EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2010)
M. Alexa and E. Do (Editors)

Sketching Variational Hermite-RBF Implicits

E. Vital Brazil[1,2], I. Macedo[1,2], M. Costa Sousa[2], L. H. de Figueiredo[1], L. Velho[1]

[1]IMPA – Instituto Nacional de Matemática Pura e Aplicada, Brazil
[2]University of Calgary, Canada

Abstract

We present techniques for modeling Variational Hermite Radial Basis Function (VHRBF) Implicits using a set of sketch-based interface and modeling (SBIM) operators. VHRBF Implicits is a simple and compact representation well suited for SBIM. It provides quality reconstructions, preserving the intended shape from a coarse and non-uniform number of point-normal samples extracted directly from the input strokes. In addition, it has a number of desirable properties such as parameter-free modeling, invariance under geometric similarities on the input strokes, suitable estimation of differential quantities, good behavior near close sheets, and both linear fitting and reproduction. Our approach uses these properties of VHRBF Implicits to quickly and robustly generate the overall shape of 3D models. We present examples of implicit models obtained from a set of SBIM language operators for contouring, cross-editing, kneading, oversketching and merging.

1. Introduction

An important class of sketch-based interface and modeling (SBIM) systems is known as *constructive* SBIM systems, which directly map a set of 2D sketched input strokes to a 3D model without any previous knowledge about the model's geometry or its topology [OSSJ09]. Constructive SBIM systems can be categorized by the two fundamental types of geometry being reconstructed: linear (i.e lines, planes and polyhedra) or free-form. The former is typically oriented towards CAD and architecture applications [ZHH96, JSC03], whereas the latter is used in applications requiring modeling of more organic, natural structures [IMt99, SWSJ05].

One important goal and challenge of constructive SBIM systems is to preserve the original modeling intent expressed by the user's 2D input sketch (i.e. "What You Sketch Is What You Get"). These systems aim to provide robust mechanisms to efficiently and effectively *interpret* the user's input strokes and *map* them to quality representations of both the geometry and topology of the intended 3D object. This is particularly important during the initial stages of conceptual, free-form prototype modeling, where the main goal is to construct the overall shape of the object for later refinement and augmentation.

Different representation structures have been proposed for free-from SBIM constructive systems, including triangle meshes [IMt99, KH06, CS07, NISA07], parametric [CSSJ05] and implicit surfaces [KHR02, AJ03, AGB04, SWSJ05, TZF04, BPCL08]. In particular, prototype free-form SBIM systems can benefit from using implicit surfaces, since they provide a compact, flexible and mathematically precise representation.

In this paper, we introduce a representation for implicit surfaces and show how it can be used to support a collection of free-form modeling operations. The main contributions of this work are: (1) an energy-minimizing Hermite interpolation scheme for sketch-based modeling (SBM) of free-form surfaces, and (2) a collection of SBM operators inspired by traditional illustration techniques to demonstrate the usefulness of the proposed representation.

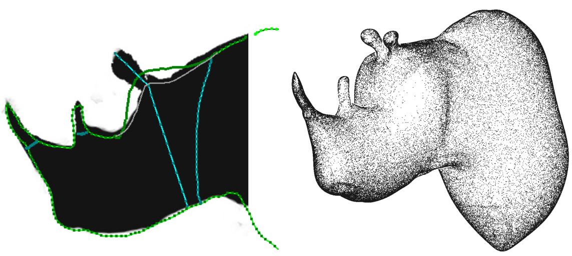

Figure 1: *Modeling a rhinocerus head with our system: the user interactively sketches construction curves over a silhouette image, resulting in a stippled implicit model.*

DOI: 10.2312/SBM/SBM10/001-008

2. Related work

Previous and related works concerning the SBM method proposed in this paper encompass the following areas: interactive modeling and visualization of implicit surfaces; sketch based interfaces; and surface reconstruction techniques from points and normals.

Early approaches for providing interactive implicit modeling and visualization involved creating a separate polygonization for each model primitive [DTG96]. Further work explored level-of-detail (LOD) implicit surface representations [BW90, BGA04], as well as the use of particle systems for both surface editing and rendering. Some of the main references along this line of research include [WH94, HBJF02, AJS07]. Other modeling techniques resort to a volumetric representation of implicit surfaces, such as the ones based on discrete volume datasets [ONNI03], adaptive distance fields [FCG00, PF01], and more recently using level set methods [MBWB02]. A different approach is based on interactively functional composed models [BMDS02]. Finally, we should mention the pioneering work in variational implicit surface modeling proposed by Turk and O'Brien [TO99].

Regarding sketch-based interfaces, implicit-based SBIM systems have contribute lately to provide effective and efficient modeling and visualization for implicits. Major efforts in this direction target adapting variational implicit surface models such as the works in [KHR02, AJ03, AGB04]. Sketch-based interfaces have also been applied to distance fields combined with mesh subdivision techniques [MCCH99, IH03], as well as to hierarchical implicit models [SWSJ05] and to convolution implicit surfaces [TZF04, BPCL08].

