Visual Comput (2007) 23: 219–229
DOI 10.1007/s00371-006-0095-2

Guy Sela
Gershon Elber

Generation of view dependent models using free form deformation

Published online: 24 January 2007
© Springer-Verlag 2007

G. Sela (✉) · G. Elber
Technion – I.I.T.,
Haifa 32000, Israel
{guysela, gershon}@cs.technion.ac.il

Abstract We present a scheme which, given two 3D geometric models, creates a third, synergetic model with resemblance to one input model from one viewing direction and the other input model from another, orthogonal, viewing direction. Our scheme automatically calculates the necessary constraints needed to deform the first model's silhouette into the second model's in 2D, and creates a 3D deformation function based on these constraints while minimizing the object's distortion in all areas but the silhouette. The motivation of this work stems from the artwork of conceptual artists such as Shigeo Fukuda [9] and Markus Raetz [19].

Keywords Free form deformation · 3D modeling · Non-photorealistic rendering · Computer graphics art

1 Introduction

Working in the field of non-photorealistic rendering (NPR), researchers have tried to find different ways to automatically create artistic images that, traditionally, only humans could produce. In the same way, we use an approach, which we coin as non-realistic modeling (NRM), to create visually interesting 3D models for mostly artistic purposes. Our effort is geared towards creating models that resemble works of art such as Shigeo Fukuda's "Duet" [9] (a sculpture that, from one direction, looks like a pianist playing a piano, and from another direction like a violinist playing a violin), or Markus Raetz's "Metamorphose" [19] (a man with a hat from one direction and a rabbit from another). One of the most basic examples of such models is the "sqriancle" (a combination of a SQuare, a tRIANgle and a cirCLE), shown in Fig. 1. We use tools from traditional 3D geometric modeling and 3D soft body deformation fields in order to create such models.

Our aim is to create 3D models that resemble one object from one viewing direction, and another, completely different object from another, orthogonal to the first, viewing direction. Following psychological experiments that demonstrated the human brain's dependency on silhouettes for 3D object recognition [3], we focus on the objects' silhouettes. The use of silhouettes is also seen in the creation of the artwork mentioned above.

In our NRM approach, the user is prompted to load two 3D models. One model will only be used for silhouette extraction, while the other will be deformed so that, from a specific viewing direction, its silhouette matches the silhouette extracted from the first model. While the problem is conceptually a 3D problem, it can be reduced to a 2D one. We use a bivariate B-spline surface, S, to represent the deformation function that transforms one silhouette into the other, in the plane. This deformation surface will be the result of a surface fairing problem, with (silhouette) deformation points as the constraints. S will then be extruded in the third dimension to form a 3D deformation function. This, in turn, will be used to deform the first object.

The rest of this work is organized as follows. In Sect. 2, we give an overview of previous work, mostly in the fine

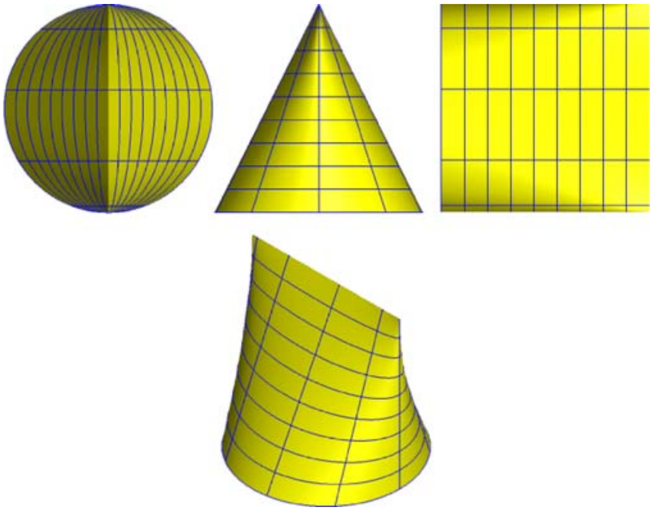


Fig. 1. The sqriancle (SQuare, tRIANgle and cirCLE) is an object that resembles a square, a triangle and a circle when viewed from different angles

arts. Section 3 gives some background on surface fairing and modeling with constrained free form deformations. In Sect. 4, we describe the proposed scheme; Sect. 5 presents a few examples and finally, we conclude in Sect. 6.

2 Related work

Little work can be found in the geometric modeling community on artistically-oriented modeling. Most contemporary modeling is done using computers for the CAD/CAM fields, making, for example, models for manufacturing. These models usually need to be very accurate. Hence, most available modeling tools (e.g., 3DStudioMax, Maya) prompt the user to specify a complete set of constraints in order to receive a precise output model. Some effort has been made to model inaccurate objects, usually by sketching, although the aim is not necessarily artistic. One notable example is Teddy, by Igarashi et al. [16], which allows the user to model simplistic 3D shapes using mouse movements. The result is, of course, a rough 3D sketch only, as the user cannot specify actual coordinates or other accurate constraints, such as angles or sizes.

Another noteworthy effort is the "Escher for Real" project [7]. Here, 3D models resembling M.C. Escher's impossible drawings are created, such that they appear to be identical to the original drawings (and hence, impossible) from one single viewing direction. From any other direction, they are revealed to be fraudulent (see Fig. 2 for a simple example). In addition, the "Beyond Escher for Real" project [6] includes, among other things, a few models such as the ones we are exploring here. One example is shown in Fig. 3: a Menorah (the state emblem of Israel) is fused with a Star of David. The models in both projects were created manually.

In contrast to the lack of work in the geometric modeling field, examples in the art world for NRM do exist.

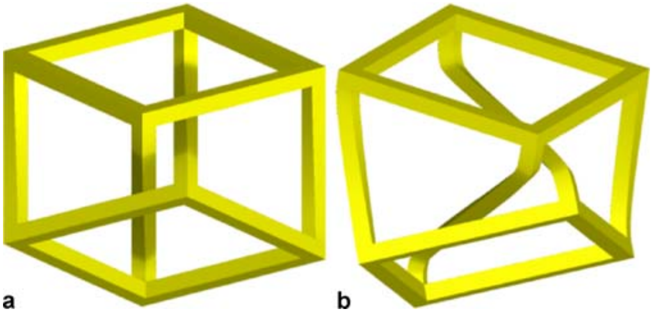


Fig. 2a,b. The "Escher for Real" Escher's Cube [7]. © Copyright Gershon Elber 2005. **a** Shows the seemingly impossible cube, while in **b** it is shown to be an illusion

