

Approximate Algebraic Methods for Curves and Surfaces and their Applications

Bert Jüttler*, Pavel Chalmovianský, Mohamed Shalaby and Elmar Wurm
Institute of Applied Geometry, Johannes Kepler University, Linz, Austria†

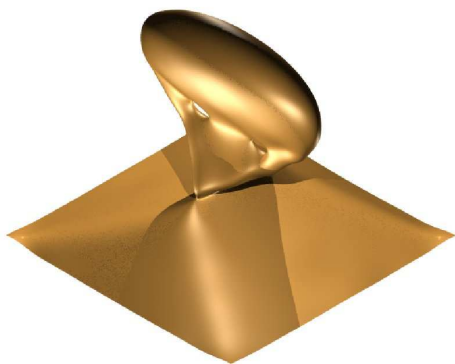


Figure 1: Approximate implicitization

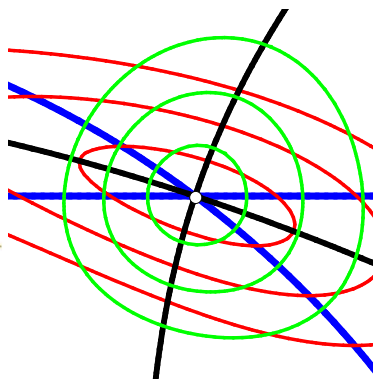


Figure 2: Preconditioning intersections

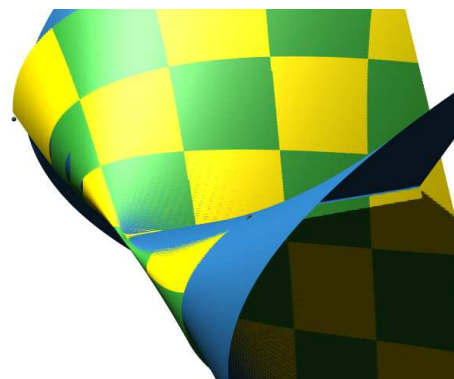


Figure 3: Approximate parameterization

Abstract

We report on approximate techniques for conversion between the implicit and the parametric representation of curves and surfaces, i.e., implicitization and parameterization. It is shown that these techniques are able to handle general free-form surfaces, and they can therefore be used to exploit the duality of implicit and parametric representations. In addition, we discuss several applications of these techniques, such as detection of self-intersections, ray-tracing, footpoint computation and parameterization of scattered data for parametric curve or surface fitting.

CR Categories: G.1.2 [Mathematics of Computing]: Numerical Analysis—Approximation; I.3.5 [Computing Methodologies]: Computer Graphics—Computational Geometry and Object Modeling; J.6 [Computer Applications]: Computer-Aided Engineering

Keywords: approximation, implicitization, parameterization, distance bounds, intersections and self-intersections, ray-tracing, footpoint computation, parameterizing scattered data.

1 Introduction

Curves and surfaces in Computer Graphics and Computer Aided Design can be modelled in several ways. They can be defined by embedding an interval $I \subset \mathbb{R}$ or a two-dimensional parameter domain $\Omega \subset \mathbb{R}^2$ (such as a rectangle) into space, by using a para-

metric representation. This approach has almost exclusively been used in the classical differential geometry of curves and surfaces. In Computer-Aided Design, the mapping is often described by rational spline functions. This leads to NURBS curves and surface, which have become a universally accepted standard.

Another technique, which is dual to the first one, originated in algebraic geometry. It is based on the representation of the geometry by iso-curves or iso-surfaces of functions (scalar fields) defined on the underlying space. Recently, this representation has attracted much interest in connection with so-called level-set techniques [Osher and Fedkiw 2003], where the function is given only at a 3D grid.

Both representations are ambiguous. Parametric representations can be modified via parameter transformations, and the scalar fields defining implicitly defined curves or surfaces can be multiplied by arbitrary (not necessarily constant) factors, provided that this does not introduce new branches in the region of interest.

Both representations are particularly well suited for certain applications. For example, parametric representations are very useful, for (e.g.) variational design, where aesthetic criteria are modelled via fairness functionals. On the other hand, implicit representations are especially useful for reconstructing curves or surfaces from unstructured point clouds, since fitting procedures for these representations do not need to assume the existence of an initial parameterization of the data (i.e., a mapping which identifies the given data with points in the parameter domain). In many applications, the simultaneous availability of both representations would be desirable, e.g., for computing surface-surface intersections.

