CONSTRAINED FITTING — A KEY ISSUE IN REVERSE ENGINEERING CONVENTIONAL PARTS

Pál Benkő
Tamás Várady
Computer and Automation Research Institute, Budapest
benko@sztaki.hu

Abstract Constrained fitting is the process of approximating segmented point sets simultaneously by multiple surfaces while certain geometric constraints, such as tangency, perpendicularity, parallelism, concentricity are satisfied. This technique is particularly important in the context of reverse engineering, where geometrically and topologically consistent, near to perfect CAD models must be created. In this paper various representational and numerical problems are discussed in order to make constrained fitting computationally efficient even for large, multiple point clouds. Special emphasis is taken to resolve contradicting constraints. Simple examples of handling smooth profile curves and tangentially connected face sets are also presented.

Keywords: reverse engineering, analytic surface fitting, constraints, minimisation

Introduction

Reverse engineering is the process of converting 3D, multiple view measured data into an evaluated boundary representation model [9]. There are many publications concerning the reconstruction of geometrically demanding free-form objects [4], but in this paper we concentrate on topologically complex solid models bounded by simple analytic surfaces, such as planes, cylinders, cones, spheres, tori and small blending surfaces [5]. While in many graphical and computer vision applications approximate models are sufficient, our main interest is to create such boundary representation models, which can directly be used in commercial CAD/CAM systems through standard data exchange interfaces, such as IGES or STEP. In this way, the benefits of using the exist-

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35492-7_50
G. L. Kovács et al. (eds.), *Digital Enterprise Challenges*

ing CAD/CAM technology can be directly applied for the reconstructed objects as well, including (re)design, analysis, simulation and NC manufacture.

The reverse engineering procedure can be decomposed into several phases, including data acquisition (typically by laser scanners), merging multiple point clouds, triangulating and decimating data sets, segmenting a point cloud, fitting surfaces for individual regions, building B-rep models and reconstructing blends. As it has been analysed in [1], an ideal reconstructed model must satisfy several requirements in order to be usable in an engineering environment. The bounding surfaces must be accurate, their surface type and the related parameters must be reliably estimated. Complex linear extrusion surfaces and surfaces of revolution must be recognised, and represented as special, smoothly connected surface elements.

For B-rep model building *smooth edges* must be explicitly detected and reconstructed, otherwise not only imperfect models are generated, but the reconstruction process may crash at the intersection of nearly tangential surfaces. In various cases, such as vertices with more than three valencies, special *topological constraints* need to be satisfied. Difficulties emerge partly due to noise, which is always present in the measured data, partly because the estimation of the various geometric entities, which also carry certain errors. The final model must be consistent from both topological and geometric points of view; this is why a higher level intelligence is necessary to incorporate certain 'likely' constraints into the reconstructed model. A final group of requirements concerns various — local or global — engineering constraints [3, 6], such as enforcing parallelism, perpendicularity, tangency, concentricity, or symmetry amongst various geometric entities.

In this paper the concept of constrained fitting is introduced, which is a technique to overcome the above mentioned difficulties. While there is a vast literature on constraints in solid modeling and on individual surface fitting, there is only one project known to the authors, where these two areas have been coupled for reverse engineering applications — see [10]. The novel method presented here is computationally efficient and capable of *resolving conflicts* between constraints according to a given priority. Computational considerations of applying so-called *auxiliary elements*, and speeding up the computation by separating the terms holding the point data and the surface parameters are also discussed. Finally, several application examples are given, such as, the reconstruction of smooth, constrained profile curves, the decomposition of smooth, multiple surface regions and enforcing various types of general constraints.

1. Mathematics of constrained fitting

Given a set S of curves and surfaces, and for each $s \in S$ a point set P_s to be approximated. Each entity is characterised by a hypothetical curve or surface type and unknown parameters a_s, which are concatenated in a vector a. There is a given set of constraints in the form of $c_i(a) = 0$ what we want to satisfy. The constraints are specified by the user or derived by the reverse engineering system itself. Our goal is to approximate simultaneously several surface elements while the constraints are also satisfied.

