Automatic Multiscale Meshing Through HRBF Networks

Stefano Ferrari, Iuri Frosio, Vincenzo Piuri, *Fellow, IEEE*, and N. Alberto Borghese*, Member, IEEE*

***Abstract*—A procedure for the real-time construction of three-dimensional (3-D) multiscale meshes from not evenly sampled 3-D points is described and discussed in this paper. 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 data points. The network goal is to achieve a uniform reconstruction error, equal to measurement error, by stacking noncomplete grids of Gaussians at decreasing scales. It is shown here how the HRBF properties can be used to develop a configuration algorithm, which produces a continuous surface in real time. In addition, the model is extended to automatically convert the continuous surface into a 3-D mesh according to an adequate error measure.**

***Index Terms*—Adaptive meshing, cloud of points, multiscale surface, noise filtering, real-time meshing, three-dimensional (3-D) scanner.**

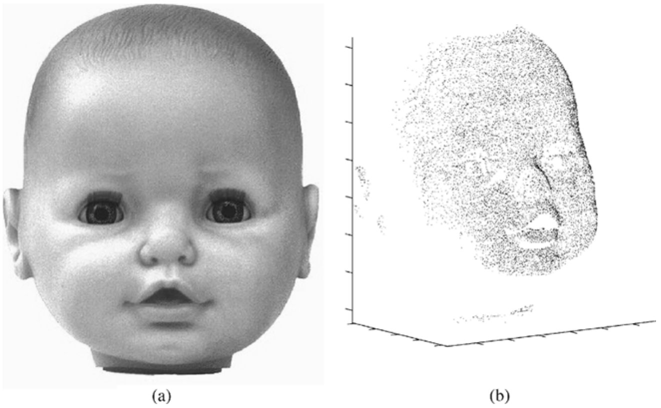

Fig. 1. Doll face in panel (a) has been scanned through the Autoscan digitizer [7], obtaining a data set of 16 851 range data points reported in panel (b).

I. Introduction

THREE-DIMENSIONAL (3-D) scanning of real objects is becoming a common technique to obtain 3-D models. The procedure is composed of two-steps: 1) measuring a set of (not evenly sampled) data points on the object's surface (cloud of points, cf. Fig. 1) and 2) generating a 3-D colored mesh, which is the *de-facto* standard in 3-D visualization. Although sampling can be indeed fast, the generation of a 3-D mesh requires a considerable amount of time as the processing chain usually requires combining a set of partial 3-D scans (each taken with a different location and attitude of the sensors) to produce a complex 3-D model. This is achieved by looping through three steps: view planning, scans alignment (registration), and surface reconstruction through merging [1]. Each partial scan can be constituted of several hundreds of thousands of data points and represents a portion of the scanned object. To acquire the entire object human intervention is required to plan additional scans, which would add to the model missing details or the object's portions not acquired in the previous scans. As a consequence of this evaluation planning procedure, the acquisition of a single model can take a very long time.

The process can be speeded up, if a visual feedback is made available in real time. In this case, subsequent scans can be efficiently planned within a short time. A solution in this direction has been recently proposed by [2], where a simplified rendering of the surface was introduced to evaluate image quality. In this technique, called splatting, an oriented planar circular shape is rendered around each sampled point to convey an idea of the surface geometry.

The approach presented here is aimed at obtaining in real-time multiscale meshes of high quality. It is based on the hierarchical radial basis function networks model (HRBF) [3], [4]), where a linear combination of Gaussian units is adopted to represent the surface. The HRBF model was derived in the connectionist domain, where the problem of fitting a mesh to range data is studied into the broad domain of multivariate approximation [5]. The main characteristic of the model is the ability to reconstruct a 3-D surface with no iteration on the data, therefore allowing fast computation of the configuration parameters. The closest approach to HRBF is based on stacking grids of B-splines [6]. In the HRBF configuration algorithm, the parameters are computed through algebraic operations carried out locally on the data: the computation of each HRBF's parameter requires considering only a subset of the data. To add finer details of the surface, which are often circumscribed in a few regions, a multiscale adaptive scheme has been developed. This schema automatically identifies these regions and inserts clusters of Gaussians at smaller scales there. This local nature of the computation can be exploited to obtain a real-time procedure for network configuration. Besides this, it is also shown how the differential properties of the Gaussian units allow defining

Manuscript received June 15, 2004; revised April 4, 2005. The research was supported in part by the Italian Ministry of Education, University of Research (MIUR) under Grant 449/97, "RoboCare: A MultiAgent System with Intelligent and Mobile Robotic Components."

