

Global Self-Intersection Detection and Elimination in Freeform Curves and Surfaces with Applications in Metamorphosis

Diana Pekerman¹
Dept. of Mathematics
Technion – IIT
Haifa 32000, Israel

Gershon Elber
Dept. of Computer Science
Technion – IIT
Haifa 32000, Israel

Myung-Soo Kim
School of Comp. Science and Eng.
Seoul National University
Seoul, Korea

Abstract

We present an algorithm for global self-intersection detection and elimination in freeform curves and surfaces. A bi-normal line criterion is used for the detection of antipodal points that characterize a closed loop. The problem is reduced to solving a system of five multivariate polynomial constraints, including one inequality. Employing an optimization procedure for constraints, we can gradually eliminate all self-intersections, while flipping the locations of antipodal points. The resulting surface is globally free of self-intersections. Our algorithm provides a practical solution to the problem of global self-intersections, which occurs in the application of arbitrary metamorphosis between freeform curves.

Keywords: Global self-intersection; surface bi-normal line; antipodal points; multivariate polynomial constraints; self-intersection-free metamorphosis.

1 Introduction and Background

Evaluating the intersection of freeform parametric curves and surfaces is a recurring operation in computer graphics, geometric and solid modeling, and computer-aided design. Hoffmann [1] listed the intersection problem as one of the most fundamental problems in the integration of geometric and solid modeling systems. There has been very extensive research in this field, resulting in numerous articles presenting different methods for intersecting freeform curves and surfaces. Nevertheless, only very few papers address the computation of self-intersections of freeform curves and surfaces, and in particular, in the context of self-intersection-free metamorphosis of freeform curves and surfaces.

The intersection of two surfaces can be topologically complex in general, comprising several univariate components that we will refer to as closed loops

and open branches. In the intersection of two freeform curves, no such components are present and it is clear that the intersection of two freeform curves is a simpler task. In order to detect endpoints of an open branch in the intersection of surfaces, we can simply intersect the boundary curves of one surface with the other surface. On the other hand, the detection of closed loops has been a major challenge in surface intersection algorithms.

Sinha et al. [2] presented a parallel normal criterion for guaranteeing the preclusion of closed loops in the intersection of two surfaces. The criterion states that if two surfaces intersect in a closed loop, at least one line that is normal to one surface must be parallel to a line that is normal to the other surface. The drawback of the parallel normal criterion is that there are, in general, an infinite number of parallel normal lines within a closed loop. The same criterion can be applied to the intersection of two planar curves. Sederberg and Meyers [3] presented a similar criterion based on cones. The criterion states that if a cone that tightly bounds all the normal vectors on one surface lies outside of a cone that tightly bounds all the normal vectors on the other surface, no closed loops exist. The authors noted that their loop detection test has been observed to require a considerable amount of computation time for some straightforward cases. In addition, Sederberg and Meyers [3] reasoned that if two space curves intersect at all, while their bounding cones do not overlap, they intersect exactly once.

Finally, a decisive collinear normal criterion for loop detection was presented by Sederberg et al. [4]: if two surfaces intersect on a closed loop and their normal vectors do not deviate more than 90° , then there exists a line that is perpendicular to both surfaces. In contrast to the parallel normal criterion of Sinha et al. [2], there are only a finite number of lines that are normal to the two surfaces and each loop in the intersection must encircle at least one collinear normal line. While Sederberg et al. [4] presented a correct theory for a loop

¹Currently working at Samsung Telecom. Research Israel.
e-mail: d.pekerman@samsung.com

detection method, an efficient implementation has thus far eluded the authors.

Galligo and Pavone [5] presented two different contributions to the determination of a self-intersection locus for a Bézier bicubic surface. The first one uses a specific sparse resultant and produces an implicit equation of a plane projection of this locus. The second one accurately computes the coordinates of critical points on this locus, by solving a system of four polynomial equations in four variables, derived from a previously computed plane projection of the self-intersection locus. Seong et al. [6] presented a scheme to trim both local and global self-intersections of offset curves and surfaces. The scheme is based on the derivation of an analytic distance map between the original curve/surface and its offset. By solving one/three bivariate polynomial equations for an offset curve/surface, respectively, all the local and global self-intersection regions in the offset curves/surfaces can be detected. The trimming of these regions is completed by projecting the zero-set of polynomial equation(s) into the desired parametric domain.

