Evolutionary Surface Reconstruction Using CSG-NURBS-Hybrids

Klaus Weinert, Tobias Surmann, and Jörn Mehnen
Dept. of Machining Technology
University of Dortmund
D-44227 Dortmund, Germany
{weinert,surmann,mehnen}@isf.mb.uni-dortmund.de

Abstract

The aim of evolutionary surface reconstruction is to find mathematical descriptions of surfaces of physical objects. These objects are represented only by discrete sampling points. The reconstructions have to render the original shape of the objects as realistic as possible. Therefore, complex discrete approximation problems must be solved. In order to process surface reconstructions in CAD systems, the descriptions have to render the construction logic of the scanned object. Here, Nonuniform Rational B-Splines (NURBS) and Constructive Solid Geometry (CSG) are combined in one hybrid reconstruction system that fits each surface type automatically using one homogenous data structure.

1 INTRODUCTION

Discrete approximation and pattern recognition are fundamental problems in mathematics and computer science. Mechanical engineering, medical science and object design have an urgent need for realistic and computer adequate descriptions of surfaces of real physical objects. In order to reproduce the shape of an object in a computer the object surfaces are sampled using either optical or contact scanning technics. Depending on the scanning system, thousands ranging to millions of discrete points with either regular or irregular distributions are generated. The task of surface reconstruction is to identify the properties of the original surface from the discrete sampling points and to generate a mathematical surface description that can be used for further processing in CAD systems.

In CAD, two typical modeling techniques can be distinguished. On one hand, Nonuniform Rational B-Splines [PT97] are commonly used to describe free-formed surfaces. On the other hand, complex technical structures can be composed using elementary geometries (e.g. spheres, boxes, cylinders, etc.). This structure is called Constructive Solid Geometry. Both models have their own specific advantages and disadvantages regarding their constructional and computational effort. However, CAD systems are often designed to support (mainly) only one construction technique. Technical demands as well as the natural appearance of real objects make it necessary to combine both approaches in one hybrid surface model. See Figure 1.

Figure 1: Turbine Blade

Both, discrete surface approximation and pattern recognition can be interpreted as optimization problems. Evolutionary algorithms belong to a class of probabilistic optimization strategies that already proved to be robust and able to find surprisingly good solutions even applied to complex multimodal, high dimensional and multiobjective problems [SNL00]. Evolution Strategies (ES) [Sch95, Bäc96] performed well in NURBS surface approximations [WM00b, WM00a]. Genetic Programming (GP) [Koz92] has been successfully applied to design optimization [Ben99] and surface reconstruction [KMBW99, Meh01].

In this article, the classic approach to use separate data structures and specialized optimization strategies is replaced by one evolutionary hybrid surface reconstruction algorithm that deals with NURBS as well as CSG structures.

2 DATA STRUCTURE

2.1 CONSTRUCTIVE SOLID GEOMETRY

Mechanical machine components are typically composed using basic components such as boxes, spheres, cylinders, etc. Guo, Menon showed, that it is possible to build arbitrarily complex bodies using elementary elements and half spaces [GM96]. Every complex construction can be composed using simpler CSG objects that interact geometrically depending on the binary relations join ($\cup$), intersect ($\cap$) and subtract ($-$). Typically, a binary tree forms the basic data stucture used in CSG-CAD-systems. The primitive geometries (quadrics, boxes, tori, NURBS-solids, etc.) are used to form the terminal nodes of the binary tree. Correspondingly, the binary relations are used as inner nodes of the CSG tree. Due to its recursive structure, a CSG tree can be interpreted as a word from a context free language. Standard interfaces like STEP or IGES make use of this property. Figure 2 illustrates the composition of the CSG structure displayed at the top of the graph.