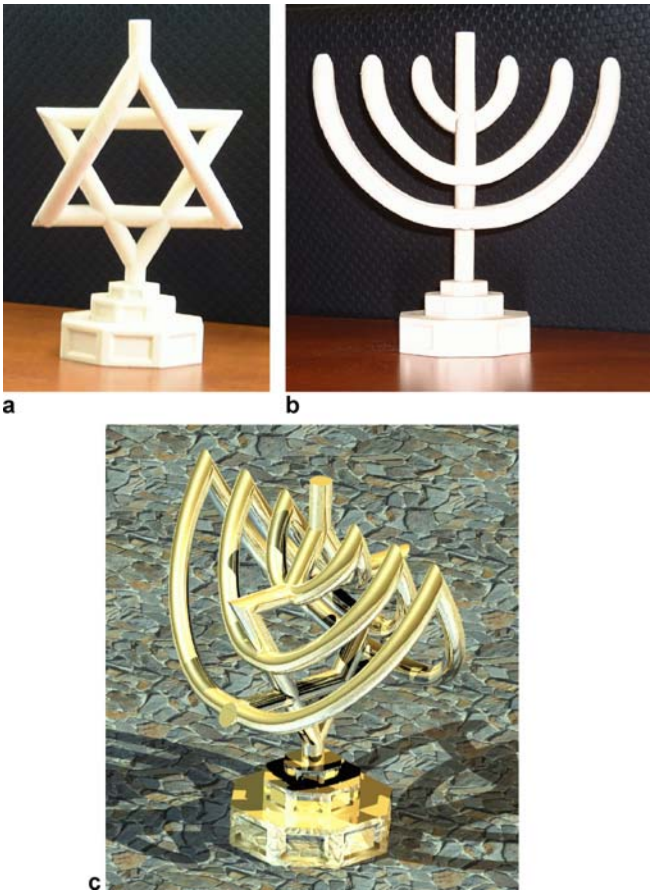


Fig. 3a–c. The "Beyond Escher for Real" Star of David/Menorah [6]. © Copyright Gershon Elber 2005. **a** Shows the view of the Star of David while **b** shows the view of the Menorah. **c** Shows a general view of this model. Parts **a** and **b** show photographs of an actual manufactured part

Shigeo Fukuda [9] created such works as "Duet", "Love Story" and "Cat/Mouse" [9], all resembling two different objects from two orthogonal directions. Another relevant artist is Markus Raetz, who also sculpts such works, for example "Metamorphose" [19], and also a series of pieces showing one word from one direction, and another word, usually the antonym, from another. Examples are "Yes/No" (in many languages), "This/That" and one interesting piece titled "Same/Same" [19], which reads "Same" from the front, and "Same" read backwards from the side. Hence, its reflection also reads "Same" on a well placed mirror.

In other pieces, artists use a different technique to achieve a similar effect. A collection of objects is placed in no apparent order, only to reveal a certain shape when viewed from a special direction. From any other viewing angle, they appear to be just a jumbled set of objects. Examples include Shigeo Fukuda's "Underground Piano" [9], in which an arbitrary collection of parts appear to form a piano when viewed from a particular viewing angle, and "Lunch with a Helmut On" [9], in which a heap of welded forks and spoons form a shadow in the shape of a motorcycle.

3 Background

This section provides the necessary background needed for understanding the proposed method. In Sect. 3.1, we briefly introduce the problem of surface fairing, which will be used for creating a 2D deformation function from a set of incomplete constraints. In Sect. 3.2, we describe the modeling tool known as FFD and its uses. FFD will be created from the 2D deformation function, and will be used to deform our model. In Sect. 3.3, we briefly explain the curve matching problem. Curve matching will be used to attain correspondence between the 2D representations of the objects' silhouettes.

3.1 Surface fairing

Surface fairing is essentially a constrained optimization problem. Typically, the constraints are points that the surface must interpolate. The optimization generally appears in the shape of some functional that should be minimized (or maximized), e.g., curvature properties or bending or stretching energy. Most works seek to minimize bending energy via the curvature functional [10]:

$$F(S) = \int\limits_{\Omega} \kappa_1^2(S) + \kappa_2^2(S) \mathrm{d}A,$$

where Ω is the surface region of the object, κ_1 and κ_2 are the principal curvatures and dA is a surface area element. Unfortunately, this functional is highly non-linear, and requires a computationally expensive numerical solution. Thus, most schemes use a linear approximation of F. One popular option is to use the following, second order approximation to the bending energy (or thin plate energy) functional [10]:

$$F(S) = \iint\limits_{u\ v} \left(\frac{\partial^2 S}{\partial u^2}\right)^2 + 2\left(\frac{\partial^2 S}{\partial u \partial v}\right)^2 + \left(\frac{\partial^2 S}{\partial v^2}\right)^2 \mathrm{d}u\mathrm{d}v. \quad (1)$$

The functional of Eq. 1 is derived from the sum of the second derivatives of S:

$$\frac{\partial^2 S}{\partial u^2} + \frac{\partial^2 S}{\partial u \partial v} + \frac{\partial^2 S}{\partial v \partial u} + \frac{\partial^2 S}{\partial v^2}.$$

According to Clairaut's theorem [24], if a function, $\mathcal{F}$, has continuous second partial derivatives at any given point, then $\frac{\partial^2 \mathcal{F}}{\partial u \partial v} = \frac{\partial^2 \mathcal{F}}{\partial v \partial u}$. Uniform B-spline functions of orders larger than 3 are at least C^2 everywhere. In other words $\frac{\partial^2 S}{\partial u \partial v} = \frac{\partial^2 S}{\partial v \partial u}$, and so we arrive at the functional of Eq. 1.

For the sake of brevity, we will only survey a small subset of results on the subject. Surface fairing algorithms use either global or local optimization schemes [10]. Moreton and Séquin [18] interpolated a system of point constraints and connectivity information into curve constraints, which they used to solve for the desired surface. The interpolation is done patch by patch, but each patch is solved, in one pass. Vassilev [23] also used a global scheme in which he added additional degrees of freedom in the shape of new constraints in order to reduce the number of unknowns to a minimum. The additional constraints are added where there is an insufficient number of constraints.

Hadenfeld [11] locally modified the position of one control point at a time, iteratively, solving for an optimal position in order to minimize thin plate energy. Hahmann and Konz [12] also used a local scheme. In each step of their scheme, a knot is removed and then reinserted in order to locally smooth the surface. As the main drawback of local schemes is that they may converge to local minima, [12] used simulated annealing to help the search for the global minimum. The work in [26] tried to minimize an energy functional (mainly thin plate) by an iterative algorithm that can perform more than one editing operation per step, editing the control points that cause irregularities in the surface's smoothness.

3.2 Modeling with FFDs

Free form deformation (FFD) is a strong and versatile geometry manipulation tool. Sederberg and Parry first introduced FFDs in [22]. A major drawback in using FFD is that the control is over the FFD's control points and not over the deformed model. Hsu et al. [13] tried to circumvent this problem by using direct manipulation. Direct manipulation allows the user to move the vertices of the object to be deformed, and not the control points of the B-spline volume. Shi-Min et al. [14] improved the work in [13] by introducing an explicit and more efficient solution to the direct manipulation problem. Other researchers explored different ways to control the FFD. Yoshizawa et al. [25] calculated the model's skeletal mesh, allowing the user to manipulate it, and finally update the original geometry using an FFD constructed by calculating the difference between the original and edited skeletal meshes. Hua and Qin [15] embedded their objects in scalar fields defined by implicit functions. The scalar fields are then manipulated via level sets in order to deform the object. Raviv and Elber [20] used a hierarchy of 3D B-spline scalar functions, while the zero set still defines the surface of their model. By modifying the hierarchy of scalar functions, the zero set is altered, and the desired model is created, with multi-resolution control.