In order to fully exploit the potential of both representations, efficient and robust algorithms for conversion between them are needed. The existing exact techniques (which are restricted to the cases of rational curves/surfaces and to scalar fields which are defined by polynomials) from algebraic geometry and symbolic computation cannot play this role; they face many difficulties, e.g., when dealing with coefficients given by floating point numbers. In addition, they may produce unexpected results, see Fig. 4 for an example. In applications, the use of approximate (numerical) methods will therefore often be preferred.

*e-mail: bert.juettler@jku.at

†Web-site: www.ag.jku.at

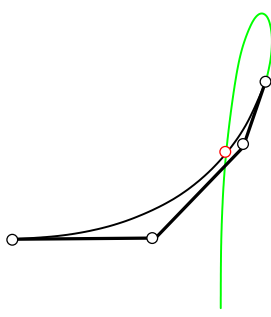


Figure 4: The exact implicitization of a planar cubic may introduce another branch (green) and a singular point (red) in the region of interest.

This paper gives a survey of various approximate techniques for the two conversion operations: implicitization and parameterization. In addition, we discuss the generation of a posteriori error bounds in the case of spline representations and applications of the results to intersections, ray tracing, footpoint computation, and to the parameterization of scattered data.

2 Approximate implicitization of surfaces

Given a parametric surface $\mathbf{p}(u, v)$ with parameter domain $\Omega \subset \mathbb{R}^2$, we compute an implicitly defined surface

$$0 = f(\mathbf{x}) = \sum_{i=0}^n \phi_i(\mathbf{x}) c_i \quad (1)$$

with basis functions ϕ_i (e.g., Bernstein polynomials, tensor-product B-splines, radial basis functions, or others) and unknown coefficients $c_i \in \mathbb{R}$, such that

$$f(\mathbf{p}(u, v)) \approx 0 \text{ for all } (u, v) \in \Omega. \quad (2)$$

Sometimes, the surface is only known at sampled points $\mathbf{p}_j = \mathbf{p}(u_j, v_j)$, $j = 0, \dots, N$. In this case, Eq. (2) is to be satisfied only at these points.

Note that Eq. (2) does not suffice to obtain a meaningful result. On the one hand, the trivial solution ($f = 0$) has to be excluded. In addition, one has to ensure that regular points of the parametric surface correspond to regular points of the implicitly defined surface. More precisely, the implicitization procedure should not generate additional branches of the surface which pass through the region of interest.

2.1 Related work

In the case of discrete data, the methods of [Pratt 1987] and [Taubin 1991] are based on certain normalizations in the coefficient space, in order to exclude the trivial solution. Pratt’s simple linear normalization, which is not geometrically invariant, leads to a linear system of equations for the unknown coefficients c_i . Taubin’s technique is based on a quadratic normalization, which produces a generalized eigenvalue problem. Both methods may have problems with unwanted branches. [Carr et al. 2001] use additional values of the function which are specified at so-called off-surface points.

In the case of given spline surfaces, the technique of [Dokken 2001] uses the factorization

$$f(\mathbf{p}(u, v)) = \mathbf{b}(u, v)^\top \mathbf{A} \mathbf{c} \quad (3)$$

where \mathbf{b} is a vector of basis functions spanning a linear space that contains the functions $\phi_i(\mathbf{p}(u, v))$, and $\mathbf{c} = (c_i)_{i=0}^n$. By applying singular value decomposition to the matrix \mathbf{A} , a right eigenvector corresponding to a small singular value is used to define an approximate implicitization. One may try to address the problem of additional branches by combining two or more eigenvectors (associated with small singular values).

Except for the technique of [Carr et al. 2001], all methods described in this section are able to reproduce the exact implicitization of a rational surface by a polynomial f . However, in practice one may prefer to work with lower degrees, since the number of coefficients may be very high otherwise.

2.2 Simultaneous fitting of points and normals

This method has been originally formulated in [Jüttler and Felis 2002]. Meanwhile, it has been implemented as prototype software within the frame of the European project GAIA II [Dokken et al. 2002–2005]. More details, including benchmarking and qualitative comparisons, can be found in [Wurm et al. 2005b; Shalaby et al. submitted].