In terms of surface reconstruction methods, most related to our work are techniques presented in [KHR02, AJ03], derived from classical-interpolation theory by Duchon [Duc77] and subsequently introduced in graphics by Turk and O'Brien [TO99]. In these works, the creation of artificial offset-points is required to properly fit an implicit surface from a given set of points and normals. A recent approach proposed in [MGV09] reconstructs implicits from points and normals, based on a generalized interpolation framework, and exhibits increased robustness for coarse and nonuniform samplings, as well as to close-sheets without the need of artificial offsets. The key ingredient to this method is the treatment of normals as hard-constraints on the gradient field of the fitted implicit function. In [MGV09], the authors introduce a representation suitable for SBM of implicit surfaces built upon the works of Macêdo et al. [MGV09] and Duchon [Duc77], which combines the advantages of Hermite reconstruction of surfaces and the variational characteristic of those representations based on Duchon's work, most notably, the ability to extrapolate the surfaces well on void regions, thus filling unsampled areas.

3. Variational HRBF Implicits

Recently introduced by Macêdo et al. in [MGV09], HRBF Implicits provide a powerful tool to reconstruct implicitly-defined surfaces from points and normals. The authors present a theoretical framework for generalized interpolation with radial basis functions and specialize their results to first-order Hermite interpolation, which they subsequently employ to recover implicit surfaces passing through prescribed points with corresponding given normals.

We decided to investigate the use of their technique with SBM systems for implicits surfaces. We were motivated by the quality reconstructions they obtain from coarse and nonuniform samplings. This is especially important when compared to the offset-based methods often employed in sketch-based implicit modeling systems. Although we obtained good results in our experiments, the sparsity and coarseness of the strokes' curves rendered pointless the use of compactly-supported functions, and their associated radius parameter, since their main advantage lies in exploiting sparsity of the interpolation linear system when dealing with large datasets. This structure could not be exploited since the radius had to be taken large enough to compensate for the large areas not covered by the sparse set of strokes.

These shortcomings motivated us to look for alternatives which better exploited the structure of our application: sparsely-drawn strokes leading to small- to medium-sized datasets with large unsampled areas. We found an answer to these issues in Duchon's seminal paper [Duc77], which contains derivations analogous to those in [MGV09] for a particular class of function spaces. It is noteworthy that, even though the classical-interpolation part of Duchon's work has influenced the offset-based methods used for variational implicit modeling today, we focus on his Hermite interpolation result and introduce Variational HRBF Implicits.

In the following, we provide a brief description of the Variational HRBF Implicits interpolant and its key properties which motivated its use for SBM of implicit surfaces.

3.1. Description

Macêdo et al. [MGV09] posed the problem of reconstructing an implicit surface from points $\{\mathbf{x}^j\}_{j=1}^N \subset \mathbb{R}^3$ and normals $\{\mathbf{n}^j\}_{j=1}^N \subset \mathbb{S}^2$ as a Hermite interpolation problem, in which a function $f : \mathbb{R}^3 \to \mathbb{R}$ was sought such that $f(\mathbf{x}^j) = 0$ and $\nabla f(\mathbf{x}^j) = \mathbf{n}^j$, for each $j = 1, \ldots, N$. The authors presented a framework which allowed them to derive a concrete form for such an interpolant, which they named *HRBF Implicit*, upon the specification of a radial basis function $\psi : \mathbb{R}^3 \to \mathbb{R}$.

The variational Hermite interpolant deduced by Duchon in [Duc77] has essentially the same form of a HRBF Implicit with $\psi(\mathbf{x}) = \|\mathbf{x}\|^3$, differing only by an additional low-degree polynomial term which comes accompanied by additional conditions to ensure well-posedness of the interpolation problem. It should be noted that this RBF does not

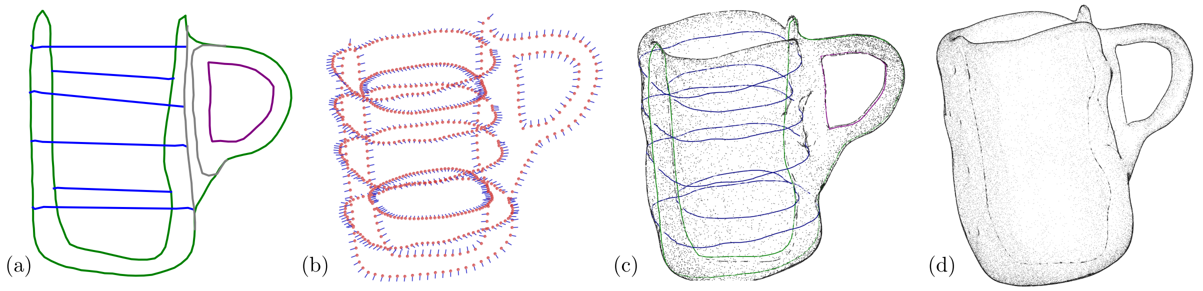

Figure 2: *Pipeline: (a) user input sketches, (b) samples (points and normals), (c) preview rendering during a prototype modeling session, computed after preprocessing* 4*K primitives on about* 5 *seconds, (d) final rendering with* 100*K primitives processed in about* 5 *minutes. Green curve is the external countour, magenta is the internal countour, the blue are the cross editing curves and the grey are curves not used after the oversketching operations.*

satisfy the sufficient conditions required by the framework presented in [MGV09]; however, Duchon's results ascertain the solvability of the interpolation system. For this reason, we named this representation a *Variational HRBF Implicit*.

