Hermite Interpolation of Implicit Surfaces with Radial Basis Functions

Ives Macêdo*
ijamj@impa.br

João Paulo Gois†
joao.gois@ufabc.edu.br

Luiz Velho*
lvelho@impa.br

Vision and Graphics Laboratory, Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de Janeiro, RJ, Brazil
†*Centro de Matemática, Computação e Cognição, Universidade Federal do ABC (UFABC), Santo André, SP, Brazil*

***Abstract*—We present the *Hermite radial basis function (HRBF) implicits* method to compute a global implicit function which interpolates scattered multivariate Hermite data (unstructured points and their corresponding normals). Differently from previous radial basis functions (RBF) approaches, HRBF implicits do not depend on offset points to ensure existence and uniqueness of its interpolant. Intrinsic properties of this method allow the computation of implicit surfaces rich in details, with irregularly spaced points even in the presence of close sheets. Comparisons to previous works show the effectiveness of our approach. Further, the theoretical background of HRBF implicits relies on results from generalized interpolation theory with RBFs, making possible powerful new variants of this method and establishing connections with previous efforts based on statistical learning theory.**

***Keywords*-Implicit surfaces, Hermite data, radial basis functions, Hermite-Birkhoff interpolation, scattered data approximation, geometric modeling, surface reconstruction**

I. Introduction

The computation of implicit surfaces that approximate or interpolate scattered data is a classical problem in Computer Graphics and Geometric Modeling. In particular, the interpolation of scattered data sets is frequently desired in various specific applications, e.g., modeling [1], tracking of time-dependent surfaces [2] and polygon soup interpolation [3].

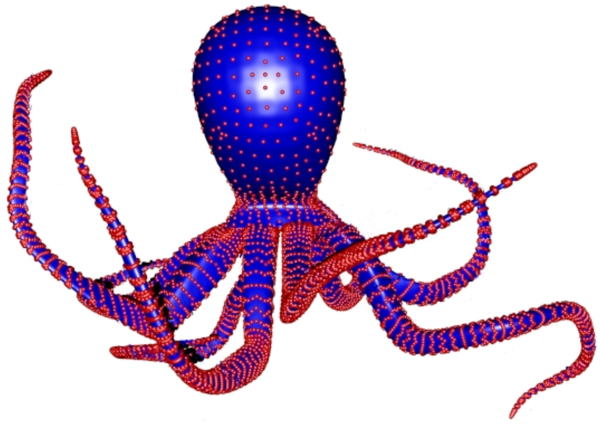


Figure 1. Hermite Radial Basis Function Implicits (octopus dataset, 29714 points/normals): Hermite Interpolation of an irregularly spaced dataset with close sheets and thin parts (plot of normals are omitted).

In this work we tackle the problem of computing an implicit function interpolating Hermite scattered data defined as a set $\mathcal{P}$ of unorganized points with their corresponding normal vectors. To that end, we propose a scheme named *Hermite Radial Basis Function (HRBF) Implicits* which interpolates, on its zero-level surface, simultaneously a given set of points and – differently from previous Radial Basis Function (RBF) approaches [4], [5] – their normal vectors. To develop this scheme, our formulation exploits theoretical results from Hermite-Birkhoff interpolation with RBFs [6]. Since we construct our method from such a general framework, new variants can be derived for additional flexibility.

In most previous methods based on RBFs, normals are only used to define *offset-points* which ensure the existence of a non-null interpolant implicit function. In one of the most used approaches to compute offset points is, two offset points are created at $\mathbf{x}^i \pm \epsilon\mathbf{n}^i$, for each input point $\mathbf{x}^i \in \mathcal{P}$, where $\mathbf{n}^i$ is the normal on $\mathbf{x}^i$, with associated offset scalars $\pm\epsilon$. It is not hard to see that such approaches tipicaly do not interpolate the given normals.

Another drawback is that those offset-points lead to numerical instabilities for small ϵ, whereas a large ϵ subjects the methods to generate poor results. In fact, a single optimal choice for ϵ does not exist in general. Other heuristics have been proposed [1], but all of them (to the best of our knowledge) are based on hand-tuned parameters and still subject to produce poor results.

Since HRBF implicits interpolate both points and normals, it not only ensures the existence of a non-null implicit function without the need of offset points, but it is also capable of generating effective results. In fact, HRBF implicits have the good properties shared by previous RBF methods, as being able to deal with quite irregular datasets, while experiments indicate it is also superior to related approaches [2], [4], [7] when dealing with *close sheets* [2], [7] (Fig. 2).

Many surface reconstruction methods deal with close sheets employing combinations of spatial data structures, geometrical heuristics (e.g. normal clustering), or even statistical inference to select points belonging to the same sheet before evaluating the implicit function [8], [9]. In this aspect, HRBF implicits are most similar to methods which build into the interpolation/approximation scheme a capability of identifying and handling close sheets without the need of additional information [2], [4], [5], [7].

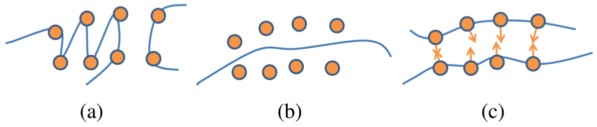


Figure 2. The issue of *close sheets* (see [2], [7]): when two parts of the surface are very close, many interpolation/approximation techniques behave poorly. Case (a) is an example of undesired interpolation, whereas case (b) is an example of undesired approximation. HRBF implicits, since it interpolates both points and normals, produces results similar to (c).

Properties and contributions of HRBF implicits

In the following, we enumerate the main properties and contributions of HRBF implicits.

Global implicit interpolant surface of Hermite data: HRBF implicits aim at computing a global implicit function whose zero-level interpolates given points and their derivatives, in our case, the normal vectors.

Offsets-free: Differently from previous RBF interpolants, HRBF implicits do not require any heuristics for creating off-surface points improving robustness.

