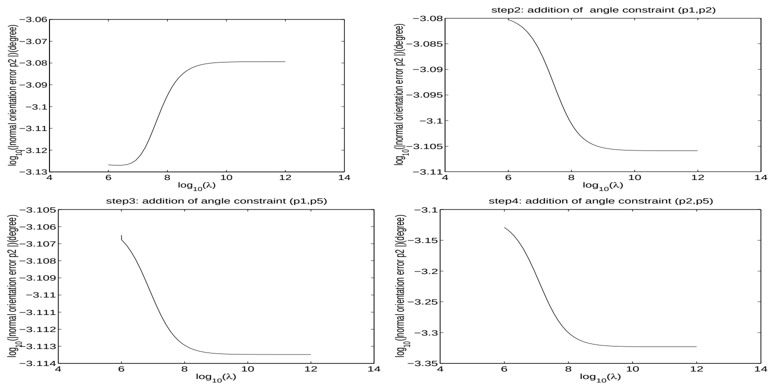


Figure 5: Variation of the orientation error related to ($\vec{p_2}$) along the four steps.

is added and to keep the weight of the previous constraints at the fixed maximum value of λ. The experiments confirmed that a previous constraint is almost not affected when a new constraint is added.

The optimization procedure used here produces solutions in a few minutes or less, which is suitable for CAD work.

The work here assumed that the range measurements were already segmented into groups associated with features. This is a reasonable assumption, but how to achieve this in difficult cases is an open problem.

Finally, real parts usually have more than just the constrained developable surfaces. The optimization procedure discussed above manipulates the constrained surface positions and shapes, but not the other surfaces. Consequently, a complete system would need to consider how to move and transform the other connected surfaces as the constrained features move.

Acknowledgements

The work presented in this paper was funded by UK EPSRC grant GR /H86986.

References

[1] A.Blake, A.Zisserman *Some properties of weak continuity constraints and the GNC algorithm.* CVPR, pp.656-661, 1986.

[2] K.L.Boyer, M.J.Mirza, G.Ganguly *The Robust Sequential Estimator.* IEEE Trans. PAMI, Vol.16, No.10, pp.987,1001 October 1994.

[3] Y.Chen, G.Medioni *Object Modeling by Registration of Multiple Range Image.* Proc. IEEE Int. Conf. Robotics and Automation, Vol.2 pp.724-729, April, 1991.

[4] J.De Geeter, H.V.Brussel, J.De Schutter, M. Decreton *Recognising and Locating Objects with an ultrason and infra-red sensor .* Proc. IMACS, pp.587-592, Lille, France, 1996.

[5] P.J.Flynn, A.K.Jain *Surface Classification: Hypothesizing and Parameter Estimation.* Proc. IEEE Comp. Soc. CVPR, pp. 261-267. June 1988.

[6] S.Kumar, S.Han, D.Goldgof, K.Boyer *On Recovering Hyperquadrics from Range data.* IEEE Trans. PAMI, Vol.17, No.11, pp.1079-1083 November 1995.

[7] S.L.S. Jacoby, J.S Kowalik, J.T.Pizzo *Iterative Methods for Nonlinear Optimization Problems.* Prentice-Hall,Inc. Englewood Cliffs, New Jersey, 1972.

[8] J.Porrill *Optimal Combination and Constraints for Geometrical Sensor data.* International Journal of Robotics Research, Vol.7, No.6, pp 66-78, 1988.

[9] M.Soucy, D.Laurendo *Surface Modeling from Dynamic Integration of Multiple Range Views.* Proc 11th Int. Conf. Pattern Recognition, pp.449-452, 1992.

[10] H.Y.Shun, K.Ikeuchi, R.Reddy *Principal Component Analysis with Missing Data and its Application to Polyhedral Object Modeling.* IEEE Trans. PAMI, Vol.17, No.9, pp.855-867.

[11] B.C.Vemuri, J.K Aggrawal. *3D Model Construction from Multiple Views Using Range and Intensity Data.* Proc. CVPR, pp.435-437, 1986.