The problem can be stated as minimising a function $g : \mathbb{R}^n \to \mathbb{R}$ on the zero set of another function $c : \mathbb{R}^n \to \mathbb{R}^m$ (the concatenation of c_i's). In our case g is formulated as

$$g(a) = \sum_{s \in S} \alpha_s \sum_{p \in P_s} f(p, a_s)^2,$$

where $\alpha_s \in \mathbb{R}$ is a weighting term and $f(p, a_s)$ approximates the distance of point p from surface s.

The unconstrained minimisation by Newton-Raphson iteration would go by picking a starting point, and computing a step from the second order Taylor-approximation at the current point. Since g is a sum of squares, this approximation is positive semi-definite. The usual method for marching on a given surface is stepping on the tangent plane and projecting the result to the surface. Projection to the implicit surface $c^{-1}(0)$ can be solved again by Newton-Raphson iteration using the first order Taylor-approximation of c at the current point. To avoid double iteration, these methods are combined in our constrained minimisation scheme (as described in details in [2]). In each iteration step of the minimisation just the first step of the projection iteration is made; an approximation of the tangent plane is computed, and the minimisation step is performed on it. Computing the tangent plane approximation is done sequentially, by traversing all constraints. In this phase it is recognised whether a constraint follows from the previous ones or contradicts them. In these cases the constraint is ignored: in the former case nothing needs to be done, in the latter case nothing can be done.

At the end of the iteration the result is on the constraint surface, and the gradient of g is orthogonal to it. Initial values for the iteration are generally obtained from independent unconstrained fits.

1.1. Object representation

Regular surfaces are represented in implicit form: $\{p \in \mathbb{R}^3 : f(p, a_s) = 0\}$. a_s contains a well-defined set of variables, which uniquely define the

surface, such as dimensions, positions, radii, angles, etc. Such a description can be used for minimising $\sum_{p\in P} f(p,a_s)^2$. In order to get a meaningful result, such descriptions are requested, which well approximate the Euclidean distance $d(p,a_s)$. A useful notion is the *faithful* approximation of the Euclidean distance [7]: f approximates d faithfully iff for any fixed parameter vector a_s they are equal up to order one on the surface described by f.

During the minimisation of $\sum_{p\in P} f(p,a_s)^2$ the set of points P is fixed and a_s is modified in each iteration step. To make the iteration *efficient*, f is searched in a special form. When the sum of squares is computed, P should not be 'traversed' in each iteration step, but some quantities (depending on P) should be computed in advance in a preprocessing phase, thus the sum can be calculated in constant time. This is possible if f is given in the form $f_1(p)^T f_2(a_s)$ (f_1 and f_2 are vector valued functions of the same dimension); in this case

$$\begin{aligned}\sum_{p\in P} f(p,a_s)^2 &= \sum_{p\in P} f_2(a_s)^T f_1(p) f_1(p)^T f_2(a_s)\\ &= f_2(a_s)^T \left(\sum_{p\in P} f_1(p) f_1(p)^T\right) f_2(a_s) = f_2(a_s)^T A f_2(a_s).\end{aligned}$$

For a Newton-Raphson iteration, the first two derivatives of f must be computed. If f is in this form, then

$$\begin{aligned}\sum_{p\in P} f'(p,a_s) &= 2 f_2(a_s) A f_2'(a_s)\\ \sum_{p\in P} f''(p,a_s) &= 2\left(f_2'(a_s) A f_2'(a_s) + f_2(a_s) A f''(a_s)\right).\end{aligned}$$

These formulae may still be difficult to compute, so a form where f_2 is as simple as possible is requested. A simple solution is to use another description vector $a_s' = f_2(a_s)$. In that case $f_2(a_s') = a_s'$, and the derivative formulae become really simple:

$$\begin{aligned}\sum_{p\in P} f'(p,a_s') &= 2Aa_s'\\ \sum_{p\in P} f''(p,a_s') &= 2A.\end{aligned}$$

Note that the above transformations must be performed with care: if the dimension of a_s' is much greater than that of a_s, then the size of the system grows, which of course makes computation slower; moreover, it

may be difficult to compute the new constraints which must have been introduced.

Later, simple examples will be given to show how efficient representations can be created.