S. Ferrari and V. Piuri are with the Department of Information Technologies, University of Milano, 26013 Crema, Italy (e-mail: ferrari@dti.unimi.it; piuri@dti.unimi.it).

I. Frosio and N. A. Borghese are with the Department of Computer Science, University of Milano, 20135 Milan, Italy (e-mail: frosio@dsi.unimi.it; borghese@dsi.unimi.it).

Digital Object Identifier 10.1109/TIM.2005.851471

a particular error measure, which can be used to guide the adaptive resampling of the continuous HRBF surface to obtain the 3-D mesh.

The paper is organized as follows. The HRBF model is briefly summarized in Section II. In Section III, the procedure to obtain real-time surface computation is described, and in Section IV, it is shown how to sample the surface to obtain a 3-D mesh. Results are reported in Section V and a few concluding remarks in Section VI.

II. HRBF Model

Let us suppose that the measured points can be expressed as a two-dimensional (2-D) data set, that is, as a height field $\{(P_i, z_i)\,|\, z_i = S(P_i),\, P_i = (x_i, y_i),\, 1 \leq i \leq N\}$. In this case, the surface will assume the explicit analytical shape $z = S(P)$. The output of the HRBF network is obtained by adding the output of a stack of hierarchical layers $a_l(P)$, at decreasing scales, as follows:

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

where σ_l determines the scale of the lth layer, with $\sigma_l > \sigma_l + 1$. When the Gaussian $G(\cdot;\, \sigma) = 1/(\pi\sigma^2)\exp(-\|\cdot\|^2/\sigma^2)$ is taken as basis function, the output of each layer can be written as

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

where M_l is the number of Gaussian units of the lth layer. The $G(\cdot)$ are equally spaced on a 2-D grid, which covers 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 that can be reconstructed.

The shape of the surface in (2) depends on a set of parameters: the *structural parameters*, which are the number $M = \sum_l M_l$, the scale ensemble $\{\sigma_l\}$, the position $\{P_{l,k}\}$, and the weights associated to each Gaussian $\{w_{l,k}\}$. Each grid l realizes a low-pass filter, which is able to interpolate and reconstruct the surface up to a certain scale, determined by σ_l. 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 $\{P_{l,k}\}$ [8]. In this schema, the weights $\{w_{l,k}\}$ represent the surface height measured in the grid crossings $w_{l,k} = S(P_{l,k})$. As measured points are usually not equally spaced, $S(P_{l,k})$ is not available and should be estimated. To that purpose, a weighted average of the measured points $\{P_m\}$, where the weight decreases with the distance of P_m from $P_{l,k}$, can be adopted. The estimate can be carried out locally in space, by using only those points lying in an appropriate neighborhood of $P_{l,k}$. This neighborhood, called *receptive field* $A(P_{l,k})$, can be chosen as a circular region centered in $P_{l,k}$, which have the radius proportional to the grid side ΔP_l. A possible weighting function is

$$\tilde{S}(P_{l,k}) = \frac{\sum\limits_{P_m \in A(P_{l,k})} S(P_m)\, e^{-\|P_{l,k} - P_m\|^2/\sigma_l^2}}{\sum\limits_{P_m \in A(P_{l,k})} e^{-\|P_{l,k} - P_m\|^2/\sigma_l^2}}. \tag{3}$$

Although a single layer with Gaussians of very small scale can reconstruct the finest details, this would produce an unnecessary dense packing of units in all those regions that ure a low