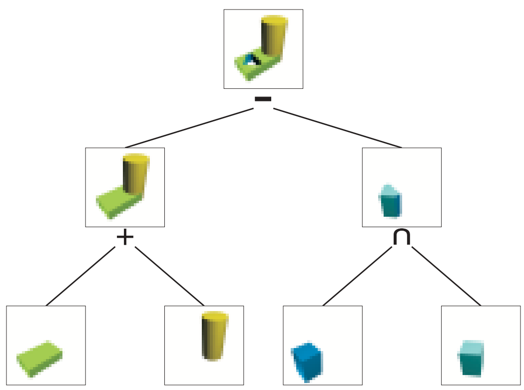

Figure 2: Composition of a CSG Structure

Note, that for each CSG object, there is an infinite number of representing CSG trees, which means the CSG object representation is not unique. Following the demands of a designer, a surface reconstruction algorithm has to find the most simple constructive setup, i.e. the CSG tree with a minimum number of elements that resembles the digitized objects as realistic and exact as possible. Due to the fact that only objects have to be reconstructed that once have been constructed using a finite set of elements, it is always possible to find a (not unique) finite reconstruction. Thus, the search space is finite but with *a priori* unknown dimension. The reconstruction algorithm has to find the CSG construction that fits best into a given point set. Thus, an approximation problem and a combinatoric problem has to be solved in parallel. The search space consists of a combination of real value vectors (position and orientation of each CSG element in $\boldsymbol{R}^3$) and a variable dimensional graph structure (CSG tree).

2.2 FITNESS FUNCTION

Concerning its structure a CSG tree is well suited to represent the genome of an individual. The fitness function is confronted with the problem to map the genome to its phenotypic representation and to compare this representation with the given sampling points. Since the reconstructions are restricted to objects without undercuts, the fitness function can be simplified to a comparison of each scanning point with its projection on the CSG object parallel to the z-axis. Therefore, it is necessary to implement a function that calculates an intersection between a ray and the CSG object.

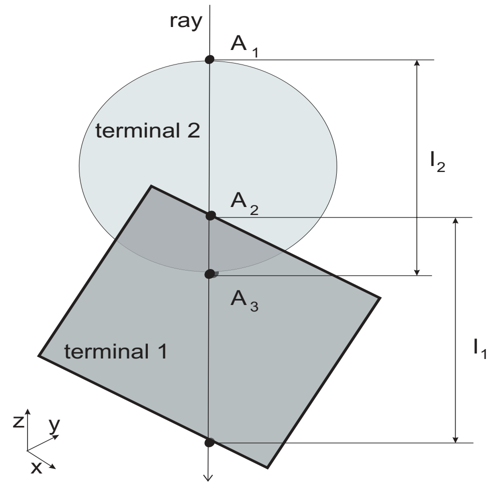

Figure 3: Calculating Intersections of a Ray and a CSG Object

Figure 3 shows two CSG elements (terminals) and a ray. The intersections of the ray with each terminal defines the intervals I_1 and I_2. In order to model the surface of the resulting CSG structure, the corresponding point A can be calculated from the intersection of the two intervals depending on the corresponding boolean operation in the CSG tree [FvDFH90]. In the example A equals A_1 or A_2 or $A3$ if terminal 1 and terminal 2 are joined, intersected or subtracted.

In order to evaluate the quality of a reconstruction, a comparison between the digitized points and the corresponding points on the CSG object surface has to be performed. The following criteria are used to form the multiobjective fitness function:

Δ describes the distance between each sampling point and its corresponding point A on the surface of the individual.

ABN (angle between normals) represents the deviation of the normal vectors at each sampling point position from the corresponding vectors position on the surface of the CSG object. This value is calculated by the dot product of the two vectors. The total ABN result is normalized to an interval of 1 (best fit) and 0.

CT (curvature type) is a criterion that helps the algorithm to find the optimal terminal types to construct a CSG object. This function compares the curvature type of each point with the corresponding one on the individual's surface. The curvature type – like the values of the normal vectors – have been stored in the preprocessed sampling data. Thus, the evaluation time can be reduced drastically. CT yields 1 for an optimal match and 0 otherwise.