Capability of handling irregularly-spaced data: Similarly to previous RBF-based methods, the HRBF implicits method is also able to compute reasonable interpolations even in the presence of irregular data distributions.

Flexibility for true Hermitian data sets: Although we consider in the present formulation "Hermitian data" as a set of scattered points and their associated normal vectors, the HRBF implicits method is more general since it allows constraining arbitrary gradient vectors for the implicit function on the sample points.

Capability to handle close sheets: Our results show that HRBF implicits allow for computing surfaces with close sheets [2], [7], and indicate that our Hermite-interpolatory method is superior to previous solutions in this situation.

Simple implementation: Our formulation and subsequent treatment build upon theoretical results from scattered data approximation theory [6] and concepts from functional analysis [10], yet it leads to a simple matrix-based algorithm that is a direct translation of the mathematical results. This allows a simple computational implementation general enough to be independent of the ambient space dimension. Moreover, the theoretical framework supporting HRBF implicits indicates directions to build variants of the basic method which may allow further flexibility.

II. Theoretical framework

We pose the problem of fitting an implicit surface defined by a function $f : \mathbb{R}^n \to \mathbb{R}$ to given Hermite data $\left\{\left(\mathbf{x}^i, \mathbf{n}^i\right)\right\}_{i=1}^N \subset \mathbb{R}^n \times \mathbb{R}^n$ as looking for a function satisfying both $f\left(\mathbf{x}^i\right) = 0$ and $\nabla f\left(\mathbf{x}^i\right) = \mathbf{n}^i$, for each data sample, from a certain space $\mathcal{H}$ of sufficiently smooth functions. In approximation theory, this problem is classified as an instance of first-order Hermite interpolation.

In this section, we present some results from a general theoretical framework of Hermite-Birkhoff interpolation with radial basis functions and its application to the first-order Hermite interpolation of multivariate scattered data. Our presentation is strongly influenced by Wendland's wonderful book [6] and, at least for this theoretical reasoning, a few basic results from functional analysis would be helpful [10].

A. Hermite-Birkhoff interpolation with RBFs

Hermite-Birkhoff interpolation is a generalized interpolation problem in which data consists of information regarding point evaluations of differential operators acting on a function, e.g. $f(\mathbf{x}) = c_0^{\mathbf{x}}$, $\frac{\partial f}{\partial z}(\mathbf{y}) = c_z^{\mathbf{y}}$ and $\frac{\partial^2 f}{\partial x \partial y}(\mathbf{z}) = c_{xy}^{\mathbf{z}}$. Differently from Hermite interpolation, Hermite-Birkhoff interpolation doesn't require one to provide the full set of values for each sample point, e.g. we might provide first-order information without the need to provide the function's value. This provides a flexible framework employed even for the numerical solution of partial differential equations [11].

More formally, given a Hilbert space of functions and a dataset $\{(\lambda_i, c_i)\}_{i=1}^N$ consisting of continuous linear functionals $\lambda_i \in \mathcal{H}^*$ and real numbers $c_i \in \mathbb{R}$, we are looking for a function $f \in \mathcal{H}$ such that $\lambda_i(f) = c_i$, for each i. Such a function is called a *generalized interpolant* of the data in $\mathcal{H}$. However, there can be infinitely many generalized interpolants for a dataset in $\mathcal{H}$. This uniqueness issue can be solved by taking that one with minimal $\mathcal{H}$-norm. The theory provides us with a characterization for this solution ensuring that it is a linear combination of the Riesz representers $v^i \in \mathcal{H}$ of the data functionals, i.e. $f^* = \sum_{i=1}^N \alpha_i v^i$ and the v^i are such that $\lambda_i(u) = \left\langle v^i, u\right\rangle_{\mathcal{H}}$ for every $u \in \mathcal{H}$.

Under the assumption of linear independence of the data functionals, the coefficients in the linear combination can be uniquely determined by enforcing the interpolation constraints, leading to a linear system $\mathbf{A}\boldsymbol{\alpha} = \mathbf{c}$, where the interpolation matrix is given by $(\mathbf{A})_{ij} = \lambda_i\left(v^j\right) = \left\langle v^i, v^j\right\rangle_{\mathcal{H}}$, hence it is an inner-product (Gramm) matrix, therefore symmetric and positive definite.

In Hermite-Birkhoff interpolation, the data functionals have the form $\lambda = \delta_{\mathbf{x}} \circ D^{\gamma}$, where $\delta_{\mathbf{x}}(u) = u(\mathbf{x})$ is the point-evaluation functional on $\mathbf{x} \in \mathbb{R}^n$ and D^{γ} is a differential operator in which the multi-index $\boldsymbol{\gamma} \in (\mathbb{N} \cup \{0\})^n$ indicates how many times (γ_j) the function will be differentiated with respect to the j-th variable (x_j). In this way, we have $\lambda_i(u) = \left(D^{\gamma^i} u\right)\left(\mathbf{x}^i\right)$. In words, the application of the functional λ_i on a function u consists of differentiating u according to $\boldsymbol{\gamma}^i$ and then evaluating the result at $\mathbf{x}^i$.

All we need is a construction of suitable spaces $\mathcal{H}$ in which the representers of the Hermite-Birkhoff functionals, as well as their inner-products, can be effectively computed. Indeed, it can be shown that such spaces can be implicitly constructed from certain radial functions. Formally, given a *positive definite radial basis function* $\phi : \mathbb{R}_+ \to \mathbb{R}$ such that $\psi := \phi(\|\cdot\|) \in C^{2k}(\mathbb{R}^n) \cap \mathbb{L}_1(\mathbb{R}^n)$ $\left(k \geq \left|\boldsymbol{\gamma}^i\right|, \forall i\right)$, there exists a unique (up to isometric isomorphisms) *native* Hilbert space $\mathcal{H}$ in which the λ_i are continuous and, as long as they are pairwise distinct, they are also linearly independent. Moreover, their representers have the form $v^i(\mathbf{x}) = -\left(D^{\gamma^i}\psi\right)\left(\mathbf{x} - \mathbf{x}^i\right)$ and their inner-product given by $\left\langle v^i, v^j \right\rangle_{\mathcal{H}} = -\left(D^{\gamma^i} D^{\gamma^j}\psi\right)\left(\mathbf{x}^i - \mathbf{x}^j\right)$ [6]. The most notable examples of such RBFs are the Gaussians and the compactly supported (piecewise polynomial) Wendland's functions [12], which we employ in our implementation.