3.3 Curve matching

Curve matching algorithms attempt to find some correspondence between two free form curves or polylines, and possibly alter one of them such that their matched features occur at similar parameter values. Besl and McKay [1] performed curve matching using an iterative closest point (ICP) algorithm, mainly for registration purposes. Later on, Cohen et al. [5] used first order differential analysis on the input curves to try to find a correspondence that best fits their features. The latter was extended using second order differential analysis in [17]. This was intended more for morphing, geometry construction and animation purposes. Rodriguez et al. [21] adapted an algorithm from the string matching field for use with 3D vectors, in order to measure the similarity between two 3D curves.

4 The non-realistic modeling scheme

Let M_1, M_2 be two 3D geometric models selected by the user, and let $\mathcal{C}_1$, $\mathcal{C}_2$ be the 3D model's silhouettes represented as free form curves describing their *outer* general outlines according to two user defined viewing directions, $\mathcal{V}_1$ and $\mathcal{V}_2$. Our view dependent model's (VDM) construction scheme consists of three parts. In the first, the planar silhouettes $\mathcal{C}_1$ and $\mathcal{C}_2$ are extracted from M_1 and M_2, respectively. This stage is described in Sect. 4.1. The second stage, after the curves are matched, calculates the necessary constraints needed to deform $\mathcal{C}_1$ into $\mathcal{C}_2$, in the plane. A planar B-spline surface S is constructed, which interpolates the constraints $S(\mathcal{C}_1) = \mathcal{C}_2$, and serves as a planar deformation function. The construction of this set of constraints is described in Sect. 4.2. In order to solve for S, we reformulate the problem as a set of linear equations. This stage is described in Sect. 4.3. The result of the second stage is a planar B-spline surface S: $\Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$,

$$S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{i,j} B^{d_u}_{i,\boldsymbol{\tau}^u}(u) B^{d_v}_{j,\boldsymbol{\tau}^v}(v), \quad u, v \in [0, 1], \tag{2}$$

where the $P_{i,j}$s are the surface control points, d_u and d_v are the degrees of the B-spline basis functions, $B^d_{i,\tau}$, in the u and v directions, and $\boldsymbol{\tau}^u$ and $\boldsymbol{\tau}^v$ are the knot vectors over which the basis functions are defined. S should be a surface that satisfies the required constraints that deform $\mathcal{C}_1$ into $\mathcal{C}_2$ while minimizing a certain fairing functional, such as that of Eq. 1. In the third and final stage, discussed in Sect. 4.4, S is extruded along its normal direction to form a B-spline volume FFD, which is, in turn, applied to M_1, resulting in the desired VDM model.

4.1 Silhouette extraction

Consider model M_i (see Fig. 4a). In order to extract M_i's silhouette, we use the following simple method. First, render M_i into an image, without the use of shading, from the viewing direction $\mathcal{V}_i$, selected by the user. This yields a binary image I_i in which one color marks the background, and another marks the body of M_i, as shown in Fig. 4b. Then, employ an edge detector to extract the boundary. Here, we used the simple Sobel edge detection filter [4]. The output of the filter is another binary image. In this binary image, black pixels mark edge pixels of I_i, i.e., the *outer* silhouette of M_i, and gray pixels cover the rest of I_i. This step is shown in Fig. 4c. In the final stage, we traverse this outer silhouette, assuming that it is completely contained in I_i. During this traversal, we sample the outer silhouette at regular intervals, and then approximate the samples by a periodic B-spline curve, $\mathcal{C}_i$, as shown in Fig. 4d. We complete this section by noting that any other silhouettes' extraction scheme could be employed as part of this NRM effort.

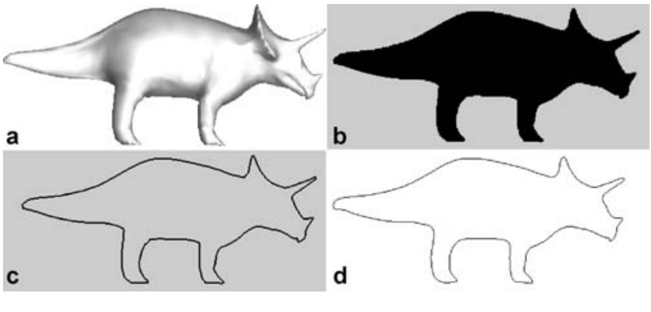


Fig. 4a–d. Silhouette extraction: **a** is the original triceratops model shown from the desired viewing direction. **b** Shows the image I that is the result of rendering the model with no shading information. **c** Shows the screen buffer after the application of the Sobel edge detection filter, and finally **d** shows the final B-spline silhouette curve $\mathcal{C}$

4.2 Calculating the constraints set

We are now ready to define a set of constraints that will warp $\mathcal{C}_1$ into $\mathcal{C}_2$, and construct a B-spline surface S that complies with these constraints. While the problem of exactly mapping $\mathcal{C}_1$ to $\mathcal{C}_2$ is continuous and hence, quite difficult, we benefit from our need to only have an approximate solution and solve a discrete problem instead. We approximate the desired deformation by calculating a discrete set of constraints, the size of which is set by the user, according to the smoothness of $\mathcal{C}_1$ and $\mathcal{C}_2$. The smoother the curves, the smaller the set of constraints needs to be in order to approximate $\mathcal{C}_i$ to within a prescribed tolerance. Each constraint is a pair of 2D points of the form $(u_i^1, v_i^1) \in \mathcal{C}_1$, $(u_i^2, v_i^2) \in \mathcal{C}_2$, such that the 2D point on S with parameter values (u_i^1, v_i^1) will be mapped to (u_i^2, v_i^2), or simply,

$$S\left(u_i^1, v_i^1\right) = \left(u_i^2, v_i^2\right) = Q_i^2. \tag{3}$$

The term $\mathcal{C}_2$ is the outer silhouette that we seek for M_1 from view direction $\mathcal{V}_1$ following the deformation. As of now, there is no correlation between $\mathcal{C}_1$ and $\mathcal{C}_2$. This means that if we sample $\mathcal{C}_i$ at regular parameter values and match the resulting coordinates, there is no guarantee that the samples will be appropriately matched in any way. In order for the constraints to be meaningful, we must first apply some form of curve matching [5] between $\mathcal{C}_1$ and $\mathcal{C}_2$, so that the two curves have the same orientation, and, as much as possible resemble each other in terms of the parameter values at their features; see Fig. 5 for a simple example. With proper matching, the two curves are mapped to a common planar domain. To this end, mainly for the sake of simplicity when performing the matching, each curve is non-uniformly scaled so that its bounding box is the domain $[0, 1]^2$. This non-uniform scaling deforms the silhouette curves, and will later be compensated for, and recovered in the created VDM.