For the sake of completeness and ease of reference, we provide the concrete form of a Variational HRBF Implicit,

$$f(\mathbf{x})=\sum_{j=1}^{N}\left\{\alpha_j\|\mathbf{x}-\mathbf{x}^j\|^3-3\langle\beta^j,\mathbf{x}-\mathbf{x}^j\rangle\|\mathbf{x}-\mathbf{x}^j\|\right\}+\langle\mathbf{a},\mathbf{x}\rangle+b.$$

with coefficients $\alpha_1,\ldots,\alpha_N,b\in\mathbb{R}$ and $\beta^1,\ldots,\beta^N,\mathbf{a}\in\mathbb{R}^3$. These coefficients can be uniquely determined by enforcing the Hermite interpolation conditions above along with

$$\sum_{j=1}^{N}\alpha_j=0,\qquad \sum_{j=1}^{N}\left\{\alpha_j\mathbf{x}^j+\beta^j\right\}=0$$

as long as the $\mathbf{x}^j$ are pairwise-distinct (a mild assumption). This implies that we can find an implicitly-defined surface passing through given points with prescribed normals just by solving a symmetric indefinite system of linear equations on the Variational HRBF Implicit's coefficients. In the following, we discuss the salient properties of this representation which are relevant for sketch-based implicit modeling. Later, we discuss some computational aspects of the numerical solution of our interpolation systems.

3.2. Variational HRBF for sketch-based modeling

The Variational HRBF Implicits representation has a number of interesting properties, either verifiable through theoretical arguments or indicated by experimentation, most notably:

Energy-minimizing Hermite interpolation. Duchon's deductions, which resulted in the variational Hermite interpolant, sought a function minimizing an energy measure on its derivatives under the pointwise Hermite interpolation constraints. This energy-minimizing property on the function's derivatives induces an interpolant which penalizes spurious oscillations, thus producing well behaved surfaces obeying both point and normal constraints from the strokes.

Invariance under geometric similarities. It can be shown that the implicit surface recovered as a Variational HRBF Implicit is invariant to geometric similarity transformations on the input samples. Intuitively this means that the shape of the reconstructed surface is the same, no matter how we consistently move, rotate, reflect or uniformly scale the input samples. Even though this property is obviously desirable in any method, many of them cannot provide such a guarantee.

Parameter-free. The Variational HRBF Implicit interpolant does not depend on any kind of support-radius or normal-offset parameters usually required by current RBF-based representations.

Smoothness. It is guaranteed that the implicit function computed is globally C^1, is infinitely differentiable in the whole domain but the sample positions, and has bounded Hessian near the interpolated points. This property allows suitable estimation of differential quantities such as normals, curvatures and principal directions at regular points.

Robustness under sparse samplings. Since the recovered implicit function's gradient interpolates the unit normals provided along with the input points, it behaves robustively in their vicinity even under nonuniform and coarse samplings, while extrapolating the reconstructed surface well over large unsampled regions. This observed property is one of the main reasons Variational HRBF Implicits provide a powerful representation for SBM of implicit surfaces.

Well-behavior near close-sheets. By directly interpolating normals with the implicit function's gradient, the HRBF surface reconstruction can deal with close-sheets, which usually pose a difficulty for offset-based schemes.

Linear fitting. In order to fit a Variational HRBF Implicit from given points and normals, all that is required is to solve a symmetric indefinite linear system on $4N+4$ unknowns.

As we discuss later, this can be accomplished by employing off-the-shelf linear algebra packages.

General data. To ensure well-posedness of the Hermite interpolation process and the invertibility of the associated linear system, the only and mild requirement on the dataset is that the input points need to be pairwise-distinct.

Linear reproduction. If the data is sampled from an affine function (defining a hyperplane), the recovery is exact. Intuitively, the interpolant approximates the data as well as it can by an affine function then corrects the errors with the HRBF-expansion.

After describing our basic surface representation, we now proceed with its actual use in our proof-of-concept sketch-based implicit surface modeling system.

4. Modeling pipeline

Our pipeline is divided in two main steps. First, as preprocessing step, we handle the user input strokes and process them to create samples (points and normals) in $\mathbb{R}^3$ space. We then use a Hermite RBF fitter to obtain our final implicit surface (Figure 2). At this second stage, despite the mathematical theory being well-established (Section 3), there are some computational issues concerning numerical stability and performance. In the following subsections, we describe these issues, as well as the sketch preprocessing and the fitter used in our system.

4.1. Sketch preprocessing

The process starts from a blank canvas, where the user sketches the curves for defining the final model. These curves are labeled as one of the sketch-based operators (Section 6). Next, all curves are processed (noise filtering) on the plane, and transported to model space. Normals are then created for each curve point while observing the specific rules of the associated operator. We assume all curves are parametrized in $[0,1]$.