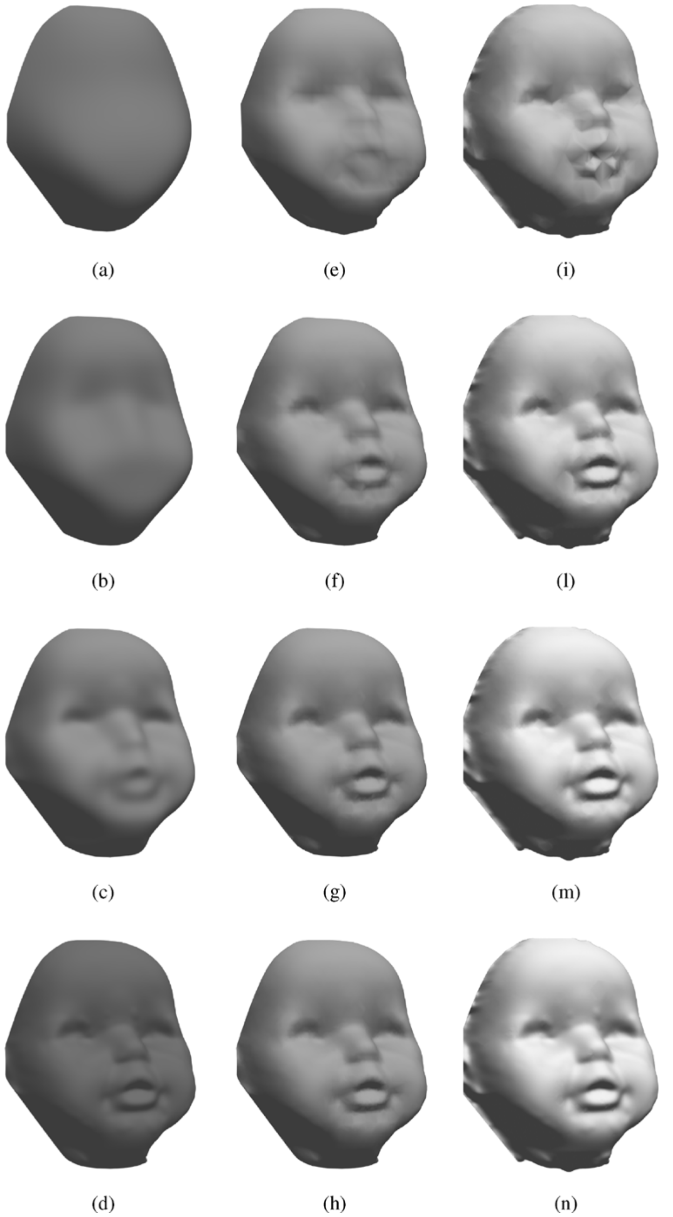

Fig. 2. Surfaces of the doll face in Fig. 1 obtained by multilayer HRBF reconstruction. The surfaces in (a)–(d) are obtained from the fast HRBF algorithm by using one to four layers; the ones in (e)–(h) are obtained from the fast meshing algorithm by restricting the meshing on the centers of, respectively, the first up to the fourth layer; the surfaces in (i)–(n) are generated by the fast meshing algorithm, with different threshold θ (0.5, 0.1, 0.05, 0.01, respectively). The surfaces in (a)–(d) are obtained by densely resampling the HRBF surface (61161 triangles). The meshes in (e)–(h) are composed by 1276, 4321, 12578, and 28 296 triangles, respectively. The surfaces in (i)–(n) are composed by 8094, 28296, 39797, and 63 172 triangles, respectively.