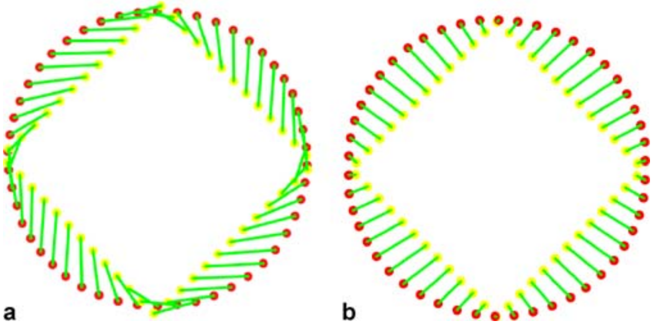


Fig. 5. a Shows the sampled constraints needed to deform the circle (*red points*) to the square (*yellow points*), without matching the two curves. The orientations of the curves are similar, as is their speed, but their starting points do not match. **b** Shows the samples after the curves have been properly matched

In order to discretize the problem, we uniformly sample both silhouette curves f times, at parameter values

$$t_k = \frac{k}{f}, \quad k = 0 \dots f-1.$$

The constraints are as shown in Eq. 3, where

$$\left(u_k^1, v_k^1\right) = \mathcal{C}_1(t_k),$$

and

$$\left(u_k^2, v_k^2\right) = \mathcal{C}_2(t_k).$$

An example of such a set of constraints for a properly matched pair of curves is shown in Fig. 6.

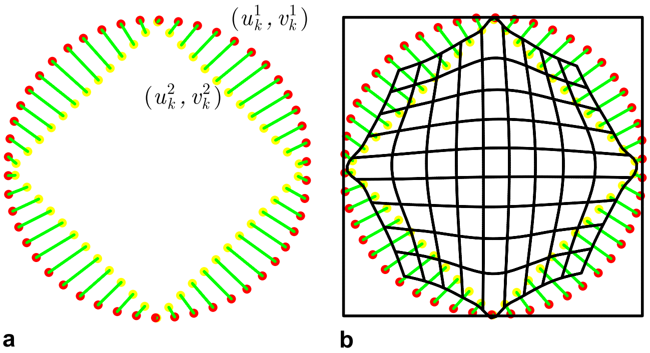


Fig. 6. a Shows the points on the silhouettes of a sphere and a cube, (*red* and *yellow points*, respectively), and the correspondence between them (*green lines*). **b** Adds the resulting deformation surface S (in *black*) and its parametric domain (*black square*)

4.3 The optimization problem

The unknowns in this optimization problem are the control points of the B-spline surface, S, namely, $P_{i,j} : i = 1 \dots n,\ j = 1 \dots m$, as seen in Eq. 2. Unfortunately, the deformation constraints of Eq. 3 are not distributed evenly across the parametric domain of the surface ($[0, 1]^2$). This may cause some regions of S to be either under-constrained or over-constrained. The over-constrained case can be avoided by adaptively adding knots to the surface S, thus adding degrees of freedom, or alternatively by solving for the control points, using Least Squares (LS). In order to deal with the under-constrained regions, we add a minimization requirement, and force the surface to be as smooth as possible globally, by minimizing the following functional [10]:

$$F = \iint\limits_{u,v} \nu_1 \left(\frac{\partial^2 S}{\partial u^2}\right)^2 + \nu_2 \left(\frac{\partial^2 S}{\partial u \partial v}\right)^2 + \nu_3 \left(\frac{\partial^2 S}{\partial v^2}\right)^2 \mathrm{d}u\mathrm{d}v, \tag{4}$$

where ν_i, $i = 1, 2, 3$, are user defined parameters that allow better control of the resulting surface. By default, we use the natural coefficients $\nu_1 = \nu_3 = 1$ and $\nu_2 = 2$, as explained in Sect. 3.1.

The functional in Eq. 4 with coefficients $\nu_1 = \nu_3 = 1$ and $\nu_2 = 2$ is the simplified thin plate energy functional (or bending energy functional, Eq. 1) [10] of a parametric surface. The resulting surface should uphold the constraints of Eq. 3 while minimizing the global bending functional Eq. 4. Areas that are under-constrained will minimize bending, and consequently, will move as little as possible from the initial regular grid, which has no bending energy at all. (Note Eq. 4 also measures bending in the plane).

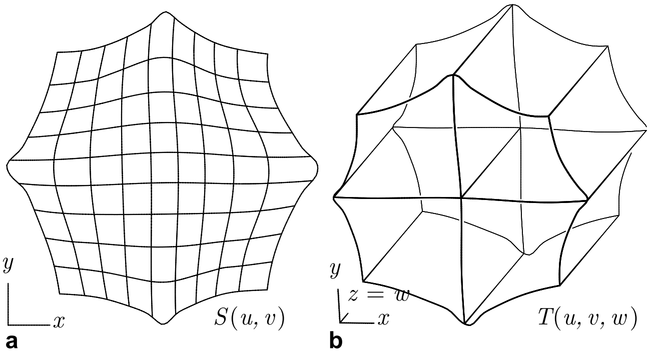


Fig. 7. a Shows the planar deformation surface, S, needed to deform a circle into a square (see Fig. 6). **b** Shows the same surface extruded into a 3D FFD, F. Figure 8 shows the final VDM model of this example

We now construct the Lagrange operator L:

$$\begin{aligned}
L = &\iint_{u,v} \nu_1 \left\| \frac{\partial^2 S}{\partial u^2} \right\|^2 + \nu_2 \left\| \frac{\partial^2 S}{\partial u \partial v} \right\|^2 + \nu_3 \left\| \frac{\partial^2 S}{\partial v^2} \right\|^2 \mathrm{d}u\mathrm{d}v \\
&+ \sum_{k=0}^{f-1} \lambda_k \left\| Q_k^2 - S\left(u_k^1, v_k^1\right) \right\| \\
= &\iint_{u,v} \nu_1 \left\| \frac{\partial^2 S}{\partial u^2} \right\|^2 + \nu_2 \left\| \frac{\partial^2 S}{\partial u \partial v} \right\|^2 + \nu_3 \left\| \frac{\partial^2 S}{\partial v^2} \right\|^2 \mathrm{d}u\mathrm{d}v \\
&+ \sum_{k=0}^{f-1} \lambda_k \left\| Q_k^2 - \sum_{i,j=0}^{n,m} P_{i,j} B_{i,\boldsymbol{\tau}^u}^{d_u}\left(u_k^1\right) B_{j,\boldsymbol{\tau}^v}^{d_v}\left(v_k^1\right) \right\|, \qquad (5)
\end{aligned}$$

where λ_k are the Lagrange multipliers. In order to solve Eq. 5 we differentiate L with respect to $P_{i,j}$, $(i = 0 \ldots n, \quad j = 0 \ldots m)$ and λ_k $(k = 0 \ldots f - 1)$ and find the extremal points.