Our noise filtering process is done in two steps. First, we supersample the input curve with a maximum distance of 0.5 pixels between two consecutive points. We then run 5 times the simplification algorithm described in [SB04]. This filter provides both good overall approximation of the intended stroke path and distribution of points along the curve (Figure 3).

The next step is to create the samples (points and normals) which will be interpolated by our implicit surface. A set of SBM operators was created for testing our implicit representation. These operators are grouped into two classes for creating *contour* and *inflation* curves. In contrast to inflation curves, which are immersed in $\mathbb{R}^3$ but not necessarily on a plane, contours are planar curves. All strokes are sketched on the plane $\Pi_d : (x,y,0)$.

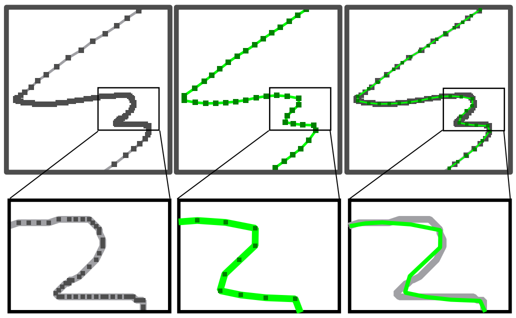

Figure 3: *Input stroke: raw input (left), after filtering (middle) and overlapping for comparison (right).*

Since the contour curves C are directly transported by an affine transformation to the final position in space, we can consider the contour points as in Π_d. We also use this plane as a reference frame for the inflation curves. It is worth noticing that $\mathbf{e}_z = (0,0,1)$ is the normal of the Π_d. We treat two different types of contour curves: *incomplete* and *complete* contours (Section 6.1). To assign normals to contour samples, we compute the cross-product between the normal to the drawing plane $\mathbf{e}_z$ and the unit tangent vector $\mathbf{t}$ at the given sample point on the curve, and assigning to that point the normal $\mathbf{n} = \mathbf{e}_z \times \mathbf{t}$.

Inflation curves are used to create curves outside the drawing plane. We have two operators to create them: *kneading* and *cross-editing* (Section 6.2). A kneading curve is transported to a user-defined plane, and its normal assigned to each sample. For simplicity, we define this plane as a translated version of the original drawing plane Π_d, allowing the user to flip its normal.

For each cross-editing curve $I(t)$ drawn, we associate two plane curves $\gamma_1(t)$,$\gamma_2(t) : [0,1] \to [0,1] \times [0,1]$. With x_t^i and y_t^i being the coordinates of γ_i at t, we define two 3D curves $I_i(t) = I(x_t^i) + y_t^i \cdot \mathbf{e}_z$ for each $i = 1,2$. The normal assigned to $I_i(t)$ is the normal of $\gamma_i(t)$ transported to the plane defined by the tangent of $I(x_t)$ and $\mathbf{e}_z$. The curves γ_i can be defined either automatically, as fixed functions of the arc-length of the curve I, or by input sketches. Both I_i are then sampled to generate the input points and normals.

4.2. Fitting Variational HRBF Implicits

To compute the coefficients of a Variational HRBF Implicit, we need to solve a symmetric indefinite system of linear equations with $4N+4$ unknowns. Relying on the quality reconstructions obtained from very sparse samplings, we employ LAPACK's dense linear algebra subroutines [LAP10] to perform the fitting computations.

We performed experiments with three basic strategies available in LAPACK for solving the Variational Hermite RBF interpolation system: a *LU* solver for general square matrices (xGESV), a LDL^T-based solver for symmetric indefinite systems (xSYSV) and a version of this last routine for matrices stored in packed form (xSPSV). Our experiments compared which solver would be more appropriate to compute the HRBF's coefficients in our SBM system. We assessed three basic aspects: performance, memory requirements and computational stability. We concluded that the full LDL^T solver xSYSV provided the best trade-offs among these three aspects.

5. Rendering approach

In our work, the resulting implicit models are rendered using a simplified version of the technique presented in [VMC*10]. This method was designed to depict shape and tone for surfaces represented as HRBF Implicits by relying on a few basic geometric operations to place and modulate the tone of point-based (stippling) primitives. To keep the modeling sessions interactive, we use coarse samplings to feed our stippling renderer to achieve fast visualization previews of the implicit surfaces. Figure 2(c) shows a typical model preview computed after a 5s-long preprocessing to place 4K points. If a higher quality depiction of the modeled surface is desired, the user can incrementally refine the stippling. A very useful feature of this technique is its ability to depict hidden structures in the surface, providing the user a good perception of occluded regions of the model as well as the spatial relations between the final shape of the model and the strokes being sketched. Figure 2(d) shows a frame from a simple final rendering computed after preprocessing 100K primitives for about 5min.

6. Sketch-based modeling operators

In this section, we present a set of SBM operators to create and test a variety of implicit objects modeled using our Variational HRBF scheme. The design of these SBM operators were inspired by traditional techniques and tools used by artists and illustrators [And06].

