Robert E. Keller, Wolfgang Banzhaf, Jörn Mehnen and Klaus Weinert

Surface reconstruction is a hard key problem in the industrial core domain of computer-aided design (CAD) applications. A workpiece must be represented in some standard CAD object description format such that its representation can be efficiently used in a CAD process like redesign. To that end, a digitizing process represents the object surface as a weakly-structured discrete and digitized set of 3D points. Surface reconstruction attempts to transform this representation into an efficient CAD representation. Certain classic approaches produce inefficient reconstructions of surface areas that do not correspond to construction logic. Here, a new reconstruction principle along with empiric results is presented which yields logical and efficient representations. This principle is implemented as a Genetic-Programming/Evolution-Strategy-based software system.

3.1 Introduction

Genetic programming (GP) [Banzhaf et al., 1998] is an evolutionary search process that generates structures of arbitrary shape and size. The most prominent special case of such a structure is the representation of an algorithm, for instance, as a computer program in a common language like C. The present contribution, however, focuses on GP as an apt tool for the evolution of representations of three-dimensional objects.

The aim of advanced **surface reconstruction (SR)** is the transformation of a physical object into a data representation that meets the high technological requirements of a construction engineer. Such a physical object may be a hand-modeled prototype of a machine part to be produced, and the corresponding data representation may be a computer-aided design (CAD) 3D representation. As an essential part of the associated production process, the engineer modifies this representation with a CAD system so that the resulting representation can be used to produce the corresponding physical object. A CAD system is typically used in mechanical engineering where a physical object, like a prototype of a forging die, often does not have a CAD representation. Especially, the physical object often lacks an exact geometrical representation as a CAD object. This is either because a corresponding CAD data base does not exist, or because the physical object has been changed manually in the course of the production process.

In order to obtain a CAD object, optical or tactile digitizing hardware can be used. For instance, a tactile sensor may systematically scan the physical object's surface. For a practically relevant physical object, a digitizing process generates several megabytes of weakly-structured discrete 3D point data at least. "Weakly-structured" means the 3D point data set does not allow for a trivial recognition of the represented physical object by an automated process.

Modern rapid-prototyping production processes, like high-speed cutting or stereotype lithography, accept surface- or volume-oriented CAD objects and manufacture the corresponding physical object. A surface-oriented CAD object is constructed by combining

CAD surfaces, like a saddle surface, while a volume-oriented object is constructed by combining CAD volumes, like a sphere. Accordingly, there are surface- or volume-oriented CAD systems and hybrid systems used by a construction engineer for operating on such CAD objects. Many CAD objects mainly consist of primitive CAD objects like spheres, cylinders, cuboids or tori. Thus, a CAD system provides corresponding object libraries and supports the manipulation of such objects.

A prominent class of volume-oriented CAD systems employs **constructive solid geometry (CSG)**. The CSG principle is to construct complex CAD objects from primitive objects. The resulting CSG objects represent physical, that is solid, objects. Curved surfaces, like certain parts of car bodies, are typically represented by triangulations, that is a surface approximation by plane triangles. Another well-known representation uses non-uniform rational B-splines (NURBS), which are especially apt for the construction of a smooth curved surface by smoothly joining curved surfaces [Piegl and Tiller, 1997]. NURBS have become the *de facto* standard for smooth-surface representations in the CAD world. They are a powerful tool for geometric design tasks, because they are fast to calculate, numerically stable and allow a rather intuitive use.

Surface reconstruction, that is the automatic construction of a CAD object from data, is a hard and industrially relevant problem. The task being considered in this chapter is the reconstruction from a given 3D point data set. The problem core is that, in $\mathbb{R}^3$, any given set represents infinitely many different *geometrical* surfaces, that is those and only those surfaces that have the set in common. However, the data set represents only one *physical* surface, which is the surface of that physical object from which a digitizing process generated the data set.

Thus, a surface-reconstruction system must reconstruct a CAD object that approximates the physical object. This corresponds to the task of recognizing a physical object in a 3D point set, which is a special case of pattern recognition. The system must perform this task such that a construction engineer can start working with the resulting CAD object without being forced to introduce an expensive manual modification to the representation. Pattern recognition problems are classically covered by artificial intelligence and machine learning. Here, a new and evolutionary approach using a genetic programming/evolution strategy (ES) hybrid is used to reconstruct a CSG object from a non-empty discrete 3D point data set. The approach is represented as the software system SURREAL (Surface Reconstruction by Evolutionary Algorithms).

3.2 Classic context

3.2.1 Digitizing and preprocessing

A digitizing process generates a point-data set, which can be imagined as a point cloud, from a physical-object surface for use by a CAD system. Usually, such a cloud has too many points and is too weakly topologically structured for efficient handling by a CAD system. Thus, preprocessing, like data reduction by chordal deviation [Friedhoff, 1996], depth-pass filtering [Müller and Mencl, 1997], or mesh optimization [Hoppe et al., 1993], is needed. The preprocessed point cloud represents an *approximated* physical-object surface. Preprocessing generates topological information relevant to surface reconstruction. This information has to be calculated only once due to the static nature of the point cloud. Topological information includes normal vectors and Gaussian curvatures which will be explained in section 3.2.2.

Surface reconstruction consists of two key tasks. The first task is to obtain topological information from the preprocessed point cloud by analytic methods. The second task is to construct an *approximating* geometrical surface, that is a surface that approximates the physical surface, from this point cloud by use of the topological information. Typically, a surface-reconstruction method assumes digitizing and preprocessing are topology-preserving, that is the approximated physical surface has a topology close to that of the physical surface. Note that, subsequently, the approximated physical surface will be identified with the physical surface since the former is the best available computer-accessible representation of the latter.

3.2.2 Gridded representation and topological information

A classic surface-reconstruction method uses the idea of constructing an approximating surface with a usually very large number of plane pieces. Triangulation, for instance, yields an approximating surface consisting of triangles as plane pieces. In this context, an intuitive idea of smoothness of the approximating surface is used: the surface is considered "smooth" in a certain area if the angles between the area's plane pieces are not "too wide". The formalized search for a smooth triangulation of a point cloud is hard. Different methods yielding smooth triangulations can be found in [Müller and Mencl, 1997; Weinert et al., 1997a; Schumaker, 1993; Weinert et al., 1997b].

There are several topological properties of a surface that may be used by an advanced classic surface-reconstruction method. In order to determine the peculiar properties of a given surface, a gridded triangulation may be calculated from a smooth triangulation. A triangulation is called "gridded" if and only if the triangle corners are in the normal vectors of the points of a uniform 2D grid. An instance of a gridded triangulation is shown in figure 3.1.

