Reconstructing Occlusal Surfaces of Teeth Using a Genetic Algorithm with Simulated Annealing Type Selection

Vladimir V. Savchenko
Faculty of Computer and Information Sciences,
Hosei University
3-7-2, Kajino-cho, Koganei-shi, Tokyo 184-8584,
Japan
Phone: 81-42-387-4537
e-mail: vsavchen@k.hosei.ac.jp

Lothar M. Schmitt
School of Computer Science and Engineering
The University of Aizu
Aizu-Wakamatsu City, Fukushima, 965-8580, Japan
Phone: 81-24-237-2577
e-mail: lothar@u-aizu.ac.jp

ABSTRACT

In this paper, we present an application of numerical optimization for surface reconstruction (more precisely: reconstruction of missing parts of a real geometric object represented by volume data) by employing a specially designed genetic algorithm to solve a problem concerning computer-aided design in dentistry. Using a space mapping technique the surface of a given model tooth is fitted by a shape transformation to extrapolate (or reconstruct) the remaining surface of a patient's tooth with occurring damage such as a “drill hole.” Thereby, the genetic algorithm minimizes the error of the approximation by optimizing a set of control points that determine the coefficients for spline functions, which in turn define a space transformation. The fitness function to be minimized by the genetic algorithm is the error between the transformed occlusal surface of the model tooth and the remaining occlusal surface of the damaged (drilled) tooth. The algorithm, that is used, is based upon a proposal by Mahfoud and Goldberg. It uses a simulated-annealing type selection scheme, which is applied sequentially (pair-wise, or one-by-one) to the members in the parent generation and their respective offspring generated by mutation-crossover. We outline a proof of convergence for this algorithm. The algorithm is parallel in regard to computing the fitness-values of creatures.

Keywords

Constructive solid geometry, volume modeling, space mapping, surface reconstruction, computer-aided restoration design, genetic algorithm, simulated annealing.

INTRODUCTION

In this paper, we present an application of 3D-space mapping technique and numerical optimization with a specially designed genetic algorithm to a problem concerning computer aided design (CAD) in dentistry.

In the application considered here, the optimization task is to find a functionally transformed occlusal surface (an inlay part) of a model tooth that best matches the remaining occlusal surface of a treated tooth following an approach proposed in [25].

Many practical surface reconstruction techniques based on measured data points require the solution of optimization problems in fitting surface data. The application considered in this paper, i.e., CAD in dentistry, is a particularly attractive research direction: it makes it possible to both shorten the treatment time down to a single patient visit and address the demand for more esthetic dentistry using ceramics. One of the important aspects of restoration design is the careful reconstruction of occlusal surfaces of teeth that not only fit to the existing articulation, but also preserves the main topological features of the occlusal surface. Reconstruction of the occlusal surface for onlay/inlay is an even more demanding task because the missing part of the surface taken, for example, from a generic (model) tooth must smoothly join with the real surface at the common boundary; i.e., such a surface is constructed by blending the missing part and the remaining one. The principal difference between standard industrial applications and occlusal surface modeling of teeth is as follows: in industrial applications the designers take into account primary natural, simple mathematical properties such as curvature or continuity, while for design or reconstruction of a missing part of a tooth, the reconstructed surface has primarily to provide existing, natural shapes with sometimes irregular topological features.

For the genetic algorithm used, the difference or error for a smooth joint between the remaining occlusal surface of a treated tooth and the corresponding area of the functionally transformed occlusal surface of a model tooth defines the fitness function, which is minimized. A creature used in the genetic algorithm consists of a collection of control points that determine the coefficients for spline functions, which in turn define a space transformation for the occlusal surface of a model tooth. As usual, creatures are subject to mutation, crossover, and selection. The selection operation used here is based upon a proposal by Mahfoud and Goldberg in [16, 17]: the creatures in the current parent population compete sequentially either as pairs, or one-by-one with their respective offspring generated via mutation-crossover for survival based upon a simulated-annealing type selection scheme.

2. RELATED WORK: OPTIMIZATION TECHNIQUES IN CAD AND GENETIC ALGORITHMS

Important examples of fitting surface data have been intensively investigated in the past few years. An overview can be found in the paper [09] by Hermann et al. Various strategies were proposed to minimize user interaction; however, it is noticed in [09] that surface fitting of complex free-form shapes still remains a largely unsolved problem of great practical importance. In current practice in the automobile industry, due to the popularity of isophotes and reflection lines, fairing curves and surfaces are designed visually to obtain esthetically smooth surfaces. Another example of surface fitting of complex free-form shapes is designing a lens-system such as a contact lens, which requires an accurate mathematical description of the corneal surface, cf., Chu and Barsky [04]. Recently, a number of papers have been published in which the authors consider the long-standing problem of CAD for geometric shapes with constraints: that of finding optimal designs for optical systems, cf., Halstead et al. [08], Loos et al. [14, 15]. The goal of the end-user is to define a desired photometric distribution and geometric constraints, and then to design automatically an optical system satisfying them. In techniques based on the error functional minimization the question how to solve highly non-linear problems may occur, leading to the minimization of quadratic functionals as considered in [15].