6.1. Contouring

To outline the exterior and interior contours of a model, the user has two different options: *incomplete* and *complete* contours (Figure 4). The first one is used to suggest the contour of the model using a few number of open curves [And06]. The second one is used to define the entire contour of the model using closed curves. In both operators, the drawing direction defines the sample normals. This allows the user to define either exterior or interior contours (i.e holes) by drawing clockwise or counterclockwise, respectively.

6.2. Inflation

There are many ambiguities in how to inflate a model with only coplanar samples [KH06]. After the user delimits the overall shape contour, its inflation can be controlled by using two classes of curves: *kneading* and *cross-editing*.

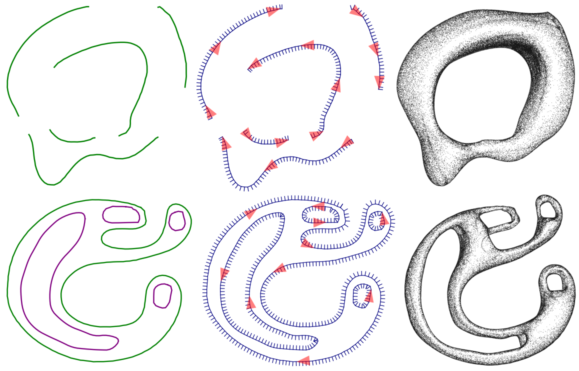

Figure 4: *Contouring operator: open and closed curves defined by incomplete (top row) and complete (bottom row) contours. Left to right: input strokes, normals and final stippled model. Red triangles depict curve orientation.*

Kneading curves allows to freely mold regions across the model. This operator is inspired by traditional kneaded erasers made of pliable material that can be shaped by hand. By using this operator the user has additional control on the inflation of the model (Figure 5), most notably when combined with the cross-editing operator (Figure 6).

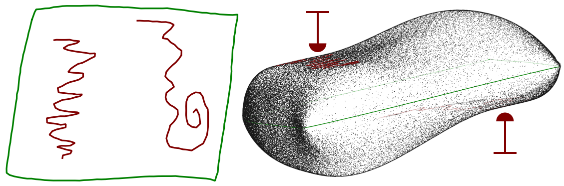

Figure 5: *Using kneading to control the inflation. (Left) Input strokes of the external contour of the model (green) and the free-form curves (red) defining the regions to be kneaded. (Right) The resulting 3D model with red arrows indicating the kneading orientation.*

Cross-editing curves define the profiles of the final model. These curves are built in two steps: first the user draws a path over Π_d and then defines the profiles of the curves. Each path is associated with two curves defining the top and bottom regions in relation to the drawing plane. These paths represent the projection of the final 3D curve on the drawing plane. Our system provides the users two options to define these profiles. The first option automatically creates two semicircles while the second one allows the user to define a profile by sketching it on an auxiliary canvas. Cross-editing curves can be used to define either linear (Figure 7(a)) or free-form cross-sections (Figure 7(b)).

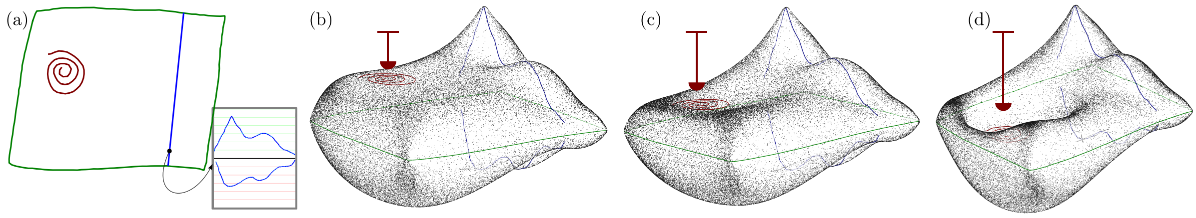

Figure 6: *Combining different kinds of inflation: (a) sketched curves defining the external contour (green), cross-editing (blue) and kneading (red); (b), (c) and (d) final models with different levels of kneading.*

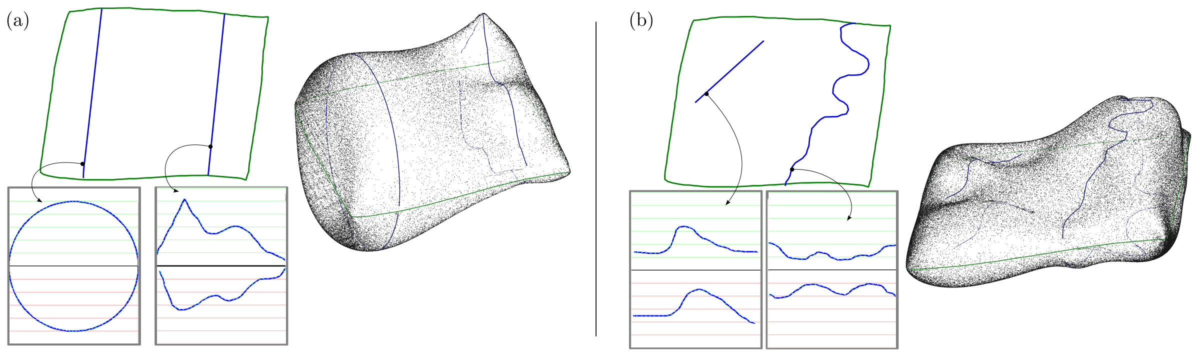

Figure 7: *Using the cross-editing curves to define the inflation: (a) cross-section paths, with left and right curves for the default and free-form profiles, respectively; (b) free-form paths, both curves using free-form sketched profiles.*