Samoilov and Elber [7] introduced two new methods for eliminating self-intersections. Both their algorithms exploit the matching algorithm of Cohen et al. [8]. The first algorithm investigates building and employing a homotopy between the two original curves, which is a composition of a ruled surface with an appropriate subjective continuous function that causes the curve of homotopy to be self-intersection-free. The second algorithm constructs the best correspondence of the relative parameterizations of original curves. Then, it eliminates the remaining self-intersections and flips back the domains that self-intersect.

To the best of our knowledge, there is no known method in the literature that can efficiently detect self-intersections in arbitrary metamorphosis of freeform curves and then eliminate the detected self-intersections.

The rest of this paper is organized as follows. Section 2 presents the detection algorithm, which uses a bi-normal line criterion for global self-intersection detection in freeform surfaces. In Section 3, we present the global self-intersection elimination algorithm. Section 4 provides some examples of metamorphosis between freeform curves, employing the two proposed algorithms. Finally, we conclude this paper in Section 5.

2 Bi-Normal Line Criterion for Global Self-Intersection Detection

Let $S(u, t)$, $(u, t) \in [u_0, u_1] \times [0, 1]$, denote a regular surface that represents a locally self-intersection-free metamorphosis between two simple, regular, planar freeform curves, $C_0(u)$ and $C_1(u)$. Global self-intersection of $S(u, t)$ can be composed of several components in the parametric domain of the surface: open

branches, closed loops (see Figure 1) and singular intersections (tangencies with closed loops of size zero). If the curves $C_0(u)$ and $C_1(u)$ are open, the end points of the open branch are easy to detect by intersecting the surface $S(u, t)$ with its boundary curves $S(u_0, t)$ and $S(u_1, t)$. In contrast to open branches, the detection of closed loop intersection is a complex task.

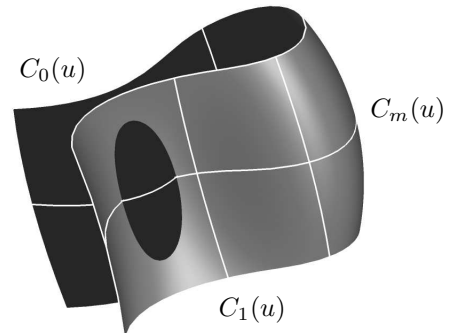


Figure 1: A globally self-intersecting surface with a closed loop intersection component.

While we pose the problem using the metamorphosis application, the presented algorithms are all general. We start with an algorithm for loop intersection detection in freeform curves. We now define some terms that are used in this section:

Definition 2.1 A *surface normal line* at the parameter (u_1, t_1) is a line through $S(u_1, t_1)$ that is parallel to the surface normal, $N_S(u_1, t_1)$, where $N_S(u_1, t_1) = \frac{\partial S(u_1, t_1)}{\partial u} \times \frac{\partial S(u_1, t_1)}{\partial t}$.

Definition 2.2 The line through $S(u_1, t_1)$ and $S(u_2, t_2)$ is denoted as a *surface bi-normal line* at parameters (u_1, t_1) and (u_2, t_2) , where $(u_1, t_1) \neq (u_2, t_2)$, if it is a surface normal line at both $S(u_1, t_1)$ and $S(u_2, t_2)$.

Definition 2.3 Let L be a surface bi-normal line of the surface $S(u, t)$ at parameters (u_1, t_1) and (u_2, t_2) , such that $\langle N_S(u_1, t_1), N_S(u_2, t_2) \rangle < 0$. Then, $S(u_1, t_1)$ and $S(u_2, t_2)$ are called the *antipodal points*.

Figure 2 presents the middle isoparametric curve, $C_m(u)$, of the surface from the Figure 1 that contains the bi-normal lines with antipodal points.

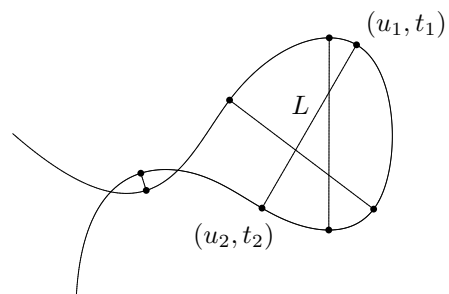


Figure 2: The isoparametric curve of the surface seen in Figure 1, shown with its bi-normal lines.

