Towards object modelling by incorporating geometric constraints

Naoufel Werghi Robert Fisher Craig Robertson Anthony Ashbrook
University of Edinburgh
Department of Artificial Intelligence
5 Forrest Hill, EH1 2QL Edinburgh, UK
naoufelw@dai.ed.ac.uk

Abstract

This paper describes a new technique for global shape improvement based upon features' positions and geometrical constraints. It suggests a general incremental, framework whereby constraints can be added and integrated in the model reconstruction process, resulting in an optimal trade-off between minimization of the shape fitting error and the constraints' tolerances. The objects treated are industrial parts containing planes and quadric surfaces.

1. Introduction

There has been a recent flurry of effort on reconstructing 3D geometric models of objects from single [3, 6, 8] or multiple [2, 4, 12, 11, 13] 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. However, an important aspect had not been fully investigated which is the exploitation of the geometric constraints defining the spatial or topological relationships between object features. Taking these constraints into account may be of great usefulness since it allows the reduction of the error in registration and mis-calibration effects. Consequently this leads to a more faithful model of the object.

The main problem encountered in this purpose is how to integrate the constraints in the shape fitting process. The problem is particularly crucial in the case of nonlinear constraints. In his pioneering work, Porrill [10] suggests a linearization of the nonlinear constraints and their combination with the Kalman filter equations. This method takes advantage of the recursive linear estimation of KF, but it guarantees satisfaction of the constraints only to linearized first order. Additional iterations are needed at each step if more accuracy is required. This last condition has been taken into account in the work of De Geeter *et al* [5] by defining a "Smoothly Constrained Kalman Filter". The key idea 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. However, the characteristics of Kalman filtering makes these methods essentially adapted for iteratively acquired data and many data samples. Moreover, there were no criteria mentioned in both of the above works for knowing to what level the constraints have been satisfied at the end of the algorithm.

In this paper we describe a new approach for global shape improvement based on feature position and shape constraints. This approach avoids the drawbacks of linearization techniques since the constraints are integrally implemented and is more general in scope and application. The key of 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. The constraints may 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.

This paper extends the work reported previously in [14] by adding new constraints, allowing one to specify the relationships between quadratic surfaces and planes and include such information in the optimization process.

2. 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.

A state vector $\vec{p}$ is associated to the object, which includes the set of parameters related to the patches. The vector $\vec{p}$ has to best fit the data while satisfying the constraints. 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 function:

$$E(\vec{p}) = F(\vec{p}) + C(\vec{p}) \tag{1}$$

where $F(\vec{p})$ is the objective function defining the relationship between the set of data and the parameters and $C(\vec{p})$ is a function representing the constraints. The objective function could be the likelihood of the range data given the parameters (with negative sign since we want to minimize) or the least squares error function. The likelihood function has the advantage of accounting for the statistical aspect of the measurements. In a first step, we have chosen the least squares function. The integration of the data noise characteristics in the LS function can be done afterwards with no particular difficulty, leading to the same estimation of the likelihood function in the case of the Gaussian distribution.

The LS function can be represented by :

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

where $\mathcal{H}$, $\vec{h}$ and K are respectively a matrix, a vector and constant depending on the data. Moreover, the matrix $\mathcal{H}$ is positive definite. More details about how such equation is obtained are expanded in [14].

Given M constraints, the constraint function is represented by the following equation:

$$C(\vec{p}) = \sum_{k=1}^{M} \lambda_k C_k(\vec{p}) \tag{3}$$

where λ_k are weighting coefficients (known in the literature as Lagrange multipliers) used to control the contribution of the constraints in the parameters' estimation.

2.1. Optimization of shape satisfying the constraints

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 we can show easily that the matrix $\mathcal{H}$ is positive definite.

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 [9].

In some particular cases it is possible to get a closed form solution for (1) (the eigenvalues methods). This depends on the characteristics of the constraint functions and whether it is possible to combine them efficiently with the objective function. 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 investigating, whether it is possible to get a solution that satisfies a specified tolerance. So the objective is to determine the vector $\vec{p}$ which satisfies the constraints to the desired specification and which fits the data to a reasonable degree.