The existence of a solution to the interpolation system relies on the fulfillment of the Schoenberg–Whitney condition [2]. For any set of non-self-contradicting con-

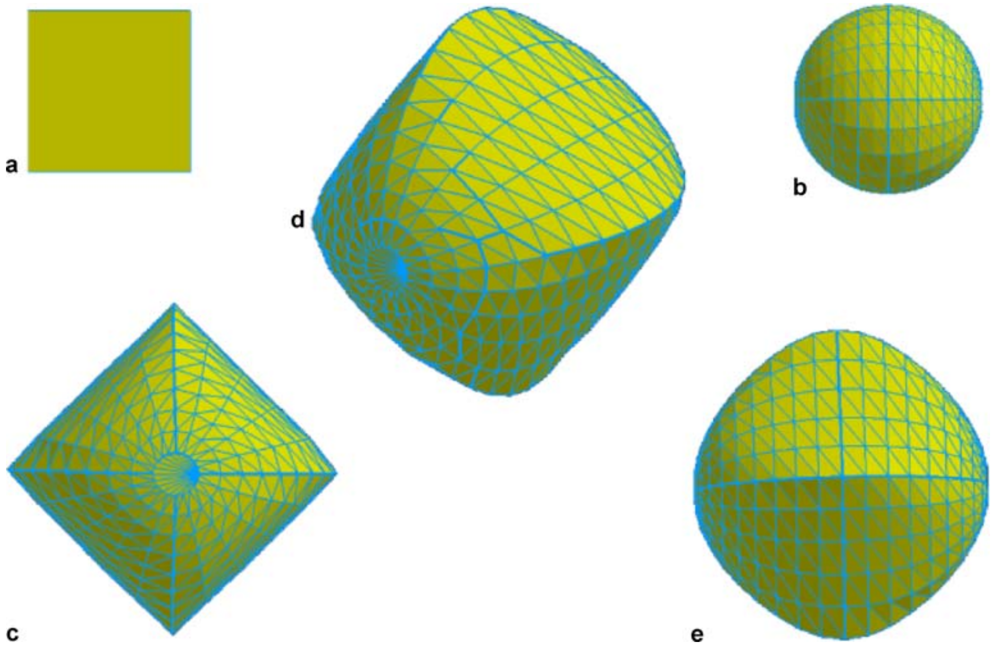


Fig. 8a–e. The Sphube, a sphere–cube merge. **a** Shows the original cube model from the front and **b** shows the original sphere model. These are the desired outcome's silhouettes. The parts **c**,**d** and **e** show the resulting VDM model from the front, a general view and side, respectively. See also Fig. 7 showing the FFD that created this VDM model

straints, there exists a fine enough grid at which the system has at least one solution.

4.4 Creating and applying the FFD

While our current deformation function, $S(u, v)$, is only 2D, remember we are seeking a 3D deformation volume. The volumetric FFD is created so that it is an identity along the w axis, using an extrusion: $T(u, v, w) = \big(s_x(u, v), s_y(u, v), w\big)$ where $S = (s_x, s_y)$. The w axis is parallel to $\mathcal{V}_1$ (the viewing direction of M_1) and spans the w range of M_1 (see Fig. 7). As a final step, the FFD, T, is applied to the base model's vertices, deforming it to its new desired, VDM, shape.

5 Results

Figures 8–14 present some examples created using the presented approach for NRM realization of VDMs. All examples are organized in the same manner. The top images show the input models, M_1 and M_2, from the desired view directions, $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. The center image is a general view of the resulting VDM model. Finally, the bottom images show the resulting model from the front and side, displaying the desired silhouettes. One noticeable artifact appears in the Tricerhino (Fig. 11), where the rhinoceros' ears (marked (1)) and horns (6) were matched to the Triceratops' frill (4) and horns (5), respectively. As the rhinoceros' silhouette requires that its ears and horns be at the same height, the front view of the resulting Tricerhino reveals that its frill and horns have been brought to the same height as well ((2) and (3)). This problem is discussed in Sect. 6.

Figure 13 is a bit different than the other examples, as it is an attempt to emulate a wire-sculpture by Markus Raetz, titled "Hasen-Spiegel" (Hare-Mirror in German). The sculpture is actually a thick wire, which resembles the silhouette of a hare from one direction, and that of a bust from another. In our example, M_1 was created by taking a "wire" representation of the bust model, stretching it by a factor of $\sqrt{2}$ in the X direction, and then rotating it by 45 degrees around the Y axis, so that its silhouette is that of a "wire" bust from both the side and the front. M_2 is a simple "wire" bunny. The silhouettes of the bunny and

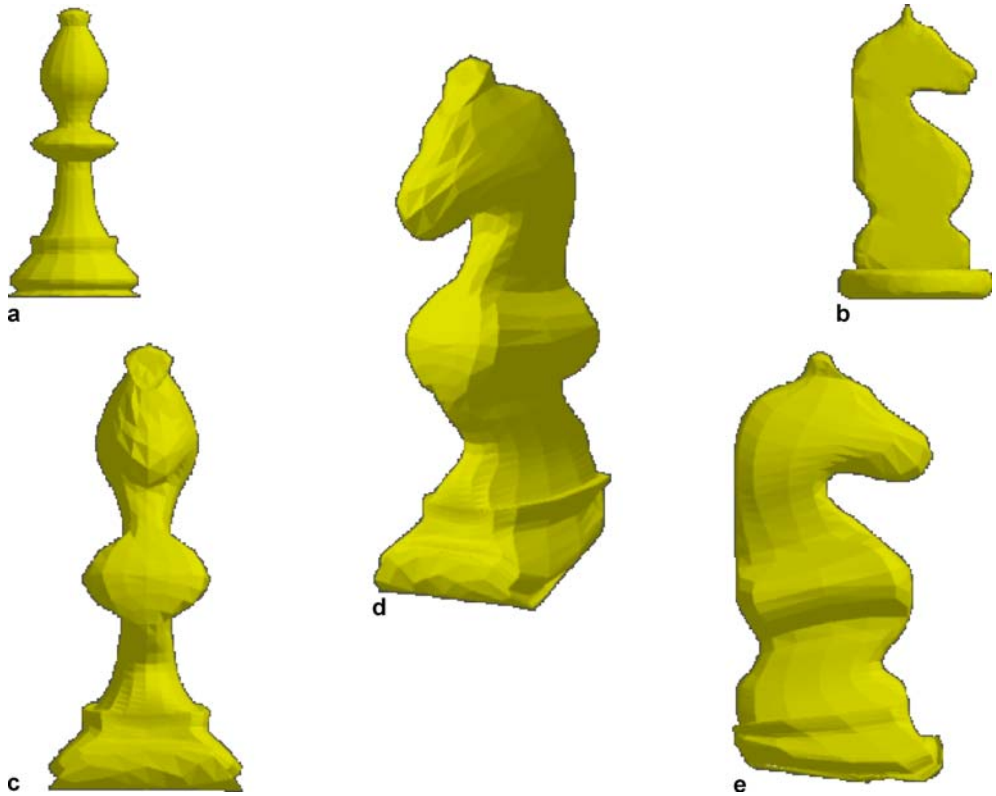


Fig. 9a–e. The Knishop, a knight–bishop merge. **a** Shows the original bishop model from the front, and **b** shows the original knight model from the side. These are the desired outcome's silhouettes. The parts **c**, **d** and **e** show the resulting VDM model from the front, a general view and side, respectively. The bishop model was retrieved from the Princeton 3D model search engine, http://shape.cs.princeton.edu/search.html. The knight model was retrieved from the 3DCafe website, http://www.3dcafe.com