1.2. Auxiliary objects

Certain constraints can be described just in a very roundabout way, but can be naturally decomposed into simpler constraints by introducing an *auxiliary* object: this is an artificial object not present in the original problem formulation, and it does not need to approximate data points. For example, the constraint 'two cylinders, a cone and a torus go through a common point' may be needed. It can be circumvented by introducing a *point* object and four simple constraints that each surface goes through that point. Coplanarity of some points can be described by introducing an auxiliary plane and prescribing the points to lie on it. The main difficulty with auxiliary objects is that as they do not fit onto data points they cannot be initialised by unconstrained fitting.

1.3. Simple objects and constraints

Lack of space prevents us to present all 2D and 3D object descriptions and the related constraint definitions (see [2]). Nevertheless, it was felt important to provide simple examples to show how efficient and faithful representations can be formulated. For simplicity's sake, here only 2D lines and circles are described, however, similar formulations can be found after some algebra for the 3D surface elements as well.

The Euclidean distance for lines can be given in an efficient form: if the unit normal is denoted by n and the signed distance from the origin by δ, the signed Euclidean distance of a point p is $\langle p \mid n \rangle + \delta$, which is in the desired form, choosing $f_1(p) = (p_x, p_y, 1)$ and $f_2(a_s) = (n_x, n_y, \delta)$. The normalisation condition $n^2 = 1$ completes the description.

The Euclidean distance for circles (spheres) cannot be written in an efficient form, but a faithful approximation given in [8] can be:

$$\begin{aligned} |p-o| - r &= \frac{(p-o)^2 - r^2}{|p-o| + r} \approx \frac{(p-o)^2 - r^2}{2r} \\ &= \frac{p^2}{2r} - \frac{\langle p \mid o \rangle}{r} + \frac{o^2 - r^2}{2r}; \end{aligned}$$

we obtain the desired form by choosing

$$\begin{aligned} f_1(p) &= (p^2, p_x, p_y, 1), \\ f_2(a_s) &= (\frac{1}{2r}, -\frac{o_x}{r}, -\frac{o_y}{r}, \frac{o^2 - r^2}{2r}). \end{aligned}$$

This is still a bit complicated because of the divisions, however, by introducing a new description vector we can get rid of them. $a'_s = f_2(a_s) = (\kappa, o'_x, o'_y, \mu)$. Now $f_2(a'_s) = a'_s$, so computing $\sum f$ and its derivatives is as simple as for a plane. Because the dimension of a'_s is greater by one than that of a_s, a further constraint is needed, which can be derived from the formulae in $f_2(a_s)$:

$$o'^2 - 4\kappa\mu = 1.$$

This description has the advantage that straight lines are just special (and not singular) cases of circles: instead of r being ∞, $\kappa = 0$. In that case o' is the normal of the line, μ is δ, and the normalisation condition is the same as that for lines. From that description the radius and the centre can be computed easily: $r = 1/2\kappa$, $o = -o'/2\kappa$. These formulas are repeatedly used for setting up the constraint equations. For example, the constraint that a line crosses the centre of a circle is

$$\begin{aligned} \langle o \mid n \rangle + \delta = \left\langle \frac{-o'}{2\kappa} \mid n \right\rangle + \delta &= 0 \\ \langle o' \mid n \rangle - 2\kappa\delta &= 0. \end{aligned}$$

As mentioned before, there are many more object types and constraints in 3D, these are considerably more complex than the planar entities. The complexity is increased by the fact that cones and tori cannot be represented in a form which is efficient *and* faithful. In [2] we gave both a faithful representation and an efficient representation for these types.

2. Examples

In this section a few examples are presented to illustrate the importance of constrained fitting in the course of reverse engineering CAD models.

2.1. Contradictory constraints

This example shows the usage of auxiliary objects and demonstrates how the final solution depends on the order (priority) of the constraints when they contradict each other.