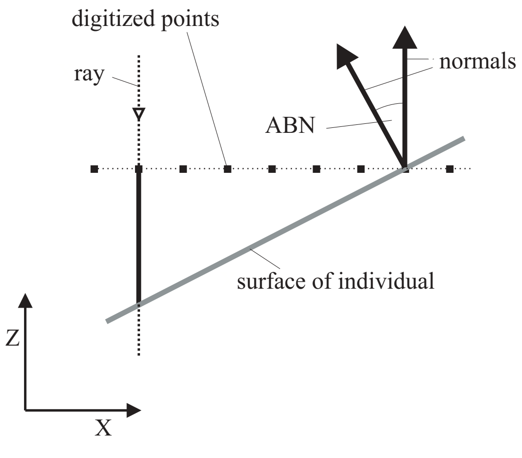

Figure 4: Δ and ABN Criteria

There is also a fitness criterion that does not refer to the points on the CSG object but to the individual´s structure itself. This is the number of nodes N in the CSG tree. A good reconstruction excels not only by minimal Δ, ABN and CT values but also by its simplicity. Thus the number of elements of a genome as to be introduced into the fitness function. These four criteria can be weighted individually for each reconstruction to get results which focus at the desired quality of the reconstruction. The formula for calculating the fitness F_p of the individual at the point p is as such:

$$F_p = w_\Delta \frac{1}{1+\Delta} + w_{ABN}\, ABN + w_{CT}\, CT - \frac{N}{c}$$

The parameters w_x are the weights of the criteria and c is a constant, which controls the influence of the number of nodes inserted into the individual. If the value of c is high, the influence of the number of nodes will be small, thus, an individual may be constructed from many terminals. The overall fitness is the average over all F_p. The lower bound of the fitness is $-\frac{N}{c}$ indicating the worst case while $w_\Delta + w_{ABN} + w_{CT} - \frac{N}{c}$ indicates an optimal reconstruction.

2.3 NONUNIFORM RATIONAL B-SPLINES

Due to technical or design purposes many objects show smooth surfaces. E.g. car bodies, houseassets and even turbine blades are typically designed using NURBS. NURBS surfaces are elementary construction elements in many CAD systems [PT97]. NURBS can model smooth surfaces as well as edges or peaks. Their mathematical structure allows intuitive manual modifications. NURBS are a very powerful tool in CAD and, thus, this mathematical model has become a quasi-standard representation in CAD systems. International data exchange interfaces like STEP or IGES support these rational B-Spline formats.

NURBS are a superset of Bézier polynomials and nonrational B-splines. They are efficient regarding space and time complexity and numerically stable. NURBS surfaces are parametric tensor products that map the vector $(u,v)^T \in \boldsymbol{R}^2$ to $(x,y,z)^T \in \boldsymbol{R}^3$. NURBS surfaces of order (p,q) are composed of the following components:

- the control point net with $n \times m$ vertices $\mathbf{P} = \{\mathbf{P}_{i,j} \in \boldsymbol{R}^3, i = 1,...,n, j = 1,...,m\}$,
- the knot vector U and V, where $\mathbf{U} = (\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{r-p-1}, \underbrace{1,\ldots,1}_{p+1})^T$, $\mathbf{V} = (\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_{s-q-1}, \underbrace{1,\ldots,1}_{q+1})^T$, and $r = n+p+1$ and $s = m+q+1$. The

u_i and v_j have an ascending or descending order, respectively.

- the weight vector $\mathbf{W} = \{w_{i,j} \in \boldsymbol{R}^+, i = 1, ..., n, j = 1, ..., m\}$.

The parametric surface function is defined over the domain $(u, v) \in [0, 1] \times [0, 1]$ and expressed by

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P_{i,j}}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}}.$$

$N_{i,p}(u) : \boldsymbol{R} \rightarrow \boldsymbol{R}$ are the ith basis functions of order p for the parameter $u \in \boldsymbol{R}$ computed on a knot vector $(u_0, ..., u_m)^T$. The basis functions $N_{i,p}(u)$ can be defined recursively by

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$
$$N_{i,p}(u) = \frac{(u-u_i)N_{i,p-1}(u)}{u_{i+p}-u_i} + \frac{(u_{i+p+1}-u)N_{i+1,p-1}(u)}{u_{i+p+1}-u_{i+1}}$$

and evaluated efficiently by the *Cox - de Boor* algorithm for example. The parameters of the control net vertices, knot vectors and the weight vectors can be joined to one vector $\varphi = (\mathbf{P}, \mathbf{T}, \mathbf{W})$ of the space $\boldsymbol{R}^\gamma$ with dimension $\gamma = 4nm + (n + p + m + q)$.