The main advantage of employing spaces implicitly constructed by these RBFs is that they provide explicit means to compute both the Riesz representers and their inner-products, making computationally feasible the solution of a minimum norm generalized multivariate interpolation problem with scattered Hermite-Birkhoff data functionals. Other advantages come from the vast literature on both theory and practice of scattered data approximation with radial basis functions, opening many avenues for further research, some of which we discuss in the last section.

B. First-order Hermite interpolation with RBFs

We now exploit the theoretical developments on general Hermite-Birkhoff interpolation with RBFs to our model of implicit surface reconstruction from exact Hermite data.

In our problem statement, we assume to have a dataset consisting of N points on the surface we are interested in recovering and their associated normals. Posing the problem in the framework above, we have $n+1$ kinds of data for each sample: $(\delta_{\mathbf{x}^i}, 0)$, $\left(\delta_{\mathbf{x}^i} \circ \frac{\partial}{\partial x_1}, \left(\mathbf{n}^i\right)_1\right)$, ..., $\left(\delta_{\mathbf{x}^i} \circ \frac{\partial}{\partial x_n}, \left(\mathbf{n}^i\right)_n\right)$. Therefore, the representation results above ensure that the interpolant we are interested in can be written in the form:

$$f^*(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \psi\left(\mathbf{x} - \mathbf{x}^j\right) - \sum_{j=1}^{N} \left\langle \boldsymbol{\beta}^j, \nabla\psi\left(\mathbf{x} - \mathbf{x}^j\right)\right\rangle \quad (1)$$

where each of the coefficients $\alpha_j \in \mathbb{R}$ and $\boldsymbol{\beta}^j \in \mathbb{R}^n$ can be uniquely determined by enforcing the interpolation conditions $f^*\left(\mathbf{x}^i\right) = 0$ and $\nabla f^*\left(\mathbf{x}^i\right) = \mathbf{n}^i$, $\forall i$. Leading to

$$\sum_{j=1}^{N} \alpha_j \psi\left(\mathbf{x}^i - \mathbf{x}^j\right) - \sum_{j=1}^{N} \left\langle \boldsymbol{\beta}^j, \nabla\psi\left(\mathbf{x}^i - \mathbf{x}^j\right)\right\rangle = 0 \quad (2)$$

$$\sum_{j=1}^{N} \alpha_j \nabla\psi\left(\mathbf{x}^i - \mathbf{x}^j\right) - \sum_{j=1}^{N} \mathrm{H}\,\psi\left(\mathbf{x}^i - \mathbf{x}^j\right)\boldsymbol{\beta}^j = \mathbf{n}^i \quad (3)$$

where H is the Hessian operator defined by $(\mathrm{H})_{kl} := \frac{\partial^2}{\partial x_k \partial x_l}$. Here, the requirement that ψ be at least twice as much continuously differentiable as the maximum order of the differential operators in the data functionals is more concrete. Therefore, for our implicit surface reconstruction problem, we need to choose a RBF at least C^2, resulting in a function at least C^1 (naturally, unless zero is a regular value of f^*, we can't guarantee *a priori* that our "surface" will be C^1).

We are now able to numerically implement a method for Hermite interpolation of implicit surfaces with radial basis functions. We discuss computational aspects of this method in the next section. Nevertheless, we first comment on an existing alternative method to *approximate* given Hermite data and its connection to our interpolatory scheme.

C. A note on a regularization-based approach

In the framework we presented, we have taken an interpolatory approach to recover an implicit surface from Hermite data. However, for other important applications, the input data is corrupted by noise and an approximative approach would be more adequate. In such a setting, the traditional strategy is to penalize the solution's deviation from the input data as well as the magnitude of its $\mathcal{H}$-norm (otherwise there would be infinitely many interpolatory solutions). Such a regularization-based approach was taken by Walder et al. [13] building upon statistical learning theory. While our interpolatory approach can be variationally stated as

$$\min_{f \in \mathcal{H}, \lambda_i(f) = c_i} \|f\|_{\mathcal{H}},$$

theirs can be written as (here, slightly more general actually)

$$\min_{f \in \mathcal{H}} \left\{ \|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{N} \rho_i \cdot \left(\lambda_i(f) - c_i\right)^2 \right\} \quad (4)$$

where $\rho_i > 0$ are penalization parameters.

It turns out that (4) also admits a unique solution as a linear combination of the data functionals' representers. By restricting the optimization problem to this finite-dimensional space, it can be shown that the coefficients for the regularized solution can be computed by solving the system $\left(\mathbf{A} + \mathbf{D}^{-1}\right)\boldsymbol{\alpha} = \mathbf{c}$, where $\mathbf{D} = \operatorname{diag}(\rho_1, \ldots, \rho_N)$ and everything else as above.

The insights gained on this method from the previous developments we made indicate that not only the regularization approach is well posed but also the limiting interpolatory case, when each $\rho_i \to \infty$ and $\mathbf{D}^{-1} = 0$. Also, this analysis indicates that more flexibility is feasible by interpolating just a subset of the data, where the corresponding entry in $\mathbf{D}^{-1}$ must be zeroed, without sacrificing well-posedness of the combined interpolation/regularization system (since, $\mathbf{A}$ is guaranteed to be positive definite and $\mathbf{D}^{-1}$ positive semi-definite). This might be interesting for example when the normals come from an inexact process, as when they are just estimated from given surface points.