The example consists of a square and three circles in its interior. The constraints are the following:

- the circles have equal radii
- the centres of the circles are collinear, and their line is aligned horizontal

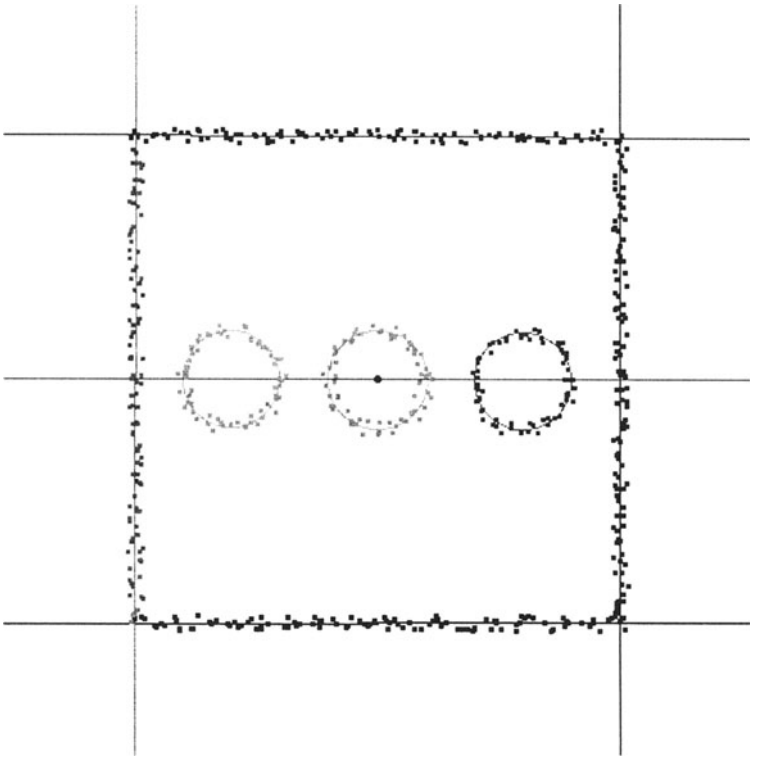

Figure 1. Initial state

- the centres subdivide the width of the square at equal distances of 30 units
- the length of the edge of the square is 100 units

To describe the collinearity of the centres, an auxiliary line object is introduced, and all centres are constrained to lie on it. The initial state does not depend on the constraints, so it is the same for all possible sequences of the constraints; as it can be seen in figure 1.

If the constraints are given in the sequence above, the last constraint will not be satisfied and the length will be 120 units, as shown in figure 2. If the 'centre spacing is 30 units' constraint is given last and ignored, then the result is figure 3. If the 'line of centres is aligned' constraint is given last, the result is shown in figure 4. If the 'centres are collinear' constraint is given last, the result is figure 5.

In practical reverse engineering the number of constraints may be much higher than those of the degrees of freedom, so the well assigned priorities are particularly important to get the right results.