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

$$E = \vec{p}^T \mathcal{H} \vec{p} + \vec{h}^T \vec{p} + K + \lambda \sum_{k=1}^{M} C_k(\vec{p}), \quad \lambda \geq 0 \tag{4}$$

As λ will be driven (effectively) to $+\infty$, this will force all constraints to be satisfied. Since we are not placing a relative importance on the different constraints, it is then convenient to just use a single λ.

The problem now is how to find $\vec{p}$ that minimizes (4). Because (4) 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 (4) for small λ, in which the constraints are weakly expressed. Then we iteratively increase λ slightly and solve for a new optimal parameter $\vec{p}^{[n+1]}$ using the previous $\vec{p}^{[n]}$. 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 becomes 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 algorithm is illustrated in Figure 1.a.

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 sep-

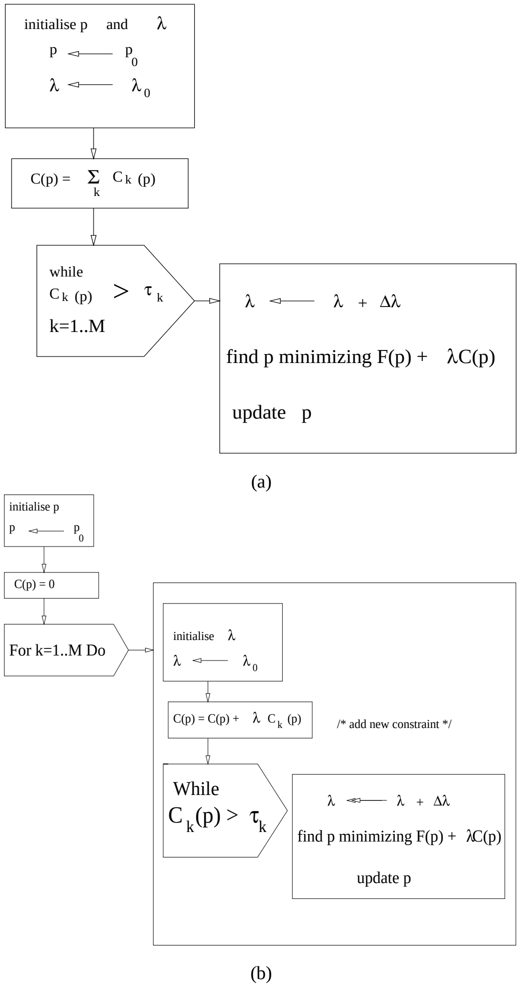


Figure 1. (a): $optim1$ - batch constraint optimization algorithm. (b): $optim2$ - sequential constraint introduction optimization algorithm.

arately 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 to avoid the above trivial solution and to give the constraints a certain weight. A convenient value for the initial λ is :

$$\lambda_0 = \frac{F(\vec{p}^{[0]})}{C(\vec{p}^{[0]})} \tag{5}$$

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 1.b. 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. We have experimented with both algorithms and found that the first "batch" approach to be very slightly less accurate but faster.

3. Object shape constraints

Polyhedral objects involve 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. These constraints are discussed in Sections 3.1 and 3.2. Quadratic surfaces allow additional constraints, of which the simple ones: 1)fixed angle between quadratic surfaces' axes; 2) fixed angle between quadratic surface axis and plane normal. Some of these constraints are addressed in Section 4.3. Other constraints can be defined as well for the quadratic surfaces such as the variety related to the common case of surfaces of revolution. We do not explicitly consider these constraints in this paper but the representation given in section 2 can be applied to them as well.