III. Computational aspects

After presenting a theoretical framework to deal with Hermite-Birkhoff interpolation problems and specializing those results to the first-order Hermite interpolation of our surface fitting model, we will discuss some aspects and issues of its computational implementation. More specifically, we deal with the assembly of the interpolation system defined by equations (1) and (2) and its numerical solution.

A. Assembling the interpolation system

Prior to the direct solution of the interpolation system, we need to assemble the corresponding matrix. For this end, notice that equations (1) and (2) can be rewritten as

$$\sum_{j=1}^{N}\begin{bmatrix}\psi\left(\mathbf{x}^i-\mathbf{x}^j\right) & -\nabla\psi\left(\mathbf{x}^i-\mathbf{x}^j\right)^T\\ \nabla\psi\left(\mathbf{x}^i-\mathbf{x}^j\right) & -\mathrm{H}\,\psi\left(\mathbf{x}^i-\mathbf{x}^j\right)\end{bmatrix}\begin{bmatrix}\alpha_j\\ \boldsymbol{\beta}^j\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{n}^i\end{bmatrix}$$

with each block in the sum as a $(n+1)\times(n+1)$-matrix. Therefore, the $(n+1)N\times(n+1)N$ linear system can be assembled one block at a time, corresponding to a pair (i,j) of samples. All we need to compute such a block is to know how to evaluate ψ, its gradient and its Hessian on a point.

At this point, we notice that the cost $O\left([(n+1)N]^2\right)$ of assembling such a system on globally supported RBFs is prohibitive for many applications. Indeed, our first implementation used dense matrix packages and this limited our experiments to about $8K$ points in $\mathbb{R}^2$ and $6K$ points in $\mathbb{R}^3$, on a commodity laptop with 2GB of RAM.

We have addressed this problem by using compactly-supported RBFs and employing sparse matrix packages. In our implementation, we used Wendland's $\phi_{3,1}$ function which defines a family $\psi_r(\mathbf{x}):=\psi\left(\frac{\mathbf{x}}{r}\right)$ indexed by a radius $r>0$, where each ψ_r is a C^2 positive definite function, for $n\leq 3$, $\psi(\mathbf{x}):=\phi_{3,1}(\|\mathbf{x}\|)$ and $\phi_{3,1}(t)=(1-t)^4(4t+1)$, for $t\in[0,1]$, and $\phi_{3,1}(t)=0$, otherwise [12]. That's all one needs to compute formulae for the gradient and Hessian of the ψ_r and, consequently, assemble the interpolation matrix.

Nonetheless, even with compactly-supported basis and efficient numerical methods, the assembly procedure can became the bottleneck when a very large number of samples are used. Theses cases demand a data structure for accelerating the range queries and retrieving only those pairs for which the blocks were non-zero. However, this introduces another problem, the trade-off between system sparsity and the width of the neighborhood in which the surface is defined. This issue is well known in methods which use compactly-supported kernels (e.g. RBF and moving least squares). Although we do not elaborate on this problem in this work, in the last section we discuss how we intend to build upon previous methods developed for classical interpolation problems with RBFs [14], [15].

B. Solving the interpolation system

It is well known and reported elsewhere that RBF interpolation systems suffer from numerical conditioning problems when the sampling is large and too dense. Many methods for improving the condition numbers have been developed to circumvent this issue and allow faster convergence of iterative methods for the sparse systems [6], [14].

With this in mind, we chose a simpler implementation for our first prototypes and decided to employ direct methods tailored for sparse systems. Since our analysis ensures that the interpolation system is symmetric and positive definite, we employ a sparse Cholesky factorization as implemented in the PARDISO solver [16] and the efficient codes in the BLAS and LAPACK linear algebra packages as implemented by the ATLAS project [17] in our core mathematical routines for evaluating gradients and Hessians.

Using these numerical packages along with the C++ language, we were able to develop a code completely independent of dimension, including the range-query acceleration data structure, which we chose to be a kd-tree. This prototype allowed us to solve Hermite interpolation problems with up to $500K$ samples in $\mathbb{R}^3$ with a suitably chosen sparsity structure, limited by the memory requirements of the PARDISO solver.

IV. Results

We first examine how changes on radius in ψ_r influence the family of HRBF implicits (Fig. 3 and 5).

In Fig. 3-(a), as we consider radius smaller than the distance between two points, HRBF implicits consists of four disjoint line segments. It can be noticed in the following images (from 3-(b) to 3-(f)) that, the larger the radius, the smoother HRBF implicits turns out.

Fig. 5 depicts the equivalent three-dimensional case of previous experiment, where from a set of six equally spaced points on the unity sphere, our method produces a family of HRBF implicits. In that case, the sequence of interpolants ranges from six disjoint discs (when radius is smaller than the inter-point distance), through a cube, to a sphere.

Since normals are also interpolated, in Fig. 4 we show examples of how the normals affect results. In Fig. 4-(a) we observe a zero-contour which does not interpolates any point due to the normal orientations. Such a zero-contour arises due to the Intermediate Value Theorem. That issue is avoided when normals are accordingly oriented (Fig. 4-(b)). Notice that oscillations occur in the corner of the mouth in Fig. 4-(c). In that case, since we set arbitrary normals in the corners, which are non-differentiable regions, it is expected such oscillations as HRBF implicits need to satisfy all constraints of normals and points.