2.4 CSG-NURBS-HYBRID

For the reconstruction of realistic workpieces the application of regular geometric objects is often not sufficient. Design parts are often composed of both sculptured surfaces and regular geometries. To find a remedy, NURBS patches have to be integrated into the CSG model. Figure 5 shows a CSG object consisting of some boxes, cylinders and a NURBS patch. Due to the fact that the fitness function only computes the distance between a digitized point and a CSG object along a parallel to the z-axis, it is possible to use NURBS patches as terminals in the CSG model. It is assumed that the NURBS patch represent a surface of a solid which is infinite in the negative direction of the z-axis. Thus, the primary functionality a NURBS terminal has to provide is to compute an intersection of a ray which has a direction parallel to the z-axis. This is done by a binary search for the (u, v) koordinates accoding to the actual sampling point. To avoid multiple evaluation of the surface function, the algorithm stores intermediate data which is updated only if made necessary by a genetic operator. The second intersection (for CSG it is necessary to have an even number of intersections) which is expected for volumetric objects is set to a default value which lies below the minimum z-value of the digitized points. This ensures the correctness of the intersection routine of the CSG data structure. See Figure. 6.
The evolutionary mutation operator of the terminals works on the control points, the location and the tilt angles of the NURBS patches. The insertion of new control points and the variation of the degree of the NURBS are not implemented yet.

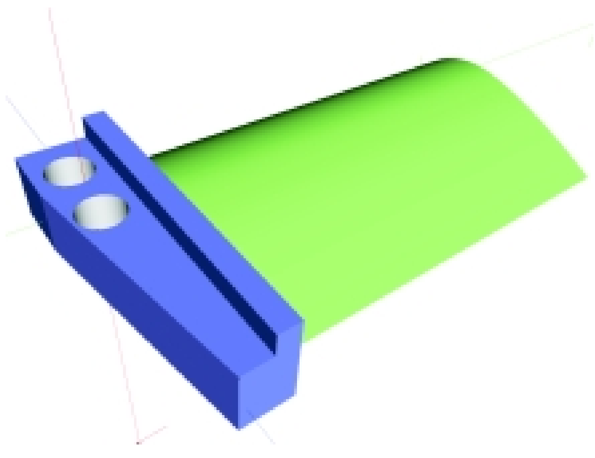

Figure 5: CSG Model with a NURBS Component

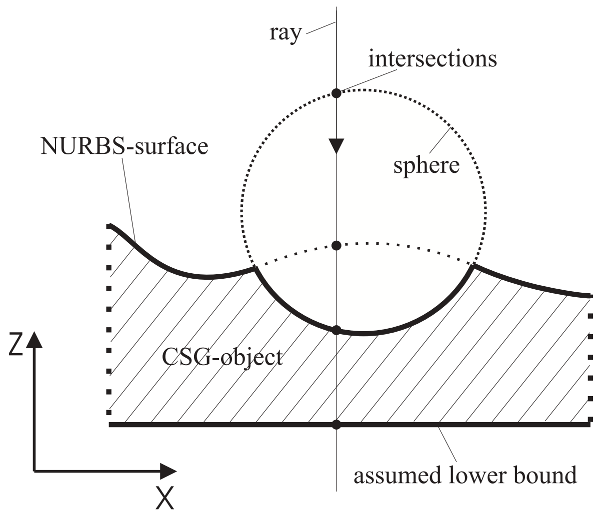

Figure 6: Intersection of a Ray and a CSG-NURBS-Hybrid

2.5 GENETIC OPERATORS

In an object oriented view for an individual it is necessary to provide operators which perform mutations and recombinations. Therefore several operators have been implemented which vary the attributes of the terminal elements and the inner nodes of the CSG tree.

