IMTC 2004 - Instrumentation and Measurement
Technology Conference
Como, Italy, 18-20 May 2004

The Accuracy of the HRBF Networks

S. Ferrari[1], I. Frosio[2,3], V. Piuri[1], N.A. Borghese[2]

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

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

[3]Department of Bioengineering, Politecnico of Milano.

Abstract *– A procedure for the construction of 3D surfaces from range data in real-time is here described and discussed. The main goal is to observe the accuracy of the reconstruction through the phases of the 3D model reconstruction. The process is based on the connectionist model, named Hierarchical Radial Basis Functions Network (HRBF), which has been proved effective in the reconstruction of smooth surfaces from sparse noisy points. The network goal is to achieve a uniform reconstruction error, equal to measurement error, by stacking non-complete grids of Gaussians at decreasing scales. The HRBF properties allow reconstructing meshes from range data in real-time.*

Keywords *– 3D scanner, Multi-scale surface, Noise filtering, Real-time meshing.*

I. INTRODUCTION

The three-dimensional scanning of real objects is becoming a common technique used to obtain a 3D model. The procedure is composed of two-step: the sampling of a set of range data points on the surface to be scanned and the generation of a 3D colored mesh. Although sampling can be indeed fast, the generation of a 3D mesh requires a considerable amount of time, as the processing chain, usually require combining a set of partial 3D scans to produce a complex 3D model. This is achieved by looping through three steps: view planning, aligning the scans (registration) and reconstructing a unique surface (merging) [1]. To produce results of good quality, human intervention at different stages is required both to provide initialization data and to evaluate the effectiveness of the operation. This registration / view planning cycle is very time consuming: automated registration procedure and a fast visual feedback, although not accurate, may speed up the object acquisition.

The approach analyzed here uses a linear combination of Gaussians to represent the surface. The surface is supported by a 2D grid, and the parameters are computed through algebraic operations carried out locally on the data, and therefore it is suitable to fast processing. Moreover, to add finer details of the surface, which are often circumscribed in few regions, a multi-scale adaptive scheme has been developed. This schema automatically identifies these regions and inserts clusters of Gaussians at smaller scales there. The model has been termed Hierarchical Radial Basis Function Networks (HRBF, [2]; [3]). The spatial-frequency locality property of the HRBF paradigm can be effectively exploited both in the configuration procedure and in the registration/fusion cycle. Besides, the differential properties of the Gaussian allow using an error prediction procedure to state an adaptive resampling of the surface, useful for the fast visual feedback purpose. Hence, the HRBF paradigm can be used through the whole surface reconstruction process.

In this paper we will describe the surface reconstruction process chain based on HRBF with the main goal of observing the reconstruction accuracy through the different phases of the process.

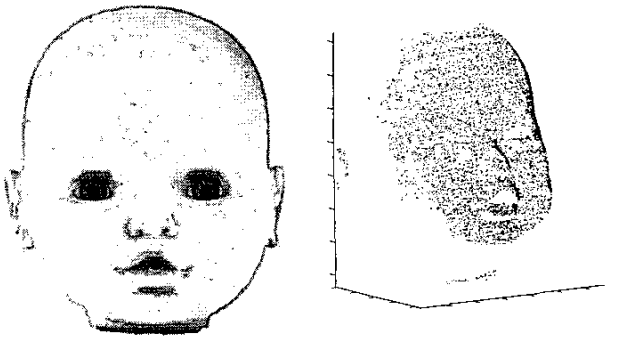
II. THE HRBF MODEL

Let us suppose that the set of range data, which can be expressed as a 2½D data set, that is as a height field: $\{z = S(x, y)\}$. In this case, the surface will assume the explicit analytical shape: $z = S(P)$. The output of a HRBF network is obtained by adding the output of a stack of hierarchical layers, $a_l(P)$, at decreasing scale:

$$s(P) = \sum_{l=0}^{M} a_l(P; \sigma_l) \qquad (1)$$

where σ where σ_l determines the scale of the l-th layer, and $\sigma_l > \sigma_l + 1$ holds. When the Gaussian, $G(\cdot)$, is taken as basis

0-7803-8248-X/04/$17.00 ©2004 IEEE



(a) (b)

Fig. 1. The model in panel (a) has been scanned through the Autoscan digitizer [4] obtaining a daset of 16.851 range data points reported in panel (b).

function, the output of each layer can be written as:

$$a_l(P;\sigma_l) = \sum_{k=0}^{N} w_{l,k} G(P - P_{l,k};\sigma_l) \qquad (2)$$

The $G(\cdot)$ are equally spaced on a 2D grid, which cover the input domain of the range data: that is the $\{P_{l,k}\}$s are positioned in the grid crossings. 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 shape of the surface in (2) depends on a set of parameters: the *structural parameters*, which are the number, N, the scale, σ_l, and the position, $\{P_{j,k}\}$ of the Gaussians, and the weights $\{w_{l,k}\}$. Each grid, l, realizes a low-pass filter, which is able to reconstruct the surface up toa certain scale, determined by σ_l. Some considerations, grounded on the signal processing theory, allow, given a certain scale, σ_l, to set the grid side, ΔP_l, and consequently N and the $\{P_{l,k}\}$ [5].

