HAVE 2007 - IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa, Canada, 12-14 October 2007

Refining Hierarchical Radial Basis Function Networks

S. Ferrari[1], F. Bellocchio[1], N.A. Borghese[2], V. Piuri[1]

[1] Department of Information Technologies,
University of Milano
via Bramante, 65 – 26013 Crema, Italy
E–mail: {ferrari, bellocchio, piuri}@dti.unimi.it.

[2] Department of Computer Science
University of Milano,
via Comelico, 39-41 – 20135 Milano, Italy
E–mail: borghese@dsi.unimi.it.

Abstract *– The Hierarchical Radial Basis Function (HRBF) Network is a neural model that proved its ability in surface reconstruction problem. The algebraic error is used to drive the HRBF configuration procedure and for evaluating the reconstruction ability of the network. While for function approximation the algebraic distance is the appropriate error metric, for computer graphics applications, such as model reconstruction by 3D scanning, the geometric distance is a more suitable error metric. In this paper, we propose a modified HRBF model which makes use of the geometric error as a measure of the reconstruction accuracy.*

Keywords *– Radial Basis Function Networks, HRBF, geometric distance.*

I. INTRODUCTION

A 3D scanning device measures the geometric properties of a real object and produce a 3D model of the object itself. This digitizing procedure is performed in two step: the measuring of the position of some points of the object (sampling) and the processing of the sampled data for obtaining a representation of the surface of the object. In computer graphics, the most used paradigm for representing the geometry of the model is the mesh of triangles. Subsequent processing may be operated on the mesh to obtain a more sophisticated representation in order to speed up processing such as editing or visualization or to diminish the size of the representation.

Multiresolution representations [1][2][3][4] are widely used for this scope: as they represent the surface at different level of detail, they usually allow to operate at the resolution required by the specific application or to perform locally by processing only a small fraction of parameters that represent the surface.

The Hierarchical Radial Basis Function (HRBF) model is a neural network paradigm that can be used for surface reconstruction from a cloud of points affected by noise [5][6]. It uses a linear combination of Gaussians to represent the surface as an explicit function defined in the $\mathbb{R}^2$ domain. As its configuration procedure is not iterative and uses only local operation, it is suitable to fast processing. The multi-scale adaptive scheme adds Gaussians only in those regions that features the details of the surface, allowing a compact representation of the surface.

In its original formulation, the HRBF uses the algebraic error between the dataset points and the actual approximation for evaluating the presence of residual details and the need of adding new Gaussians to the network. As for practical use in computer graphics the accuracy of a model is better measured by means of the geometric error, the use of this error measure in the HRBF configuration for surface reconstruction merits to be investigated.

In this paper, we will formulate a modified HRBF model which makes use of the geometric error. In section II, the original formulation of the HRBF model is described, while in section III a variant is proposed. A comparison with the original HRBF model is carried out for a problem of surface reconstruction in section IV. In section V the results of the comparison are discussed, while in section VI conclusions are drawn.

II. HRBF NETWORKS

Let us assume that the manifold to be approximated can be described as a $\mathbb{R}^D \rightarrow \mathbb{R}$ function. This allows to consider the input dataset as a height field: $\{(P_i, z_i) \,|\, z_i = S(P_i),\ P_i \in \mathbb{R}^D,\ 1 \leq i \leq N\}$, and the manifold, output by the network, will assume the explicit analytical shape: $z = S(P)$. The output of a HRBF network is obtained by adding the output of a pool of Radial Basis Functions (RBF) networks, organized as a stack of hierarchical layers, each of which is characterized by a decreasing scale:

$$S(P) = \sum_{l=1}^{L} a_l(P; \sigma_l) \tag{1}$$

where σ_l determines the scale of the l-th layer, with $\sigma_l > \sigma_l + 1$. If we suppose that the units are equally spaced on a grid support and a normalized spherical Gaussian function, $G(\cdot; \sigma) =$

$\frac{1}{\sqrt{\pi\sigma^2}^D} exp\left(-\frac{||\cdot||^2}{\sigma^2}\right)$, is taken as basis function, the output of each layer can be written as a linear low-pass filter:

$$a_l(P;\sigma_l) = \sum_{k=1}^{M_l} w_{l,k} G(||P - P_{l,k}||;\sigma_l) \quad (2)$$

where M_l is the number of Gaussian units of the l-th layer. The $G(\cdot)$ are equally spaced on a D-dimensional grid, which covers the input domain of the data points: that is the $\{P_{l,k}\}$s are positioned in the grid crossings of the l-th layer. The side of the grid is a function of the scale of that layer: the smaller the scale, the shorter is the side length, the denser are the Gaussians and the finer are the details which can be reconstructed.