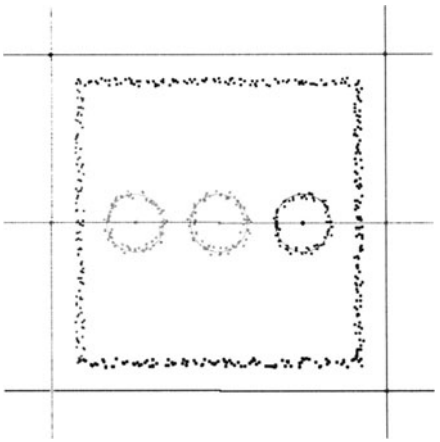

Figure 2. length of square edge not 100

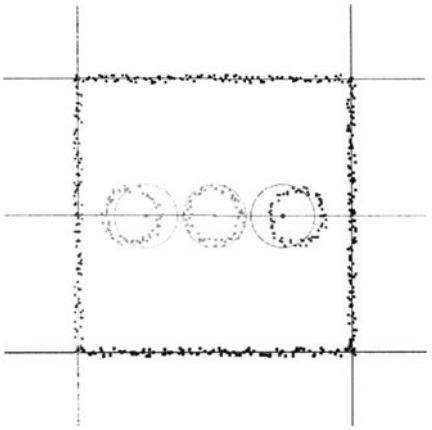

Figure 3. spacing of centres not 30

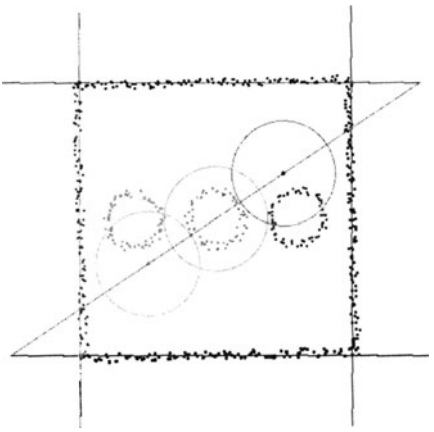

Figure 4. line of centres not aligned

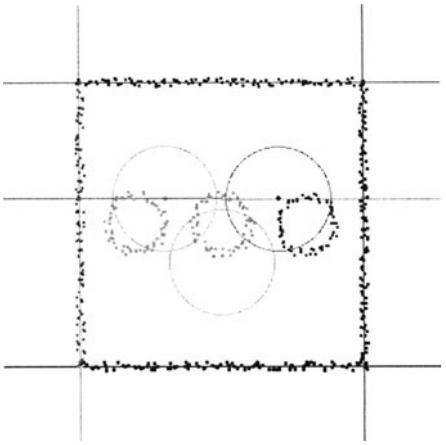

Figure 5. centres not collinear

2.2. Profile fitting

A very important subproblem in reverse engineering is the reconstruction of linear extrusions and surfaces of revolution. Both involve fitting a profile curve made up of smooth straight line - circular arc sequence to approximate a noisy and 'thick' planar point set. Thickness originates not only from the noise, but also from the inaccurate estimation of the best translational direction or the best rotational axis. Assume the point set is segmented; then circles and straight lines are fitted to the individual segments while adjacent elements are *constrained* to be

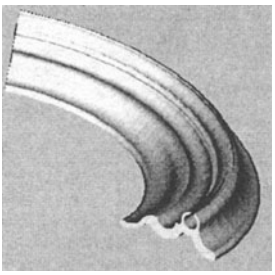

Figure 6. Rotational object

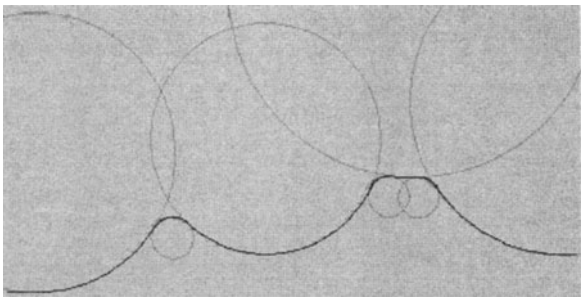

Figure 7. Profile of smoothly connected arcs

tangential. We have found that the use of special auxiliary elements is useful for profile fitting as well. We can stabilise the system by forcing each shared meeting point to approximate some data points where the join is expected.

A simple object is presented in figure 6. After determining the best rotational axes, the smooth profile from the bottom of the object has been reconstructed, see figure 7. The thick point set and the full circles are also shown.

2.3. 3D examples

Figure 8. Decimated point set

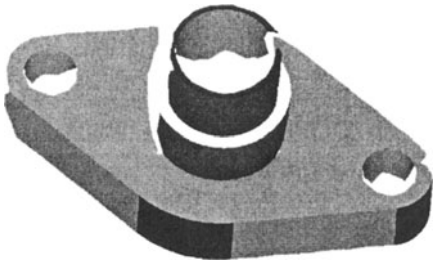

Figure 9. Composite smooth region

Finally, two simple 3D test objects have been selected. The first one shows a single view data set, a decimated triangular mesh is given in figure 8. As can be seen, the side is an extruded, composite face set, the separating edges are smooth, these were computed based on the profile of extrusion. The other edges are sharp, and these were determined

using surface-surface intersections, see figure 9. This object is a good illustration to show that incomplete models can also be reconstructed in a reasonably robust manner.

In many engineering objects, there are smooth, composite regions, where the boundaries of the primitive surfaces cannot be found directly by locating the abrupt changes of the estimated normal vectors — see also [1]. In such situations again the constrained fitting technique can provide an adequate solution.

The next example is a subset of a larger object. A decimated mesh can be seen in figure 10: there are seven regions which must join smoothly; three planes, three cylinders and a torus. In addition to the above 'face' objects further auxiliary objects need to be used, such as planes of the inner edges, or point-with-directions for ensuring the tangential connection at the innner vertices, or a line to ensure the coaxiality of two cylinders and the torus. Note, that in order to get a topologically consistent B-rep model, we must assure that the two inner vertices with valency four will represent a *single* 3D point. This fact must also be incorporated into the related constraint system together with various constraints including orthogonality and parallelism of planes, parallelism or orthogonality between planes and cylinder/torus axes, equality of radii, and so on. The final result is shown in figure 11.