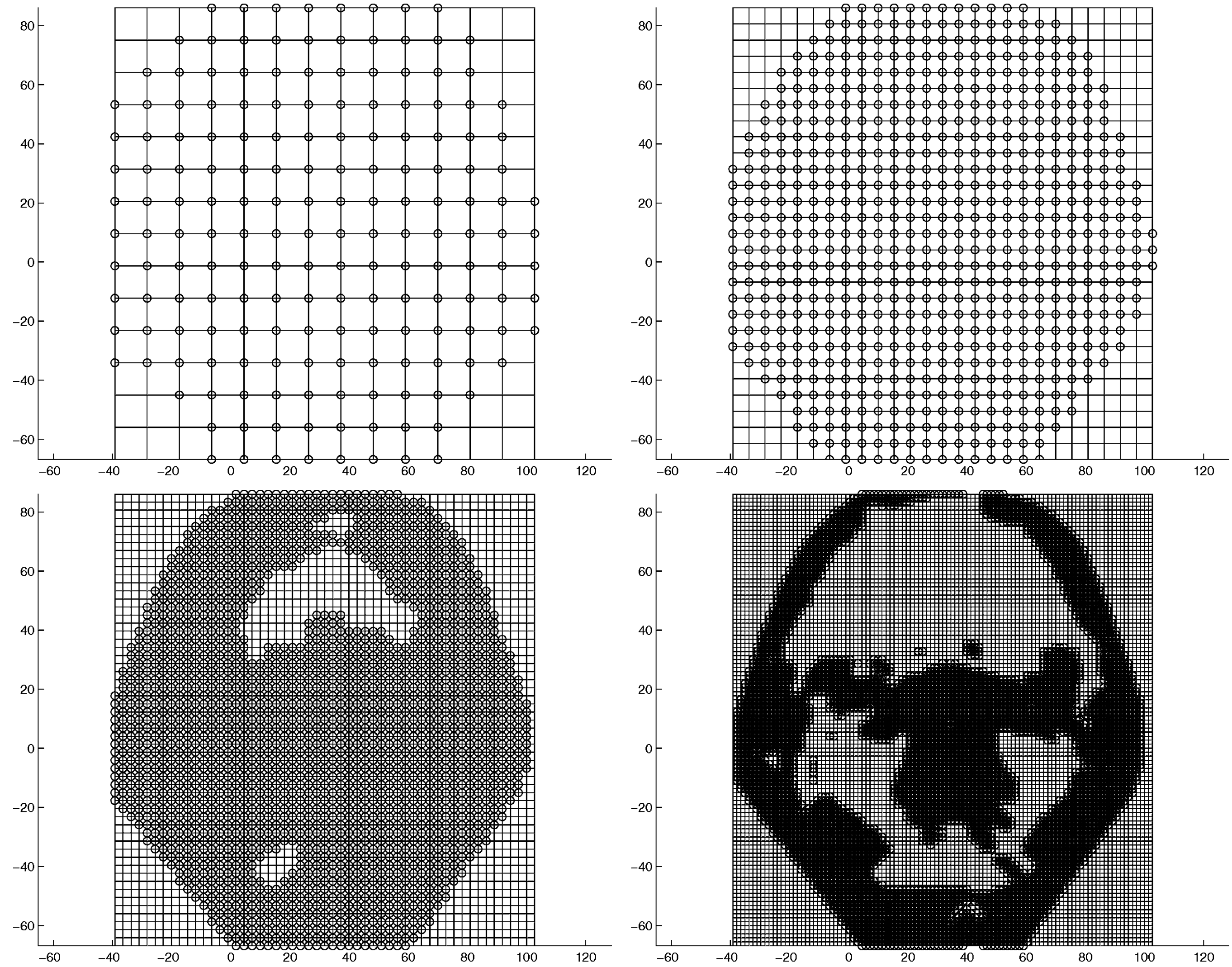

Fig. 3. Four hierarchical grids used to support the Gaussian reconstruction of a face. Circles represent the presence of a Gaussian unit in a crossing. Notice that the first grid is complete, while the grids associated to smaller scales are more dense but sparse.

scale. 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 as explained in the following.