A common approach in CAD involving complex shapes is to use optimization techniques. However, requirements or constraints become increasingly difficult to meet using traditional optimization techniques, since simple analytical solutions do not exist for complicated shapes. Recently, genetic algorithms or evolutionary optimization techniques have found wide application in CAD. Goel and Thompson [06] present an elegant example of the use of evolutionary optimization based on the simple genetic algorithm developed by Holland et al. [10] to recreate the optical system, i.e., the curved refractive interface, in the eye of a certain type of trilobite, the phacopid. (Phacopids had eyes containing large biconvex lenses made of durable minerals. The lens consists of two subunits separated by a concave surface.) Genetic algorithms have also been applied to curve fitting (for more references see the paper by Manela et al. [18]). An attempt to solve a practical problem in spline interpolation was made by Markus et al. in [19]. The paper proposes a new method that determines the number of knots and their locations simultaneously and automatically by using a genetic algorithm and Akaike's information criteria [02]. Different examples of engineering applications of evolutionary design and form growth for constructive solid geometry modeling can be found in a book edited by Bentely [03].

Occlusal surface modeling for restoration, based on jaw articulation requirements is discussed in the dental literature. A brief overview can be found in the paper by Myszkowski et al. [20], where elements of a system involved in occlusal surface adjustment are discussed. The main idea (in particular for crown restoration) is to use a generic tooth (see the paper by Duret et al. [05]) acquired from the library of standardized teeth, which must fit the existing articulation. After positioning and scaling the generic tooth, initial matching with neighboring teeth is achieved, cf., Sohmura and Takahashi [32]. Usually, some corrections of the occlusal surface are needed. In [20], 3-D space transformations for correction of tooth shape is used; [25] uses the same technique for a rough restoration of missing parts of a tooth.

As already indicated above, genetic algorithms have been used successfully as (heuristic) optimization methods in a wide variety of applications. For an overview in that regard see, e.g., the books by Goldberg [07], and Koza [12, 13]. A theoretical foundation describing the convergence or asymptotic behavior of (scaled) genetic algorithms is developing at a slower rate. The book by Vose [34], and the article [28] should provide an overview of theoretical aspects of genetic algorithms. One important question raised in the conclusion of [28] is the convergence of the genetic algorithm proposed by Mahfoud and Goldberg in [16, 17], where a simulated-annealing type selection scheme is applied sequentially to the creatures in the parent generation and the creatures in an (intermediate) offspring generation generated by mutation-crossover. We shall include an outline of proof for the asymptotic convergence of this algorithm below.

3. OVERVIEW ON APPLICATION OF GEOMETRIC MODELING TECHNIQUES

Traditionally, constructive solid geometry modeling uses simple geometric objects for a base model, which can be further manipulated by implementing a certain collection of operations such as set-theoretic operations, blending, or offsetting. The operations mentioned above and many others have found quite general descriptions or solutions for geometric solids represented as points (x,y,z) in space satisfying $f(x,y,z)\geq 0$ for a continuous function f. Such a representation is usually called an *implicit model* or a *function representation*. Set-theoretic solids have been successfully included in this type of representation with the application of R-functions and their modifications (see the papers by Shapiro [31], and Pasko et al. [21]). In [27], an approach to volume modeling is proposed which combines the voxel representation and function representation. It supposes that the two main representations (voxel/function) are given rather autonomously and a rich set of operations can be used for modeling volumes. Whatever complex operations have been applied to a geometric object, which can be given by a voxel raster, by elevation data, or by a function representation, equivalent volume data, can be generated. Consequently, in the present paper, the geometric representation used is a *volume model*. Objects defined in this manner are not regularized solids, as is required in traditional constructive solid geometry modeling, but have a function description.

The goal in occlusal surface modeling for restoration is to reproduce a “reasonable” reconstructed shape surrounding a missing part of the restoration area. In practice, the technician must use relatively simple polygonal shapes to avoid problems in the production of an inlay with high-quality and long-life restoration materials such as porcelain or ceramic. The following example demonstrates how one can use an exact analytic description (a polygon) to reproduce inlay parts of treated teeth. A variety of techniques can be applied to let the user create volume models applying geometric operations. One of the possible ways is to apply a monotone formula [27] to design a hole in a reconstructed object, as shown in Figure 1(b). Other applicable methods may also be used (see the references in [27]).

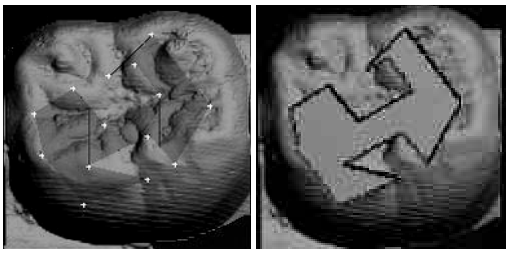

(a) (b)

Figure 1. Defining a “reasonable” boundary shape of the restoration area: (a) Control points of concave polygon. (b) Result of set-theoretic subtraction.