6.3. Oversketching

The oversketching operator is applied over contour curves and it is organized in two classes. The first class corresponds to a correction or augmentation to a single curve (Figure 8(a)). The user draws a new curve S near an existing curve C to be modified; $S(0)$ and $S(1)$ are then projected onto the line C to define two pieces of C between these two projected points, one of which will be replaced by S. The ambiguity on which piece will be replaced is solved by simply enforcing coherence on the orientation of the final curve. The second class of oversketching operator allows to merge two existing curves (Figure 8(b)). The user initiaslly selects these two curves, C_1 and C_2, to be merged and then draw two paths, P_1 and P_2. Next, $P_1(0)$ and $P_2(0)$ are both projected onto C_1, while $P_1(1)$ and $P_2(1)$ onto C_2. The curve segments between those points are eliminated and the new curve is built by 'gluing' C_1, P_1, C_2 and, P_2. For this class of operator, the ambiguities are solved by both the orientation criterion used for the first class of oversketching operator and the order in which the paths are drawn.

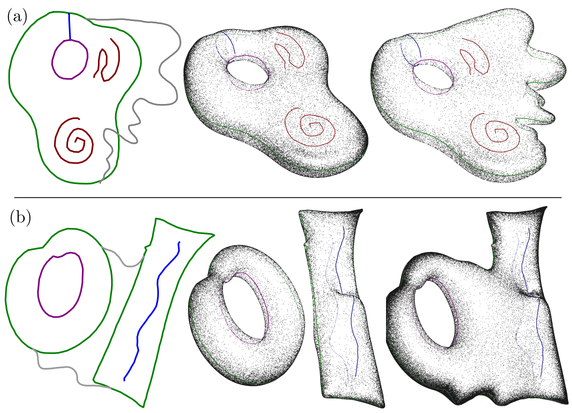

Figure 8: *Oversketching (grey curves): (a) editing a single contour, (b) blending two contours.*

7. Results and discussion

Our SBM techniques successfully model Variational HRBF Implicit surfaces by interpolating the sketched curves, thus preserving the intended shape to be modeled. All the results were generated on a 2.66 GHz Intel Xeon W3520, with 4 gigabytes of RAM and an OpenGL/nVIDIA Quadro FX 3800 graphics card. The user sketches directly on a Wacom Cintiq interactive pen display. All input strokes preprocessing, Variational HRBF surface fitting, and run-time rendering were computed on the CPU only.

Results were generated for a variety of shapes and objects, sketched by users with different drawing abilities. We observed that the overall modeling time (from the initial input strokes to the finished model) took around 15 minutes. As expected more samples result in more complex Variational HRBF computations. However, we observed that these computations took less than 5 seconds. Our approach produces promising results requiring few strokes from the user to construct a model while preserving the intended shape. We evaluated our results by observing the timings for the overall modeling stages and the quality of the final model.

Figures 4 to 8 show examples of each of our sketch-based modeling operators. Figure 2 shows the result of modeling a simple mug. The user sketched 17 strokes (including cross-editing and oversketching curves), resulting in 720 samples, fitted in less than 2 seconds. The overall modeling (from initial strokes to finished model) took 12 minutes. Figures 1 and 10 show the result of sketching directly over a reference image to reconstruct its 3D shape. For the rhinocerus head (Figure 1), 11 strokes were sketched, resulting in 490 samples, fitted in less than 1 second. The overall modeling took 9 minutes. For the rubber duck model (Figure 10), the user sketched 19 strokes, resulting in 989 samples, fitted in less than 3 seconds. The overall modeling took around 14 minutes. Figure 9 shows the result of modeling a terrain in 22 strokes, 838 samples, fitted in less than 3 seconds. The overall modeling took about 16 minutes.

8. Conclusion and future work

In this work, we introduced a representation for variational implicit surfaces suitable for sketch-based modeling. This representation retains the good properties of the currently used techniques for variational implicits while additionally being more robust to the sparse and coarse samplings resulting from sketch modeling sessions even in the presence of close-sheets. Moreover, the representation does not depend on artificial parameters commonly found in other RBF-based methods, e.g. offsets and support-radii.

We employ this representation as the basis for a proof-of-concept sketch-based modeling system for which we introduce a small, albeit powerful, set of simple modeling operators for contouring, cross-editing, kneading, oversketching and also merging different parts of a model.

There are still many avenues for improvement and further investigation. On the modeling side, we plan to extend the language and operators to support more complex edition tasks as well as specific application-domains. Also, there is the need for a systematic usability evaluation in order to assess the language and operators and point directions to enhance them as well as our system's user interface.