The grid represents a physical plane area on which the physical object rests during the digitizing process. A grid point represents a point that is aimed at by a sensor, like a pin of a tactile digitizing hardware. The sensor aims at this grid point along a vector that is orthogonal to the plane area. For a gridded triangulation, the indicated topological properties can be easier determined than for a non-gridded triangulation. Two examples of such properties are a "normal vector" and "Gaussian curvature".

A normal vector in a surface point is orthogonal to the surface. Since the approximating

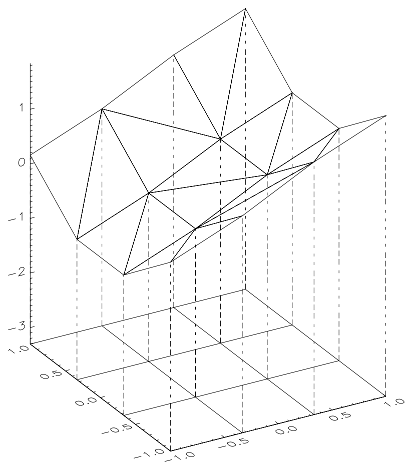

Figure 3.1
Gridded triangulation of a curved surface. Dotted lines represent normal vectors from uniform grid cross points to corresponding triangle corner points.

geometrical surface differs from the approximated physical surface, the normal vector in a point of the approximating geometrical surface must be approximated itself. This approximation can be done, for instance, by help of differential geometrical considerations [Suk and Bhandarkar, 1992] or simulation [Weinert et al., 1998].

The Gaussian curvature $K(p) \in \mathbb{R}$ at any point p on a surface S classifies the curvature of S in p [Gray, 1993]. Put vividly, the value says how strong the curvature is and whether it is convex, concave or saddle-like. $K(p)$ is the product of the smallest and largest normal curvatures in point p. In order to calculate the normal curvatures in p, S is intersected by the normal planes in p. The situation is shown in figure 3.2. One instance e of the infinitely many normal planes in p is given. The curvature of the 2D curve resulting from the intersection is called "normal curvature in p". The product of the smallest and the largest normal curvature in p can be positive (convex or concave surface), negative (saddle surface), or zero (plane surface).

Thus, considering an $n \times m$ grid, the result of the first key task can be represented as an $n \times m$ matrix of vectors $s_{n.m} \in \mathbb{R}^{3+3+1}$. $s_{n.m}$ consists of three subvectors: the coordinate vector $c \in \mathbb{R}^3$ of $p_{n.m}$ which represents a physical surface point over the grid point $n.m$; the normal vector $n \in \mathbb{R}^3$ in $p_{n.m}$; the physical-object surface's Gaussian curvature in $p_{n.m}$.

The topological information is used by a typical classic surface-reconstruction method. It is also used by the evolutionary approach that will be presented next.

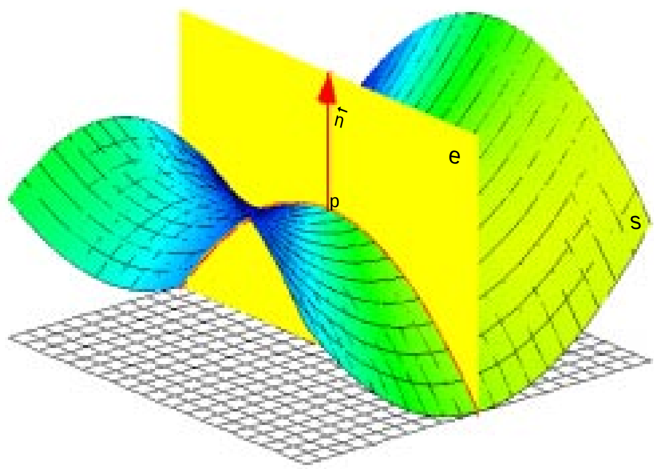

Figure 3.2
A normal plane e with normal vector n intersects saddle surface S, given by function $f(x, y) = x^2 - y^2$, at $p = (0, 0, 0)$.

3.3 SURREAL– a Genetic programming/Evolution strategy hybrid

An evolution strategy is an evolutionary algorithm introduced by Rechenberg and Schwefel in the 1960s [Rechenberg, 1994; Schwefel, 1995]. Its original form and modern variants are powerful tools for solving hard parameter optimization problems as they are ubiquitous in engineering domains.

3.3.1 The approach

A CAD system supports one or more internationally standardized formats for CAD-data exchange, like STEP, VDA-FS, or IGES [Reed et al., 1991]. A geometrical surface can be represented as a CAD expression that obeys one of these formats, while the expression represents an algorithm that constructs the surface. The basic idea of SURREAL is to have genetic programming evolve a CAD expression for a given preprocessed point cloud such that the expression represents the physical surface that underlies the cloud. A significant advantage over classic approaches is that curved physical-surface parts can be approximated by curved geometrical surface parts instead of plane geometrical surface pieces like triangles. This is an important feature since a curved physical-surface part, like the surface of a tube, is frequently used in an industrial production process. Approximating such a part by plane pieces is very inefficient in terms of computing resources. Also, approximating curved physical parts by curved geometrical parts may yield a better approximation. Finally, the resulting approximation looks more natural than a plane-piece approximation that often has a "jagged" appearance.

3.3.2 Overview

When designing an evolutionary algorithm, the following major issues must be dealt with:

- algorithmic structure (generational vs. steady-state model, etc.)
- a problem-specific genetic representation of an individual
- a quality measure, implemented as a fitness function, applicable to each evolvable individual
- a search operator set including creation operator(s) and variation operators like mutation and recombination
- a selection of individuals as operands for the genetic operators
- a selection of individuals as members of the next generation
- parameters like maximal run time, maximal leaf number, population size, operator execution probabilities, etc.

These issues will be discussed subsequently with respect to SURREAL. Especially, parameters will be introduced along with the description of the SURREAL parts they control.

3.3.3 Algorithmic structure

The algorithmic structure is a GP/ES hybrid. After creation of a population with fixed population size $\mu > 0$, a generational-cycle model takes over. A parental generation is transformed into an offspring generation of λ individuals by application of variation and reproduction. After the generation of an offspring generation has been completed, a selection method selects μ individuals from the offspring generation into the next parental generation. Then, the cycle restarts and continues until a user-defined fitness value is scored by an individual or the user terminates the run.

Mutation and recombination are tagged with particular execution probabilities m, r by the user, so that, for instance, recombination will be applied with 0.3 probability as next variation. The sum of all execution probabilities must equal 1, so that the reproduction-execution probability is computed by SURREAL as $1 - m - r$.

3.3.4 Genetic representation

3.3.4.1 Constructive solid geometry

The CAD construction of an object is realized either via line- or surface- or volume-oriented (3D) construction methods. A 3D-CAD system has some advantages in comparison to a surface-oriented CAD system. For instance, a 3D-CAD system gives a realistic visual representation of a physical object, which is a major reason why an increasing acceptance of 3D-CAD systems can be expected.Constructive solid geometry (CSG) is a prominent object-construction method used by several 3D-CAD systems. A CSG object is either a CSG primitive, like a cube, or it is a CSG complex which is represented as a CSG sequence of construction operators and CSG objects. Addition, subtraction, and intersection of CSG objects are typical construction operators. A CSG primitive is characterized by its parameters like size, position, and orientation.