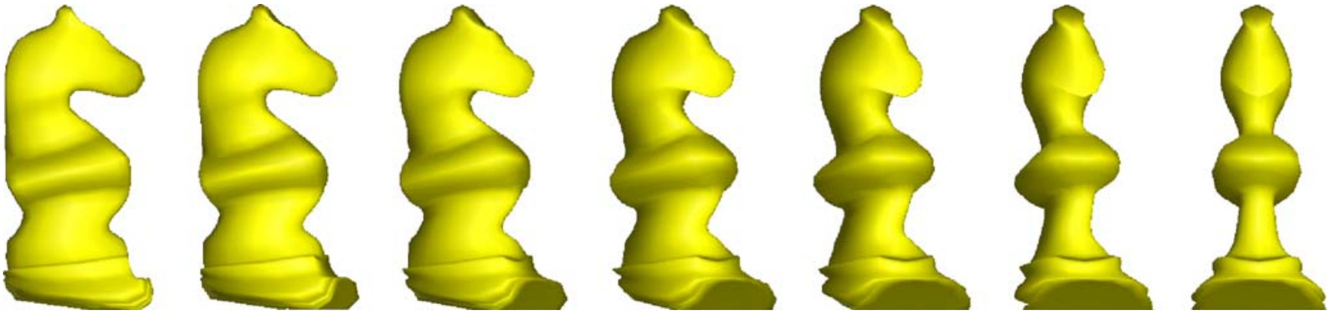


Fig. 10. The Knishop, rotated around the Y-axis, 90 degrees

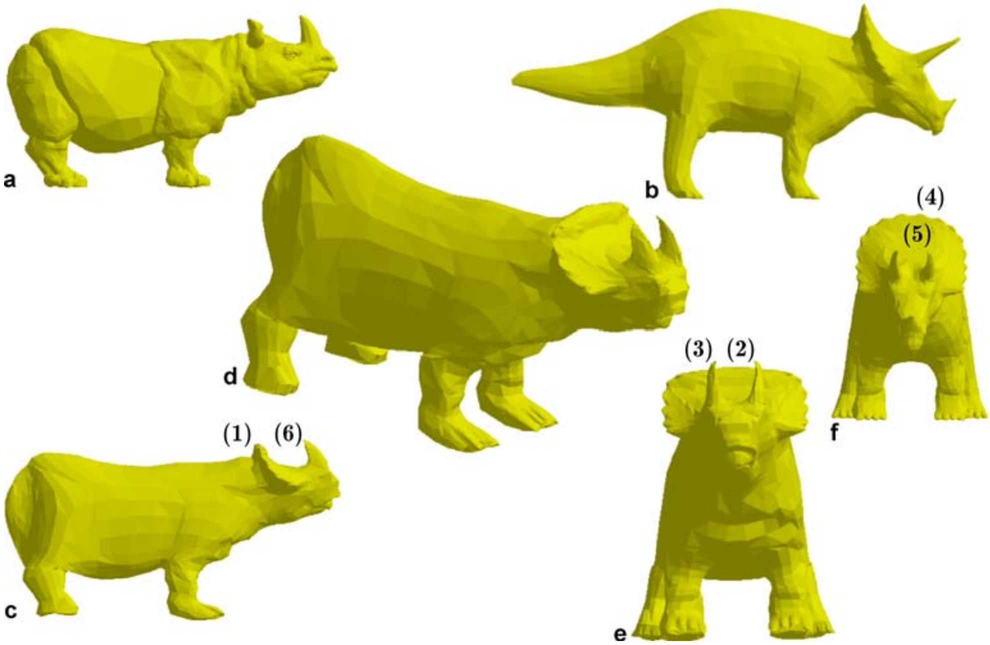


Fig. 11a–f. The Tricerhino (a triceratops–rhinoceros). **a** Shows the original rhinoceros model from the side and **b** shows the original triceratops model from the side. These are the scheme's considered silhouettes. The parts **c**, **d** and **e** show the resulting VDM model from the front, a general view and side, respectively, while **f** shows the original triceratops model from the front for comparison with **e**. Note that ideally, **a** and **c** should be identical, as should **e** and **f**. The triceratops model was retrieved from the Princeton 3D model search engine, http://shape.cs.princeton.edu/search.html. The rhinoceros model was retrieved from the 3DCafe website, http://www.3dcafe.com

bust were both extracted from the photographs of the original "Hasen-Spiegel" sculpture. Interestingly enough, the result ended up very similar to the actual wire sculpture, even when viewed from a general direction.

6 Conclusions and future work

We have presented an approach to non-realistic, artistic, modeling (NRM), in which the output is a visually intriguing 3D object, resembling two different objects when viewed from orthogonal angles. The work draws its inspiration from works of conceptual artists, in the art world.

There is a noticeable drawback to our scheme. When viewed from a direction perpendicular to the deforming direction, the silhouette of the original model does not remain intact. See, for example, the crest of the Triceratops in Fig. 11. In this example, the rhinoceros' ears (marked (1)) and horns (6) were matched to the Triceratops' frill (4) and horns (5), respectively. As the rhinoceros' silhouette requires that its ears and horns be at the same height, the front view of the resulting Tricerhino reveals that its frill and horns have been brought to the same height as well ((2) and (3)). Thus, the resulting