3.1. Planes with a fixed orientation relationship

A plane surface can be represented by this equation:

$$p_x x + p_y y + p_z z + d = 0 \tag{6}$$

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

$$p_x x + p_y y + p_z z = 0 \tag{7}$$

Consider N planes, where the angles between planes' normals are known. $(N-1) * N/2$ angle relationships can be defined between the planes. The orientation relationship between the different planes and the unity constraint of each normal can be written under a matrix formulation:

$$A_k(\vec{p}) = (\vec{p}^T \mathcal{A}_k \vec{p} - 2cos(\alpha_k))^2 = 0,\ k \in [1..(N-1) \times N/2] \tag{8}$$

$$U_i(\vec{p}) = (\vec{p}^T \mathcal{U}_i \vec{p} - 1)^2 = 0,\ i \in [1..N] \tag{9}$$

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

$$\mathcal{U}_i = \begin{bmatrix} 0_3 & . & 0_3 \\ 0_3 & (I_3)_{ii} & 0_3 \\ 0_3 & . & 0_3 \\ 0_3 & . & 0_3 \end{bmatrix}, \ \mathcal{A}_k = \begin{bmatrix} 0_3 & . & 0_3 \\ . & . & (I_3)_{ij} \\ 0_3 & (I_3)_{ji} & 0_3 \\ . & . & 0_3 \end{bmatrix}$$

and I_3 and 0_3 are the 3×3 identity and null matrices.

We noticed here that although the use of matrices may be computationally expensive, the matrix notation leads to a compact form and avoid expressions with many variables. This allows a fast and automatic implementation of the constraints.

3.2. 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 observed data points and separated by the specified geometric distance d_{pq}. Since the two planes have a common orientation, a single normal $\vec{n_{pq}}$ can be associated with them. Each pair of points $(X_i^p, X_j^q),\ \ X_i^p \in S_p,\ X_j^q \in S_q$ has to satisfy the following equation:

$$(X_i^p - X_j^q)^T \vec{n_{pq}} - d_{pq} = 0 \tag{10}$$

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

$$\sum_{i,j}^{N_p,N_q} ((X_i^p - X_j^q)^T \vec{p} - d_{pq})^2 = \vec{p}^T H_{pq} \vec{p} - 2 d_{pq} \vec{h}_{pq}^T \vec{p} + N_{pq} d_{pq}^2 \tag{11}$$

where

$$\begin{aligned} H_{pq} &= \sum_{i,j}^{N_p,N_q} (X_i^p - X_j^q)(X_i^p - X_j^q)^T \\ \vec{h}_{pq}^T &= \sum_{i,j}^{N_p,N_q} X_i^p - X_j^q = N_q \sum_i^{N_p} X_i^p - N_p \sum_j^{N_q} X_j^q \\ N_{pq} &= N_p N_q \end{aligned}$$

Here, the distance constraint is considered as a part of the objective equation since it involves both data and parameters.

3.3. Constraints on quadric surfaces

A general quadric surface is represented by the general quadratic equation:

$$ax^2 + by^2 + cz^2 + 2hxy + 2gxz + 2fyz + 2ux + 2vy + 2wz + d = 0 \tag{12}$$

The above mentioned types of constraints can be extended for quadric surfaces. However the implicit equation (12) is not sufficient for extracting a concrete formulation of such constraints. Other representations involving geometric properties of the quadric (centre, axis orientation, etc.) are more appropriate for this purpose. An example related to the cylindrical surface case will be studied in section 4.3

4. Experiments

A first series of tests was applied on a synthetic object, a step model object (Fig.2). The aim was to check the behaviour and the convergence of algorithm and compare the performances of the *optim1* and *optim2* algorithms .

The second series of experiments was carried out with two real objects, a tetrahedron (Fig.5) and half cylinder (Fig.6). The data was acquired with a 3D triangulation range sensor. The range measurements were already segmented into groups associated with features by means of the *rangeseg* [7] program.

4.1. The step model object

This object contains sets of parallel planes. The prototype object is composed of seven faces. We have studied the case when five faces are visible (Fig. 2.a). For this view we assigned a single normal for each set of parallel planes. 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 vector 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 were corrupted with Gaussian noise of $2mm$ standard deviation. Using equations (2) and (11) the least squares function is:

$$F(\vec{p}) = \vec{p}^T \mathcal{H} \vec{p} - 2 \vec{h}^T \vec{p} + K, \tag{13}$$

where $\mathcal{H} = \begin{bmatrix} H_1 + H_4 + H_{14} & 0 & 0 \\ 0 & H_2 + H_3 + H_{23} & 0 \\ 0 & 0 & H_5 \end{bmatrix}$

and $\vec{h}^T = [d_{14} \vec{h}_{14}^T, d_{23} \vec{h}_{23}^T, 0, 0, 0]$, $K = N_{14} d_{14}^2 + N_{23} d_{23}^2$