A CSG sequence is a word from a context-free language. For instance, the simplified CSG sequence “sphere $\cup$ sphere” represents the left CSG complex in figure 3.3, obtained by joining spheres. Actually, a real-world CSG sequence as it is used by a CAD system contains parameters like the sphere’s radius and center position. A CSG sequence can be represented as a hierarchical structure, that is a CSG tree, which corresponds to an infix expression. For instance, the infix expression “cube $\cup$ sphere” can be represented as a tree with a $\cup$-labeled root node that has one cube-labeled and one sphere-labeled child node.

Note that, for each CSG object, there is an infinite number of representing CSG trees, which means the CSG-object representation is not unique. For instance, a sphere can be represented as “sphere”, “sphere $\cup$ sphere”, “sphere $\cup$ sphere $\cup$ sphere”, etc., with all spheres having the same radius and position.

3.3.4.2 Terminal and function set

SURREAL’s terminal set contains the CSG primitives “box”, “sphere” and “quadric”. A quadric is a generic geometric object that can be instantiated as, for instance, a cone or a cylinder. As a special case of a finite quadric, the terminal set contains a cylinder. Other stereometrical primitives like a torus, an obelisc, barrel-shaped bodies, and primitives with non-trivial topologies can be included into the terminal set, if desired. The function set contains the binary construction operators “union”, “subtraction”, and “intersection”.

An example of three CSG complexes can be seen in figure 3.3.

3.3.4.3 Search space

Due to the genetic representation, the search space is the set of all CSG trees with a user-defined maximal leaf number c. SURREAL does not evolve trees with more leaves, that is CSG primitives. The user must set c such that a SURREAL run does not exhaust the available memory.

Note that the CSG-based genetic representation ensures that each CSG object can be represented as an individual. This again ensures that almost any practically relevant physical

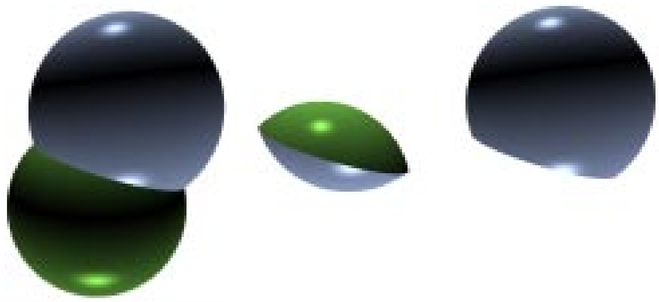

Figure 3.3
Union, intersection, and subtraction of two spheres

object can be represented as an individual, since CSG has been designed for the representation of such physical objects. Note also that an arbitrary search space point (a CSG tree) can be generated by an apt sequence of variation operators. This ensures that, in principle, every potential solution can be found by the search process, no matter where in space the search starts. Finally, the genetic representation of an individual can be transformed by a tree traversal into a CSG sequence which can be directly processed by a corresponding CAD system.

3.3.5 Quality measure

Surface reconstruction is an instance of a multi-criterion optimization problem. In the context of surface reconstruction, one obvious optimization criterion is the quality of the approximation of a physical object by a corresponding CSG object resulting from the reconstruction. Another criterion is the parsimony of the CSG tree that represents the approximating CSG object. Since the CSG-object representation is not unique, CSG trees of vastly different sizes represent the same CSG object. In order to save CAD system resources, the representing CSG tree should be small.

Further optimization criteria are related to topological properties of the CSG object and will be explained next. The quality measure is a combination of all optimization criteria, and it is used to assess the overall quality of an individual. It is implemented as a fitness function used by SURREAL. An instance of such a function is shown in section 3.4.2.

3.3.5.1 Distance criterion DELTA

This criterion represents the idea that a well-approximating CSG-object surface should contain the points from the preprocessed point cloud. Subsequently, a z-value of a cloud point is the point's height over the corresponding grid point. If this value is greater or equal to a digitizing-hardware-dependent $\epsilon > 0$, this means the digitizing process detected a physical-object surface over that grid point. A vanishing z-value $0 \leq z < \epsilon$ means the digitizing process detected no physical-object surface over that point.

The concept of a vanishing value must be introduced due to the unavoidable imprecision of the physical measuring a digitizing process must perform. For instance, the surface of the table on which a physical object rests during digitizing may have a small unevenness. If this unevenness is digitized, the resulting 3D point data represent noise since the unevenness is not part of the physical object. However, if one sets ϵ high enough with respect to table imperfections and low enough with respect to the physical object, only the physical object will be digitized.

The distance criterion is realized via the sum σ over all grid points of the differences between a physical-object surface point's z-value and the corresponding CSG-object-surface point's z-value. Thus, the time complexity of the computation of DELTA is linear in the number of all $n \cdot m$ grid points. The result is normalized to the interval $0 < x <= 1$ by use of

$$DELTA := (\sigma + 1)^{-1}.$$

Thus, a perfect CSG object results in $DELTA = 1$ since, in this case, σ equals zero.

Note two special cases that come into existence because not each grid point is necessarily covered by a CSG-object. A grid point with no corresponding CSG-object surface point and a vanishing z-value means SURREAL models the situation in this grid point correctly: no physical-object surface – no CSG-object surface. A grid point with a non-vanishing z-value and no corresponding CSG-object surface point means the situation has not yet been modeled correctly: a physical-object surface – no CSG-object surface. Thus, the error in this grid point, represented as an addend of σ, equals the z-value.

3.3.5.2 Angle criterion ABN

This criterion represents the idea that the CSG-object surface should have the same spatial orientation as the physical-object surface. It is realized by taking into consideration the normal vectors of the CSG-object surface points and of the physical-object surface points over all grid points. The physical-object normal vectors have to be computed only once at the beginning of a SURREAL run since the physical-object representation (the preprocessed point cloud) is constant. The criterion is realized as a normalized sum over all grid points. The time complexity of the computation of ABN is linear in the number of all $n \cdot m$ grid points.

An addend of the normalized sum is the absolute cosine value of the angle α_i between the CSG-object normal vector and the physical-object normal vector over the same grid point.

$$ABN = \frac{1}{n \cdot m} \sum_{i=1}^{n \cdot m} |cos\ \alpha_i|.$$

The cosine for each normal-vector pair yields "one" for parallel normal vectors and zero for orthogonal normal vectors, which implies that ABN is in $[0, 1]$ and a perfect CSG object has ABN value "one".