We make several assumptions regarding the given metamorphosis surface $S(u, t)$. To begin with, we assume that $S(u, t)$ is locally self-intersection-free. In addition, we assume that if the surface $S(u, t)$ has a global self-intersection at some location, then there are exactly two points in the parametric domain $(u_1, t_1) \neq (u_2, t_2)$ that are mapped to the same location. We further assume that if $S(u, t)$ has global self-intersections, then the normal directions of the surface inside each loop component do not vary by more than 90° .

Denote the loop intersection component as \mathcal{C} , and the region in the parametric domain of $S(u, t)$ that is interior to \mathcal{C} as $Int(\mathcal{C})$. Then, under the last assumption, any normal line at $(u, t) \in Int(\mathcal{C})$, intersects $S(u, t)$ exactly once in the domain $Int(\mathcal{C})$. In addition, any normal line at $(u, t) \in Int(\mathcal{C})$ intersects $S(u, t)$ exactly once in the *antipodal domain* of $Int(\mathcal{C})$, denoted by $Int(\mathcal{AD}(\mathcal{C}))$ (see [4] for proof), where the antipodal loop component $\mathcal{AD}(\mathcal{C})$ is defined as follows:

$$\mathcal{AD}(\mathcal{C}) = \{(u_1, t_1) | \exists (u_2, t_2) \in \mathcal{C}: S(u_1, t_1) = S(u_2, t_2), \text{ where } (u_1, t_1) \neq (u_2, t_2)\}.$$

Lemma 2.4 *If a metamorphosis surface $S(u, t)$ intersects itself in loop components, \mathcal{C} and $\mathcal{AD}(\mathcal{C})$, then there exists a bi-normal line at two antipodal points (u_1, t_1) , (u_2, t_2) in the parametric domain of the surface, where $(u_1, t_1) \in Int(\mathcal{C})$ and $(u_2, t_2) \in Int(\mathcal{AD}(\mathcal{C}))$.*

Proof: See [9].

Clearly, Lemma 2.4 is a fundamental and important part in any global self-intersection detection algorithm. The self-intersection detection problem is now reduced to determining all the antipodal points in the parametric domain of $S(u, t)$. In the next section, we explain how to detect the bi-normal lines with antipodal points of a given surface $S(u, t)$.

2.1 Computing Bi-Normal Lines with Antipodal Points

An algebraic approach to detecting the surface bi-normal lines with antipodal points to a surface $S(u, t)$ is presented by solving the following system of five constraints:

$$\begin{aligned} (a) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(u, t)}{\partial u} \right\rangle = 0, \\ (b) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(u, t)}{\partial t} \right\rangle = 0, \\ (c) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(v, s)}{\partial v} \right\rangle = 0, \\ (d) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(v, s)}{\partial s} \right\rangle = 0, \\ (e) \quad & \left\langle \frac{\partial S(u, t)}{\partial u} \times \frac{\partial S(u, t)}{\partial t}, \frac{\partial S(v, s)}{\partial v} \times \frac{\partial S(v, s)}{\partial s} \right\rangle < 0, \end{aligned} \quad (1)$$

with four unknowns, u, t, v, s .

Notice that (a) and (b) in Equation (1) seek surface normal lines at the parameter location (u, t) , while (c)

and (d) seek surface normal lines at the parameter location (v, s) . Equation (e) constrains the normal of $S(u, t)$ to be in the opposite direction to the normal of $S(v, s)$, in order to find the antipodal points. By employing only the first four equations in Equation (1), it is easy to see that the whole parametric domain of $S(u, t)$ is a solution when $u = v$ and $s = t$. The fifth equation ensures that our solution contains no points of the form $(u, t) = (v, s)$.