In Fig. 6 we present the capability of HRBF implicits in dealing with close sheets. For comparisons, in Fig. 7 and 8, we present results from RAMLS surfaces [2] and FastRBF implicits [4], respectively. This experiment considers points distributed along two orthogonally interlaced tori, with inner radius 1 and outer radius 3, distanced 0.25 from each other. In Fig. 6-(a)-left we plot points as red spheres (we omit the arrows representing normal vectors) and its HRBF implicits, whereas in (a)-right we plot only HRBF implicits. Comparing to RAMLS surfaces in Fig. 7-(a), one can observe that interpolation does not occur, as was expected,

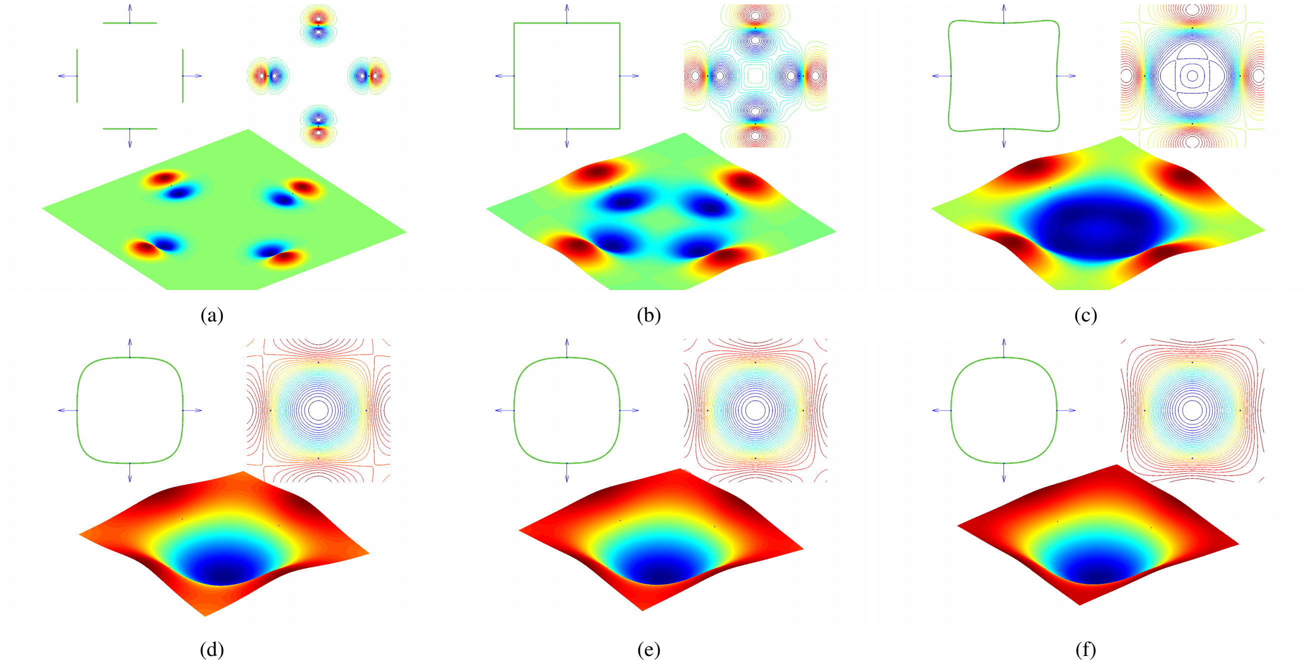

(a) (b) (c) (d) (e) (f)

Figure 3. Varying the radius in ψ_r (HRBF implicits in green): subfigures are composed of zero-level and Hermite samples (top-left), different isocontours (top-right) and HRBF implicits' graph (bottom). Data consists of four points equally spaced on the unity circle and their normals. Increasing $r > 0$, the family of HRBF implicits ranges from disjoint line segments (when radius is smaller than inter-point distance), through a square, to (almost) a circle.

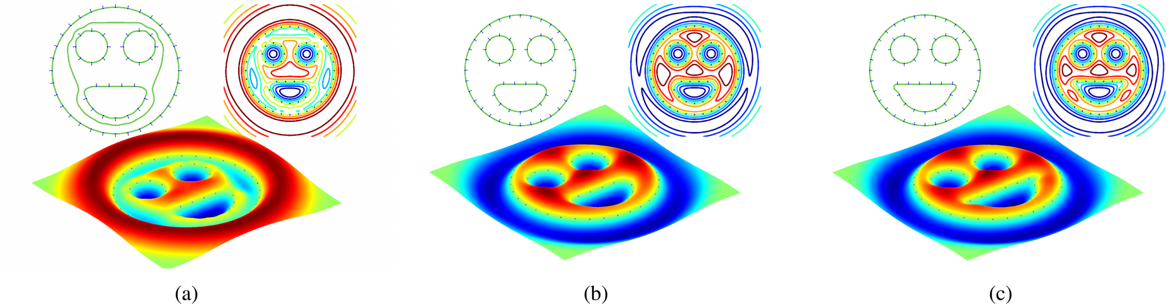

(a) (b) (c)

Figure 4. Examples of how normals can affect results: (a) inconsistent orientation causes an additional zero-level. This can be avoided by flipping the outer normals (b). In (c), arbitrarily chosen normals at non-differentiable regions (corner of the mouth) can cause oscillatory results.

since RAMLS surfaces employs an approximation-based approach. In addition, the region where the two tori become closer, the RAMLS surface comes to be thinner than the same region in the HRBF implicits method.

The reader could figure that a smaller radius in the weight function for the RAMLS method could lead to a more precise approximation. However, the use of smaller radius makes the method to fail, since the minimum number of points can't be attained (see Gois et al. [2] for further details on selecting radii).

We ran FastRBF creating offset points at $\mathbf{x}^i + \epsilon\mathbf{n}^i$, where $\epsilon = 0.1$ (left) and $\epsilon = 0.01$ (right) for each $\mathbf{x}^i \in \mathcal{P}$ with normal $\mathbf{n}^i$ (Fig. 8-(a)). It is observed that FastRBF implicits tend to be more oscillatory. We believe this occurs due to issues of numerical instabilities and farther offsets. Assuming $\epsilon \geq 1$ we experienced results that do not resemble at all the two tori.