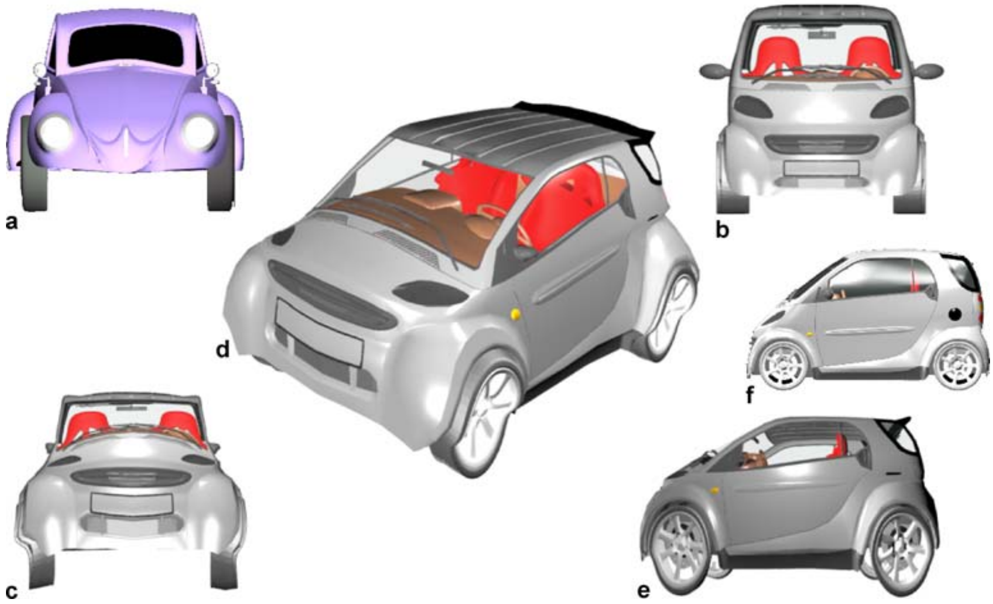


Fig. 12a–e. The Smart Beetle, a Mercedes–Benz Smart–VW Beetle. **a** Shows the original VW Beetle model from the front and **b** shows the original Smart model from the front. These are the scheme's considered silhouettes. The parts **c**, **d** and **e** show the resulting VDM model from the front, a general view and side, respectively, while **f** shows the original smart model from the front for comparison with **e**. The Smart model was retrieved from the 3DS resources website, http://www.3dsresources.com. The VW Beetle model was retrieved from the 3DCafe web site, http://www.3dcafe.com

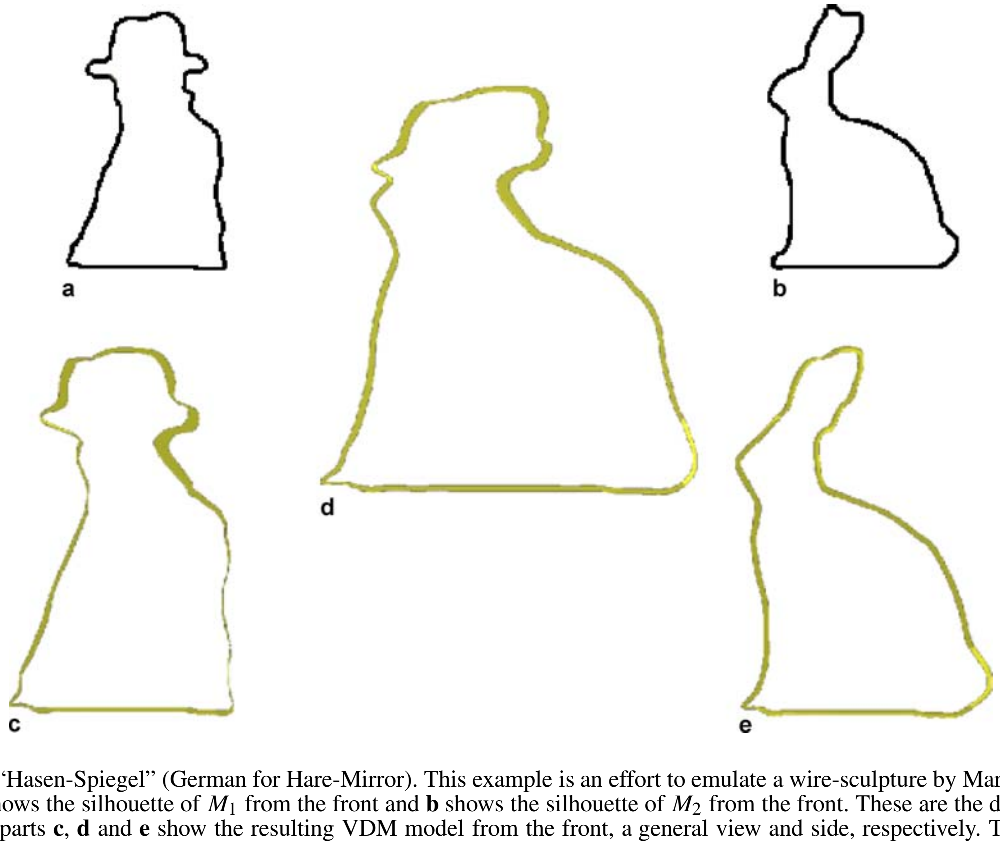


Fig. 13a–e. The "Hasen-Spiegel" (German for Hare-Mirror). This example is an effort to emulate a wire-sculpture by Markus Raetz, of the same name. **a** Shows the silhouette of M_1 from the front and **b** shows the silhouette of M_2 from the front. These are the desired outcome's silhouettes. The parts **c**, **d** and **e** show the resulting VDM model from the front, a general view and side, respectively. The silhouettes of the bunny and bust were extracted from photographs of the original sculpture

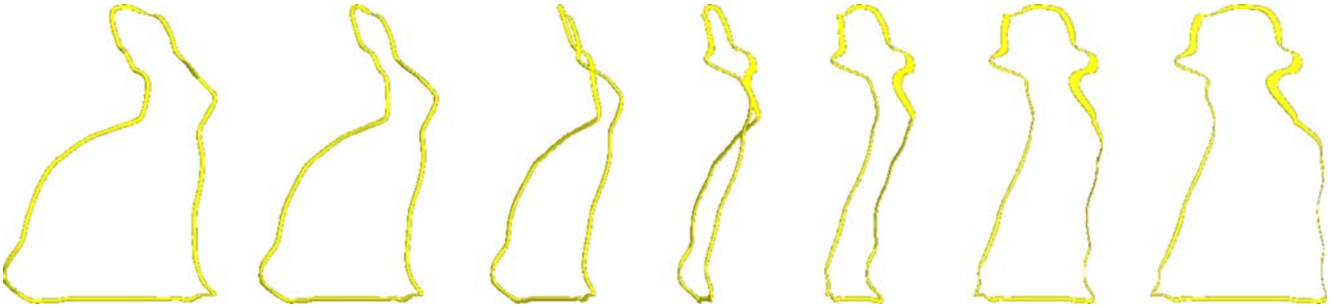


Fig. 14. The "Hasen-Spiegel", rotated around the Y-axis, 90 degrees

model does not exactly resemble the Triceratops when viewed from the front. The solution to this problem lies in a matching between $\mathcal{V}_1$ and $\mathcal{V}_2$, which would only allow movement of points on $\mathcal{V}_1$ in one direction; in this case, horizontally, towards their targets on $\mathcal{V}_2$. This kind of deformation would cause the vertices of the model to move only along the line of sight when viewing the model from the front, and eliminate this type of artifact. The drawback to this kind of matching is the loss of some freedom in the deformation.