The first series of tests were carried out with the algorithm $optim1$ in which all the constraints were applied simultaneously. Some results are shown in Fig. 2. The unit constraint function (b), the angle constraint function (c) and the estimation error of one normal (d) are mapped as a function of λ. These figures describe the behaviour of the constraint functions and the estimation errors. The unit error constraint and the angle error constraints are decreasing linearly at a logarithmic scale (b and c). It can be observed that increasing λ by factor of 10 leads to an accuracy improvement factor of almost 10 in the constraint. Thus it is possible to enforce the constraint to any level of tolerance until the numerical accuracy of the algorithm is compromised. The orientation error related to the surface normal $\vec{p_1}$ is also decreasing to a low value. The results show that the position of the optimal $\vec{p_1}$ stabilizes near the actual one. Similar behaviour is observed for the other parameter vectors. This is encouraging because it means that the part's shape and position stabilizes as a whole.

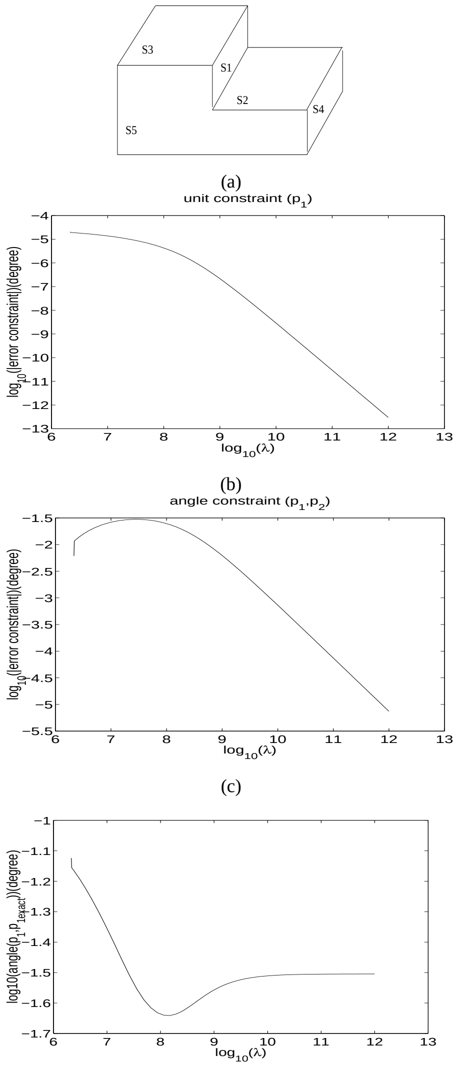


Figure 2. (a) the step model object. (b) and (c) variation of respectively the unit constraint error and the angle constraint error ($\vec{p_1}, \vec{p_2}$) in function of λ. (d) variation of estimated orientation error of the normal $\vec{p_1}$