In the case of discrete data, we assume that the given points \mathbf{p}_i are equipped with associated unit normal vectors \mathbf{n}_i . If these normals are unknown, then they can be estimated from the data via fitting locally defined planes of regression. For some of the points, this may give an ambiguous result, and therefore no normal vector can be estimated. These points are candidates for self-intersections of the surface. In the case of a given parametric surface, the normals can be sampled from the existing surface. In both cases, a local and global consistency of the sampled normals has to be guaranteed. While local consistency can be achieved by a simple region-growing process, global consistency is more difficult, in particular in the presence of singularities, see [Jüttler and Wurm 2003] for more details.

The coefficients c_i of the unknown function f are computed by minimizing a positive definite quadratic objective function of the form

$$\sum_{i=0}^N f(\mathbf{p}_i)^2 + w \|\nabla f(\mathbf{p}_i) - \mathbf{n}_i\|^2 + \text{“tension”}, \quad (4)$$

where w is a user-specified positive weight.

In the case of tensor-product B-spline functions ϕ_i , this function leads to a sparse linear system of equations. The additional tension terms are needed in order to regularize the system. The domain of interest is divided into cubes of the same size. This is done by specifying a cell-size. In order to guarantee an integer number of cells, the bounding box of the input surface is enlarged a bit. The domain of the spline functions consists of all cubes containing data, and its neighbours.

Fig. 5 shows an example. A surface with two self-intersecting curves has been implicitized by tensor-product polynomials and tensor-product spline functions. By using spline functions, one may avoid unwanted branches completely, see Figure 1. On the other hand, the local geometry of the singularities is reproduced better by the tensor-product polynomials, due to the higher polynomial degree.

The approximation result can be improved via ‘normal vector adjustment’: the normal vectors sampled from the first approximation are used to compute new one, and this can be iterated. Note that this procedure, which is similar to ‘parameter correction’ for parametric fitting [Hoschek and Lasser 1993], leads to a sequence of lin-



Figure 5: Approximate implicitization by tensor-product polynomials (this figure) and by tensor-product spline functions (see Fig. 1). Data courtesy of think3.

ear systems with identical matrices, but varying right-hand sides, which can be solved efficiently by factorizing the matrix.

3 Approximate parameterization

In the case of algebraic curves and surfaces, various symbolic-computation based techniques exist. However, these techniques have difficulties if the implicitly defined curve or surface is only given by floating point coefficients, i.e., with limited accuracy. Also, they cannot deal with general algebraic curves and surfaces, since an exact rational parameterization does not exist in the generic case. Approximate techniques, which generate a parameterization within a certain region of interest, can be used to avoid these problems. In this section we discuss approximate techniques for parameterizing planar curves, space curves (given as the intersection of two surfaces), and surfaces.

3.1 Planar curves

We consider a rational curve of the form

$$\mathbf{p}(t) = \left(\frac{p_1(t)}{p_0(t)}, \dots, \frac{p_d(t)}{p_0(t)} \right)^\top, \quad t \in I. \quad (5)$$

It can be described by homogeneous coordinates

$$p_j(t) = \sum_{i=0}^m \beta_i(t) b_{j,i}, \quad j = 0, \dots, d, \quad (6)$$

with basis functions $\beta_i(t)$ (e.g., B-splines) and coefficients $b_{j,i}$. A method for parameterizing planar curves (i.e., $d = 2$) has been described by [Jüttler and Chalmovianský 2004]. It is based on minimizing the nonlinear objective function

$$\int_I \frac{f(\mathbf{p}(t))^2}{\|\nabla f(\mathbf{p}(t))\|^2} dt + \text{regularizing terms}, \quad (7)$$

subject to suitable boundary conditions, using a damped Newton method.

The objective function (7) gives the total squared Sampson [1982] distance of the curve segment, which approximates the Euclidean

distance very well, provided that the implicitly defined curve and its approximation are sufficiently close to each other.

The regularizing terms were chosen so as to obtain an approximately uniform distribution of the parametric speed and to keep the denominator $p_0(t)$ close to one, by integrating a penalty function

$$(p_0(t) - 1)^{2k} \quad (8)$$

over the parameter domain I , with some constant $k \in \mathbb{Z}_+$.

As for all non-linear optimization techniques, the choice of the initial solution is probably the most important issue. Here, we started with a small segment (almost identical to a tangent) and force it to grow along the curve. According to our numerical experience, this approach is able to reproduce exact rational parameterizations, if these are available.