The system of equations given in Equation (1) is numerically unstable, since the value $S(u, t) - S(v, s)$ vanishes in the first four equations, at global self-intersection points $(u, t) \neq (v, s)$. Thus, we replace them by equivalent, more numerically stable, conditions as follows:

$$\begin{aligned} (\bar{a}) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(u, t)}{\partial u} \right\rangle = 0, \\ (\bar{b}) \quad & \left\langle S(u, t) - S(v, s), \frac{\partial S(u, t)}{\partial t} \right\rangle = 0, \\ (\bar{c}) \quad & \left\langle \frac{\partial S(v, s)}{\partial v} \times \frac{\partial S(v, s)}{\partial s}, \frac{\partial S(u, t)}{\partial u} \right\rangle = 0, \\ (\bar{d}) \quad & \left\langle \frac{\partial S(v, s)}{\partial v} \times \frac{\partial S(v, s)}{\partial s}, \frac{\partial S(u, t)}{\partial t} \right\rangle = 0, \\ (\bar{e}) \quad & \left\langle \frac{\partial S(u, t)}{\partial u} \times \frac{\partial S(u, t)}{\partial t}, \frac{\partial S(v, s)}{\partial v} \times \frac{\partial S(v, s)}{\partial s} \right\rangle < 0. \end{aligned} \quad (2)$$

That is, only Equations (\bar{a}) and (\bar{b}) vanish at the global self-intersection points. In Equations (\bar{c}) and (\bar{d}) , the value $S(u, t) - S(v, s)$ is replaced by the normal to the surface that never vanishes in the case of a regular surface.

The presented set of Equations (2) forms the multivariate rational spline function constraints. To find the solution, we apply a geometric constraint solver [10]. Armed with the ability to detect the existence of self-intersections, we will present, in the next section, a heuristic approach that eliminates self-intersections.

3 Global Self-Intersection Elimination by LCQP

Let $S(u, t)$ be a surface with some closed loop intersection components, and let $\{(u_i, t_i, v_i, s_i)\}_{i=0}^N$ be the set of antipodal points of $S(u, t)$, as described in Section 2. In this section, we present an algorithm that attempts to eliminate all loop intersection components in $S(u, t)$, seeking a surface $\hat{S}(u, t)$ close to $S(u, t)$, which is globally self-intersection-free.

We begin by a formal representation of the flipping constraint over \hat{S} between a pair of antipodal points (u, t) and (v, s) as follows:

$$\hat{S}(u, t) = S(v, s), \quad \hat{S}(v, s) = S(u, t). \quad (3)$$

Figure 3 illustrates the result of the flipping operation between the two antipodal points defined in Equation (3), which leads to the elimination of the intersection loop.

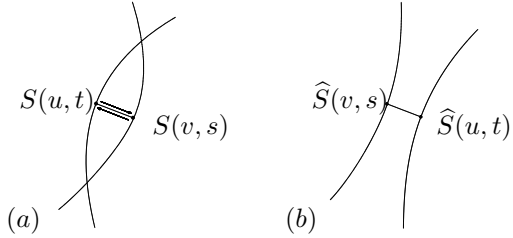


Figure 3: (a) Two isoparametric curves containing the antipodal points; (b) the resulting isoparametric curves after applying the flipping action.

To solve the flipping constraint, we consider the minimization of the following functional, over \hat{S} :

$$\sum_{i=0}^N \alpha_i \left\| \hat{S}(u_i, t_i) - S(v_i, s_i) \right\|^2 + \sum_{i=0}^N \beta_i \left\| \hat{S}(v_i, s_i) - S(u_i, t_i) \right\|^2. \quad (4)$$

In the two summations in Equation (4), we seek a new surface $\hat{S}(u, t)$ that satisfies the flipping actions (Equation (3)), for all $(N + 1)$ antipodal points. The weights α_i and β_i , $0 \leq i \leq N$, are introduced to enable non-symmetrical flipping.

In order to find the target surface $\hat{S}(u, t)$ that is close to $S(u, t)$, in the L^2 -norm sense, one should also minimize the following functional:

$$\int_{t_0}^{t_1} \int_{u_0}^{u_1} \left\| \hat{S}(u, t) - S(u, t) \right\|^2 dudt. \quad (5)$$

Combining the minimization of the functionals defined in Equations (4) and (5) with the constraint that the surface $\hat{S}(u, t)$ must describe the metamorphosis between two curves $S(u, 0)$ and $S(u, 1)$, the following constrained minimization problem is considered:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=0}^N \alpha_i \left\| \hat{S}(u_i, t_i) - S(v_i, s_i) \right\|^2 \\ & + \sum_{i=0}^N \beta_i \left\| \hat{S}(v_i, s_i) - S(u_i, t_i) \right\|^2 \\ & + \int_{t_0}^{t_1} \int_{u_0}^{u_1} \left\| \hat{S}(u, t) - S(u, t) \right\|^2 dudt \\ \text{subject to} \quad & \hat{S}(u, t_0) = S(u, t_0) = C_0(u), \\ & \hat{S}(u, t_1) = S(u, t_1) = C_1(u). \end{aligned} \quad (6)$$

Employing the B-spline representation of the surface $\hat{S}(u, t)$, one can verify that the optimization problem, defined in Equation (6), is actually a linearly constrained quadratic optimization (see [11]). Thus, it can be solved to find a local optimal solution by one of the known solvers, e.g. the ‘‘optimization toolbox’’ in [12].