In what follows, we shall discuss a class of techniques that can be used in a variety of applications to improve the quality of the restoration of missing parts of geometric objects. These techniques are applied here (as a case study) to the reconstruction and correction of the occlusal surface of a tooth with a missing part covered by the concave shape, e.g., as shown above in Figure 1(b). We start with an algorithm for rough reconstruction of a missing part using a 3-D space transformation technique. We then consider the so-called *tracking* or *correspondence problem* by applying genetic optimization for automatic correction of mapping points to avoid superfluous folding over a neighborhood area of the restoration area, i.e., the original surface and restored surface must smoothly join along a predefined boundary region of the restoration area. Finally, numerical discountinuity along the boundary line is further reduced using additional interpolation based upon a curvature analysis of existing and reconstructed surface.

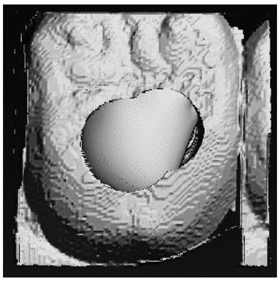


Figure 2. Illustration of the tracking problem: result of an unacceptable restoration.

To illustrate the tracking problem, consider the example in Figure 2: a rather small sphere is used to “fill” the drill hole by expanding the sphere in four directions. Human knowledge and experience is necessary to assign correspondence points to compensate for the gap between the region of restoration and the existing part of the tooth. Figure 2 illustrates the rather obvious fact that incorrect alignment of mapping points between the sphere and the restoration area results in unacceptable restoration, i.e., the result of restoration depends on a sufficiently good initial guess. To automate this procedure, we use a genetic algorithm to correct the coordinates of a limited number of corresponding points initially assigned by the user that define the space transformation. One of the most attractive features of genetic algorithms is that they are easy to interface to a model. A major drawback of, in particular, the simple genetic algorithm is that the algorithm can be trapped in a local minimum; i.e., there is a convergence problem. The convergence problem is discussed in, e.g., the paper by Goel and Thompson [06] on reconstruction of the trilobite eye. In addition, experiments in [24] show that with evolutionary optimization one can design/reconstruct the functionally defined shape of the phacoid eye, or a “four-eyed” fish matching the shape of eyes occurring in the animal. Again, a convergence problem has been encountered. The principal reason for the convergence problem is determined in [28]. As described in [28], only properly scaled genetic algorithms are guaranteed to converge asymptotically. Consequently, a genetic algorithm has to be properly scaled (using either proper scaling for both mutation rate *and* fitness function with proportional fitness selection, or using a proper cooling schedule with simulated-annealing type selection); and a genetic algorithm has to be executed for a large number of cycles (mutation-crossover-selection-reshuffle). Estimates how the limit probability distribution over only optimal creatures is approximated can be obtained along the lines of the proof of [11, p. 160, Thm. V.4.3]. In this work, we have used simulated-annealing type selection in the genetic algorithm.

4. SHAPE TRANSFORMATION AND 3D-RECONSTRUCTION

In this section, we shall give a short account of the shape transformation method used in the application considered in this paper. In order to obtain the proper surface of a treated tooth, a dentist may choose to create and modify the surface in a CAD application interactively using his own experience. Actually, this approach is currently commonplace in dental CAD systems based on parametric representation. A different approach is to use a 3-D space transformation technique, which is based on an approach to the sculpting of 3D geometric objects with arbitrary control points linked to distinct topological features of an object, such as ridges and ravines. The displacement of these control points (between model tooth and treated tooth) defines a global space mapping. To interpolate the overall displacement, we use a volume spline based on Green's function. A detailed description of the method can be found in [25, 23]. The problem is to find an interpolation spline function U Î $W_2^m(W)$, where $W_2^m(W)$ is the space of functions whose derivatives of order $\leq m$ are square-integrable over W Ì $\mathbf{R}^n$, such that the following two conditions are satisfied, as discussed in [33]: (1) $U(p_i) = r_{i,}$ $i = 1,2, . . . ,N,$ and (2) U minimizes the bending energy, if the space transformation is seen as an elastic deformation, i.e.,

$$\Phi(U) = \int_{\Omega} \sum_{|a|=m} \; m!/\alpha! \;\; (D^{\alpha}U)^2 \; d\Omega \rightarrow \min, \qquad (\blacklozenge)$$

where m is a parameter of the variational functional. The minimization of the functional $\Phi(U)$ results in a system of linear algebraic equations for the spline coefficients. An increase in the parameter m essentially leads to an increase in surface rigidity. Minimization of the bending energy in a deformation means that the structure has been deformed without introducing any “hidden” or “artificial” additional structure in the process.