The actual shape of the surface in (1) depends on a set of parameters: the *structural parameters*, which are the number, $M = \sum_l M_l$, the scale ensemble, $\{\sigma_l\}$, and the position, $\{P_{l,k}\}$; and the weights associated to each Gaussian: $\{w_{l,k}\}$. Each RBF grid, l, realizes a reconstruction of the surface up to a certain scale, determined by σ_l (low-pass filtered reconstruction). Considerations grounded on the signal processing theory allow, given a certain scale, σ_l, to set the grid side, ΔP_l, as $\sigma_l = 1.465\,\Delta P_l$ and to determine consequently M and the $\{P_{l,k}\}$ [7]. From these observations, the weights $\{w_{l,k}\}$ are set equal to the manifold height in the grid crossings: $w_{l,k} = S(P_{l,k}) \cdot \Delta P_l^D$. As the data set usually does not include the $\{S(P_{l,k})\}$, these values should be estimated. We explicitly observe that, even if the $S(P_{l,k})$ were included, they would be corrupted by noise and an estimate would be the right solution. The data points that lie in an appropriate neighborhood of $P_{l,k}$ can be used to estimate $S(P_{l,k})$ as a weighted average of such subset of data points, $\tilde{S}(P_{l,k})$. This neighborhood, called *receptive field*, $A(P_{l,k})$, can be chosen as a spherical region centered in $P_{l,k}$ with the radius proportional to the grid side, ΔP_l. A possible weighting function is:

$$\tilde{S}(P_{l,k}) = \frac{\sum_{P_m \in A(P_{l,k})} S(P_m)\, e^{-\frac{||P_{l,k} - P_m||^2}{\sigma_l^2}}}{\sum_{P_m \in A(P_{l,k})} e^{-\frac{||P_{l,k} - P_m||^2}{\sigma_l^2}}} \quad (3)$$

which is strongly related to the Nadaraya-Watson estimator and maximizes the conditional probability density when the noise is normally distributed, zero mean [8] [9].

Although a single layer with Gaussians of very small scale could reconstruct the finest details, this would produce an unnecessary dense packing of units in all those regions which feature large scale details. Moreover, there might even be not enough points inside $A(P_{l,k})$ to get a reliable estimate of $\tilde{S}(P_{l,k})$ in (3). A better solution is to adaptively allocate the Gaussian units, with an adequate scale in the different regions of the range data domain. This can be achieved by adding and configuring one layer at time, proceeding from the layer featuring the largest scale to the layer featuring the smallest one. For sake of simplicity in the configuration stage, each new layer will feature half the scale of the previous one. However, arbitrary scales could be used for the different layers.

All the layers after the first one will be trained to approximate the residual, that is the difference between the original data and the actual output of the network output by the already configured layers. Hence, the residual, r_l, is computed as:

$$r_l(P_m) = r_{l-1}(P_m) - a_l(P_m) \quad (4)$$

and it is used for estimating the parameters of the l-th layer. $r_0(P_m) = z_m$ is also assumed.

The Gaussians of a new layer are inserted only where a poor approximation is obtained from the previous layers. This is evaluated, for each Gaussian, $P_{l,k}$, through an integral measure of the residuals inside the receptive field of that Gaussian, $A(P_{l,k})$. This measure, which represents the *local residual error*, $R(P_{l,k})$, is computed as the L_1 norm of the local residual as:

$$R(P_{l,k}) = \frac{\sum_{P_m \in A(P_{l,k})} |r_{l-1}(P_m)|}{|A(P_{l,k})|}. \quad (5)$$

As the Gaussian function has an infinite support, the computation of the output of each layer, a_l may be computational very expensive. However, as the Gaussian decreases very fast to zero with the distance from its center, computational time can be saved by allowing each Gaussian to contribute to the computation of the residuals only for those points that belong to an appropriate neighborhood of the Gaussian center. This neighborhood has been called *Influence Region*.