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

$$a_1(P;\sigma_l) = \sum_{k=1}^{M_1} w_{1,k} G(\|P - P_{1,k}\|;\sigma_1). \tag{4}$$

For each sampled point, a residual is computed as the difference between the measured value of the surface height and the reconstructed one, as follows:

$$r_1(P_m) = S(P_m) - a_1(P_m). \tag{5}$$

Details at scales higher than σ_l will not be visible [cf. Fig. 2(a)–(d)]. 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 $R(P_{l,k})$ is computed as the L_1 norm of the local residuals as

$$R(P_{l,k}) = a_1(P_{l,k}) - \frac{\sum_{P_m \in A(P_{l,k})} |r_1(P_m)|}{|A(P_{l,k})|}. \tag{6}$$

When $R(P_{l,k})$ is over threshold (larger than the measurement 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. 3). 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

For each layer l, the complexity of the configuration algorithm is $O(MN)$, where M is the number of the Gaussians of the network, and N is the data set cardinality. This figure results by observing that the computation of both the residuals and the coefficients requires evaluating the distance $||P_{l,k} - P_m||$ between each measured point $\{P_m\}$ and the position of each Gaussian $\{P_{l,k}\}$ ((3) and (6)). The need to compute the distance between all data points and all Gaussian centers is required by 1) the nonzero response of the Gaussian in the whole domain $\mathbb{R}^2$ (infinite support) and 2) the extraction of the points that lie inside the receptive field.

To reduce the configuration time, several observations can be made. As Gaussian decreases rapidly toward zero, in practical applications, its influence is considered only inside a suitable neighborhood $B_{l,k}$ of the Gaussian center, which will be termed *influence region*. In the following, 3σ will be taken as radius of $B_{l,k}$, where $G(3\sigma)$ is $1.24 \cdot 10^{-4}$ times the value of the value of the Gaussian in its center.

To avoid the need of computing, for each unit (the distance from all the measured points), the key idea is to arrange the data in a data structure that allows fast retrieval of those points, which belong to the neighborhood of a unit. Easy partitioning cannot be obtained with the circular regions implicitly considered in the definition of $A_{l,k}$ and $B_{l,k}$; some approximations must be introduced, in particular, the following.

1) The receptive field of the Gaussians is approximated by the bounding square. From now on, we simply refer to these squared boxes as the *receptive field*.
2) The length of the side of the receptive field is an even multiple of the grid spacing ΔP_l. More formally, the receptive field side ΔA_l is equal to $2\rho\Delta P_l$, where $\rho \in \mathbb{N}$.
3) The Gaussian width (and, hence, the grid side ΔP_l) is halved at every layer, as follows: $\sigma_{l+1} = \sigma_l/2$.

The influence region will also be represented as a squared region of side multiple of the Gaussian spacing.

To take full advantage of locality, a quad-tree-like data storage mechanism, which organizes the data points using a recursive splitting of the input domain along both the axes and allows a fast retrieval of closed points, has been introduced [9]. Let us suppose that measured data points are stored into an array: they will be arranged such that their position in the array will reflect their position in 3-D space. In particular, points that lie inside the same influence region will be stored in close positions of the data array. To the scope, we introduce the concept of a cell, called Q, as the squared region on the support plane $x \times y$, whose vertexes are four adjacent Gaussian centers (cf. Fig. 3). Since it assumes the value of the grid side ΔP_l, the size of the side of a cell changes from layer to layer. To each Q, a data structure is associated, which contains the number of data points projecting over Q, N_Q, and the position in the array where the first point lying inside Q is stored p_Q. All the points, which lie inside a cell Q, can be retrieved easily from N_Q and p_Q. This arrangement of the data guarantees the following:

- receptive field and influence region direct mapping (the indexes of the cells associated to each Gaussian, can be directly computed from the Gaussian position index);
- quad-tree data partitioning (the cell of a higher layer can be generated by efficiently partitioning the corresponding cells of the lower layer: the union of four cells of the higher layer produces a single cell of the immediate lower layer).

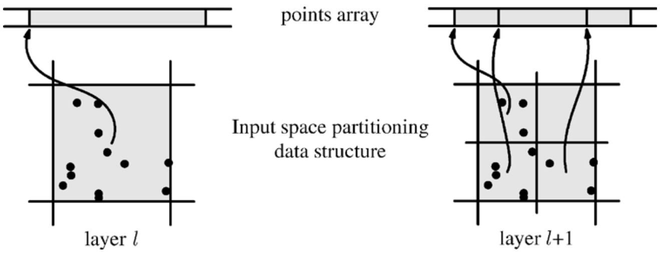

Fig. 4. Implementation of the partitioning schema into cells.