A space mapping in $\mathbf{R}^n$ defines a relationship between each pair of points in the original and deformed objects. Let an n-dimensional region W Ì $\mathbf{R}^n$ of an arbitrary configuration be given, and let W contain a set of arbitrary control points $\{\boldsymbol{q_i} = (q_1{}^i, q_2{}^i, \ldots, q_n{}^i) : i=1,2, \ldots, N\}$ for a non-deformed object, and $\{\boldsymbol{d_i} = (d_1{}^i, d_2{}^i, \ldots, d_n{}^i) : i=1,2, \ldots, N\}$ for the deformed object. It is assumed that the points $\boldsymbol{q}_i$ and $\boldsymbol{d}_i$ are distinct and given on or near the surface of an object. These points establish a correspondence between features of the two objects under consideration. The goal of the construction of the deformed object is to find a smooth mapping function that approximately describes the spatial transformation. The inverse mapping function that is needed to transform surfaces can be given in the form

$$\boldsymbol{q}_i = U(\boldsymbol{d}_i) + \boldsymbol{d}_i \,,$$

where the components of the vector $U(\boldsymbol{d}_i)$ are volume splines interpolating displacements of initial points $\boldsymbol{q}_i$. In principle, control points can be freely chosen by the user, since the algorithm does not use any regular grid. All that has to be done in applying the proposed sculpting technique is to establish a correspondence between features (for example, ridges and ravines) of the model tooth and the treated tooth.

Figure 3(a) presents the model tooth that we would like to merge with the treated tooth that has the drilled cavity shown in Figure 3(b). 21 control points linked to features were assigned manually by the user. The small crosses mark these points. Figure 3(c) presents the recovered tooth after mapping of the model tooth. In practice, one can apply several consecutive space mappings, i.e., one can sculpt the desired surface manually, by assigning control points. Unfortunately, smooth surface reconstruction along the boundary is difficult to realize even for very similar surfaces; the generated and original surfaces will typically join without numerical continuity. Hence, we apply the following strategy to adjust primary surfaces:

- we define an implicit function for the objects;

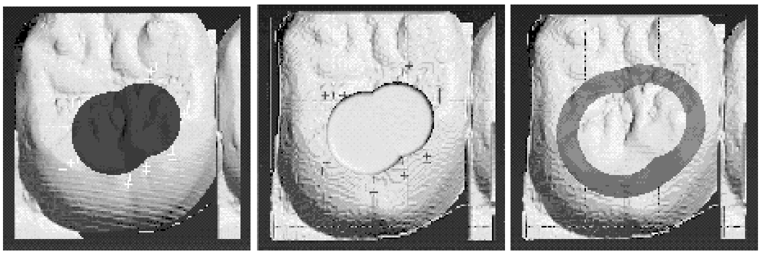
- we define control-mapping points $\boldsymbol{q}_{\mathrm{i}}$ and $\boldsymbol{d}_{\mathrm{i}}$ on the surfaces;

- we calculate a mapping function and apply it to the model tooth (Figure 3(a), shaded area);

- we apply genetic optimization (see section 6) to avoid superfluous folding over a predefined area (see Figure 3(c), extended, pre-set boundary area);

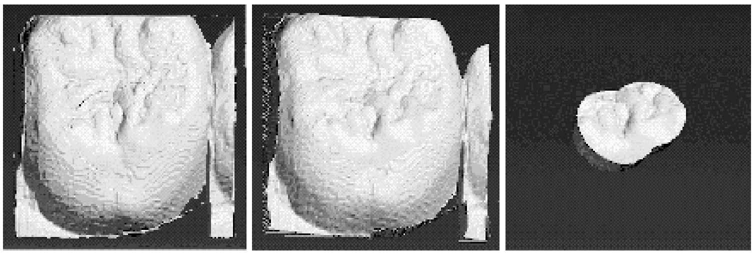
- for each iteration, we apply the same space mapping technique for points $\boldsymbol{q}_{\mathrm{i}}$ and $\boldsymbol{d}_{\mathrm{i}}$;

- in the end, we apply once again the 3-D space transformation of *contour* points of the restoration area, and finally we edit the reconstructed surface according to partial derivatives at the border of the reconstructed area to fair parts of the shape (see section 7).



(a) (b) (c)

Figure 3. (a) Model tooth. (b) Tooth to be treated. (c) Approximation area (extended, pre-set boundary area).



(a) (b) (c)

Figure 4: (a) Treated tooth after application of genetic optimization. (b) Treated tooth after final refinement. (c) Resulting approximated surface.

5. THE OPTIMIZATION ALGORITHM

In this section, we shall give a short outline of proof for the asymptotic convergence of the genetic algorithm with simulated-annealing type selection as proposed by Mahfoud and Goldberg in [16, 17]. (Actually, we shall use the Metropolis criterion for selection, whereas Mahfoud and Goldberg propose a logistic selection scheme.)