On the technical side, we are currently investigating methods to increase the scalability of the surface reconstruction and optimize the evaluations of the implicit function. Most notably, we had some encouraging preliminary results generalizing the work of Beatson et al. [BLB01] for our first-order Hermite interpolation problem.

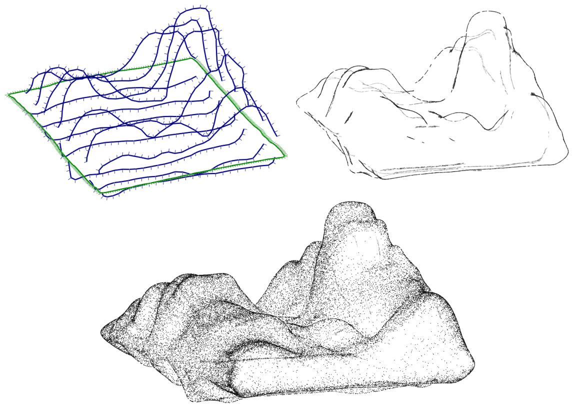

Figure 9: *Modeling a terrain: construction curves, silhouettes and stippling.*

Acknowledgements

We would like to thank Nicole Sultanum and Rafael Simon Maia for their useful discussions and advice. We also thank the anonymous reviewers for their careful and valuable comments and suggestions. This research was supported in part by the iCORE/Foundation CMG Industrial Research Chair in Scalable Reservoir Visualization, by Discovery Grants Program from the Natural Sciences and Engineering Research Council of Canada, and grants from the Brazilian funding agencies CNPq and CAPES/PDEE.

References

[AGB04] ALEXE A., GAILDRAT V., BARTHE L.: Interactive modelling from sketches using spherical implicit functions. In *Proc. of AFRIGRAPH '04* (2004), pp. 25–34.

[AJ03] ARAÚJO B., JORGE J.: Blobmaker: Freeform modelling with variational implicit surfaces. In *Proc. of 12th Enc. Português de Computação Gráfica* (2003), pp. 17–26.

[AJS07] ARAÚJO B., JORGE J., SOUSA M.: Sampling point-set implicits. In *Proc. of 4th IEEE/Eurographics Symposium on Point-Based Graphics (PBG '07)* (2007).

[And06] ANDREWS W.: Introduction to perceptual principles in medical illustration: Lines & illusions. In *Tutorial "Illustrative Visualization for Medicine and Science", Eurographics '06* (2006).

[BGA04] BARBIER A., GALIN E., AKKOUCHE S.: Complex skeletal implicit surfaces with levels of detail. In *Proc. of 12th Intl. Conf. in Central Europe on Computer Graphics, Vis. and Computer Vision (WSCG '04)* (2004), vol. 12, pp. 35–42.

[BLB01] BEATSON R. K., LIGHT W. A., BILLINGS S.: Fast Solution of the Radial Basis Function Interpolation Equations: Domain Decomposition Methods. *SIAM Journal on Scientific Computing 22*, 5 (2001), 1717–1740.

[BMDS02] BARTHE L., MORA B., DODGSON N., SABIN M.: Interactive implicit modelling based on c1 reconstruction of regular grids. *Intl. Journal of Shape Modeling 8*, 2 (2002), 99–117.

[BPCL08] BERNHARDT A., PIHUIT A., CANI M.-P., L. B.: Matisse: Painting 2D regions for modeling free-form shapes. In

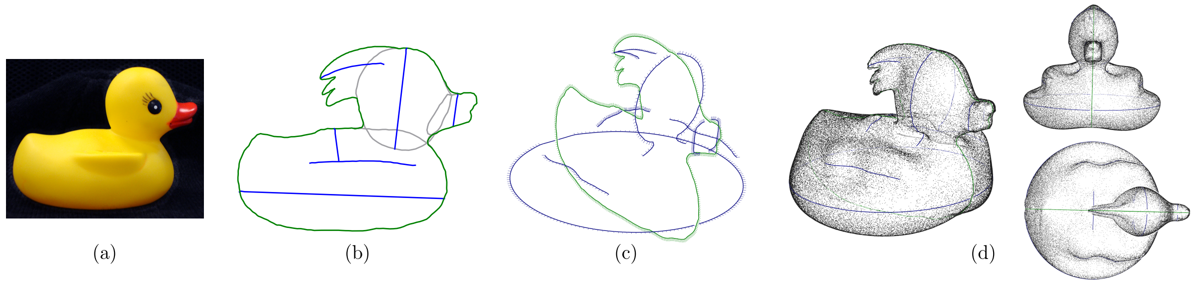

Figure 10: *Modeling a rubber duck: (a) reference photograph, (b) input strokes sketched directly over the photograph, (c) 3D construction curves with normals and (d) stipple rendering of the final model.*

Proc. of 5th Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM '08) (2008), pp. 57–64.

[BW90] BLOOMENTHAL J., WYVILL B.: Interactive techniques for implicit modeling. In *Proc. of 1990 Symposium on Interactive 3D Graphics (I3D '90)* (1990), pp. 109–116.