3.3.5.3 Curvature-type criterion CTYPE

A Gaussian curvature type (section 3.2.2) classifies a surface point as belonging to a plane, convex, concave or saddle surface. The curvature types are calculated by preprocessing (section 3.2.1) before the evolutionary algorithm starts. The CTYPE criterion represents the idea that, in corresponding points, the CSG-object surface should have the same curvature type as the physical-object surface.

The purpose of CTYPE is to guide the selection of an apt CSG-primitive for the approximation of a certain physical-object surface area such that primitive and area have identical curvature types. CTYPE is defined as the sum over all grid points of type matches, rated as "one", and mismatches, rated as "zero", divided by the number of grid points. Thus, CTYPE is in $[0, 1]$, and a perfect CSG object has CTYPE value "one", since each grid point represents a type match. The time complexity of the computation of CTYPE is linear in the number of all $n \cdot m$ grid points.

3.3.5.4 Primitive-number criterion PRIM

Evolution may lead to the generation of a CSG object that consists of a huge number of primitives ϕ. Such an object is inefficient in terms of computing resources, since each data representation of a primitive must be stored. Moreover, a CAD system must draw each primitive when drawing the CSG object, and redraw whenever the construction engineer moves or rotates the object on-screen. The more realistic a CSG object appears on screen by application of visual rendering algorithms, like ray tracing, and the higher ϕ is, the more CPU intense the redrawing gets. Even with powerful graphics hardware support, the delay due to redrawing may become annoyingly long.

Thus, the parsimony criterion PRIM is needed in order to guide the search process to small individuals. PRIM is defined as

$$PRIM := 1 - \frac{\phi}{c},$$

where $\phi \leq c \leq n \cdot m, c \in \mathbb{N}$. c has been introduced in section 3.3.4.3. $0 \leq PRIM \leq 1 - 1/c$ holds, and a perfect CSG object has $1 - 1/c$ as PRIM value. c should be set as high as the maximum number of leaves expected necessary to build an acceptable CSG object.

3.3.6 Variation

3.3.6.1 Mutation

A single mutation of an individual has exactly one of the following types:

- **primitive** It random-selects a CSG primitive (CSG tree leaf) and then randomly modifies one parameter, that is either position or size or orientation, with normal distribution for small modifications. Its objective is to tune the evolved geometric surface with many smaller and few bigger changes.

- **construction** It replaces a random-selected construction operator (inner node of a CSG tree), like intersection, by another random-selected construction operator. This mutation type usually introduces a major phenotypic change as can be seen in figure 3.3. The objective is to introduce several topologically qualitatively different phenotypes by repeated application to different individuals in the course of evolution. This, hopefully, generates a phenotype that captures the characteristics of the physical object to be approximated, so that the corresponding genotype may then be tuned.

- **replacing** It replaces a random-selected CSG primitive by another random-selected primitive. Its objective is to tune the evolved geometric surface by exchanging subareas. The parameter settings of the introduced primitives are defined by standard settings.

- **insertion** It replaces a randomly selected CSG primitive by a randomly generated CSG tree which consists of two nodes at least. Its objective is to introduce a significant change of the phenotypic shape. Especially, this operator is imperative for introducing CSG complexes into the population, since creation, as will be discussed later, only generates primitives.

- **deletion** It deletes a random-selected non-trivial subtree that is not the entire individual. Its objective is identical to the insertion objective and it counteracts the genotypic size increase introduced by insertion, thus supporting the evolution of parsimonious individuals.

The mutation-execution probability is controlled by use of an adjustment function that exponentially decreases the user-defined initial value over run time. The decreasing occurs in order to have the search process “home in” on an acceptable local optimum. If mutation is to be the next variation, one of the five presented mutation types is selected for execution. The respective five execution probabilities are user-defined. If “primitive” is selected for execution, one of the parameters (position, size, orientation) is selected for mutation. The respective three selection probabilities are user-defined.

3.3.6.2 Recombination

Recombination is defined as an exchange of two CSG subtrees between two parents. The root nodes of these subtrees are being random-selected such that each node in a parent has equal probability of being selected. This way, every combination of subtrees to be exchanged has the same probability of occurring. Thus, the phenotypic difference between parents and offspring covers the full range from almost identical to very different in shape and size. This implies that recombination, like mutation, generates and maintains genetic diversity in the population. As usual with evolutionary algorithms, genetic diversity is important for preventing SURREAL from premature convergence. Especially, diversity supports the evolution of an ideal phenotype, which is as complex as necessary with respect to approximation quality and as parsimonious as possible.

Furthermore, recombination is expected to be useful since it can construct a complex surface structure (CSG tree) from previously evolved substructures (subtrees).

3.3.7 Creation

A creation operator generates a user-defined number of individuals each of which being a randomly chosen CSG primitive with random parameter settings. This ensures, a reasonably large population size assumed, the occurrence of all primitive types in the initial population. This type diversity is helpful for the synthesis of complex topologies as they are ubiquitous in practically relevant physical objects. Note that structural diversity does not have to be introduced by creation. Instead, the insertion operator described in section 3.3.6.1 quickly introduces diversity into the initial population. Thus, the implementation and application of one or more relatively sophisticated tree creation operators like those described in [Koza, 1992] are unnecessary.

3.3.8 Selection for variation

Once creation or previous evolution have produced a generation of individuals, variation in form of recombination and mutation takes place.

In order to perform recombination, two different parents are selected from the population at random with equal probability. This fitness-independent selection mechanism guarantees that recombination is likely to access also those good substructures that are contained in mediocre or bad individuals. This purely random-based selection mechanism would result in a Monte-Carlo-like search process if it was the only such mechanism present in the system. However, in order to determine the individuals to be subjected to mutation, SURREAL offers four fitness-based selection mechanisms to the user who chooses one that will be used during a run:

- elitist selection
- ranking selection under an exponential distribution
- 2-tournament selection
- fitness-proportional selection

Elitist selection chooses the n best individuals for mutation. Ranking selection under an exponential distribution chooses individuals with an exponentially distributed probability that depends on the fitness rank of an individual within the population. The dependency is such that the lower the rank is, the exponentially lower is the probability of selecting the individual for mutation. 2-tournament selection chooses two individuals from the population at random with an equally distributed probability. The better individual will be mutated. Fitness-proportional selection chooses an individual for mutation with a probability proportional to its fitness.

3.3.9 Selection for the next generation

Once variation has taken place, selection of individuals to be contained in the next generation occurs. Here, SURREAL follows the evolution strategy. The classic evolution strategy introduces a selection pressure on the genotypes via using the plus selection and the comma selection. These selection methods *deterministically* select the μ best individuals for the next generation. Plus-selection selects from $(\mu + \lambda)$ parental and offspring individuals, and comma selection selects only from $\lambda \geq \mu$ offspring individuals. Thus, in order for the comma selection to work, the genetic operators must generate $\lambda \geq \mu$ offspring from μ parents.