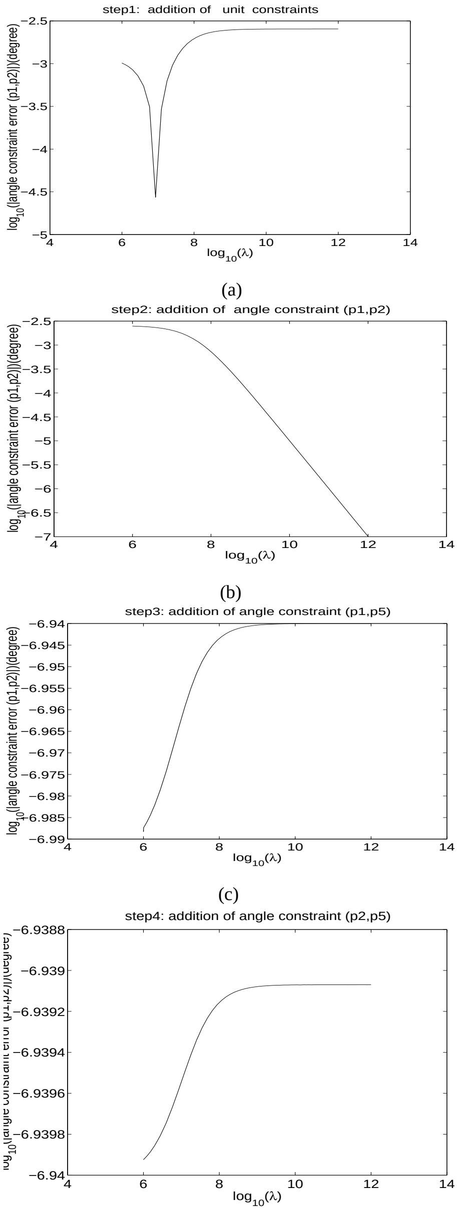
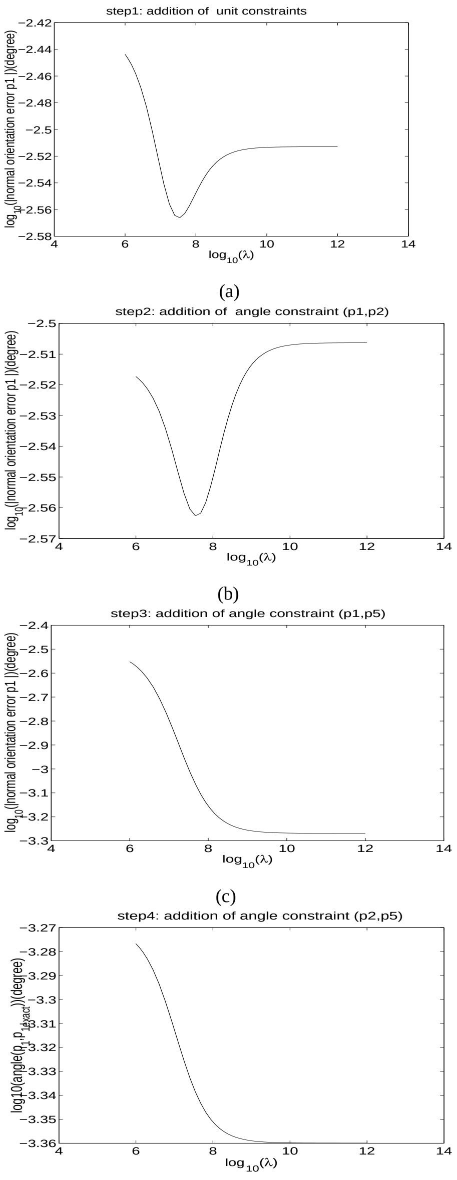


Figure 3. Variation of the angle constraint error related to $(\vec{p_1}, \vec{p_2})$ at the four steps of the algorithm $optim2$.



(a)

(b)

(c)

(d)

Figure 4. Variation of the orientation error related to $(\vec{p_1})$ at the four steps of the algorithm optim2.

In the second series of experiments, the algorithm $optim2$ was applied. In this algorithm the constraint function changes each time a new constraint 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 comprises four steps, in the first the unit constraints are considered then the three angle constraints are applied one by one. Some results are illustrated in Fig.3 and Fig.4 which show respectively the variation of one angle constraint function and the variation of the estimation error of one normal along the four steps of the algorithm. Similar results are obtained for the other parameter vectors.

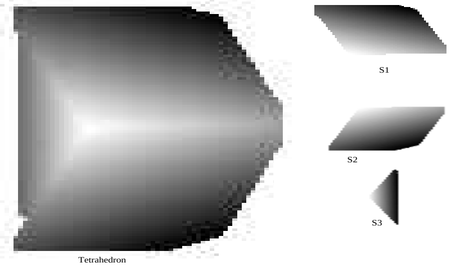
Fig.3 shows in particular the significant decrease of the constraint error (b:step 2) when the related constraint function is inferred. Moreover it shows that once the constraint is satisfied (b: end of the step 2) the addition of the other constraints (c:step 3) and (d:step 4) only affect the level of tolerance previously reached to a very small degree. Fig.4 also shows the convergence of the estimated $\vec{p_1}$ towards the actual one, moreover we observe that the final position is better than the one reached in $optim1$ by a factor of 100 (compare Fig.2.d and Fig.4.d). The unit constraints and the angle constraints are highly satisfied in both algorithms, *optim1* and *optim2* .