When $R(P_{l,k})$ is over a given threshold, ϵ, the Gaussian is inserted in the corresponding grid crossing of the current layer under construction. As a result, Gaussians at a smaller scales are inserted only in those regions where there are still some missing details, forming a sparse approximation of the data. The introduction of new layers ends when the residual error is under threshold over the entire domain (uniform approximation).

This approach has been compared with classical multiresolution analysis through wavelet basis, and it has proved superior when approximation of noisy data is required [5].

III. REFINED HRBF

Although the HRBF model can potentially reconstruct every manifold that can be expressed as a function, it requires a low scale layer for reconstructing those regions that feature a near vertical surface. In fact, from the functional representation point of view, in those regions the function changes rapidly, and, hence, they are characterized by a high frequency content. As each layer plays as a low-pass filter with the cut-off frequency inversely proportional to the scale, σ, of the layer, a low scale layer is required to cope with a near vertical region. It should be noted that it happens even if the region is

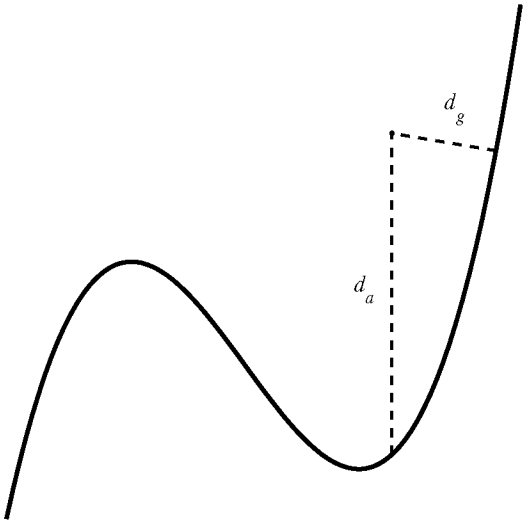


Fig. 1. Algebraic distance, d_a, vs. geometric distance, d_g.

flat, e.g., it does not feature low scale details. However, this error may be more apparent than real, as the surface can be closer to the data points than the reported error. This error may depend on the orientation of the data, as, if the data were properly rotated, probably the error achieved by the HRBF model would be much smaller. The same consideration applies if the domain of the HRBF, instead of being the $\mathbb{R}^2$ plane, would be a 2D manifold suitably bended.

A. Algebraic error vs. geometric error

Two models of error can be used to express the distance between a function, $f : A \rightarrow B$ and a point, $p = (p_a, p_b) \in A \times B$: the algebraic and the geometric error (Fig. 1). The algebraic error, d, is used to express the error when the function is given in explicit form, i.e., when a dependent variable can be expressed in term of independent variable: $d(p) = ||p_b - f(p_a)||$. The geometric error, g, is used when the function is given in implicit form (e.g., $f(\cdot) = 0$), and it is defined as the distance between a point and the closest point of a surface: $g = \min_{x:f(x)=0} ||p - f(x)||$.

Hence, the algebraic error is measured with respect to the the point projected onto the domain of the function, while the geometric error is measured with respect to the projection of the point onto the surface of the function.

The use of algebraic error is often used even in application where the geometric error would be more appropriate because its minimization can be operated by linear algorithm. The minimization of a geometric error is, in general, a non-linear problem, which requires iterative solutions (and a good starting point). However, it worth to notice that in some fields (such as computer vision) the geometric error has a physical interpretation and it is invariant under rototraslation.

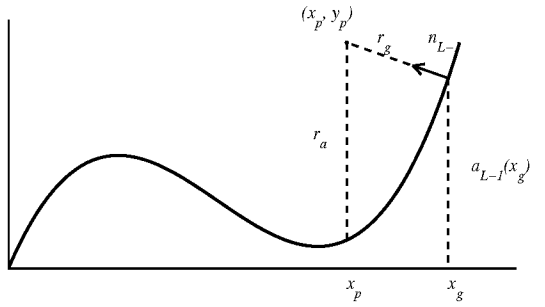


Fig. 2. Computing the residual for the original and for the refined HRBF model: for the same point $(x_p,\ y_p)$, the original HRBF will consider the point $(x_p,\ r_a)$ as the residual, while the refined HRBF will consider $(x_g\ r_g)$.