SURREAL offers the use of either the plus-selection or the comma-selection in a run. Let us call a SURREAL run a $(\mu + \lambda)$ run when it employs a plus-selection.

An evolutionary algorithm using these selection methods allows for a simple and effective tuning of the selection pressure by adjusting the μ/λ ratio: the larger the ratio, the higher the selection pressure. A high selection pressure yields fast but unsafe search process convergence to an acceptable genotype. A low pressure leads to the opposite: a Monte-Carlo-like slow search process featuring safe convergence when being given very much run time. A theory yielding a good choice for the selection pressure is currently unknown. A good rule of thumb is to choose a ratio of $5/7$ with $\lambda >> 50$ for difficult problems.

Note that, simultaneously, there is beneficial and detrimental potential in both selection methods. Plus-selection may result in a very long existence of a good but still not acceptable individual. This may lead to a critical and fast collapse of genotypic diversity via iterated reproduction of this individual. However, it may also lead to an increase of hi-fitness genetic information which finally may result in the evolution of an acceptable genotype. Comma-selection may result in the situation that the best genotype of the next parental generation is worse than that of the previous one. However, the risk of a population takeover by a good but not acceptable individual is lower than with plus-selection [Bäck, 1994].

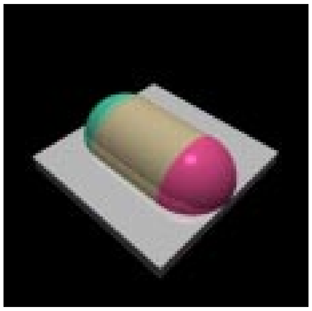

Figure 3.4
Manually constructed CSG object representing a dowel. The plane surface represents the digitizing table.

3.4 Results

3.4.1 Problem: dowel reconstruction

Subsequently, SURREAL will be applied to the surface reconstruction of a simple but practically relevant physical object: a dowel, which is a ubiquitous part in mechanical engineering. Figure 3.4 shows a manually constructed CSG object representing a dowel. The CSG object can be constructed from a cylinder and a half-sphere at each cylinder end.

There are infinitely many CSG trees representing a dowel. A parsimonious one consists of 5 nodes: three leaves for the primitives “left half-sphere”, “right half-sphere”, “cylinder”, and two inner nodes for the construction operators “join”.

3.4.2 Parameters

The required grid dimension, that is resolution, depends on the structural complexity of the physical object to be reconstructed. If the resolution is too low, a critical physical surface area may not be digitized. If the resolution is too high, redundant data is being generated during the digitizing process. This redundancy is detrimental, since it results in an unnecessary use of the limited computing resources. For the problem of the dowel reconstruction, a grid dimension of 20×20, which appears to be beneficial to the search process, has been found experimentally.

The structural complexity of the physical object influences the maximum number of leaves a CSG tree should have in order to allow for a successful evolution. A number c, as introduced for the PRIM definition, of 220 is used for the problem. This value has been found empirically like the grid dimension.

The fitness function f is defined as

$$f = 1.5 \cdot CTYPE + 0.5 \cdot ABN + 0.5 \cdot DELTA + 0.5 \cdot PRIM.$$

Thus, an f value is in $0 < f \leq 3 - 0.5/c$, and a perfect CSG object scores the upper-interval-limit value. Judging by the results of numerous trials, the constants used in the fitness function improve convergence speed and reliability of the search process.

Experiments with $(50 + 50)$, $(100 + 300)$, and $(500 + 800)$ runs all resulted in elitist strategies yielding best solutions, shortly followed by tournament selection strategies. Fitness-proportional selection and ranking selection schemes did not perform well which endorses results reported in [Project team SURREAL, 1998].

Each selected individual is either mutated or recombined with 0.5 probability. Thus, in the beginning of a run, the reproduction-execution probability equals zero. This leads to an initially strong generation of genetic diversity, since reproduction cannot yet lead to a premature population take-over by relatively good individuals. Gradually, reproduction starts taking effect, when the mutation-execution probability is decreased by use of the adjustment function.

The mutation-execution probabilities (section 3.3.6.1) of primitive, construction, replacing, insertion and deletion are 0.09, 0.01, 0.1, 0.5 and 0.3 respectively. Construction mutation usually introduces vast phenotypic changes, so that its low execution probability is needed to home in slowly on a local optimum in order not to lose track of it. The probabilities for selecting position, size or orientation during a primitive mutation are 1/3 each.

3.4.3 Discussion

3.4.3.1 Incremental optimization

Note that, in the fitness function, CTYPE clearly has the largest weight, which is due to the experience that an adequate amplification of the curvature-type weight usually yields better results. This high weight forces an early optimization of the CSG-object shape and size, while orientation, evaluated by ABN, and position, evaluated by DELTA, are targets for final tuning. Accordingly, when watching an animated picture sequence assembled from the best-in-generation phenotypes from the first to the last generation of a typical run, one often sees a corresponding structural evolution. This change is reflected by the genotypic size progression, which can be seen in figure 3.5.

One can interpret the sequence of the best-in-generation phenotypes from the first to the last generation of a run as snapshots taken of the structural and spatial evolution of a meta-individual. Initially, this individual assumes the topology of the physical object. During this period, the genotypic size increases rapidly, while the surface of the physical object is being approximated roughly by a set of primitives. These primitives have coarse curvature-type relations with physical-surface areas. In this phase, the CTYPE criterion

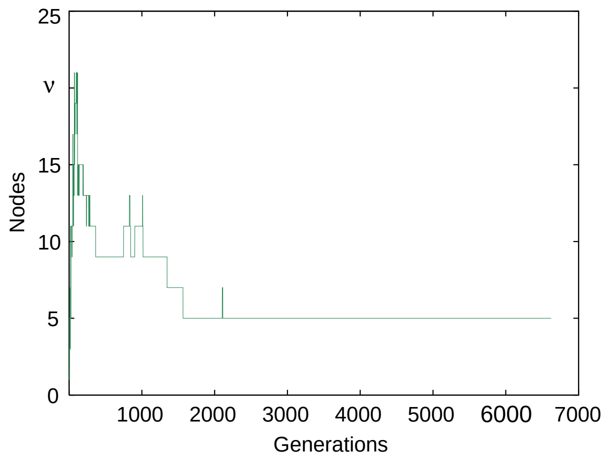

Figure 3.5
Progression of the genotypic size of the best-in-generation genotypes during dowel reconstruction. Genotypic size measured in number of CSG tree nodes.

implies topology optimization of the CSG object. Then, the meta-individual rotates into the orientation the physical object had when it was scanned. Due to the PRIM criterion, the number of primitives is being reduced rapidly while the structure of the CSG object is being refined. In this phase, the ABN criterion implies orientation optimization of the CSG object. Finally, after the structure of the physical object has been recognized, the CSG object floats into the position the physical object had relative to the grid. The distance criterion DELTA implies position optimization.