In Fig. 6-(b), 7-(b) and 8-(b) we repeat the previous test, but considering a denser sampling. In that case, we observe that RAMLS surfaces behaves quite similarly to HRBF implicits. Despite less perceptible than in the previous test, FastRBF again presents an oscillatory behavior.

Fig. 1 and 9 show the ability of our method to interpolate irregularly spaced datasets. One may observe that, even in the presence of clusters and "rings" of points, the method robustly interpolates the data. This property, also present in previous RBF interpolants [4], [5], is not shared by RAMLS surfaces [2]. Further, in Fig. 1, one can again notice the capability of HRBF implicits of handling close sheets and irregularly spaced data simultaneously.

Finally, in Fig. 10, we can observe that HRBF implicits are able to enhance object details. We believe this prop-

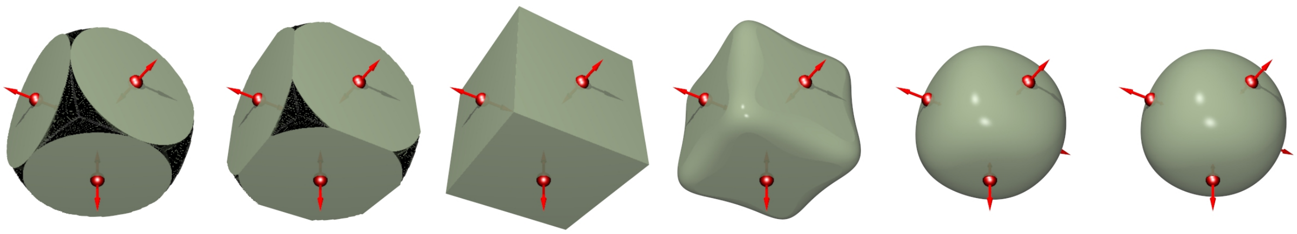


Figure 5. Varying the radius in ψ_r (HRBF implicits in gray): six points (antipodaly placed on the sphere) and their normals. Increasing $r > 0$, the family of HRBF implicits ranges from six disjoint discs (when the radius is smaller than inter-point distance), through a cube, to (almost) a sphere.

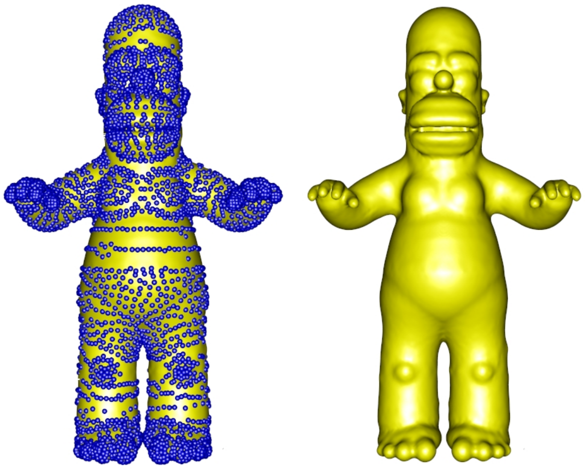


Figure 9. Ability of HRBF implicits to interpolate even irregularly spaced datasets even in the presence of clusters and "rings" of points. Such a property is also shared by previous RBF interpolants [4], [5] (Homer Simpson dataset with 5103 points/normals).

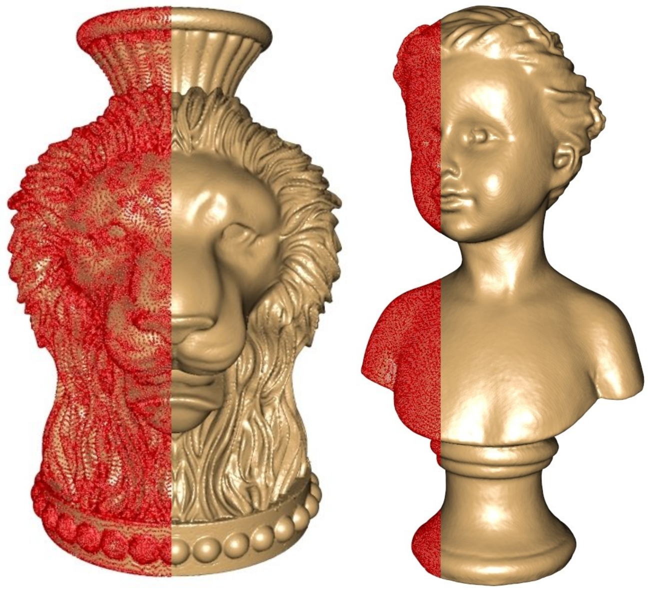


Figure 10. HRBF implicits enhance details in the interpolated surfaces (on the left half of each image we plot the sample points): Lion Vase dataset (left, 154906 points) and Buste Model (right, 255358 points).

erty is due to its Hermite-interpolatory character and the smoothness-degree of the chosen RBF, capturing continous variations of the normal field along the surfaces.

V. Final remarks

We present Hermite-RBF Implicits, a method to interpolate implicit curves and surfaces from Hermite data without recurring to introduction of offset points. In our way to tackle this problem, we present some basic results from a theory of Hermite-Birkhoff interpolation with radial basis functions. This provides a flexible theoretical and computational framework to pose many relevant problems and which we believe not to be broadly known by the graphics community.

With our proof-of-concept prototype, we were already able to solve interpolation problems consisting of $500K$ point/normal samples in $\mathbb{R}^3$ within a couple minutes. Its implementation is an almost direct translation of the mathematical results and is general enough to be independent of the space dimension, apart from the visualization module, yet including the main acceleration data structure.

As byproduct of our theoretical developments, we gained insight on a related regularization-based scheme, which was deduced from a statistical learning perspective, and this allowed us to enhance flexibility of both methods by ensuring well-posedness of a promising combined interpolation/regularization approach which we haven't explored yet.