First, one starts by noticing that in computer simulations with model programming languages such as C, there is only a discrete set of real numbers available. Consequently, we shall identify a discrete set of real numbers in a certain interval with the underlying alphabet *A,* which is used to encode the genetic information of creatures, i.e.,

$$A = \{ x + k\,\delta \ \ | \, x \in \mathbf{R}, \delta > 0, \ \alpha \geq k \geq 1 \},$$

where $\alpha \in \mathbf{N}$ is the size of the alphabet. Creatures in the genetic algorithm shall be modeled as λ-tuples of elements of A. In addition, we suppose that the optima which the genetic algorithm is supposed to find lie away from the boundary region of the l-dimensional cube $C = A^{\lambda}$, which contains all creatures and is the domain of definition of the fitness function. In this situation, the selection operation will drive the algorithm away from the boundary of *C.*

Let V_1 be the free vector space over *A*. The next step is to model mutation at the level of a single letter of the underlying alphabet *A.* The simplest possibility is to assume that *change* of a letter in *A* occurs with equal probability to all letters in *A* which are different from the current letter; i.e., the stochastic symmetric matrix (the basic mutation matrix) **m** modeling a change of letter is given by

$$< \mathbf{m}\, a, b > = 1/(\alpha -1), \quad \text{if } a,b \in A,\ a \neq b, \text{ and } \quad < \mathbf{m}\, a, b > = 0, \quad \text{if } a = b.$$

Here, *A* is identified with the base vectors in V_1. If *A* is large, and one applies a simulated-annealing type selection scheme, the above definition of **m** is not to be preferred. In that situation, one choses a symmetric, stochastic matrix **m** with zero diagonal such that changes to letters in a *small* neighborhood of the current letter occur with high probability, and change to other letters in *A* occurs with lower probability. For reasons of mathematical convenience and simplicity, *A* is thereby made cyclic, i.e., identified with $\mathbf{Z}_{\alpha} = \mathbf{Z} / \alpha\mathbf{Z}$. Hence, both endpoints of the interval are considered "close" or in the neighborhood of each other. Since under the above assumptions the selection operation will drive the algorithm away from the boundary of *C*, this does not influence the practical applicability of the algorithm.

A population of size $s \in \mathbf{N}$ is simply an *s*-tuple of creatures. Consequently, it is an *L*-tuple of letters of *A* where $L = s\lambda$. Multiple spot mutation acts on such an *L*-tuple of letters of *A* as follows: for every spot or position, it is decided separately whether to change the current letter with probability (mutation rate) *m*, or otherwise to keep the current letter. If a decision to *change* the letter is made positively, then the new letter is chosen in accordance with the transition probabilities set by the basic mutation matrix **m**.

The vector space V_L on which the stochastic matrices describing the optimization algorithm act is now defined as the *L*-fold tensor product of V_1. Consequently, the stochastic matrix describing multiple-spot mutation is given by

$$M(m) = \prod \mathbf{1} \otimes \mathbf{1} \otimes \ldots \otimes ((1-m)\mathbf{1} + m\,\mathbf{m}) \otimes \mathbf{1} \otimes \ldots \otimes \mathbf{1},$$

where each term in the above matrix product contains *L* tensor products, and the non-trivial term $(1\text{-}m)\mathbf{1}+m\ \mathbf{m}$ occurs at tensor spots 1 . . . L respectively. As a result, we see that *M(m)* is a fully positive, symmetric, stochastic matrix. (As an alternative to the above, and respecting the idea of "change in a small local neighborhood" as basis of the simulated annealing algorithm , one may chose a more localized version of mutation called single-spot mutation which is discussed in detail in [28, 29].)

Let $\mathbf{p} = (c_1, c_2, c_3, \ldots, c_s)$ be an even-sized population. The crossover operation acts on **p** as follows: (1) creatures are sequentially paired, (2) for every pair $(c_{2\sigma}, c_{2\sigma+1})$ it is decided with probability (crossover rate) *c* whether or not crossover should be applied to the pair, and (3) if crossover is to be applied, then the heads of creatures are exchanged before a randomly chosen cutpoint. The crossover operation and its interplay with mutation has been extensively investigated in the literature. See [28, 29] for a comprehensive collection of results as well as a listing of references. The following facts are important:

- crossover is represented by a symmetric, stochastic matrix *C(c)* which commutes with the mutation matrix *M(m),*

- the combined crossover-mutation matrix is a fully positive, symmetric, stochastic matrix.

Suppose for a moment that s = 1; i.e., that we consider creatures subject to mutation, and a fitness function $f: C \rightarrow \mathbf{R}$, which has to be minimized. The usual simulated-annealing selection criterion

[01] between a creature c Î C and its offspring d Î C generated via mutation in generation t Î $\mathbf{N}$ is given as follows:

- if $f(c)$³ $f(d)$, (i.e., d is better), then the algorithm proceeds with d.

- if $f(c)<f(d)$, then the algorithm proceeds with creature d with probability $\exp((f(c)-f(d))/q(t)) \to 0$, where $q(t) = \mathrm{K}/\log(t) \to +0$ for a suitable constant $K>0$, and $t\in \mathbf{N}$, $t\to\infty$.