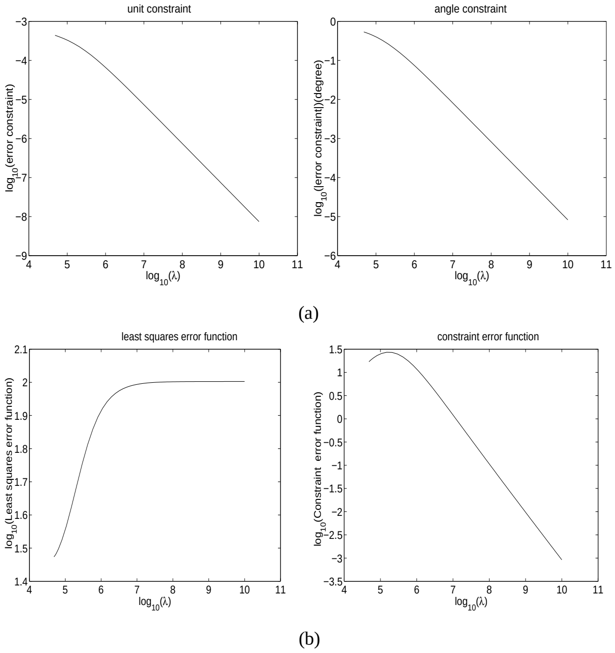
4.2. The tetrahedron

Figure 5 shows a real tetrahedron image with three faces visible. This object involves three angle 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 application of the paradigm developed in Section 3.1 is immediate for this object. We get thus easily the following error function:

$$E(\vec{p}) = \vec{p}^T \mathcal{H}\vec{p} + \lambda(\sum_{k=1}^{3} A_k(\vec{p}) + \sum_{l=1}^{3} U_l(\vec{p})) \qquad (14)$$

All constraints were applied simultaneously according to algorithm $optim1$. The results are the average of 100 trials, with the initial vector $\vec{p}^{[0]}$ corrupted by a uniform deviation of scale 5%. Fig.5a shows the decrease of the unit constraint error and the angle constraint errors. The final errors are small. In Fig.5.b, it is seen also that the least squares function converges to a stable value, and the constraint function decreases to zero at the end of the estimation. Thus, the final part shape now satisfies the shape constraints at a slight increase in the least squares fitting error.





(a)

(b)

Figure 5. (a) decrease of the unit vector and the angle constraint error functions with respect to λ. (b) variation of the LS function and the constraint function with respect to λ.

4.3. The half cylinder object

This object is composed of four surfaces. Three patches S_1, S_2 and S_3 have been extracted from the view represented in Figure.6. They correspond respectively to the base plane S_2 , lateral plane S_1 and the cylindrical surface S_3 . The least squares error function is given by:

$$F(\vec{p}) = \vec{p}^t H \vec{p} \tag{15}$$

where $H = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ 0 & 0 & H_3 \end{bmatrix}$, $\vec{p} = [\vec{p_1}, \vec{p_2}, \vec{p_3}]$

$\vec{p_1}$ and $\vec{p_2}$ involve two unit constraints as well as an angle constraint:

$$U_1(\vec{p}) = (\vec{p}^T \mathcal{U}_1 \vec{p} - 1)^2 \quad = \quad 0 \tag{16}$$

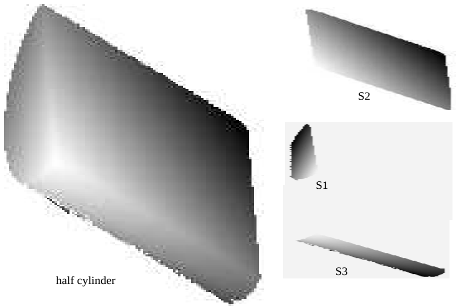


Figure 6. A single view of the cylinder object and the extracted surfaces.