A. *Ongoing work*

Currently, we are tackling two issues of our formulation by exploiting the full machinery of Hermite-Birkhoff interpolation we developed, in contrast to the presented which reduces to the special case of first-order Hermite interpolation. The first one deals with the constraint that the gradient field of the implicit function interpolates the given normal samples (all that might be ensured by the Implicit Function Theorem is that they are parallel). While the second one is concerned with points where the normals are not available or even not defined *a priori* as occurs on points along features. Solving both issues might improve flexibility of the reconstruction problems we would be able to deal with and expressiveness of the range of representable surfaces. The theoretical framework that we present provides enough flexibility to deduce the form of an RBF-interpolant for these full Hermite-Birkhoff interpolation problems as it did for our special Hermite case.

B. Future directions

We envision a number of avenues for further investigation. It would be interesting to try proving theoretical guarantees on the reconstructed surfaces by combining the many results from the theory of generalized interpolation with RBFs [6] and the techniques from surface reconstruction theory [18].

On the computational side, we are interested in adapting existing methods for the classical interpolation problem which try to aleviate the trade-off between sparsity and the support of the implicit function. More specifically, we are evaluating three main approaches: multiple-scale methods [19], preconditioning of Krylov iterations [14] and domain decomposition [15]. This would allow for a better control on the trade-off between running time and memory footprint, which are very related as we noticed that memory consumption of our direct solver for large datasets was so big that much of the running time was spent in swapping.

With respect to applications, we intend to work on dynamic implicit surfaces represented by Lagrangian particles augmented with normal data in a manner similar to [2], but with a resampling scheme suitably adapted from [20].

Since we had access to software from the authors of [4] and [2], we were able to compare our method to theirs. Our next step in this direction is to implement and compare Alexa and Adamson's recent interpolatory approach based on moving least squares surfaces [21].

Acknowledgment

The authors are grateful to Emilio Vital Brazil, Thiago Pereira and Julio Daniel M. Silva for participating in fruitful and insightful discussions and also for kindly providing computational resources. This work was partially supported by grants from CNPq, CAPES, FAPESP, FINEP and FAPERJ.

References

[1] G. Turk and J. F. O'Brien, "Modelling with implicit surfaces that interpolate," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 855–873, 2002.

[2] J. P. Gois, A. Nakano, L. G. Nonato, and G. C. Buscaglia, "Front tracking with moving-least-squares surfaces," *J. Comput. Phys.*, vol. 227, no. 22, pp. 9643–9669, November 2008.

[3] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and approximating implicit surfaces from polygon soup," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 896–904, 2004.

[4] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *Proceedings of SIGGRAPH'01*, 2001, pp. 67–76.

[5] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. Subramanian, "Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions," in *Proceedings of SMA'01*, 2001.

[6] H. Wendland, *Scattered data approximation*. Cambridge University Press, 2005.

[7] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, p. 23, 2007.

[8] J. P. Gois, V. Polizelli-Junior, T. Etiene, E. Tejada, A. Castelo, L. G. Nonato, and T. Ertl, "Twofold adaptive partition of unity implicits," *Vis. Comput.*, vol. 24, no. 12, pp. 1013–1023, 2008.

[9] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 544–552, 2005.

[10] G. Bachman and L. Narici, *Functional analysis*. Dover Publications Inc., 2000.

[11] G. E. Fasshauer, "Solving partial differential equations by collocation with radial basis functions," *Surface Fitting and Multiresolution Methods*, pp. 131–138, 1997.

[12] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Adv. Comput. Math.*, vol. 4, no. 4, pp. 389–396, 1995.

[13] C. Walder, B. Schölkopf, and O. Chapelle, "Implicit surface modelling with a globally regularised basis of compact support," *Computer Graphics Forum*, 2006.

[14] R. K. Beatson, J. B. Cherrie, and C. T. Mouat, "Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration," *Adv. Comput. Math.*, vol. 11, no. 2-3, pp. 253–270, 1999, radial basis functions and their applications.

[15] R. K. Beatson, W. A. Light, and S. Billings, "Fast solution of the radial basis function interpolation equations: domain decomposition methods," *SIAM J. Sci. Comput.*, vol. 22, no. 5, pp. 1717–1740 (electronic), 2000.

[16] O. Schenk and K. Gärtner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.

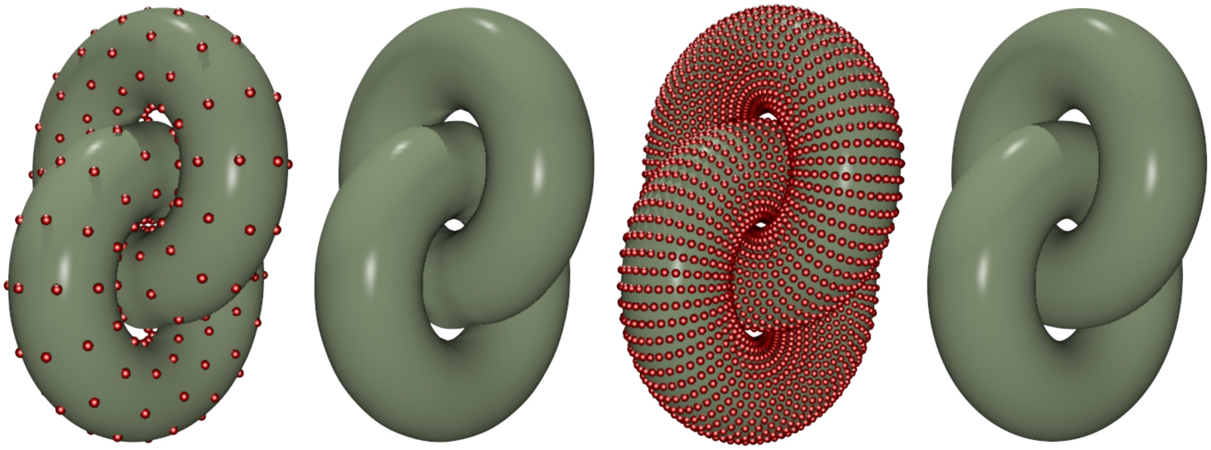
[17] "Automatically Tuned Linear Algebra Software (ATLAS)," http://math-atlas.sourceforge.net/, last access July 2009.