$q(t)$ is called the cooling schedule for the algorithm. A sufficient setting for K is max(f)-min(f). One remarkable feature of the simulated annealing algorithm is that the steady-state distribution of an individual step of the algorithm is the explicitly-known Boltzmann distribution $B(t)$. Let c_0ÎC be an element of minimal fitness. Then, the c-component of $B(t)$ is given by:

$$< B(t), c > = k(t) \exp((f(c_0)-f(c))/q(t))$$

where $k(t)$ is the normalization constant such that $||B(t)||_1 = 1$. If one applies the modified Mahfoud-Goldberg algorithm sequentially-pairwise, i.e., every parent-pair competes with its offspring-pair obtained via mutation-crossover using the probabilistic selection criterion described above, then a proof of convergence can be obtained as follows: first, one notes the fact that the steady-state distribution of one individual step $G(t)$ is given by the s-fold tensor product $b(t) = B(t)$Ä $B(t)$Ä…Ä $B(t)$ as a computation based upon the tensor product decomposition of the underlying vector space shows. The next step is to establish weak ergodicity of the inhomogeneous Markov chain describing the algorithm. First, one finds a lower bound $g(t)$ for the smallest entry in the stochastic matrix $G(t)$. The coefficient of ergodicity $\tau_1(t)$ of $G(t)$ can then be estimated using $g(t)$ and [30, p. 137, Thm. 4.8]. This yields a logarithmic cooling schedule. To obtain strong ergodicity of the inhomogeneous Markov chain describing the algorithm one proceeds in two steps: (1) summability of the sequence $||b(t+1)-b(t)||_1$, $t \in \mathbf{N}$, is established using the convergence of the normalizing factors $k(t)$ defined above, the monotone behavior of $b(t)$ in its components, and a telescoping argument, and (2) one applies [11, p. 160, Thm. V.4.3]. Finally, the explicit form of $b(t)$ shows convergence against optima in the limit.

If one lets the creatures in the parent generation compete with those in the offspring generation sequentially one-by-one, then one can obtain a proof of convergence by following the steps in the proof of [29, Thm. 8.2.4]: weak ergodicity is obtained as outlined above. Strong ergodicity is obtained by using that the steady-state distributions can be computed using Cramer's rule which yields differentiable and thus summable components in the parameter $t \in \mathbf{N}$. This allows to apply [11, p. 160, Thm. V.4.3]. Convergence is then shown by considering the flow between the overall probability $w(t)$ over populations containing an optimal creature, and the complement $(1 - w(t))$ as in [29, eq. (43)].

As the final step in one cycle of the genetic algorithm, the creatures in the current population are randomly rearranged (renumbered).

6. APPLICATION OF THE GENETIC ALGORITHM

The application of the genetic algorithm starts with initially selecting a set of M variable control points $\{\boldsymbol{d}_i=(d_1{}^i,d_2{}^i,d_3{}^i)$: $i=1,2, . . . ,M\}$ for the definition of the space transformation generating the deformed object. (Actually, the user selects points on the surface of the model tooth $\boldsymbol{q}_i$ with corresponding points $\boldsymbol{d}_i$ on the treated tooth.) The collection of coordinates of $\boldsymbol{d}_i$ defines a creature, i.e., λ = 3M. The algorithm begins by randomly distorting the initial creature, and generates s creatures, which are collected to form the initial population. Now, the genetic algorithm with sequential simulated- annealing type selection as described in the previous section is applied to this initial setup to minimize the fitness function.

Our main premise here is to avoid superfluous folding over a predefined area (the extended, pre-set boundary area of Figure 3(c)), where we calculate the fitness to get the "best approximated surface." A good physical analogy for this procedure is to consider a "sheet" of plastic material, which one wishes to deform while avoiding a superfluous folding over the a specific given area where it is attached (glued) to an existing surface. The "sheet" is supposed to be flexible, so that it deforms in conformity with the given surface area (of the treated tooth). Recall that the spline (determined by the set of variable control points $\boldsymbol{d}_i$ that constitute a creature), which is used for the global space mapping, provides a minimization of the bending energy. Following this analogy, we define the fitness function f as follows:

$$f(\boldsymbol{c}) = \sum_{i=1}^{k} \sum_{j=1}^{l} r_{ij} \text{ for i,j} \in \Omega_{\mathrm{f}},$$

where Ω_{f} can be an arbitrarily chosen "gluing" area (here: the extended, pre-set boundary area of Figure 3(c)), and r_{ij} are residuals (between interpolated and given surface) of elevation teeth data. This definition of the fitness function expresses the idea that the more smoothly we glue a "sheet" of plastic to a surface, the fewer residuals of elevation data between the model tooth and the treated one can be observed.