3.2 Intersection curves

Currently, the generalization of this technique to intersection curves of two implicitly defined surfaces $f(\mathbf{x}) = 0$, $g(\mathbf{x}) = 0$ is under investigation [Chalmovianský and Jüttler 200x]. In this case, the squared Euclidean distance l^2 of a point \mathbf{p} to the intersection curve can be approximated by

$$\frac{F(\mathbf{p})^2}{\|\nabla F(\mathbf{p})\|^2} + \frac{G(\mathbf{p})^2}{\|\nabla G(\mathbf{p})\|^2} \quad (9)$$

where the auxiliary functions F and G are defined as

$$F = \|\nabla f\|g + \|\nabla g\|f \quad \text{and} \quad G = \|\nabla f\|g - \|\nabla g\|f. \quad (10)$$

This can be seen as a preconditioning step, similar to the preconditioning of large linear systems in numerical analysis, which is used to speed up the convergence of iterative solvers.

This observation is illustrated by Fig. 2, which shows two implicitly defined planar curves $f(\mathbf{x}) = 0$, $g(\mathbf{x}) = 0$ (in blue), the iso-curves (level sets) of

$$\frac{f^2}{\|\nabla f\|^2} + \frac{g^2}{\|\nabla g\|^2} \quad (11)$$

(red), the preconditioned curves $F(\mathbf{x}) = 0$, $G(\mathbf{x}) = 0$ (black), and the iso-curves of the right-hand side in Eq. 9 (green ovals). The two preconditioned curves intersect orthogonally and the distance ovals are approximately circles.

A surface example is shown in Figure 6. In this case, the given surface are two quadrics intersecting in a conic section in the region of interest (there should be another conic section elsewhere).

Note that this techniques fails if the two original curves resp. surfaces intersect each other tangentially!

This distance measure has been used to generate the parameterization of the intersection curve of a sphere and a cylinder, see Figure 6. More details of the method will be presented in [Chalmovianský and Jüttler 200x].

3.3 Surfaces

Meanwhile, [Wurm et al. 2005a] have generalized this approach to the surface case. In addition to terms controlling the distance and the denominator, the nonlinear optimization also takes the inner geometry into account. More precisely, we use additional terms of the objective function which measure the deviation from an isometric

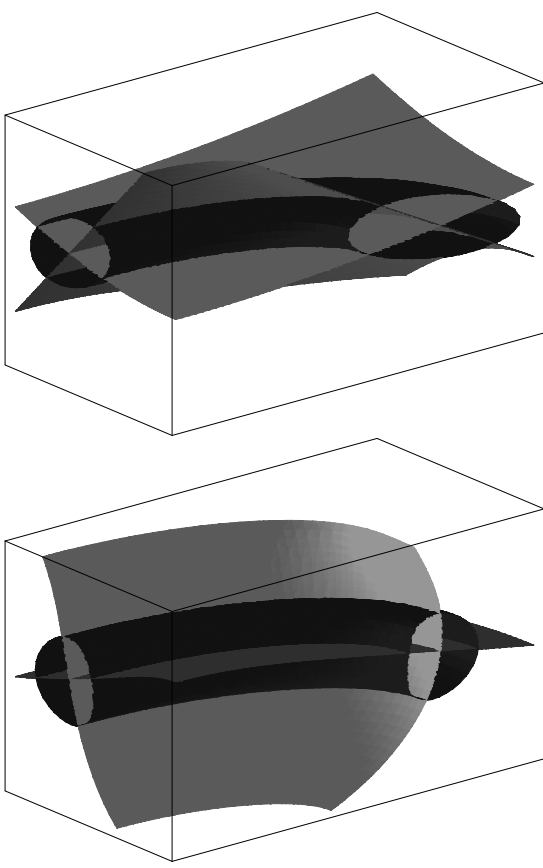


Figure 6: Preconditioning the intersection of two quadric surfaces. The two surfaces and an iso-surface of the functions (9) resp. (11) before (top) and after (bottom) preconditioning are shown. After preconditioning, the latter surface becomes more similar to a tubular surface.

parameterization. As an example, Fig. 3 shows a surface patch on an algebraic surface with a double line (‘Whitney’s umbrella’). The quality of the parameterization of the surface patch is visualized by the checkerboard pattern on the surface.