A typical structure evolution of a meta-individual, evolved during a $(50 + 50)$ run with elitist selection, can be seen in figure 3.6.

This type of structure evolution is an instance of incremental optimization, which is a powerful approach to many multi-criteria optimization problems. The idea is to water down the difficulty of the entire task by sequentially solving subproblems. This is effected by a quality measure that is composed of differently weighted elements. We return to this issue below in the context of fitness progression.

3.4.3.2 Fitness progression

The following discussion is concerned with an experiment that consists of 10 $(50 + 50)$ runs with elitist selection, initialized with different randomizer seeds. Each run lasted for, at least, $3,000$ generations and found a perfect CSG object. Fig. 3.7 shows the result of a representative run that was terminated after $6,600$ generations.

Visually, no differences between the physical object and the CSG object can be detected. Especially, SURREAL has recognized the construction logic of a dowel and implemented

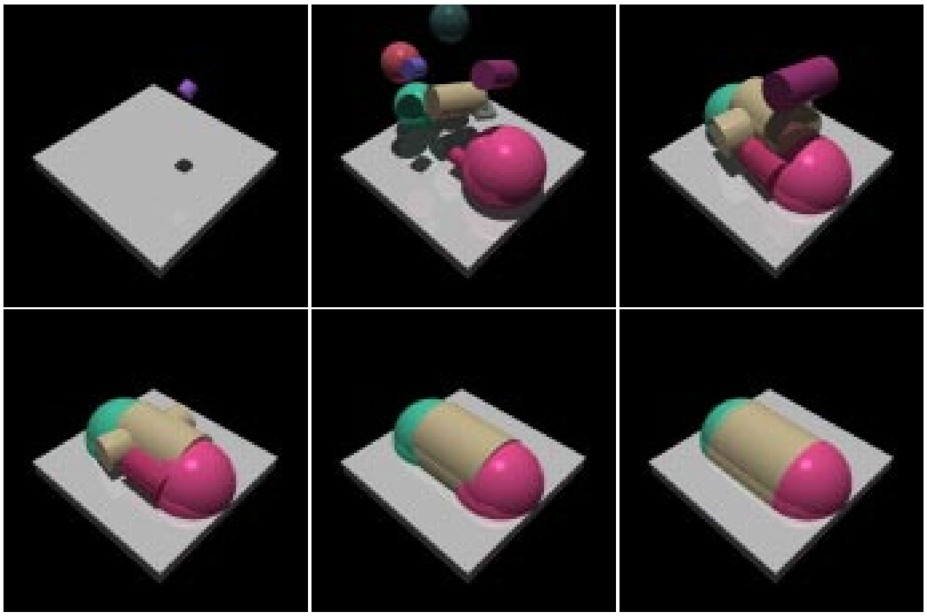

Figure 3.6
From upper left to lower right: snapshots of meta-individual taken in generations 1, 200, 500, 800, 1100, 1872 of a $(50 + 50)$ run with elitist selection. The plane areas represent the digitizing table. Note: in the bottom right picture, the dowel seems to rest on top of another dowel, which would be a bad reconstruction. Actually, however, the alleged lower dowel is a mirror image produced by the visual rendering algorithm that assumes the digitizing table to reflect light. Thus, the bottom right picture shows a perfect reconstruction. An animation of this reconstruction resides at `http://ls11-www.informatik.uni-dortmund.de/people/keller/surreal.html`.

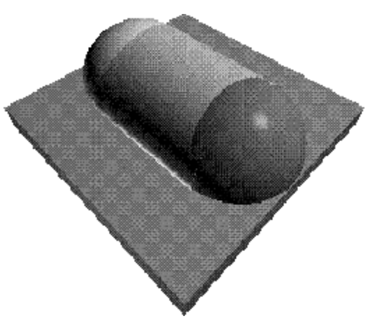

Figure 3.7
Perfect SURREAL reconstruction of a dowel in generation 5,159 of a $(50 + 50)$ run with elitist selection. The plane under the dowel represents the table of a digitizing unit where an object is placed for scanning.

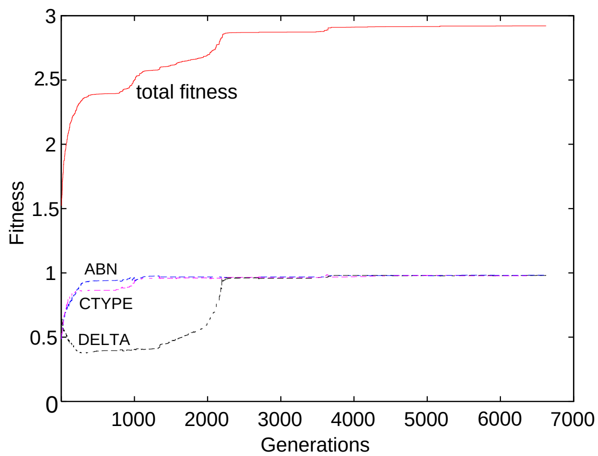

Figure 3.8
Single-criterion-value progressions and the total-fitness progression of the best individual in each generation, taken from the reconstruction of the dowel shown in figure 3.7

it in the evolved CSG object: a half-cylinder, closed by quarter-spheres. The good visual impression is endorsed by the single-criterion-value progressions and the total-fitness progression of the best individual in each generation with the final best values 0.979363 (DELTA), 0.980912 (ABN), and 0.982 (CTYPE). The progressions are displayed in figure 3.8.

Initially, rapidly increasing values of CTYPE and ABN reflect the structure evolution of the meta-individual during an exploring phase of the search process. Then, after the difficult topology and orientation optimization has been more or less completed, the easier position optimization is being dealt with, mirrored by the late start of the DELTA value increase.

3.4.3.3 Population size and convergence

Progressions of the best-individual-in-generation fitness of a representative $(100 + 300)$- and $(500 + 800)$ run using elitist selection are displayed in figure 3.9 on a logarithmic generations scale.

The progressions endorse the rule of thumb that larger populations strongly enhance convergence speed. Both runs evolve the same genotypic optimum, but the $(500 + 800)$ run uses only 35% run time of the time consumed by the $(100 + 300)$ run.