2.5.1 Variation Operators

- Terminal
  The terminal operator mutates the attributes of each terminal of the CGS tree with a predefined probability. The mutation values, which are specific for each individual, are self-adapted following an evolution strategy scheme [Sch95]. These attributes define the location, the size and the tilt angles of each CSG object. The effect of this operator is an average slight variation of each individual. It is mainly used in the late phase of a reconstruction to adjust the correctly structured CSG object to the digitized points.

- Relation
  This operator changes the boolean relations which are assigned to the inner nodes of the CSG tree. It does not change the structure of the tree itself but has a big effect on the phenotypic representation of the individual.

- Delete
  In order to remove redundant or incorrect data from an individual, this operator prunes randomly selected subtrees from the CSG tree.

- Insert
  In order to improve the shape of an individual and to make it more complex this operator randomly inserts terminals into the CSG tree.

- Replace
  In some cases the system fits a terminal into a subset of sampling points which represents a terminal of another type (e.g. a box instead of a sphere). For this reason there is an operator which can change the type of a terminal into another to make sure that the system does not stay in a local extremum of the fitness function.

2.5.2 Recombination Operator

Two individuals are recombined by randomly exchanging subtrees of the genome. This way it is possible to combine parts of two CSG trees to form a new one which fits better into the digitized points.

3 MAIN ALGORITHM

The main algorithm uses features of both ES and GP. The selection follows either a (μ, λ)- or a $(\mu + \lambda)$-ES scheme. Furthermore, the step size adaptation is adopted from an ES [Sch95]. The variation of the CSG tree structure is realized by application of a classic GP scheme [Koz92]. This allows the reconstruction of objects with an from the outset unknown complexity.
It has been found useful to work on multiple populations and to recombine individuals from different populations always after some generations. So every population can work out some details of the object to be reconstructed. After a certain number of generations (this number is highly dependent on the complexity of the workpiece) these partial solutions are exchanged between the populations by recombination to form new individuals which often provide new and advantageous features.

4 RESULTS

4.1 CSG RECONSTRUCTIONS

The object shown in Figure 7 has been selected to illustrate the evolutionary reconstruction process. It consists of two interleaving crosses. The physical object has been digitized using a tactile scanner. The corresponding point set was reduced and resampled using 2D-point selection schemes and optimized triangulations [WM00a]. The resampled points showed a regular and equally spaced structure which is more adequate for curvature analyses than the unstructured and dense original sampling points. A deterministic algorithm analyzed each coordinate of the point set and extended the data by the curvature value and normal vectors of the corresponding surface points. This yields a seven dimensional vector matrix which is used to evaluate the quality of the evolved CSG solutions.
The sequence in Figure 7 shows the three typical phases of a reconstruction. The algorithm starts with few randomly selected primitives. Each solution is evaluated by the multiobjective fitness function described above. In the beginning elements which approximately fit into the point set form seed points for more complex structures. During the next generations the GP algorithms increases the number of CSG elements per individual rapidly. This yields large but weakly structured CSG objects. These aggregations contain subsolutions which already roughly approximate parts of the original object structure. In the following phase these substructures are filtered from the stream of individuals. This decreases the number of elements and increases the fitness value of the solutions.
During the third phase the algorithm aligns the substructures with the structural shape of the object. During this period the distance and the orientation of the elements are adjusted while the number of elements is kept constant. The GP/ES-hybrid optimizes