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 position of the data inside the vector segment associated to the each cell [10]. The partitioning schema is illustrated in Fig. 4, and it can be efficiently used to compute the parameters in (3) and (6).

This processing greatly improves the configuration procedure performances, as each point is now involved in the computation of no more than $(2\rho)^2$ Gaussians for each layer. Hence, the computational time linearly scales with the cardinality of the data set N. For example, the reconstruction of the surface reported in Fig. 2(a)–(d) required a processing time of 1.78 s (with a standard deviation of 0.118 s) averaged over 20 trials, on a Pentium III 1-GHz machine with 256 MB of memory. The HRBF data are reported in Table II. The amount of overhead added by data structuring was negligible, being 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 multiscale continuous surface. To be visualized by graphical hardware, this surface has to be digitized, that is converted into a multiscale mesh. This is a piecewise planar approximation of the surface. One possibility is to densely sample the surface and tessellate it. However, this would produce an unnecessary dense mesh.

A better solution can be obtained by exploiting the differential properties of the HRBF surface, to produce a mesh such that the vertices are denser in those regions containing the finest details. The rationale is to start with a low-density mesh and to make it denser around the points $\{P\}$, in which the deviation from linearity is greater than a given threshold θ. Let us call Err(P) this deviation and define it as

$$\text{Err}(P) = f(|S(P) - \text{ polyg}(P)|). \tag{7}$$

If the evaluation of Err(P) were carried out by computing $S(P)$ as in (1), there would be no advantage in later discarding the point from the mesh (at least in terms of meshing accuracy, computational time, and memory allocation). However, the differential properties of the HRBF model grant us a cheap estimate of the surface height: as shown in the Appendix , since the derivatives of each Gaussian can be computed mostly reusing some of the intermediate results of the network configuration

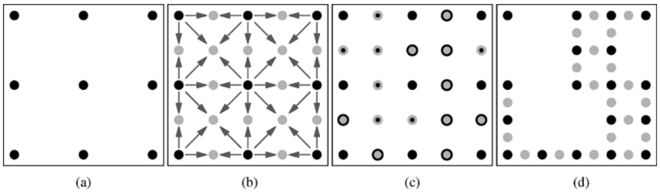

Fig. 5. HRBF fast meshing schema. In panel (a), the base points are shown. In panel (b), the set of probing points P_T is shlown in light gray. The value in P_t is estimated starting from the HRBF value computed in U_t. In panel (c), an example of residual evaluation computed for each P_t is shown. The points with the dark outer circle are over threshold and lead to the addition of new vertexes as shown in panel (d).

procedure, the computation of the derivatives of the HRBF surface is very efficient. This fact can be fully exploited when using a second-order approximation of the HRBF surface $\hat{S}(\cdot)$ in (7) instead of the HRBF value $S(\cdot)$, given by (1).

The meshing procedure starts by considering the crossings of the first grid $C_1 = \{P_{1,k}\} \subseteq \mathbb{R}^2$, which we call *base points* [cf. Fig. 5(a)]. The HRBF surface is sampled in C_1 to obtain $V_1 \subseteq \mathbb{R}^3$, which constitutes a first ensemble of mesh vertexes. Notice that the surface height in the base points is obtained as sum of the outputs of all the L layers (1) $z_{1,k} = S(P_{1,k})$ [e.g., Fig. 2(a)].

The adequacy of the resulting mesh is evaluated by analyzing the approximation error between pairs of adjacent base points: we will make the mesh denser (of vertices), where the approximation error is higher. To the scope, a probing set $\{P_t\}$ is chosen as the set of the midpoints of the segments connecting two adjacent base points. It should be noted that $\{P_t\}$ is a subset of the crossings of the second grid.