In general, the solution surface $\hat{S}(u, t)$ can still have loop intersection components because it is only locally optimal. In this case, we again apply the test of the bi-normal line criterion from Section 2 to find the antipodal points of the obtained surface $\hat{S}(u, t)$. Then, we

again solve the optimization problem in Equation (6) for the new metamorphosis surface $\hat{S}(u, t)$. We continue applying the above algorithm gradually, to achieve a globally self-intersection-free metamorphosis. Unfortunately, there is no guarantee that this optimization procedure will converge.

Clearly, if metamorphosis surface $S(u, t)$ is globally self-intersection-free, its isoparametric curves $S(u, t_0)$, t_0 constant, will all be globally self-intersection-free as well. However, a globally self-intersecting metamorphosis surface $S(u, t)$ does not guarantee the existence of a self-intersecting isoparametric curve $S(u, t_0)$. The following lemma presents a sufficient condition for a globally self-intersecting surface to possess self-intersecting iso-curves:

Lemma 3.1 *Let $S(u, t)$ be a globally self-intersecting metamorphosis surface. If the isoparametric curves $S(u, t_0)$, t_0 constant, are all planar, $S(u, t)$ holds self-intersecting isoparametric curves.*

Proof: See [9].

Even if the isoparametric curves $S(u, t_0)$ of the given surface $S(u, t)$ are planar, the presented global self-intersection elimination algorithm does not guarantee that the isoparametric curves $\hat{S}(u, t_0)$ of $\hat{S}(u, t)$ will also be planar. Should the user desire planar curves in the morphing process, one can find a simple 2D projection, if such exists, of a simple 3D curve, for example by using the projection technique in [13]. The next section presents some examples and results.

4 Examples

The first example of global self-intersection elimination is illustrated in Figure 4. Figure 4(a) shows the same example of a globally self-intersecting surface with a closed loop intersection as in Figure 1. Figures 4(b)-(e) show surfaces, gradually acquired from the presented algorithm and resulting in a self-intersection-free metamorphosis surface, after four iterations.

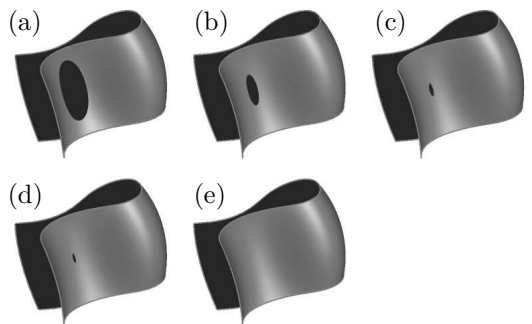


Figure 4: A sequence of surfaces, acquired from the global self-intersection elimination algorithm, resulting in a self-intersection-free surface.

Another example is shown in Figure 5. Figure 5(a) shows a globally self-intersecting metamorphosis surface with a loop intersection component between a cow and a camel (see Figure 5(c) for the globally self-intersecting metamorphosis sequence). Figure 5(b) illustrates the self-intersection-free metamorphosis surface produced by applying the proposed global elimination algorithm to the surface. Figure 5(d) shows the globally self-intersection-free metamorphosis sequence. The loop intersection component was eliminated in a single iteration of the optimization problem with large values of $\alpha = \beta = 10$ for fast convergence.