[18] T. K. Dey, *Curve and surface reconstruction: algorithms with mathematical analysis*. Cambridge University Press, 2007.

[19] M. S. Floater and A. Iske, "Multistep scattered data interpolation using compactly supported radial basis functions," *J. Comput. Appl. Math.*, vol. 73, no. 1-2, pp. 65–78, 1996.

[20] J. Schreiner, C. Scheidegger, and C. Silva, "High-quality extraction of isosurfaces from regular and irregular grids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1205–1212, 2006.

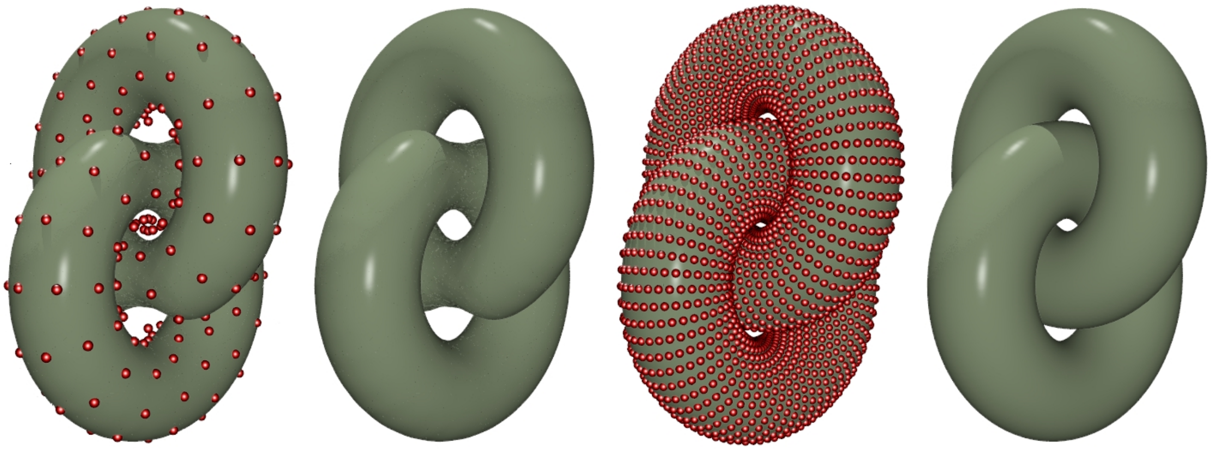
[21] M. Alexa and A. Adamson, "Interpolatory point set surfaces – convexity and Hermite data," *ACM Trans. Graph.*, vol. 28, no. 2, pp. 1–10, 2009.



(a) 256 points and Radius 3

(b) 4096 points and Radius 1.5

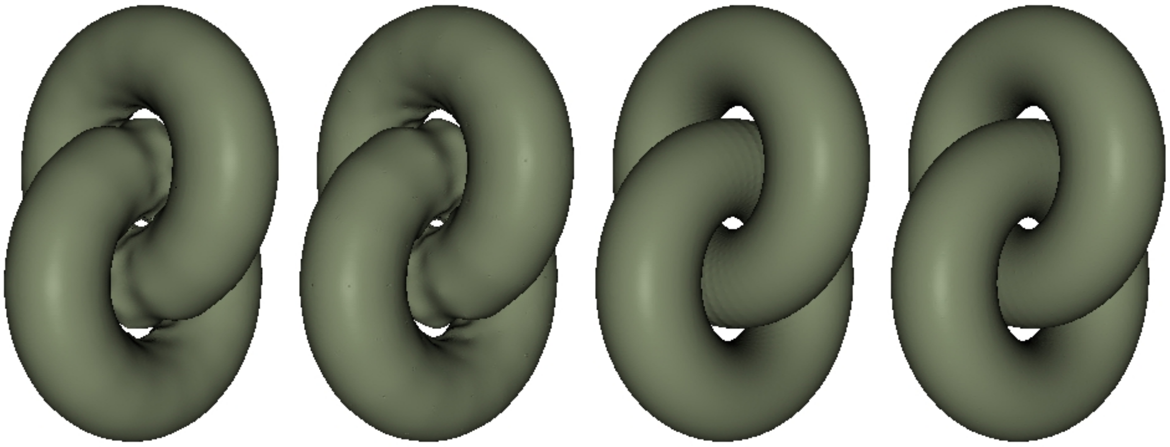
Figure 6. HRBF Implicits: Comparing with RAMLS surfaces (see Fig. 7) and FastRBF implicits (see Fig. 8), one can observe that our method produce more accurate results in the presence of close sheets, which is most noticeable in the presence of sparse samples.



(a) 256 points and weight function radius 1.5

(b) 4096 points and weight function radius 0.375

Figure 7. RAMLS surfaces (considering the weight function proposed in Gois et al. [2]): In subfigure (a), where the two tori are closer, we observe that RAMLS surfaces are far from the samples and shrink both tori. In subfigure (b), this issue is not noticeable and the result is similar to ours (Fig. 6).



(a) 256 points + 512 offsets

(b) 4096 points + 8192 offsets

Figure 8. FastRBF implicits [4] (offset points where created at $\mathbf{x}^i + \epsilon\mathbf{n}^i$, for $\epsilon = 0.1$ (left) and $\epsilon = 0.01$ (right) in both subfigures): Near close sheets one can observe an oscillatory behavior of FastRBF implicits for three of the four tests.