4 Distance bounds

In order to certify the accuracy of the results of approximate implicitization and/or approximate parameterization, distance bounds between implicitly defined and parametric curves and surfaces are needed. Under certain technical assumptions (see [Aigner et al. 2004]), the one-sided Hausdorff distance between a planar parametric curve and an implicitly defined parametric curve can be bounded by

$$\frac{\max_{t \in I} |f(\mathbf{p}(t))|}{\min_{\mathbf{x} \in \Omega} \|\nabla f(\mathbf{x})\|}, \quad (12)$$

where $\Omega \subset \mathbb{R}^2$ is the region of interest. This fact can easily be generalized to the surface case.

In the case of polynomials, upper resp. lower bound on numerator and denominator in Eq. 12 are available by using the Bernstein–Bézier representation. More precisely,

$$|f(\mathbf{p}(t))| \leq \max_j |c_j| \quad (13)$$

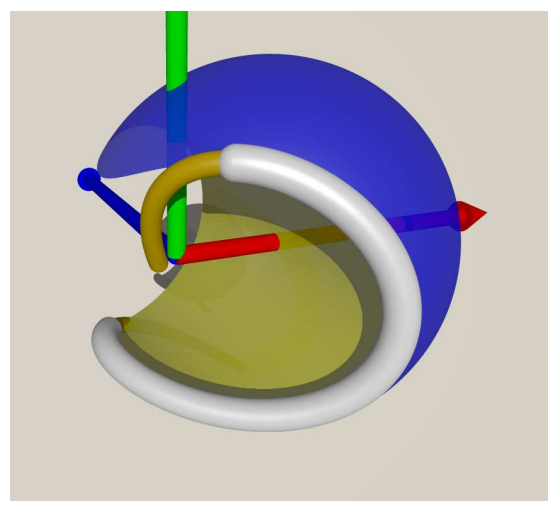


Figure 7: Parameterization of the intersection curve between two quadric surfaces.

where

$$f(\mathbf{p}(t)) = \sum_j \beta_j(t) c_j. \quad (14)$$

The norm of the gradient $\|\nabla f(\mathbf{x})\|$ can be bounded from below by the minimum distance of the ‘gradient patch’ $\nabla f : \Omega \rightarrow \mathbb{R}^3$ to the origin, which can again be bounded from below by using the convex hull of its control points with respect to a suitable Bernstein–Bézier representation.

However, in the case of piecewise polynomials f (e.g., bivariate tensor-product spline functions) and \mathbf{p} , a B-spline representation of the form (14) is hard to find, since one would need to compute the parameter values of the intersections between $\mathbf{p}(t)$ and the knot lines of f , see Fig. 8. This is even more true in the case of surfaces, where such a B-spline representation does not exist in general. In this situation, one can bound the numerator by using

$$|f(\mathbf{p}(t))| \leq \max_{i,j} |c_{i,j}| \quad (15)$$

where

$$f(\hat{\mathbf{p}}(u,v)) = \sum_{i,j} \beta_i(u) \beta_j(v) c_{i,j}, \quad (16)$$

where $\hat{\mathbf{p}}(u,v)$ is a bilinear parameterization of a bounding box of the curve $\mathbf{p}(t)$, and similarly for surfaces. Clearly, $f \circ \hat{\mathbf{p}}$ has a tensor-product spline representation, provided that f is a tensor-product spline function, too.

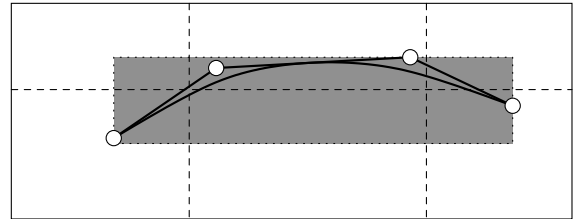


Figure 8: Generating an upper bound on $|f(\mathbf{p}(t))|$ in the case of piecewise polynomials by using the bounding box (dotted). The accuracy of the bound can be improved by splitting the curve $\mathbf{p}(t)$ into smaller segments. The dashed lines are the knot lines of f .

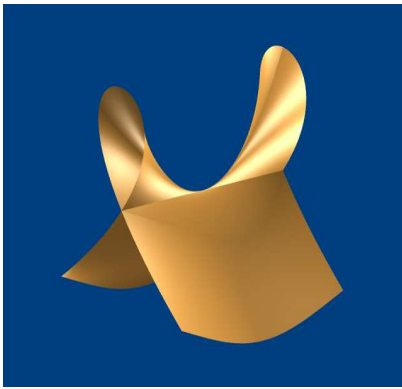


Figure 9: Ray tracing using approximate implicitization.