3.4.3.4 Interactive evolution

In the presented runs, SURREAL found the solution without human interaction. Since interactive evolution [Banzhaf, 1997] may be desired by a user in order to support the search

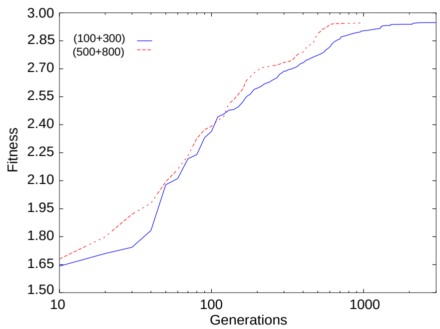

Figure 3.9
Progressions of the best-individual-in-generation fitness of a representative $(100 + 300)$ and $(500 + 800)$ dowel-reconstruction run with elitist selection on a logarithmic generations scale.

process by introducing human knowledge, the system offers a corresponding user interface with several features:

- **Genotypic redesign** The user may interrupt a run, select an evolved genotype, and order the visual rendering of the corresponding phenotype. The user may then redesign the genotype during a modification/rendering cycle until the redesigned phenotype has satisfactory quality. He or she may finally introduce the redesigned genotype to the current population and restart the halted run. Note that this feature has not been used during the reconstruction runs presented in this chapter.

- **Parameter setting** The result-enhancing setting of the run parameters, like the population size, is a difficult task for the user, requiring much experience. Successful parameter settings depend on the structural complexity of the object. Inadequate settings yield CSG objects with too many primitives that approximate the physical object only poorly. A TCL/TK interface provides a comfortable way to set, change, and save parameter settings for analysis. Especially, a setting may be changed during a run.

- **On-line visualization** The system can generate on-line pictures in TGA (Truevision Targa Image File Format) to give an on-the-fly impression of the search progress. For instance, the image sequence shown in figure 3.6 has been generated by use of the on-line visualization.

3.4.4 Problem: cross structure reconstruction

As another problem, the reconstruction of a cross structure shown in figure 3.10 shall be presented.

Cross structures are frequently needed components of automatic-transmission control units. The structure at hand is a part of a transmission case. The most parsimonious CSG

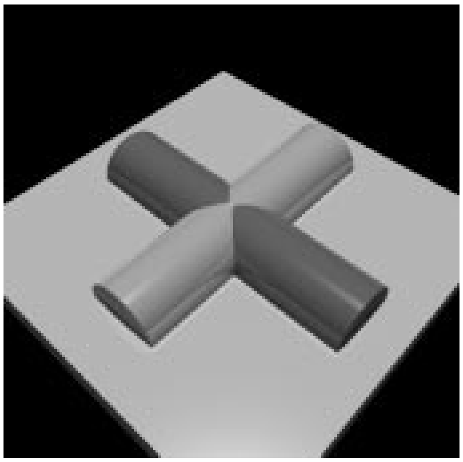

Figure 3.10
Cross structure

tree consists of 3 nodes: two leaves for the primitives “half-cylinder” and an inner node for the construction operator “union”.

Especially, $(100 + 300)$ runs are performed. The mutation probability is set to 0.01 only, while the recombination probability is set to 0.5. Thus, recombination and reproduction are the most frequently used operators. The mutation-execution probabilities (section 3.3.6.1) of primitive, construction, replacing, insertion and deletion are 0.2, 0.05, 0.1, 0.5 and 0.3 respectively. The probabilities for selecting position, size or orientation during a primitive mutation are 1/3 each. Subsequently, the results of a particular run are presented. The genotypic size progression can be seen in figure 3.11.

The according structure evolution of the meta-individual can be seen in figure 3.12.

The progressions of the total fitness of the best individual and of the average total fitness are shown in figure 3.13.

The small difference between both values per generation is due to the high reproduction probability compared to the very low mutation probability. Thus, a good individual quickly takes over the population, rising the average fitness. The close linkage of best and average fitness values is also present in the single criterion value progressions as can be seen in figure 3.14. At generation $10\,000$, the top, middle and bottom double curves show the CTYPE, DELTA, and ABN progressions, respectively.

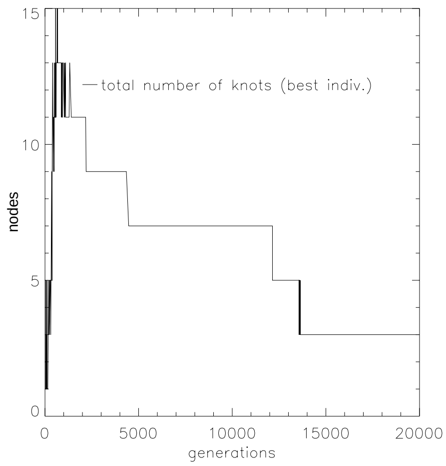

Figure 3.11
Progression of the genotypic size of the best-in-generation individuals during cross structure reconstruction with a $(100 + 300)$ run.

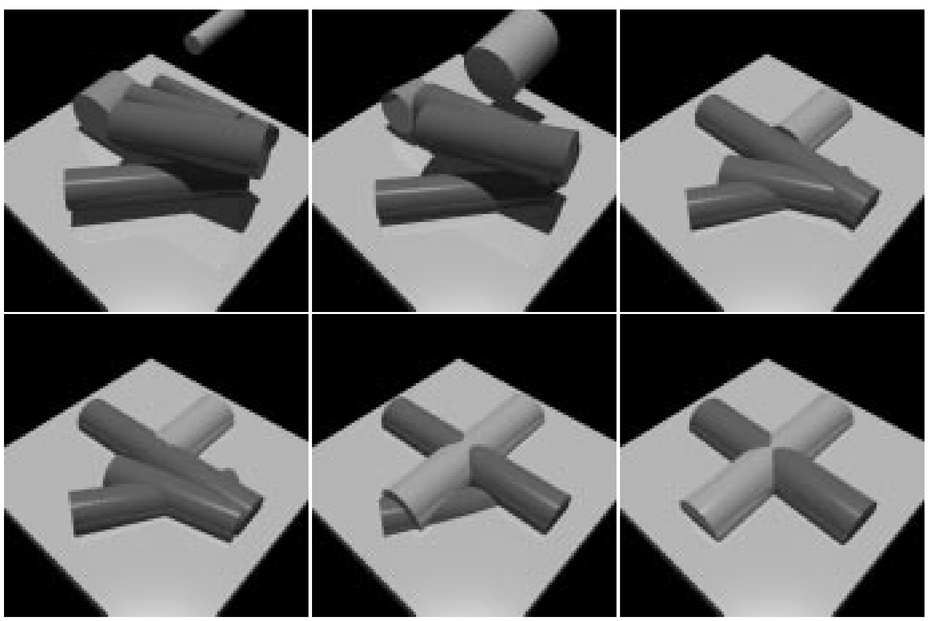

Figure 3.12
From upper left to lower right: snapshots of meta-individual taken from 6 generations. Last generation: $14,400$

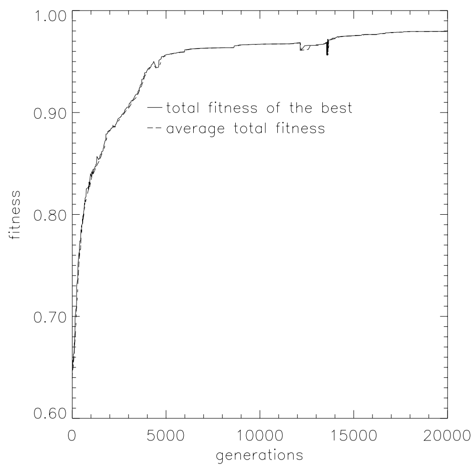

Figure 3.13
Fitness progressions. The average-total-fitness curve runs close below the best-total-fitness curve.