The height of the reconstructed surface in each probing point P_t, z_t is estimated through the second-order Taylor expansion of the HRBF $\hat{S}(P_t;\, P_c)$, evaluated in the neighboring grid crossings of P_t, $U_t = \{P_c\} \subseteq C_1$. In particular, z_t is evaluated as the average of the surface height estimated in the points which belong to U_t through $\hat{S}(\cdot,\, P_c)$, as follows:

$$z_t = \frac{\sum\limits_{P_c \in U_t} \hat{S}(P_t;,\, P_c)}{|U_t|}. \tag{8}$$

Fig. 5(b) depicts these operations. The arrows depart from an elements of C_1 and reach a probing point the neighborhood to which they belong. For each probing point, the estimated height is computed according to (8) and it is compared with the piecewise approximation in P_t, $\bar{S}(P_t) = \sum_{P_c \in U_t} S(P_c)/|U_t|$, offered by the mesh. The probing point P_t is added to the mesh only if, for a given threshold θ, the condition

$$|z_t - \bar{S}(P_t)| > \theta \tag{9}$$

holds. Hence, the difference $|z_t - \bar{S}(P_t)|$ is used to estimate the distance between the HRBF surface and its polygonal approximation in the point P_t. If this distance is higher of a given threshold θ, the point P_t is added to the base points set, and its HRBF surface height is computed [Fig. 5(c) and (d)]. This schema is iterated until a given criterion is met.

For example, the meshes in Fig. 2(e)–(h) and the meshes in Fig. 2(i)–(n) are computed using this schema but different stopping criteria. For the meshes of the first group, we restrict the meshing on the center of the different layers [the first up to the forth layer for the meshes in Fig. 2(e)–(h), respectively]. Hence, the stopping criterion is the number of iterations of the meshing procedure. Instead, the meshes in Fig. 2(i)–(n) have been constructed iterating the previously described procedure until the surface results under threshold in all the probing points.

TABLE I
Reconstruction Performance of the Original HRBF

layer	grid size	used units	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
Reconstruction Performance of the Fast HRBF

layer	grid size	used units	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
Reconstruction Performance of the Fast Meshing

layer	vertices	triangles	RMSE	ϵ_{mean}	ϵ_{std}
1	675	1276	5.83	4.68	3.47
2	2210	4321	2.60	1.81	1.87
3	6365	12578	1.31	0.796	1.05
4	14258	28296	0.786	0.439	0.651

V. Accuracy

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

In the processing chain previously described, approximations are induced in both the residual computation and the resampling stages. In the computation of the residual stage, approximations are introduced by arbitrarily limiting the contribution of a Gaussian inside a square region that surround the Gaussian itself, while in the meshing stage the choice of an estimator for predicting the surface height may result in a mesh of poor quality.

In order to monitor the 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}).

Tables I–III report the figures of merit for the reconstruction obtained from the original HRBF algorithm (Gaussians with infinite support), the HRBF trained with the fast schema (Gaussian with bounded support), and the fast meshing schema (predictor

TABLE IV
ACCURACY PERFORMANCE COMPARISON

	original	fast	fast meshing			
	HRBF	HRBF	θ=0.01	θ=0.05	θ=0.1	θ=0.5
RMSE	0. 761	0. 753	0. 761	0. 771	0. 786	0. 955
ϵ mean	0. 397	0. 414	0. 418	0. 427	0. 439	0. 564
$\epsilon_{st\ d}$	0. 649	0. 629	0. 636	0. 642	0. 651	0. 771

TABLE V
ACCURACY PERFORMANCE COMPARISON OF GENERATED MESHES

	θ=0.01	θ=0.05	θ=0.1	θ=0.5
RMSE	0. 0556	0. 0982	0. 137	0. 469
ϵ_{mean}	0. 00767	0. 0265	0. 0463	0. 219
ϵ_{std}	0. 0550	0. 0946	0. 129	0. 414
vertices	31711	20012	14258	4130
triangles	63172	39797	28296	8094

guided meshing with $\theta = 0.5$). These figures of merit of the first and the second reconstruction compare well with the measurement error of the data (0.7 mm). In Table III, instead of reporting the structural parameters of the HRBF network (which are the same of the network in Table II), the number of vertices and triangles of the resulting meshes are reported.