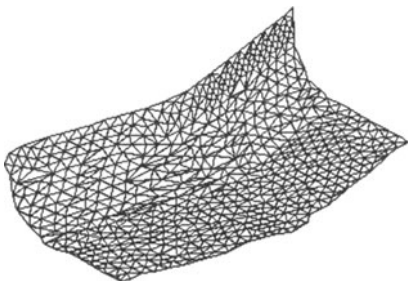

Figure 10. Decimated point set

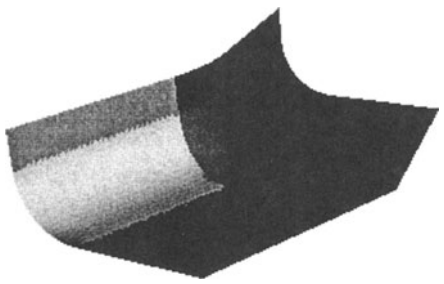

Figure 11. Composite smooth region

Conclusion

After describing a new numerical method to fit surfaces simultaneously to multiple point regions, the significance of this technique in reverse engineering was demonstrated. As it was shown, constrained fitting is necessary to obtain accurate and consistent CAD models, which can later be used in real engineering. Moreover, constrained fitting is necessary for model building: without constraints it would be hardly possible to locate smooth edges and determine coincident geometric en-

tities. The importance of handling contradicting constraints has also been pointed out. Finally, constraints are to be used for 'beautifying' reconstructed models. For example, when 'almost' perpendicular faces or 'almost' coaxial cylinders have been detected, these must be set accurately once the user approves to do so. To recognise and enforce a complex set of constraints in a consistent way is a difficult problem and will be subject of further research.

Acknowledgments

Many thanks are due to Géza Kós (CARI, Budapest) and Ralph Martin (Cardiff University) for inspiring discussions on various problems of constrained fitting.

Pál Benkő is working with the Geometric Modelling Laboratory of CARI, Budapest since 1995. He has just written his PhD thesis supervised by Tamás Várady on reconstructing conventional engineering objects.

Tamás Várady is the head of GML since 1991. He received a DSc degree with his thesis 'Vertex blending surfaces in computer aided geometric design' in 1998. In 1990, he was the cofounder of CADMUS Consulting and Development, of which he is president.

References

[1] P. Benkő, R. R. Martin, T. Várady: 'Algorithms for Reverse Engineering Boundary Representation Models', *Computer Aided Design*, accepted

[2] P. Benkő, L. Andor, G. Kós, R. R. Martin and T. Várady: 'Constrained fitting in reverse engineering', *Computer Aided Geometric Design*, accepted

[3] B. Brüderlin, D. Roller, Eds.: *Geometric constraint solving and applications*, Springer, 1998.

[4] J. Hoschek, W. Dankwort, Eds.: *Reverse Engineering*, B. G. Teubner, Stuttgart, 1996

[5] C. M. Hoffmann: *Geometric & Solid Modelling: an Introduction*, Morgan Kaufmann publishers, inc., San Mateo, 1989

[6] C. M. Hoffmann, R. Joan-Arinyo: 'Symbolic constraints in constructive geometry', *Journal of Symbolic Computation*, Vol 23, 1997, pp. 287–300

[7] A. D. Marshall, G. Lukács, R. R. Martin: 'Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy', *IEEE PAMI* Vol 23, No 3, 2001, pp. 304–314

[8] V. Pratt, 'Direct least-squares fitting of algebraic surfaces', *Computer Graphics* (SIGGRAPH 87), Vol 21, No 4, 1987, pp. 145–152

[9] T. Várady, R. R. Martin, J. Cox: 'Reverse Engineering of Geometric Models — An Introduction'; *Computer Aided Design*, Vol 29, No 4, 1997, pp. 255–268

[10] N. Werghi, R. B. Fisher, C. Robertson and A. Ashbrook, 'Object reconstruction by incorporating geometric constraints in reverse engineering', *Computer Aided Design*, Vol 31, 1999, pp. 363–399