B. Geometric error in HRBF

The HRBF model considers the algebraic error as a measure of fitting. This simplifies the formulation of the problem and the implementation of the algorithm. However, for some application such as 3D scanning, a more realistic model of error could be the geometric error.

In order to force the HRBF to fit the dataset along the normal of the HRBF surface, we propose to modify the computation of the last layer by considering the projection of the residuals onto the last configured layer surface, a_{n-1}. As illustred in Fig. 2, the residual points for the refined layer will be constituted by points having the height equal to the geometric distance from the surface (signed with respect to the normal to the surface) and having the same coordinate of the projected points onto the surface of the previous layer. The last layer of such an HRBF will have the surface of the previous layer as domain, instead of the dataset domain, giving the following reconstruction:

$$S(P) = \sum_{l=1}^{L-1} \vec{z}\, a_l(P; \sigma_l) + \vec{n}_{L-1}\, a_L(P) \tag{6}$$

where $\vec{z}$ is the vertical unit vector (e.g., [0 0 1] for $\mathbb{R}^2 \rightarrow \mathbb{R}$ functions), and $\vec{n}_k$ is the normal of the k-th layer. It should be noted that this modification reframes the HRBF to be a parametrized function, where the input domain act as the parametric domain, and the HRBF output is a point in the dataset space (e.g., $\mathbb{R}^3$, for $\mathbb{R}^2 \rightarrow \mathbb{R}$ HRBF).

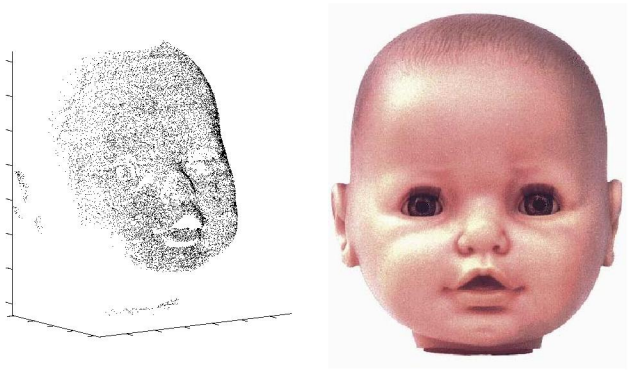
The projection of a point on a generic surface cannot be obtained in closed form. However, an iterative procedure can be used to compute an approximation of the projected point and, hence the distance between the point and the surface.

We observe that the geometric distance cannot be greater than the algebraic distance. Hence, for the point $(P,\ z)$, its projection point, $(Q,\ S(Q))$, on the surface $S(\cdot)$ have to be such that Q is in the spherical neighborhood of P having a radius $||z - S(P)||$: $Q \in I(P,\ ||z - S(P)||)$. For a generic surface, an extensive search should be performed, but the smoothness of the HRBF surface may be exploited to compute efficiently a

Given a point $(P,\ z)$ and a HRBF with l layer, having output $S(\cdot)$, find the projection point of P on $S(\cdot)$ with an accuracy α:

1. $c := P$; $r := ||z - S(P)||$; $\delta := \min(0.5\ \sigma_l,\ 0.5\ r)$;
2. while $\delta > \alpha$, do:
   (a) $T :=$ sample $(I(c,\ r),\ \delta)$;
   (b) $Q := \arg\min_{t \in T} ||(P,\ z) - (t,\ S(t))||$;
   (c) $c := Q$; $r := \min(||z - S(Q)||,\ 0.6\ r)$; $\delta := \min(0.5\ \sigma_l,\ 0.5\ r)$;
3. return $(c,\ S(c))$;

Fig. 3. Pseudocode of the procedure for approximating the projection of a point on the HRBF surface.