There are two additional issues worth investigating, in the context of this work. The first is to determine whether a solution for S exists. The interpolation of the constraints is possible once the Schoenberg–Whitney conditions [2] are met. The minimization problem, on the other hand, requires that the matrices involved in the explicit solution be invertible, the conditions for which are still unclear, and also depend on the selected minimization functional. The second issue is the uniqueness of an (optimal) solution, as there may be many VDM models that satisfy certain demands.

Future work may also include an attempt to directly calculate a fair 3D volume, and not just a fair 2D surface, using a 3D set of constraints, allowing more degrees of freedom. Another interesting related problem to examine would be the arrangement of many random 3D objects such that their collective silhouette would resemble some object (or objects) from a specific viewing direction, and nothing in particular from any other direction, as seen in Shigeo Fukuda's "Underground Piano" [9]. Yet another interesting option could be to create an object with its right half similar to one input model, and its left half similar to a different input model, all from the same viewing direction. An interesting optimization issue considers the use of a different functional as the minimization functional T, instead of the bending energy functional used in this work. One viable alternative is stretching energy, but many others can be found in the relevant literature. Also worth exploring is the use of non-orthogonal viewing directions for silhouette extraction and deformation. Clearly, one can map the two orthogonal views of the model, $\mathcal{V}_1$ and $\mathcal{V}_2$, to two arbitrary views, using a shearing transformation, but an interesting question is whether one can handle more than two views, and if so, what conditions are needed for the creation of this kind of a VDM?

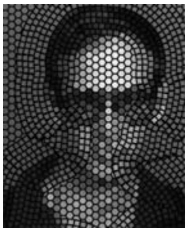
Acknowledgement This work was partially supported by European FP6 NoE grant 506766 (AIM@SHAPE) and in part by the Fund for Promotion of Research at The Technion, Haifa, Israel. The illustrations were prepared using the IRIT solid modeling system [8].

References

1. Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. IEEE Trans. Patt. Anal. Mach. Intell. **14**(2), 239–258 (1992)
2. de Boor, C.: A Practical Guide to Splines. Springer, Berlin Heidelberg New York (1978)
3. Bülthoff, H.H., Edelman, S.: Psychophysical support for a two-dimensional view interpolation theory of object recognition. In: Proceedings of the National Academy of Sciences of the United States of America **89**, 60–64 (1992)
4. Castleman, K.R.: Digital Image Processing. Prentice Hall, New York (1996)
5. Cohen, S., Elber, G., Bar-Yehuda, R.: Matching of freeform curves. Comput. Aided Des. **29**(5), 369–378 (1997)
6. Elber, G.: "Beyond Escher for Real" project. http://www.cs.technion.ac.il/~gershon/BeyondEscherForReal (2005)
7. Elber, G.: "Escher for Real" project. www.cs.technion.ac.il/~gershon/EscherForReal (2005)
8. Elber, G.: Irit solid modeler. http://www.cs.technion.ac.il/~irit (2005)
9. Fukuda, S.: "Works". http://neuro.caltech.edu/~seckel/mod/fukuda.htm (2005)
10. Greiner, G.: Variational design and fairing of spline surfaces. Comput. Graph. Forum **13**(3), 143–154 (1994)
11. Hadenfeld, J.: Local energy fairing of B-spline surfaces. In: M. Dæhlen, T. Lyche, L.L. Schumaker (eds.) Proceedings of the First Conference on Mathematical Methods for Curves and Surfaces (MMCS-94), pp. 203–212. Nashville, USA (1995)
12. Hahmann, S., Konz, S.: Knot-removal surface fairing using search strategies. Comput. Aided Des. **30**(2), 131–138 (1998)
13. Hsu, W.M., Hughes, J.F., Kaufman, H.: Direct manipulation of free-form deformations. In: SIGGRAPH, pp. 177–184 (1992)
14. Hu, S.M., Zhang, H., Tai, C.L., Sun, J.G.: Direct manipulation of FFD: Efficient explicit solutions and decomposible multiple point constraints. Vis. Comput. **17**(6), 370–379 (2001)
15. Hua, J., Qin, H.: Free-form deformations via sketching and manipulating scalar fields. In: Symposium on Solid Modeling and Applications, pp. 328–333 (2003)

16. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A sketching interface for 3D freeform design. In: SIGGRAPH, pp. 409–416 (1999)
17. Johan, H., Koiso, Y., Nishita, T.: Morphing using curves and shape interpolation techniques. In: PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, p. 348. IEEE Computer Society, Washington, DC, USA (2000)
18. Moreton, H.P., Séquin, C.H.: Functional optimization for fair surface design. In: SIGGRAPH, pp. 167–176 (1992)
19. Raetz, M.: "Markus Raetz – Metamorphose". http://www.crownpoint.com/artists/raetz/index2.html (2005)
20. Raviv, A., Elber, G.: Three dimensional freeform sculpting via zero sets of scalar trivariate functions. Comput. Aided Des. **32**(8/9), 513–526 (2000)
21. Rodrigues, W., Last, M., Kandel, A., Bunke, H.: 3-dimensional curve similarity using string matching. In: Proceedings of 3rd International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey (2001)
22. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. Comput. Graph. **20**, 151–160 (1986)
23. Vassilev, T.I.: Fair interpolation and approximation of B-splines by energy minimization and points insertion. Comput. Aided Des. **28**(9), 753–760 (1996)
24. Wikipedia: "Clairaut's theorem." http://en.wikipedia.org/wiki/Clairaut's_theorem (2005)
25. Yoshizawa, S., Belyaev, A.G., Seidel, H.P.: Free-form skeleton-driven mesh deformations. In: Symposium on Solid Modeling and Applications, pp. 247–253 (2003)
26. Zhang, C., Zhang, P., Cheng, F.: Fairing spline curves and surfaces by minimizing energy. Comput. Aided Des. **33**(13), 913–923 (2001)

GUY SELA is a Computer Science Masters Graduate. His research interests include computer graphics, 3D geometric modeling, medical simulations using computers and computer aided geometric design. Guy Sela received a BA in computer science and an MSc in computer science from the Technion, Israel in 2003 and 2006, respectively.



GERSHON ELBER is a professor in the Computer Science Department, Technion, Israel. His research interests span computer aided geometric designs and computer graphics. Prof. Elbe received a BSc in computer engineering and an MSc in computer science from the Technion, Israel in 1986 and 1987, respectively, and a PhD in computer science from the University of Utah, USA, in 1992. He is a member of the ACM and IEEE. Prof. Elber serves on the editorial board of the Computer Aided Design, Computer Graphics Forum, The Visual Computer, and the International Journal of Computational Geometry & Applications and has served in many conference program committees including Solid Modeling, Shape Modeling, Geometric Modeling and Processing, Pacific Graphics, Computer Graphics International, and SIGGRAPH. Prof. Elber was one of the paper chairs of Solid Modeling 2003 and Solid Modeling 2004. He has published over 100 papers in international conferences and journals and is one of the authors of a book titled "Geometric Modeling with Splines – An Introduction." Elber can be reached at the Technion, Israel Institute of Technology, Department of Computer Science, Haifa 32000, Israel.