The weights $\{w_{l,k}\}$ could be chosen equivalent to the surface height in the grid crossings: $w_{l,k} = S(P_{l,k})$ according to the scheme of digital filters. As range data are usually not equally spaced, $S(P_{l,k})$ is not available and should be estimated. To the purpose a weighted average of the range points, $\{P_m\}$ (where the weight is a value decreasing with the distance of P_m from $P_{l,k}$) could be adopted. The estimate can be carried out locally in space, by using only the range points lying in an appropriate neighborhood of $P_{l,k}$. This neighborhood, called receptive field, $A(P_{l,k})$, is choose as the square region centered in $P_{l,k}$, of side equal to $2\Delta P_l$. A possible estimating function is reported in (3):

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

Although the use of a single layer of very small scale Gaussian can be sufficient to reconstruct the finest details, this would produce an unnecessary dense packing of units in all those regions which feature a low scale. In these regions there might even be not enough points to get the estimate 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 as explained in the following.

The first grid outputs a rough estimate of the surface, $a_1(P)$ at a large scale as:

$$a_1(P:\sigma_l) = \sum_{k=0}^{N} w_{1,k} G(P - P_{1,k};\sigma_1) \qquad (4)$$

For each of the range data points, a residual is computed as the difference between the measured value of the surface and the reconstructed one:

$$r(P_m) = S(P_m) - a_1(P_m). \qquad (5)$$

The details will be added in the higher layers, captured by pools of Gaussians at smaller scales (cf. Fig. 4). To this scope a second grid, featuring a smaller scale than the first one is created. Somehow arbitrarily we choose $\sigma_{l+1} = \sigma_l/2$, as usually chosen in Wavelet decomposition. The Gaussians are inserted only where a poor approximation is obtained. 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, $R(Pl,k)$, is computed as the L_1 norm of the local residuals as:

$$R(P_{l,k}) = a_1(P_{l,k}) - \frac{\sum_{minRF(P_{l,k})} |r(P_m)|}{M_{RF(P_{l,k})}}. \qquad (6)$$

When $R(P_{l,k})$ is over threshold (larger than the measured noise), the Gaussian is inserted in the corresponding grid crossing of the second layer.

Grids are created one after the other until the Residual Error goes under threshold, usually defined as standard deviation of the measurement error, over the entire input domain. As a result, Gaussians at a smaller scale are inserted only in those regions where there are still some missing details, forming a sparse approximation (Fig. 2). Moreover, the number of layers is not given a-priori, but it is the result of the configuration procedure: the introduction of a new layer stops when the residual error is under threshold over the entire domain (uniform approximation).

III. FAST CONFIGURATION OF HRBF SURFACES

To obtain a fast processing scheme on a sequential machine locality has been fully exploited. The data are partitioned into voxels by a smart positioning of their coordinates in memory.

Let us suppose that the range data points are stored into an array: they will be arranged such that their position in the array will reflect their position in space. In particular, points belonging to the same voxel will lie close inside the array. Each voxel, V, will be a structure, which contains the number of data points, which lie inside that voxel, N_V, and a pointer to the position in the array where the first point of that voxel is memorized, ptr_V. The arrangement of the data guarantees that all the data points which belong to that voxel lie in adjacent positions. All the points belonging to a voxel can be retrieved easily from N_V and ptr_V.