(a) (b)

Fig. 4. The data set used to test the refined training procedure (a). The range data has been acquired by the 3D scanner from the dolly reported in (b).

reliable projection point: the surface can be iteratively sampled in an appropriate neighborhood of the closest sampled point. As the scale parameter, σ_l, gives a clue about speed at which the surface can change, it can be used as starting sampling step for the searching procedure. At each iteration, the sampled point closest to the given point, P, is chosen as the center of the searching region, and the radius of the searching region and the sampling step is decreased. The procedure can be iterated until the sampling step does not reach a predefined accuracy, α. The searching algorithm can be formalized as the pseudocode in Fig. 3.

IV. RESULTS

We applied both the original and the refined HRBF model to a problem of surface reconstruction from a cloud of sampled points.

The 3D input dataset, reported in Fig. 4b, is composed by 16,000 range data acquired by sampling the surface of the object in Fig. 4a using the 3D scanner described in [10].

In order to test the effect of the refining, we configure three pairs of networks, where each pair is composed by a network configured with the original algorithm and a refined HRBF, and each pair has an increasing number of layers. The networks of the same pair differs only by the last layer as the last layer of the refined networks is configured using the geometric error, while the last layer of the original network is trained with the algebraic error. Both these layers features a scale equal to the scale of the previous layer. The rational for this choice is that the refinement layer is added for coping with the geometric error and not for providing lower scale details. For the original HRBF, the re-use of the same scale is equivalent to consider a fraction of the high frequency information content that was filtered out by the previous layer (and that would be recovered by the next lower scale layer).

The parameters of the networks and the performance indexes are reported in the Tables I–III.

Similar figure of merits are obtained if the last layer feature

TABLE I. Performance Indexes and Parameters of Each Layer of the final HRBF Network

	l	σ_l	#Gauss. Gauss.	#eff. [mm]	RMSE [mm]	ϵ_{std}
common	1	16	210	185	29.1	5.28
	2	8	754	582	5.96	2.44
original	3	8	754	499	3.00	1.73
refined	3	8	754	507	2.11	1.45

TABLE II. Performance Indexes and Parameters of Each Layer of the HRBF Networks

	l	σ_l	#Gauss. Gauss.	#eff. [mm]	RMSE [mm]	ϵ_{std}
common	1	16	210	185	29.1	5.28
	2	8	754	582	5.96	2.44
	3	4	2850	1561	1.09	1.04
original	4	4	2850	705	0.669	0.809
refined	4	4	2907	679	0.515	0.709

TABLE III. Performance Indexes and Parameters of Each Layer of the HRBF Networks

	l	σ_l	#Gauss. Gauss.	#eff. [mm]	RMSE [mm]	ϵ_{std}
common	1	16	210	185	29.1	5.28
	2	8	754	582	5.96	2.44
	3	4	2850	1561	1.09	1.04
	4	2	11187	2072	0.332	0.565
original	5	2	11187	578	0.285	0.519
refined	5	2	11187	511	0.249	0.482

TABLE IV. Performance Indexes and Parameters of Each Layer of the HRBF Network

	l	σ_l	#Gauss. Gauss.	#eff. [mm]	RMSE [mm]	ϵ_{std}
common	1	16	210	185	29.1	5.28
	2	8	754	582	5.96	2.44
original	3	4	2850	1561	1.09	1.04
refined	3	4	2850	1556	0.760	0.870

half the scale of the second-last layer. For example, the results for the 3 layer HRBFs are reported in Table IV.

The refined HRBF algorithm requires the computation of the projection of the points onto the HRBF surface where the original HRBF requires only the computation of the HRBF surface. The computation overhead increases with the number of the Gaussians, and is inversely proportional to the scale of the last layer, σ_l, and the required accuracy, α.

For the networks reported in this sections, we experimented the need of 141 HRBF sampling for each data point for $\alpha = 0.1$ ($\alpha = 0.05\,\sigma_4$), and the need of 237 HRBF sampling for each data point for $\alpha = 0.01$ ($\alpha = 0.005\,\sigma_4$).

V. DISCUSSION

From the data reported in Tabs. I–III, it can be noted that both the RMSE and the standard deviation of the error distribution achieved by the refining layer are smaller than the corresponding original layer: in particular, the RMSE is 12.6–29.7% smaller.

The computational cost of the refined HRBF configuration algorithm is dominated by the projection procedure and it is two order bigger than the original HRBF one. However it should be pointed out that the procedure can be optimized by allowing to store all (or a selection of) the sampled points. In this case, the search of the projected point of a data paint may take advantages of the previously sampled points. For close data points, it can be a considerable saving.