7. ADDITIONAL SURFACE REFINEMENT AT THE BOUNDARY

The techniques that we have applied so far provide a reconstruction of a missing part of the given, damaged object which is visually smooth for most parts of the reconstructed area. In practice, however, a completely smooth surface reconstruction can hardly be realized, because the generated and original surfaces will typically join without numerical continuity. See Figure 4(a), where a black curved step can be observed on the left-hand side of the reconstructed area, which shows the discontinuity in the depth data. Let us notice here that after application of genetic optimization, which takes an acceptable 15-20 minutes of processing time on an SGI O^2 workstation, the maximum residual between the reconstructed area and original one on the border connecting two parts of the object is about 1.3 μm, and the average residual is about 0.3 μm.

To finally adjust the given surfaces one can apply interactive “engraving” with the help of a haptic user interface to avoid a superfluous crack over the restoration area. Such an approach is an attempt to mimic the dentist’s activity [22].

The traditional approach to finally refine the reconstructed surface with the help of geometric transformations is possibly more suitable for CAD applications. First, the optimization of space mapping for control-mapping points of the reconstructed area (see Figure 4(a)) makes it possible to reduce the numerical discontinuity. Secondly, a space mapping for *contour* points of the reconstructed area (see Figure 4(b)) makes it possible to reduce the numerical discontinuity further. The maximum residual between the reconstructed area and the original area becomes about 0.28 μm, and the average residual is about 0.023 μm, which satisfies the anticipated design conditions in practice. The final step is to use partial derivatives at the boundary of the reconstructed area to fair or to join smoothly the two distinct parts of the geometric object. We need at least one of the partial derivatives to be nonzero. In fact, this is the condition for the manifold given by *f(x,y,z)*=0 to be free from singularities such as self-crossing, cuspidal edges, or similar features. In practical applications, one simply assumes that one is dealing with “regular values” of the function *f*. For blending the surface parts one can apply a spline interpolation technique based on the Green functions, taking into account that one knows the values of functions in the *n*-dimensional area *W* of the reconstruction and the first derivatives $f¢_R$ at boundary points (for more references see [33]). In [26], the problem of 3D reconstruction of volume data without taking into account first derivatives is examined. Partial derivatives can be numerically calculated at the boundary points or in a small neighborhood of the boundary points belonging to the model tooth. In the case of an area such as that shown in Figure 1(b), the calculation of partial derivatives for some corner points becomes impossible; in this case, one adopts the simple strategy to use neighboring points.

In order to define the spline function *U* needed for the above procedure, the following functional is minimized:

$$\Phi(U) = \int_{\Omega} \sum_{|a|=m} \ m!/\alpha! \ (D^{\alpha}U)^2 \, d\Omega + \sum_{P \in p} [\ \partial u/\partial \eta_p(P) - f'_P]$$

where π is a point set for which the first derivatives $f¢_R$ are defined. The procedure is similar to the one outlined in the discussion of equation (♦).

8. CONCLUSION

In this paper, we have presented an application of numerical optimization for surface reconstruction (more precisely, reconstruction of missing parts of a real geometric object represented by volume data) by employing a specially designed genetic algorithm to solve a problem in CAD in dentistry. We have shown and proved that we have a good alliance of geometric modeling and optimization techniques using genetic algorithms to determine the reconstructed surface and assure overall smoothness. In addition, we showed how to edit a final surface by using first derivative constrains to fair parts of the shape.

There are several directions for future work. The main restriction of our method is that the user needs to assign feature points manually. Future research can be done on how topological surface properties such as ridges and ravines can be used to automatically assign starting and destination points for 3D-transformations. It would be of significant practical value to design a 3D user interface that enables the user to design inlay tooth surfaces. To enhance the practical value of the proposed technique, an additional goal is to reduce feasible undesirable effects of implementing constrains stipulated by partial derivatives which can be caused by “noise” of the real measured data points.

While many challenges remain, we believe that the presented approach for recovering of missing parts of geometric objects provides an important direction for research and practical applications.

9. ACKNOWLEDGMENTS

The authors would like to thank Professor McDonald for his help with the style of this paper.

10. REFERENCES

[1] E.H.L. Aarts and P.J.M. van Laarhoven, Simulated Annealing: An Introduction, Statistica Neerlandica **43**, 1989, 31-52

[2] H. Akakie, A new look at the statistical model identification, IEEE Transactions Automatic Control, Vol. AC-19, 6, 1974, 716-723

[3] P. Bentely, Evolutionary Design by Computers, Morgan Kaufman, 1999

[4] L. Chu and B.A. Barsky, Cylindrical coordinate representations for modeling surfaces of the cornea and contact lenses, Proc. Int. Conf. on Shape Modeling, IEEE Computer Society Press, 1997, 98-113

[5] F. Duret, J.L. Blouin, and B. Duret, CAD/CAM in dentistry, J. Am. Dent. Assoc., 117(11), 1988, 715-720

[6] M. Goel, R.L. Thompson, The evolution of the trilobite eye, Computer Simulations of Self-Organization in Biological Systems, part 11, Croom Helm, London & Sydney, 1988, 275-290

[7] D.E. Goldberg, Genetic Algorithms, in Search, Optimization & Machine Learning, Addison-Wesley, 1989