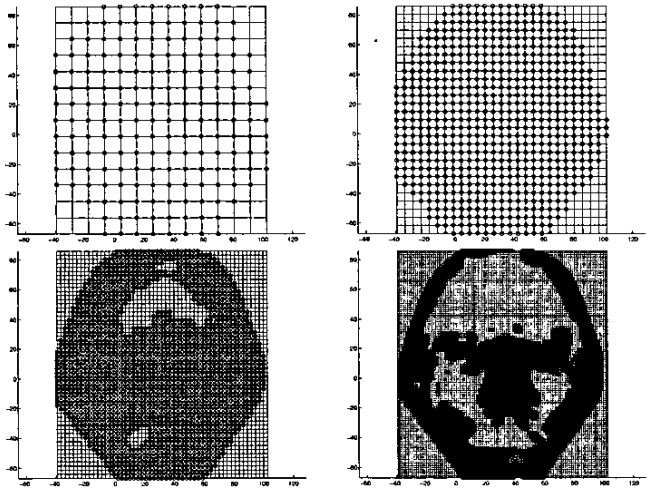


Fig. 2. Four hierarchical grids used to support the Gaussian reconstruction of a face. Circles represent the presence of a Gaussian unit in a crossing. Notice that the first grid is complete, while the grids associated to smaller scales are more dense but sparse.

This subdivision scheme can be efficiently used to compute the parameters in (3) and (6). If we accept the voxel as an approximation of the receptive field, and we align the voxels with the grid mesh supports, the points which lie inside a voxel lie also inside the receptive field of a Gaussian. The same partitioning scheme can be iterated for the higher layers using some sort of octree subdivision: each voxel (father) is subdivided into four voxels (sons) of half size, and the points beloning to each of these four voxels can be obtained by sorting only the points contained in the father voxel. The rearrangement of the points is obtained by an in-place partial sorting algorithm, a variant of Quicksort, in which the pivot value is the mean value of the cell [6]. The partitioning schema is illustrated in Fig. 5.

This processing is quite efficient, as the computation of each point does not involve more than $(\alpha+1)^2$ gaussians per layer, where α is the ratio (usually equal to 2) between the receptive field, $RF(\cdot)$, and a grid spacing ΔP.

This pre-processing allow to obtain a processing time of 1.78 s averaged over 20 trials on a Pentium III 1 GHz machine.

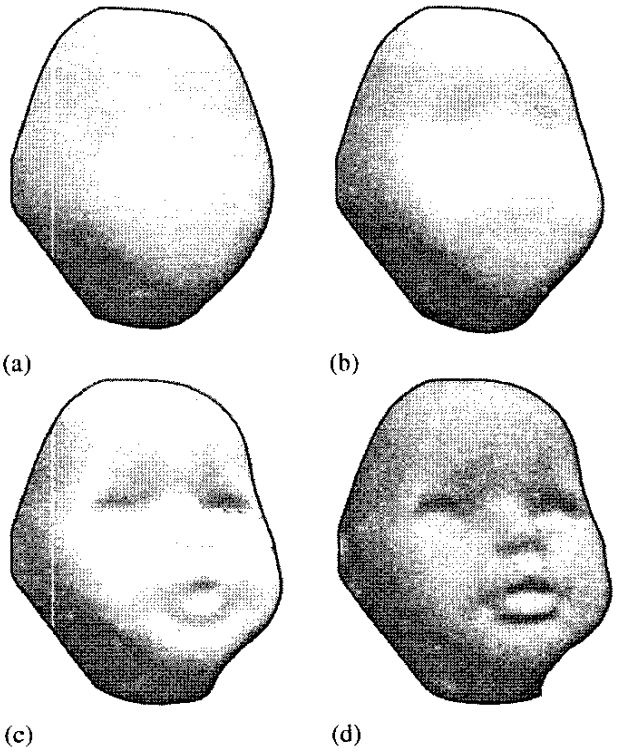


Fig. 3. The surfaces obtained by multi-layer HRBF recostruction with one (a), two (b), three (c), and four (d) layers.

The amount of overhead added by data partitioning was negligible, being the pre-processing time of one order of magnitude smaller than network configuration time.

IV. FROM HRBF SURFACES TO HRBF MESHES

The output of the HRBF network is a multi-scale continuous surface. To be visualized by graphical hardware, this surface has to be digitized, that is converted into a multi-scale mesh. One possibility is to densely sample the surface and tessellate it. This would produce an un-necessary dense mesh.

A better result, is to exploit the differential properties of the HRBF surface, and to produce mesh, which is denser in those regions where geometry contains more details.