In the field of computer graphics, hierarchical surface representation that make use of normal displacement to increase the details of the representation are called normal meshes [3][4].

Normal meshes [3] is a multiresolution representation constituted by a base model (which represent the coarsest representation of the surface) and a set of normal displacement. The representation at higher scale is obtained by subdividing the triangles which constitute the base model and displacing the new vertices in the direction of the normal of the surface. The procedure can be iterated to add details to the surface. Hence, the normal representation allows to save two coefficients for each vertices of the higher scales meshes (one displacement coefficient instead of three coordinates).

In [3] the normal representation is computed from a generic mesh in several steps: the original mesh is simplified to obtain the base model and parametrized. The parametrization is used to find the subdivision points from which the intersection of the surface normals and the original mesh can be computed. Variational normal meshes [4] differs from normal meshes algorithm in the way the normal representation is computed: normal meshes realizes interpolation of the original mesh, while the variational normal meshes finds approximations.

Refined HRBF and normal meshes are both multiresolution coarse-to-fine representation (HRBF uses Gaussians, while normal meshes uses Hat function as basis functions). However, the normal meshes construction algorithm is specialized for converting existing mesh, while the HRBF configuration algorithm operates on a noisy cloud of points.

VI. CONCLUSIONS

Considering the geometric error as the measure of the HRBF network reconstruction ability and incorporating it in the configuration procedure makes the HRBF model more complex, but more powerful. The results showed that the approximation ability increases both in terms of accuracy and uniformity.

The properties of the HRBF allow to formulate some possible topics for future works.

The HRBF surface computation allows to compute efficiently also the derivative of the surface [6]. This information may be exploited in the projection algorithm (Fig. 3) in, at least, two ways: to speed up the minimization, approximating locally the surface, and to better control the accuracy of the approximation of the projection, by comparing the normal at the surface in c and the vector $P - c$.

A full normal HRBF, i.e. a HRBF where each layer used the previous layer as domain, can be investigated. In that case, the need of parametrizing each layer of the surface may occur in order to avoid the stretching of the next layer. The computation of the derivative, and hence the curvature of the surface, may facilitate the parametrization. It worth noting that an equispaced arrangement of points in a parametrized domain in general does not result in an equispaced set of points in the original $\mathbb{R}^3$ domain. Hence, the use of the previous (parametrized) layer as support for the next layer causes the use of non-radial and non-equispaced basis functions, with unpredictable results in the approximation performance of the HRBF.

REFERENCES

[1] H. Hoppe, "Progressive meshes," *Proceedings of Siggraph 96*, pp. 99–108, 1996.

[2] M. Lounsbery, T. DeRose, and J. Warren, "Progressive meshes," *ACM Trans on Graphics*, vol. 16, no. 1, pp. 34–73, 1997.

[3] I. Guskov, K. Vidimče, W. Sweldens, and P. Schröeder, "Normal meshes," *Proceedings of Siggraph 2000*, pp. 95–102, 2000.

[4] I. Friedel, P. Schröeder, and A. Khodakovsky, "Variational normal meshes," *ACM Trans. on Graphics*, vol. 23, no. 4, pp. 1061–1073, 2004.

[5] S. Ferrari, M. Maggioni, and N. A. Borghese, "Multi-scale approximation with hierarchical radial basis functions networks," *IEEE Trans. on Neural Networks*, vol. 15, no. 1, pp. 178–188, Jan. 2004.

[6] S. Ferrari, I. Frosio, V. Piuri, and N. A. Borghese, "Automatic multiscale meshing through HRBF networks," *IEEE Trans. on Instr. and Meas.*, vol. 54, no. 4, pp. 1463–1470, Aug. 2005.

[7] N. A. Borghese and S. Ferrari, "Hierarchical RBF networks and local parameter estimate," *Neurocomputing*, vol. 19, no. 1–3, pp. 259–283, 1998.

[8] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.

[10] N. A. Borghese and S. Ferrari, "A portable modular system for automatic acquisition of 3-D objects," *IEEE Trans. on Instr. and Meas.*, vol. 49, no. 5, pp. 1128–1136, October 2000.