[CS07] CORDIER F., SEO H.: Free-form sketching of self-occluding objects. *IEEE Comput. Graph. Appl. 27*, 1 (2007), 50–59.

[CSSJ05] CHERLIN J., SAMAVATI F., SOUSA M., JORGE J.: Sketch-based modeling with few strokes. In *Proc. of 21st Spring Conf. on Computer Graphics (SCCG '05)* (2005), pp. 137–145.

[DTG96] DESBRUN M., TSINGOS N., GASCUEL M.-P.: Adaptive sampling of implicit surfaces for interactive modelling and animation. *Computer Graphics Forum 15*, 5 (1996), 319–325.

[Duc77] DUCHON J.: *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, vol. 571 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1977, pp. 85–100.

[FCG00] FERLEY E., CANI M.-P., GASCUEL J.-D.: Practical volumetric sculpting. *Visual Computer 16*, 8 (2000), 469–480.

[HBJF02] HART J. C., BACHTA E., JAROSZ W., FLEURY T.: Using particles to sample and control more complex implicit surfaces. In *Proc. of Shape Modeling Intl. (SMI'02)* (2002), pp. 129–136.

[IH03] IGARASHI T., HUGHES J.: Smooth meshes for sketch-based freeform modeling. In *Proc. of 2003 Symposium on Interactive 3D Graphics (I3D '03)* (2003), pp. 139–142.

[IMt99] IGARASHI T., MATSOUKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform. In *Proc. of SIGGRAPH '99* (1999), pp. 409–416.

[JSC03] JORGE J. A., SILVA N. F., CARDOSO T. D.: Gides++. In *Proc. of 12th Encontro Português de Computação Gráfica* (2003).

[KH06] KARPENKO O., HUGHES J.: SmoothSketch: 3d free-form shapes from complex sketches. *ACM Trans. Graph. (SIGGRAPH '06) 25*, 3 (2006), 589–598.

[KHR02] KARPENKO O., HUGHES J., RASKAR R.: Free-form sketching with variational implicit surfaces. *Computer Graphics Forum 21*, 3 (2002), 585–594.

[LAP10] LAPACK: Linear Algebra PACKage, 2010. http://netlib.org/lapack/.

[MBWB02] MUSETH K., BREEN D. E., WHITAKER R. T., BARR A. H.: Level set surface editing operators. *ACM Trans. Graph. 21*, 3 (2002), 330–338.

[MCCH99] MARKOSIAN L., COHEN J. M., CRULLI T., HUGHES J.: Skin: A constructive approach to modeling free-form shapes. In *Proc. of SIGGRAPH '99* (1999), pp. 393–400.

[MGV09] MACÊDO I., GOIS J. P., VELHO L.: Hermite interpolation of implicit surfaces with radial basis functions. In *Proc. of XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '09)* (2009), pp. 1–8.

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: FiberMesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. (SIGGRAPH '07) 26*, 3 (2007), 41.

[ONNI03] OWADA S., NIELSEN F., NAKAZAWA K., IGARASHI T.: A sketching interface for modeling the internal structures of 3d shapes. In *Proc. of 4th International Symposium on Smart Graphics* (2003), pp. 49–57.

[OSSJ09] OLSEN L., SAMAVATI F., SOUSA M., JORGE J.: Sketch-based modeling: a survey. *Computer & Graphics 33*, 1 (2009), 85–103.

[PF01] PERRY R. N., FRISKEN S. F.: Kizamu: A system for sculpting digital characters. In *Proc. of SIGGRAPH '01* (2001), pp. 47–56.

[SB04] SAMAVATI F., BARTELS R.: Local filters of b-spline wavelets. In *Proc. of Intl. Workshop on Biometric Technologies 2004* (2004), pp. 105–110.

[SWSJ05] SCHMIDT R., WYVILL B., SOUSA M., JORGE J.: ShapeShop: Sketch-based solid modeling with blobtrees. In *Proc. of 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM '05)* (2005), pp. 53–62.

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In *Proc. of SIGGRAPH '99* (1999), pp. 335–342.

[TZF04] TAI C.-L., ZHANG H., FONG J.-K.: Prototype modeling from sketched silhouettes based on convolution surfaces. *Computer Graphics Forum 23*, 1 (2004), 71–83.

[VMC*10] VITAL BRAZIL E., MACÊDO I., COSTA SOUSA M., DE FIGUEIREDO L. H., VELHO L.: A Few Good Samples: Shape & Tone Depiction for Hermite RBF Implicits. In *Proc. of 8th Intl. Symposium on Non-Photorealistic Animation and Rendering (NPAR '10)* (2010).

[WH94] WITKIN A. P., HECKBERT P. S.: Using particles to sample and control implicit surfaces. In *Proc. of SIGGRAPH '94* (1994), pp. 269–278.

[ZHH96] ZELEZNIK R. C., HERNDON K. P., HUGHES J. F.: SKETCH: an interface for sketching 3d scenes. In *Proc. of SIGGRAPH '96* (1996), pp. 163–170.