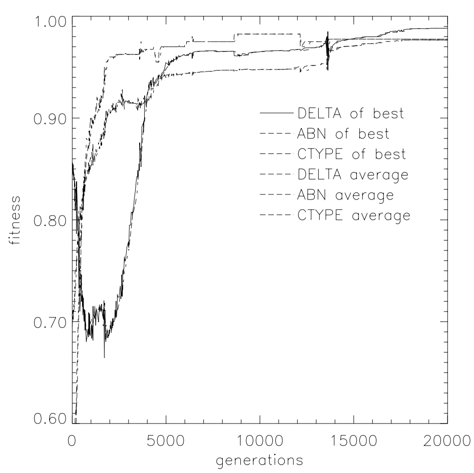

Figure 3.14
Single criterion value progressions. Each average-value curve of a criterion runs close below the best-value curve of the same criterion. At generation 10 000, the top, middle and bottom double curves show the CTYPE, DELTA, and ABN progressions, respectively.

3.5 Conclusion and future work

It has been shown that surface reconstruction by means of Genetic Programming is possible. With the software system SURREAL, a Genetic Programming/Evolution Strategy surface reconstruction approach has been introduced for the first time. SURREAL transforms a weakly-structured digitized 3D representation of a physical object into an efficient and construction-logical standard CAD-object representation. The presented approach simultaneously performs pattern recognition and structure evolution. Genetic Programming is especially apt for such tasks since it can operate in search spaces with *a priori* unknown dimensionality. Especially, the evolution of CSG trees of arbitrary size and shape is possible. Accordingly, the reconstruction of more complex objects is the major target of future work.

Connected with this primary objective, the following issues are identified:

- **Alternative genetic representations** In order to use the synergy of surface- and volume-based CAD representations for the reconstruction of arbitrary surfaces, further representations like NURBS (see section 3.1) should be evolvable, too.

- **Knowledge-based search operators** Complex objects, consisting of many small objects, dramatically increase the search-space size. A possible answer to this challenge is to supply the search process with "intelligent" search operators that use domain-specific knowledge when being applied. For instance, simple rules to be observed by an operator are "each part of a generated object must be connected to at least one other part of this object", "an object must not be completely contained within another object" and "the object to be reconstructed has a maximal height H over the grid".

- **Duplication** Many practically relevant physical objects have symmetric or fractal partial topologies. Duplication, as an additional mutation type, of an existing genotypic part should be beneficial to the evolutionary reconstruction of such objects.

Note that SURREAL could reconstruct arbitrary, also non-artifact, physical objects like human body parts, especially faces. The reconstruction of these objects represents a difficult problem for classic reconstruction approaches due to their highly irregular surface topology. Applications in, for instance, cosmetics, medicine, entertainment, and person identification are conceivable.

Acknowledgments

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (SFB 531) projects C4 and B2. Special thanks go to the students' project team SURREAL, whose members have implemented the system under the guidance of the authors.

Bibliography

Bäck, T. (1994), "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE Press.

Banzhaf, W. (1997), "Interactive evolution," in *Handbook of Evolutionary Computation*, Z. M. T. Bäck, D. Fogel (Ed.), pp C2.9,1–7, Oxford University Press.

Banzhaf, W., Nordin, P., Keller, R., and Francone, F. (1998), *Genetic Programming - An Introduction; On the Automatic Evolution of Computer Programs and Its Application*, dpunkt-Verlag, Heidelberg. Morgan Kaufmann, San Francisco.

Friedhoff, J. (1996), *Aufbereitung von 3D-Digitalisierdaten für den Werkzeug-, Formen- und Modellbau*, Essen: Vulkan Verlag.

Gray, A. (1993), "The gaussian and mean curvatures," *Modern Differential Geometry of Curves and Surfaces, Boca Raton, FL: CRC Press*, pp 279–285.

Hoppe, H., DeRose, T., and Duchamp, T. (1993), "Mesh optimization," in *Computer Graphics Proceedings, Annual Conference Series*, pp 21–26, New York: ACM Siggraph. Proceedings of Siggraph '93.

Koza, J. (1992), *Genetic Programming, On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Massachusetts.

Müller, H. and Mencl, R. (1997), *Interpolation and Approximation of Surfaces from Three-Dimensional Scattered Data Points*, Dortmund: University of Dortmund, Research Report, No. 662.

Piegl, L. and Tiller, W. (1997), *The NURBS Book*, Springer.

Project team SURREAL (1998), "Flächenrekonstruktion mit Genetischem Programmieren," Technical report, University of Dortmund, Computer Science Department, Dortmund, Germany.

I. Rechenberg (Ed.) (1994), *Evolutionsstrategie '94*, volume 1 of *Werkstatt Bionik und Evolutionstechnik*, Stuttgart, Frommann-Holzboog.

Reed, K., Kelly, J. C., Harrod, D., and Conroy, W. (1991), *The Initial Graphics Exchange Specification (IGES) Version 5.1.*, Fairfax, VA: National Computer Graphics Assoc., Administrator-IGES/PDES Organization.

Schumaker, L. (1993), "Computational triangulation using simulated annealing," *In: Computer Aided Geometry Design*, 10:329–345.

Schwefel, H.-P. (1995), *Evolution and optimum seeking*, Wiley Interscience.

Suk, M. and Bhandarkar, S. (1992), *Three Dimensional Object Recognition from Range Images*, Berlin: Springer.

Weinert, K., Mehnen, J., Albersmann, F., and Drerup, P. (1997a), *New Solutions for Surface Reconstructions from Discrete Point Data by means of Computational Intelligence*, Dortmund: University of Dortmund, Technical Report, CI-22/98, SFB 531 (Collaborative Research Center "Computational Intelligence"), ISSN 1433-3325.

Weinert, K., Mehnen, J., and Prestifilippo, G. (1997b), *Optimal Surface Reconstruction from Digitized Point Data using CI Methods*, Dortmund: University of Dortmund, Technical Report, CI-5/97, SFB 531 (Collaborative Research Center "Computational Intelligence"), ISSN 1433-3325.

Weinert, K., Müller, H., Albersmann, F., and Mehnen, J. (1998), "Efficient Rasterized Offset Computation from Tactile Digitization," in *Proceedings of the ISMTII'98, Sep. 2nd to 4th 1998*, pp 258–264, Miskolc, Hungary: University of Miskolc.