[8] M. Halstead, B. Barsky, S. Klein, and R. Mandell, Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. Conf. Proc. SIGGRAPH 96, H. Rushmeier (ed.), Addison Wesley, 1996, 335-342

[9] T. Hermann, Z. Kovacs, and T. Varady, Special applications in surface fitting, Geometric Modeling: Theory and Practice, W. Strasser, R. Klein and R.Rau (Eds), Springer, 1997, 14-31

[10] J.H. Holland, K.J. Holyoak, R.E. Nisbett, P.R. Thagard, Induction: Processes of Inference Learning and Discovery, MIT Press, 1986

[11] D.L. Isaacson and R.W. Madsen, Markov Chains: Theory and Applications, Prentice-Hall, 1961

[12] J.R. Koza, Genetic Programming, MIT Press, 1992

[13] J.R. Koza, Genetic Programming *II*, MIT Press, 1994

[14] J. Loos, G. Greiner, and H.-P. Seidel, A variational approach to progressive lens design, Computer-Aided Design, Vol. 30, No. 8, 1998, 595-602

[15] J. Loos, Ph. Slusallek, and H.-P. Seidel, Using wavefront tracing for the visualization and optimization of progressive lenses, EUROGRAPHICS'98, Computer Graphics Forum, vol. 17, No. 3, 1998, 255-265

[16] S.W. Mahfoud and D.E. Goldberg, A Genetic Algorithm for Parallel Simulated Annealing, in: R. Manner, B. Manderick (eds.), Parallel Problem Solving from Nature **2**, Elsevier, 1992, 301-310

[17] S.W. Mahfoud and D.E. Goldberg, Parallel recombinative simulated annealing: a genetic algorithm, Parallel Computing 21, 1995, 1-28

[18] M. Manela, N. Thornhill, and J.A. Campbel, Fitting spline functions to noisy data using a genetic algorithm. Proc. of the Fifth Int. Conf. on Genetic Algorithms, Morgan Kaufmann, 1993, 549-556

[19] A. Markus, G. Renner, and J. Vancza, Spline interpolation with genetic algorithms, Proc. Int. Conf. on Shape Modeling, IEEE Computer Society Press, 1997, 47-54

[20] K. Myszkowski, V.V. Savchenko and T.L. Kunii, Computer modeling for the occlusal surface of teeth, Proc. CGI'96 Conf., Pohang, Korea, June 24-28, 1996, 196-198

[21] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications, The Visual Computer, 11(6), 1995, 429-446

[22] V.V. Savchenko, 3-D Geometric Modeller with Haptic Feedback: Engraving Simulation, in proceedings 2nd PHANToM Users Research Symposium 2000 , M. Harders and S. Huber, (eds.), Zurich, Switzerland, July 6-7, 2000, 35-42

[23] V.V. Savchenko and A.A. Pasko, Transformation of functionally defined shapes by extended space mappings, The Visual Computer, 1998, 257-270

[24] V.V. Savchenko, A.A. Pasko, Evolutionary optimization of functionally defined shapes: Case study of natural optical objects, Computer Graphics Int. Conf. 1999, The University of Calgary, Canada, IEEE Computer Society, June 7-11, 1999, 20-24

[25] V.V. Savchenko, A.A. Pasko, T.L. Kunii, and A.V. Savchenko, Feature based sculpting of functionally defined 3-D geometric objects, T.S. Chua et al. (eds), Proc. of the MMM'95, Nov. 1995, 341-348

[26] V.V. Savchenko, A.A. Pasko, O. Okunev and T.L. Kunii, Function representation of solids reconstructed from scattered surface points and countours, Computer Graphics Forum, 14(4), 1995, 181-188

[27] V.V. Savchenko, A.A. Pasko, A.I. Surin, T.L. Kunii, Volume Modeling: Representations and advanced operations, Computer Graphics Int. Conf., F. Wolter and N.M. Patrikalakis (eds.), Hannover, Germany, IEEE Computer Society, June 22-26 1998, 4-13

[28] L.M. Schmitt, Theory of Genetic Algorithms, to appear in Theoretical Computer Science, 2001, 61p. (available as preprint from the author)

[29] L.M. Schmitt, C.L. Nehaniv, R.H. Fujii, Linear Analysis of Genetic Algorithms, Theoretical Computer Science 200, 1998, 101-134

[30] E. Seneta, Non-negative Matrices and Markov Chains, Springer Series in Statistics, Springer, 1981

[31] V. V. Shapiro, Real functions for representation of rigid solids, Computer Aided Geometric Design, 11(2), 1994, 153-175

[32] T. Sohmura and J. Takahashi, Improvement of CAD to produce crown by considering occlusion., Dental Materials Journal, 12(2), 1993, 190-195

[33] V. A. Vasilenko, Spline functions: theory, algorithms and programs, (in Russian), Nauka (Novosibirsk), 1983

[34] M. D. Vose, The Simple Genetic Algorithm: Foundations and Theory (Complex Adaptive Systems), Bradford Books, 1999