5 Applications

This section discusses several applications of approximate algebraic methods, such as the detection of singularities and self-intersections, ray-tracing, footpoint computation, and parameterization of scattered data.

5.1 Detecting self-intersections

Various geometric operations in Computer Aided Design, such as offsetting and blending, may produce curves and surfaces with singularities. In order to maintain the correctness of the model, the resulting object has to be trimmed at these singularities.

Such singularities (in particular self-intersections) of curves $\mathbf{p}(t)$ and surfaces $\mathbf{p}(u, v)$ can easily be detected, provided that both representations are available. They correspond to simultaneous zeros of the components of

$$\nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{p}(\cdot)} \quad (17)$$

Alternatively, as proposed by [Thomassen 2005], one may consider the univariate / bivariate functions

$$\mathbf{p}'(t)^\perp \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{p}(t)} \text{ resp.} \quad (18)$$

$$\mathbf{p}_u(u, v) \times \mathbf{p}_v(u, v) \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{p}(u, v)} \quad (19)$$

in the case of curves and surfaces, respectively. In the case of non-tangential double points or curves, sign changes of these two functions can be used to detect the singularities of the curve or surface.

5.2 Ray-Tracing

Approximate implicitization may be used to speed up algorithms for ray-tracing, since intersections with lines are much easier to compute for implicitly defined surfaces. We demonstrate this fact by an example: It took 2 seconds to generate Fig. 9, which was generated by PoVRay, using an approximate implicit representation. When using the original parametric representation for generating a similar image, the computation time was about 20 seconds.

5.3 Footpoint computation

Algorithms for computing the footpoint (nearest point) to a given point on a curve or surface is needed for various applications, e.g.,

collision detection or parameterization of scattered data. Robust techniques, which avoid the dependency on chosen initial values, can again be formulated, provided that an implicit representation is available [Aigner and Jüttler 2005].

5.4 Parameterization of scattered data

Fitting a parametric curve or surface to scattered data is a non-linear problem, which can be solved by using suitable optimization techniques, such as parameter correction [Hoschek and Lasser 1993] or Newton-type methods [Rogers and Fog 1989; Speer et al. 1998]. The result depends heavily on the choice of an initial solution. More precisely, the given data \mathbf{p}_i have to be equipped with suitable parameter values t_i resp. (u_i, v_i) , e.g., by projecting them onto a line or plane, or another auxiliary curve or surface.

A more general approach for generating such an auxiliary surface, which seems to be promising, is as follows:

1. Generate a low-degree algebraic (spline) curve or surface, which is a rough parameterization of the given points,
2. parameterize it using the techniques of approximate implicitization, and
3. find the parameter values associated with the data by computing the footpoints of the given points on this parametric curve or surface.

An example is shown in Fig. 10.

As an advantage, this approach should be able to handle data which cannot be projected onto a line or plane. In addition, the weights of the rational parameterization may also be used to initialize the weights of the approximating parametric rational curve or surface. This may be a subject of further research.

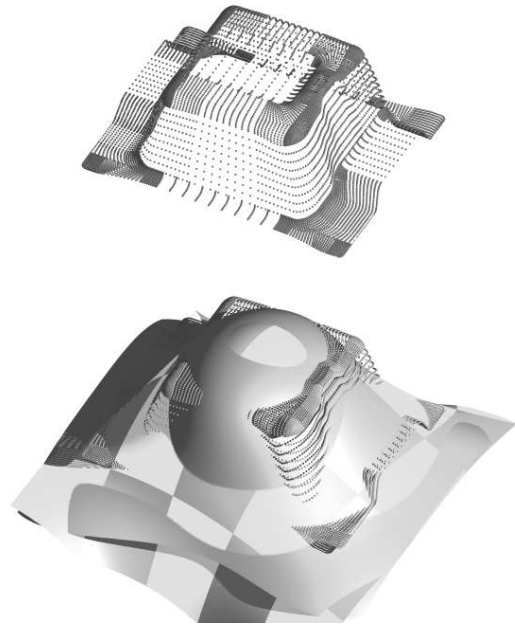


Figure 10: Generating an initial parameterization via implicit fitting and parameterization (data courtesy of Holometric Technologies, Aalen). Top: the data, bottom: implicit approximation and its parameterization (with checkerboard pattern).

This paper presented several numerical techniques for approximately converting implicitly defined curves or surfaces into parametric ones, and vice versa. In the case of approximate implicitization, this was implemented as a prototype software which is one of the deliverables of the GAIA II project [Dokken et al. 2002–2005].