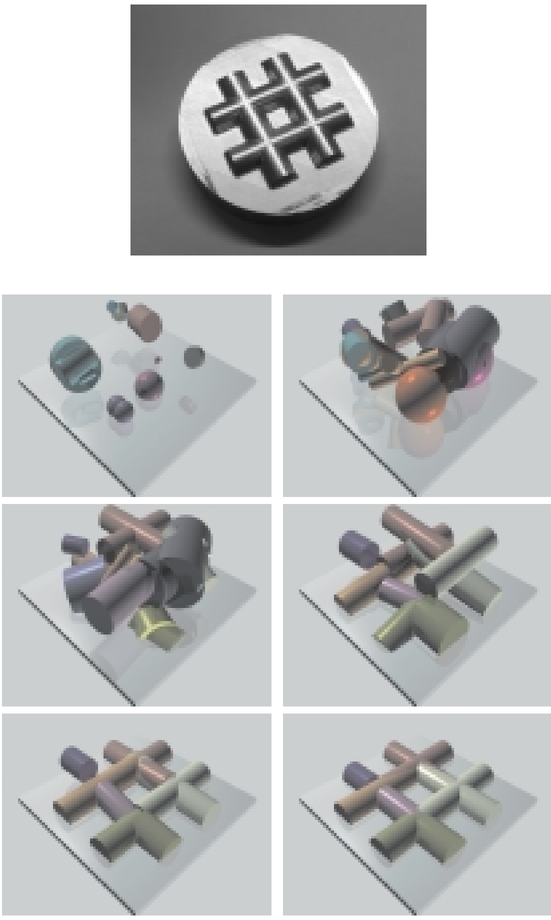

Figure 7: Reconstruction of Two Interleaving Crosses. Photo of the Original Object (Top) and the Reconstruction (Below).

the real vector parameters of the CSG elements using its ES power. The single step size of each CSG object is adjusted automatically following the self-adaptation scheme of an evolution strategy. The reconstruction sequence in Figure 8 illustrates the fact that the algorithm is also able to reconstruct shapes which can only be designed efficiently by subtraction. In this example a sphere has to be cut out of the middle of a solid square. The algorithm finds both elements, their orientation and their geometric relation. Note, that the algorithm also finds the optimal tree structure by evolutionary shifting the binary operators to their correct places in the CSG tree.

4.2 NURBS RECONSTRUCTION

In order to illustrate the application of the NURBS-feature of the algorithm, a part of a forging mold of

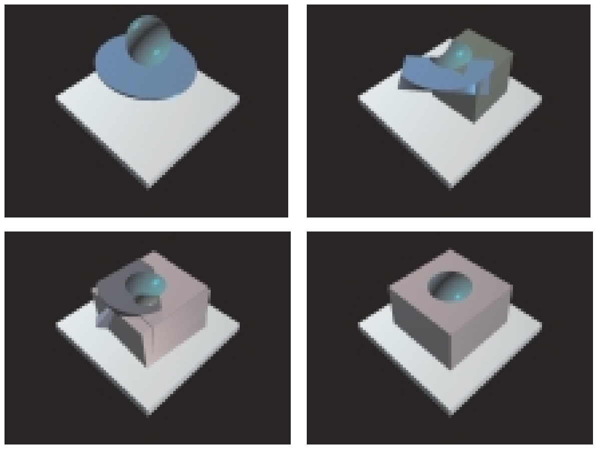

Figure 8: Reconstruction of a Cube with a Cut Out Sphere.

a turbine blade was digitized and reconstructed. The NURBS terminals used 4 by 15 control points and had degree of 3. The starting individual was a plane surface and the number of the underlying sample points was 8 by 24. The result of this reconstruction has a maximum deviation of $0.06\,mm$ and an average of $0.005\,mm$. Furthermore, the algorithm used a population of 500 individuals over 6000 generations. The runtime of about 24 hours (on an average Pentium PC) is due to the complexity of the high-dimensioned search space. I. e. each of the 60 control vertices of the NURBS-patch can move in three directions, which results in 90 parameters which have to be optimized by the system. Figure 9 shows a sequence taken from one run. The best individuals of the population of generation 30, 1000, 3000 and 6000 are displayed.