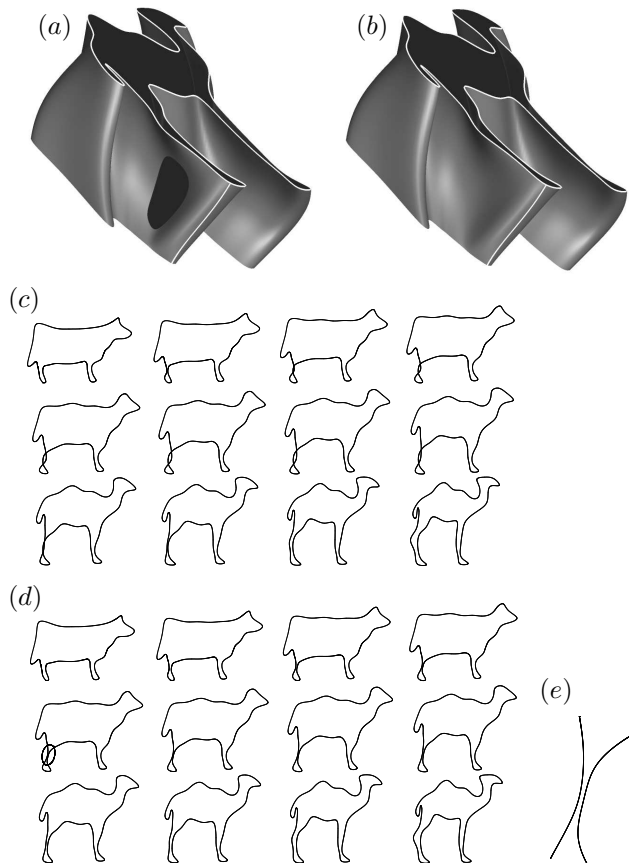


Figure 5: (a) A globally self-intersecting metamorphosis surface with loop intersection components; (b) a globally self-intersection-free metamorphosis surface, acquired from the global self-intersection elimination algorithm; (c) a morphing sequence between a cow and a camel, derived from the surface in (a); (d) a morphing sequence, derived from the surface in (b); (e) zoomed-in area of the intermediate curve from (d), which illustrates the result of global self-intersection elimination.

5 Conclusions

In this paper, we have presented an algorithm for global self-intersection detection and elimination for freeform curves and surfaces. We use a surface bi-normal line

criterion to detect antipodal points on intersection loops, which is reduced to solving a system of five multivariate polynomial constraints. Then, we gradually flip the locations of the antipodal points by setting a set of constraints, until the resulting surface is globally self-intersection-free. The presented algorithm provides the solution to the global self-intersection problem that can occur in the arbitrary metamorphosis between freeform curves.

In the future, the presented algorithm for global self-intersection detection could be extended to the case of metamorphosis volume $V(u, v, t)$, between freeform surfaces, using a 4D geometry.

REFERENCES

- [1] C. M. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann, San Mateo, CA., 1989.
- [2] P. Sinha, E. Klassen and K. K. Wang. Exploiting topological and geometric properties for selective subdivision. *Proc. of the annual symposium on Computational Geometry*, 39–45, 1985.
- [3] T. Sederberg and R. Meyers. Loop detection in surface patch intersections. *Computer Aided Geometric Design*, 5, 1988.
- [4] T. Sederberg, H. Christiansen and S. Katz. Improved test for closed loops in surface intersections. *Computer Aided Design*, 21:505–508, 1989.
- [5] A. Galligo and J. P. Pavone. Self intersections of a Bézier bicubic surface. *Proc. of the International Symposium on Symbolic and Algebraic Computation*, 2005.
- [6] J.-K. Seong, G. Elber and M.-S. Kim. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *Computer-Aided Design*, 38:283–193, 2006.
- [7] T. Samoilov and G. Elber. Self-intersection elimination in metamorphosis of two-dimensional curves. *The Visual Computer*, 14:415–428, 1998.
- [8] S. Cohen, G. Elber and R. Bar-Yehuda. Matching of freeform curves. *Computer-Aided Design*, 29:369–378, 1997.
- [9] D. Pekerman. Optimization of the metamorphosis process between freeform curves. MSc thesis, Technion, June 2006.
- [10] G. Elber and M.-S. Kim. Geometric constraint solver using multivariate rational spline functions. *Proc. of the ACM symposium on Solid Modeling and Applications*, 1–10, 2001.
- [11] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA., 1984.
- [12] *MATLAB, Version 7.0.4*, January 2005.
- [13] G. Elber. Global curve analysis via a dimensionality lifting scheme. *Proc. of the IMA Conference on the Mathematics of Surfaces*, 184–200, 2005.
- [14] *IRIT 9.5 User's Manual*, 2005, Technion. <http://www.cs.technion.ac.il/~irit>.