As demonstrated by the applications, many problems in geometric computing can be solved more easily by exploiting the duality of both representations. The investigation of suitable algorithms and their comparison with existing ones, in particular from the field of computational geometry, may be a subject for further research.

Acknowledgments

This research was supported by the Austrian Science Fund (FWF) through subproject 15 of the Special Research Area (SFB) F013 “Numerical and Symbolic Scientific Computing” at Linz, and by the European Commission through project IST 2001-35512 GAIA II, entitled “Intersection algorithms for geometry-based IT-applications using approximate algebraic methods”, which is supported by the “Future and Emerging Technologies” arm of the Fifth Framework Programme.

References

- AIGNER, M., AND JÜTTLER, B. 2005. Robust footpoint computation. In *Mathematical Methods for Curves and Surfaces: Tromsø 2004*, Nashboro Press, M. Dæhlen and L. Schumaker, Eds., in press.
- AIGNER, M., JÜTTLER, B., AND KIM, M.-S. 2004. Analyzing and enhancing the robustness of implicit representations. In *Geometric Modelling and Processing*, IEEE Press, 131–140.
- CARR, J. C., BEATSON, R. K., CHERRIE, J., MITCHELL, T. J., FRIGHT, W. R., ALLUM, B. C. M., AND EVANS, T. R. 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 67–76.
- CHALMOVIANSKÝ, P., AND JÜTTLER, B. 200x. Approximate parameterization of intersections. in preparation.
- DOKKEN, T., ET AL., 2002–2005. Intersection algorithms for geometry-based IT-applications using approximate algebraic methods (GAIA II). EU project IST 2001–35512.
- DOKKEN, T. 2001. Approximate implicitization. In *Mathematical Methods in CAGD: Oslo 2000*, Vanderbilt University Press, Nashville TN, T. Lyche and L. Schumaker, Eds.
- HOSCHEK, J., AND LASSER, D. 1993. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, Mass.
- JÜTTLER, B., AND CHALMOVIANSKÝ, P. 2004. Approximate parameterization by planar rational curves. In *Proceedings of the Spring Conference on Computer Graphics 2004*, Comenius University / ACM Siggraph.
- JÜTTLER, B., AND FELIS, A. 2002. Least-squares fitting of algebraic spline surfaces. *Advances in Computational Mathematics* 17, 135–152.
- JÜTTLER, B., AND WURM, E. 2003. Approximate implicitization via curve fitting. In *Symposium on geometry processing*, Eurographics / ACM Siggraph, L. Kobbelt, P. Schröder, and H. Hoppe, Eds., 240–247.
- OSHER, S., AND FEDKIW, R. 2003. *Level set methods and dynamic implicit surfaces*. Springer.
- PRATT, V. 1987. Direct least-squares fitting of algebraic surfaces. In *Proceedings of SIGGRAPH 1987*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 145–152.
- ROGERS, D., AND FOG, N. 1989. Constrained B-spline curve and surface fitting. *Computer-Aided Design* 21, 641–648.
- SAMPSON, P. 1982. Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing* 18, 97–108.
- SHALABY, M., THOMASSEN, J., WURM, E., DOKKEN, T., AND JÜTTLER, B. submitted. Piecewise approximate implicitization: Experiments using industrial data. In *Algebraic Geometry and Geometric Modelling 2004*, B. Mourrain and R. Piene, Eds.
- SPEER, T., KUPPE, M., AND HOSCHEK, J. 1998. Global reparameterization for curve approximation. *Comp. Aided Geom. Design*, 869–877.
- TAUBIN, R. 1991. Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 13, 1115–1138.
- THOMASSEN, J. 2005. Self-intersections and approximate implicitization. In *Computational Methods for Algebraic Spline Surfaces*, Springer, Heidelberg, T. Dokken and B. Jüttler, Eds., 155–170.
- WURM, E., JÜTTLER, B., AND KIM, M.-S. 2005. Approximate rational parameterization of implicitly defined surfaces. In *The Mathematics of Surfaces XI*, Springer, H. Bez and R. Martin, Eds., LNCS, in press.
- WURM, E., THOMASSEN, J., JÜTTLER, B., AND DOKKEN, T. 2005. Comparative benchmarking of methods for approximate implicitization. In *Geometric Design and Computing*, Nashboro Press, M. Neamtu and M. Luciam, Eds.