We start by sampling the reconstructed surface in the grid crossings of the first grid. These points, $\{V_1\}$, constitute a first ensemble of mesh vertexes (cf. Fig. 4b and 4g). Notice that these points are obtained as sum of the outputs of all the four (and in general M) layers (Eq. (1)). The adequacy of the resulting mesh is evaluated by analyzing the approximation error: we will make the mesh denser (of vertices), where the approximation error is higher (cf. Fig. 4c-e and 4h-j). To the scope, the height of the reconstruct surface: is evaluated in the mid-points between two grid crossings: $z_b = S(P_b)$, $P_b = (P_{j,k} + P_{j+1,k})/2$. z_b is then compared with the piece-wise approximation and the difference

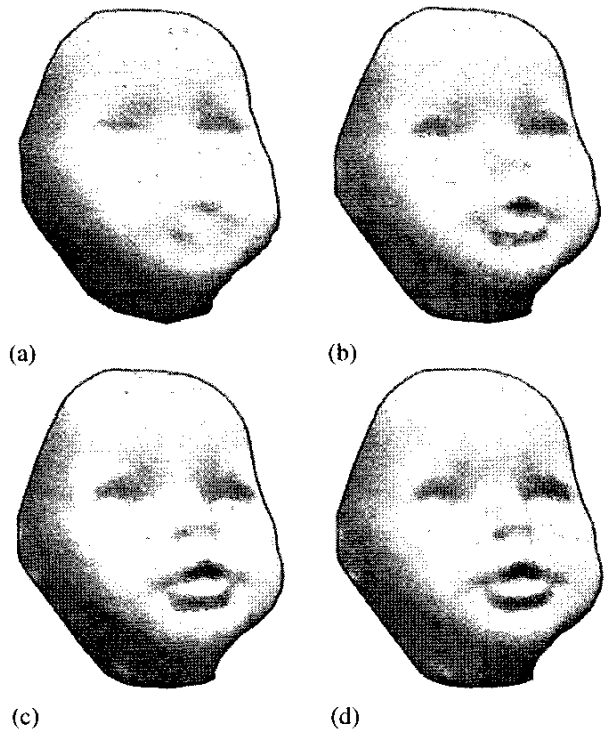


Fig. 4. The meshes obtained by the fast remeshing schema. Notice that the quality of the face is already good at the second layer, while the mouth and the nose is refined in the third and fourth layers.

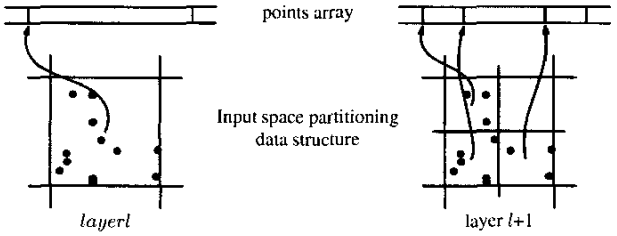


Fig. 5. The implementation of the partitioning schema into voxels.

computed as: $d_b = z_b - (S(P_{j,k}) + S(P_{j+1,k}))/2$. If this difference is over-threshold, the point (P_b, z_b) is added as a vertex of the model. This schema is iterated at the higher layers and it produces the meshes in Figs. 4c-e.

V. ACCURACY

To implement and use the HRBF processing, the underlyning theoretical foundations need to be relaxed by introducing suited approximation addressing feasibility and performance issues.

In the processing chain above described, approximations are induced in the following stages:

- residual computation.
- resampling.

In the computation of the residual stage (6), we arbitrarily limit the effects of the gaussian in a square shaped neighborhood of the gaussian center. This has been is suggested to save computation time. It is worth noting that the the approximation error of a layer can be recovered by the next layer, since it is included in the residual.

In resampling, we use a predictor to decide if the resampling should be more dense: if the prediction is comparable with the linear approximation, the region is no longer resampled.

In order to observe accuracy in the HRBF processing chain, we compare the reconstruction obtained in each stage with respect to the original data set. To evaluate accuracy we adopt the root mean square error (RMSE) and the mean and standard deviation of the absolute value of the reconstruction error (ϵ_{mean} and ϵ_{std}).

TABLE I

THE RECONSTRUCTION PERFORMACE OF THE ORIGINAL HRBF.

layer	grid size	used neurons	RMSE	ϵ_{mean}	ϵ_{std}
1	14×15	175	5.91	4.66	3.63
2	27×29	635	2.73	1.89	1.97
3	53×57	2133	1.32	0.796	1.05
4	105×113	4962	0.761	0.397	0.649

TABLE II

THE RECONSTRUCTION PERFORMANCE OF THE FAST HRBF.

layer	grid size	used neurons	RMSE	ϵ_{mean}	ϵ_{std}
1	14×15	177	5.76	4.61	3.46
2	27×29	635	2.56	1.77	1.85
3	53×57	2171	1.26	0.756	1.00
4	105×113	5104	0.748	0.411	0.625

TABLE III

THE RECONSTRUCTION PERFORMACE OF THE FAST REMESHING.

layer	grid size	used neurons	RMSE	ϵ_{mean}	ϵ_{std}
1	14×15	175	6.96	5.72	3.96
2	27×29	635	4.18	2.87	3.03
3	53×57	2133	3.52	1.87	2.98
4	105×113	4962	3.40	1.58	3.01

The tables I– III reports the figures of merit for the reconstruction obtained from the original HRBF algoritm (Gaussians with infinite domain), the HRBF trained with the fast schema (Gaussian with bounded domain), and the fast remeshing schema (predictor guided remeshing). The figures of merit of the first and the second reconstruction compare well with the measurement error of the data (0.7 mm). The fast remeshed reconstruction presents errors higher than the others due mainly to estrapolation in the the boundary regions; however, the median absolute error is 0.240 mm.