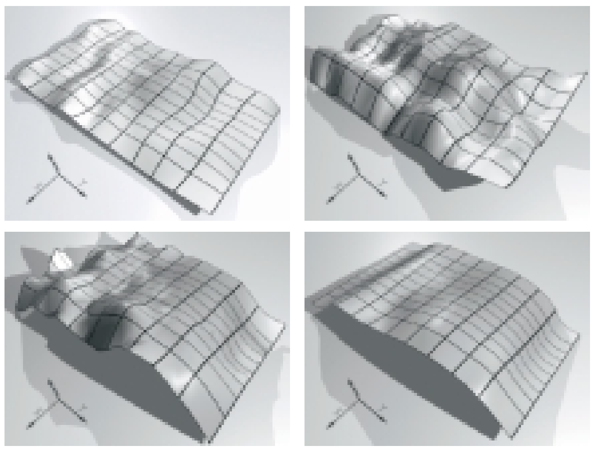

Figure 9: Reconstruction Sequence of a Sculptured Surface

5 CONCLUSION

Surface reconstruction from scanned point data is an urgent problem in the field of die and mold making. Because of its complexity (NP-hardness), surface reconstruction and shape recognition may not be realized by efficient deterministic algorithms. In cases like this, evolutionary algorithms have been proved to be very useful. Therefore, evolutionary surface reconstruction was used. The implemented algorithm uses ES and GP features in a parallel hybrid scheme.

During the reconstruction process two typical evolutionary phases can be distinguished. First the ES/GP-hybrid determines the necessary number and the type of the geometric objects, then the positions of these objects are optimized. In order to have a system which suits to real-world production purposes, it was necessary to integrate sculptured surfaces into the CSG data structure. Here, NURBS excelled for this purpose. Using a multiobjective fitness function and a parallel evolution scheme it was possible to reconstruct parts of a turbine blade.

The parallel evolutionary model uses an island model with sparse communication between the populations. This scheme performed better than panmictic models. The further work will include the heavy parallelization and the development of new and improved mutation operators. This will be done using neural networks to detect relationships between objects. Furthermore, neural networks can be used to preprocess the sampling points in order to find additional attributes which can be used by the algorithm.

Acknowledgments

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

References

[Bäc96] Th. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, NY, USA, 1996.

[Ben99] P. J. Bentley. *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, San Mateo, CA, USA, May 1999.

[FvDFH90] J. D. Foley, A. v. Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics — Principles and Practice (2nd Ed.)*. Addison-Wesley Publishing Company, Inc., 1990.

[GM96] B. Guo and J. Menon. Local shape control for free-form solids in exact CSG representations. *Computer-Aided Design*, 28(6/7):483–493, 1996.

[KMBW99] R. E. Keller, J. Mehnen, W. Banzhaf, and K. Weinert. *Surface reconstruction from 3D point data with a genetic programming / evolution strategy hybrid*, volume 3 of *Advances in Genetic Programming*, chapter 2, pages 41–65. MIT Press, MA, USA, 1999.

[Koz92] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[Meh01] J. Mehnen. *Evolutionäre Flächenrekonstruktion*. Phd thesis, Department of Mechanical Engineering, University of Dortmund, 2001.

[PT97] L. Piegl and W. Tiller. *The NURBS book*. Springer, Berlin, 1997.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley-Interscience, NY, USA, 1995.

[SNL00] R. Sarker, Ch. Newton, and K.-H. Liang. A Multiobjective Evolutionary Algorithm. In *International ICSC 2000 Congress, ISA2000 Intelligent Systems and Applications*, pages 125–131. ICSC Academic Press, Canada, 11.-15. Oct. 2000.

[WM00a] K. Weinert and J. Mehnen. Comparison of Selection Schemes for Discrete Point Data for Evolutionary Optimized Triangulations and NURBS Surface Reconstructions. In *International ICSC 2000 Congress, ISA2000 Intelligent Systems and Applications*, pages 104–110. ICSC Academic Press, Canada, 11.-15. Oct. 2000.

[WM00b] K. Weinert and J. Mehnen. NURBS Surface Approximation of Discrete 3D-Point Data by Means of Evolutionary Algorithms. In R. Teti, editor, *ICME 2000*, pages 263–268, Capri, Italy, 2000. ISBN 88-87030-21-9.