$$U_2(\vec{p}) = (\vec{p}^T \mathcal{U}_2 \vec{p} - 1)^2 \quad = \quad 0 \tag{17}$$

$$A(p) = (\vec{p}^T \mathcal{A} \vec{p} - 2cos(\pi/2))^2 \quad = \quad 0 \tag{18}$$

where

$$\mathcal{U}_1 = \begin{bmatrix} I_3 & O_3 & O_{3,10} \\ O_3 & O_3 & O_{3,10} \\ O_{10,3} & O_{10,3} & O_{10,10} \end{bmatrix} \quad \mathcal{U}_2 = \begin{bmatrix} O_3 & O_3 & O_{3,10} \\ O_3 & I_3 & O_{3,10} \\ O_{10,3} & O_{10,3} & O_{10,10} \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} O_3 & I_3 & O_{3,10} \\ I_3 & O_3 & O_{3,10} \\ O_{10,3} & O_{10,3} & O_{10,10} \end{bmatrix}$$

The cylinder axis is collinear to S_1's normal $\vec{p_1}$ and orthogonal to S_2's normal $\vec{p_2}$. Thus the vector $\vec{p_1}$ can be associated with the cylinder axis orientation. The exploitation of the above constraint needs to use the following form of a cylinder equation:

$$(x-x_0)^2+(y-y_0)^2+(y-y_0)^2-(n_x(x-x_0)+n_y(y-y_0)+n_z(z-z_o))^2-r^2 = 0 \tag{19}$$

where (x_0, y_0, z_0) is an arbitrary point on the axis, (n_x, n_z, n_y) is a unit vector along the axis and r is the radius of the cylinder. Consequently this vector can be $\vec{p_1}$.
The expansion of (19) and the identification with (12) yields

$$\begin{aligned}
a &= 1 - n_x^2 && (20)\\
b &= 1 - n_y^2 && (21)\\
c &= 1 - n_z^2 && (22)\\
h &= -n_x n_y && (23)\\
g &= -n_x n_z && (24)\\
f &= -n_y n_z && (25)\\
u &= -((n_x x_0 + n_y y_0 + n_z z_0)n_x + x_0) && (26)\\
v &= -((n_x x_0 + n_y y_0 + n_z z_0)n_y + y_0) && (27)\\
w &= -((n_x x_0 + n_y y_0 + n_z z_0)n_z + z_0) && (28)\\
d &= x_0^2 + y_0^2 + z_0^2 - (n_x x_0 + n_y y_0 + n_z z_0)^2 - r^2 && (29)
\end{aligned}$$

The unit constraint on the vector $\vec{p_1}$ leads to:

$$a + b + c = 2$$

whose impact on the vector $\vec{p}^T = [\vec{p_1}^T, \vec{p_2}^T, \vec{p_3}^T]$ can be represented by the following constraint equation:

$$C_1(\vec{p}) = (i_1^T \vec{p} - 2)^2 = 0 \tag{30}$$

where $i_1 = [0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0]^T$
The summation of the equations (23), (24) and (25) gives:

$$f + g + h = -(n_x n_y + n_x n_z + n_y n_z)$$

which leads to the following constraint :