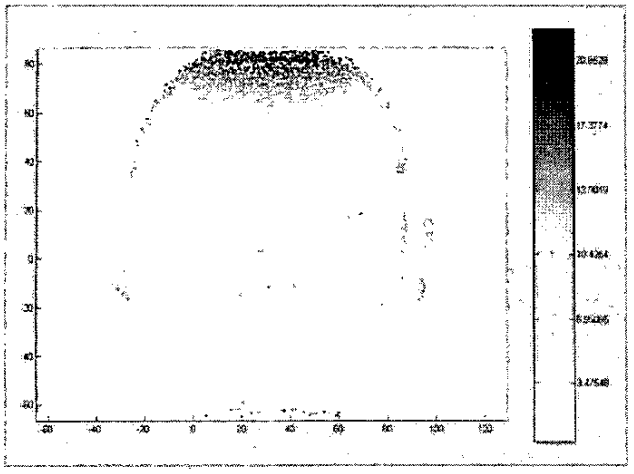


Fig. 6. The reconstruction error of the fast remeshing schema.

VI. CONCLUSIONS

The HRBF model was derived in the artificial intelligence domain, where the problem of fitting a mesh to range data is studied into the broad domain of multi-variate approximation [7]. Main characteristic of the model is the ability to reconstruct a 3D surface with no iteration on the data, therefore allowing fast computation of the configuration parameters. The closest approach to our is based on stacking grids of B-splines [8]. The main difference is that, in the HRBF model, the grids in the superior layers are not complete, but Gaussian units are inserted in clusters where the residual is over threshold. This allows coping with range of different densities and different details content and to allocate units where these are mostly requested.

Data pre-processing allows placing the data in the input array such that the points inside the receptive field of each Gaussian can be directly addressed without any sorting. This allows implementing efficiently the computation locally on the data and achieve real-time meshing on sequential machines. Computing time overhead is negligible being experimentally measured of one order of magnitude smaller than configuration time.

Accuracy is not affected by the approximation introduced by the implementation choices, as shown in Tabs. I– III.

Errors that might be introduced by the implementation choices (e.g., the quantization error [9], the approximations introduced in section III) do not decrese the reconstruction accuracy, since the constructive nature of the configuration algorithm. The figures of merit of the reconstruction error for traditional and the fast configuration algorithms are in fact almost identical, as shown in tabs. I and II. The accuracy achieved by the fast remeshing schema is sufficient for previews of the reconstruction, which allow realtime quality assesment of the scanning.

REFERENCES

[1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo project: 3D scanning of large statues," in *Siggraph 2000, Computer Graphics Proceedings*, Kurt Akeley, Ed. July 2000, Annual Conference Series, pp. 131–144, ACM Press / ACM SIGGRAPH / Addison Wesley Longman.

[2] N.A. Borghese and S. Ferrari, "A portable modular system for automatic acquisition of 3D objects," *IEEE Trans. on Instrumentation and Measurement*, vol. 49, no. 5, pp. 1128–1136, Oct. 2000.

[3] 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.

[4] N.A. Borghese, G. Ferrigno, G. Baroni, R. Savar, S. Ferrari, and A. Pedotti, "Autoscan: A flexible and portable scanner of 3D surfaces," *IEEE Computer Graphics & Applications*, vol. 18, no. 3, pp. 38–41, may–jun 1998.

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

[6] C. A. R. Hoare, "Algorithms 64: Quicksort," *Communication of the ACM*, vol. 4, no. 7, pp. 321, 1961.

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

[8] S. Lee, G. Wolberg, and S.Y. Shin, "Scattered Data Interpolation with Multilevel B-Splines," *IEEE Trans. on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, July 1997.

[9] S. Ferrari, N. A. Borghese, and V. Piuri, "Multiscale models for data processing: an experimental sensitivity analysis," *IEEE Trans. on Instr. and Meas.*, vol. 50, no. 4, pp. 995–1002, Aug. 2001.