$$C_2(\vec{p}) = (i_2^T \vec{p} + \vec{p}^T Q \vec{p})^2 = 0 \tag{31}$$

where

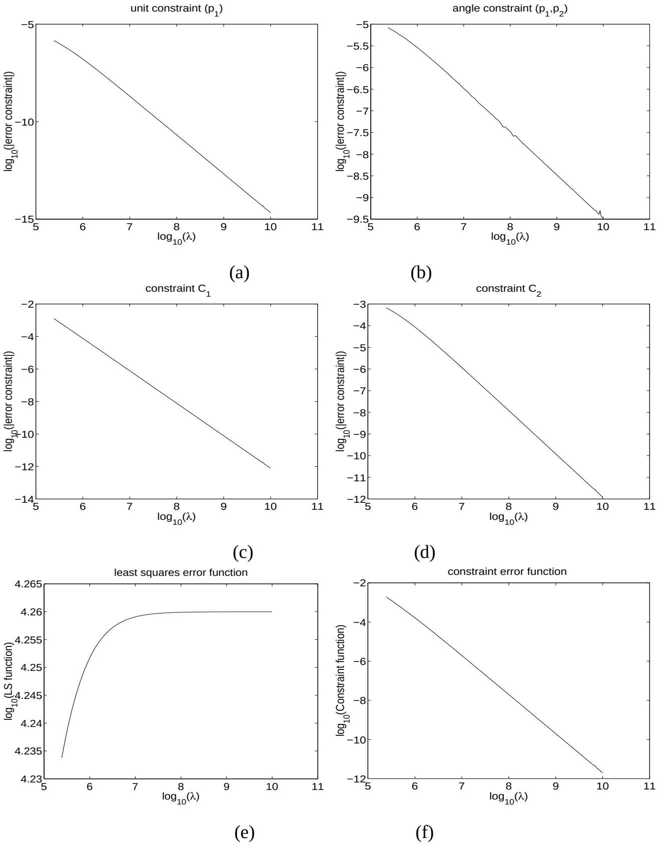
$$\begin{aligned} i_2 &= [0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0]^T & (32) \\ Q &= \begin{bmatrix} R & O_{3,3} & O_{3,10} \\ O_{3,3} & O_{3,3} & O_{3,10} \\ O_{10,3} & O_{10,3} & O_{10,10} \end{bmatrix} & (33) \\ R &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} & (34) \end{aligned}$$

Considering all the above, the optimization function is

$$E(\vec{p}) = \vec{p}^T \mathcal{H} \vec{p} + \lambda(U_1(\vec{p}) + U_2(\vec{p}) + A(\vec{p}) + C_1(\vec{p}) + C_2(\vec{p})) \tag{35}$$

The algorithm $optim1$ has also been used for this object. Results are shown in Figure.4.3. (a) and (b) show the decrease of respectively one unit constraint and the angle constraint. The constraints C_1 and C_2 ((c) and (d)) are decreasing with the same slope. All the constraints have been satisfied to a high degree at the end of the optimization.

The least squares error converges to a stable value (e) and the constraint error function is practically zero at the end of the algorithm (f).



(a) (b)

(c) (d)

(e) (f)

Figure 7. (a,b,c,d) decrease of the constraints' errors as a function of λ. Variation of least squares error function (e) and the constraint error function (f) with respect to λ.

5. 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 slightly as the constraints are applied. However, what is important is reconstructing shapes that satisfy the given constraints while also binding the remaining unconstrained shape parameters using the range information. We have also to bear in mind that the weakness of the least-squares residuals value may not reflects a good estimation in the case when the measurement errors are systematic (e.g. registration errors, miscalibration errors). The magnitude of the actual least-square error, even relative to the least-squares error of the unconstrained fit, is unimportant relative to the constraint satisfaction. The small amount of change in position of the constrained surfaces relative to the original position is similarly irrelevant.

The option of adding the constraints incrementally was also investigated. We have chosen to start from the previous optimal position when a new constraint 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 unaffected when a new constraint is added.

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

Regarding the improvement the optimization technique, we are investigating a reformulation of the constraints that might allow an exact solution using the Lagrange multipliers. One issue we are concerned with is the large number of constraints in a typical 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. The work presented in this paper can be considered as a first step toward a general methodology for object modelling integrating the constraints. Future work will consider more complicated object shapes including different types of quadric surfaces. [1]

References

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

[2] R.M.Bolle, D.B.Cooper *On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data.* IEEE Trans. PAMI, Vol.8, No.5, pp.619,638, September 1986.

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

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

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

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

[7] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, R. Fisher *An Experimental Comparison of Range Segmentation Algorithms.* IEEE Trans. PAMI, Vol.18, No.7, pp.673-689, July 1996.

[8] 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.

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

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

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

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

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

[14] N.Werghi, R.B.Fisher, A.Ashbrook, C.Robertson *Improving Model Shape Acquisition by Incorporating Geometric Constraints.* in Proc. BMVC, pp.530-539, Essex, september 1997.

[1] Acknowledgements: the work presented in this paper was funded by UK EPSRC grant GR /L25110.