Table IV reports the accuracy figures of the fourth level of the reconstruction algorithm (original and fast HRBF) compared with the ones achieved by the fast meshing over several values of the threshold θ. The generated surfaces are reported in Fig. 2(i)–(n).

Table V reports the figures of merit achieved by the fast meshing computed with respect to the approximation of the original data given by the mesh obtained from the resampling of the fast HRBF algorithm. Besides, it reports the number of vertices and triangles of the generated meshes.

VI. DISCUSSION AND CONCLUSION

As shown in Tables I–III, the accuracy is not affected by the approximation introduced by the implementation choices. Experiments operated on several data sets confirm this conclusion.

In the computation of the residual stage (6), in order to save computational time, we arbitrarily limit the contribution of a Gaussian inside a square centered in the Gaussian center. This choice is acceptable in spaces of low dimensionality D as in our case. When D increases, the volume of a D-dimensional sphere becomes negligible with respect to the volume of its bounding D-dimensional cube, and the choice is not justified anymore. Spherical neighborhood region computation requires additional processing, which makes this approach less appealing. However, the accuracy of the reconstruction is not affected as the contribution of a Gaussian is negligible outside its neighborhood region. Moreover, it is worth noting that the approximation error of a layer can be recovered by the next layer, since this is included in the residual [4], [11].

Another potential source of inaccuracies is the use of an estimator in (8) to decide if the mesh resampling should be more dense. As vertices added to the mesh depend on the surface prediction (8), inaccuracy of the prediction may result either in a mesh unnecessarily dense or in a mesh poor of details.

As pointed out by the figures of merit reported in Table IV, the use of the second-order Taylor expansion as surface predictor has proven to be effective. However, the comparison (9) between the height of predicted surface and the height of the polygonal approximation in itself may not be adequate in measuring the difference of the visual appearance of the two surfaces. Fig. 2(i)–(n) illustrates this point, as they are generated by using only the threshold θ as stopping criterion to drive the meshing procedure. Comparing them with the surfaces reported in Fig. 2(e)–(h), it is evident that they employ an unneccessarily high number of triangles to reconstruct the surface, whether a stopping criterion based on the number of vertices may balance the opposite requirements of accuracy and lightness of the mesh. For the task of measuring the difference in visual appearance, a better figure of merit could be the distance between the polygonal approximation and its projection onto the predicted surface. However, this operation cannot be used in the present application due to its high computational cost.

The relaxation of some theoretical aspects of the hierarchical radial basis functions (HRBF) algorithm allows implementing an efficient computation of the HRBF parameters, which operate locally on the data and achieve real-time meshing on sequential machines. Computing time overhead for the preprocessing is negligible, being experimentally measured of one order of magnitude smaller than configuration time.

Errors that might be introduced by the implementation choices (e.g., the quantization error [11], the approximations introduced in Section III) do not decrease the reconstruction accuracy, because of 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 Tables I and II. A comparison between the two reconstructed surfaces is shown in Fig. 6. It can be noticed that in the region covered by the points, the difference is almost negligible.

The fast meshing schema allows achieving real-time meshing, at least for a preview of the reconstruction, which allows real-time assessment of the quality of the obtained model. As reported in Fig. 7, the error is distributed mainly in the border region, while it is perceptually negligible in the inner region. It should be noted that as most of the expansions are computed along the axis, they do not require all the derivatives. In other words, once the HRBF surface $S(P)$ has been computed, with little additional effort, the function $\hat{S}(\cdot;\, P_p)$, which evaluates the surface height with respect to P_p, is available. The accuracy of the generated mesh allows correcting the scanning setup and planning additional refinement scans immediately in order to avoid artifacts or holes in the produced mesh.

Moreover, the accuracy achieved by the fast meshing schema is adjustable by the threshold θ. The value θ can be regulated on both the computational resources, which can be devoted to the meshing procedure, and the capability of the visualization device. Besides, the generated mesh can be a good starting point

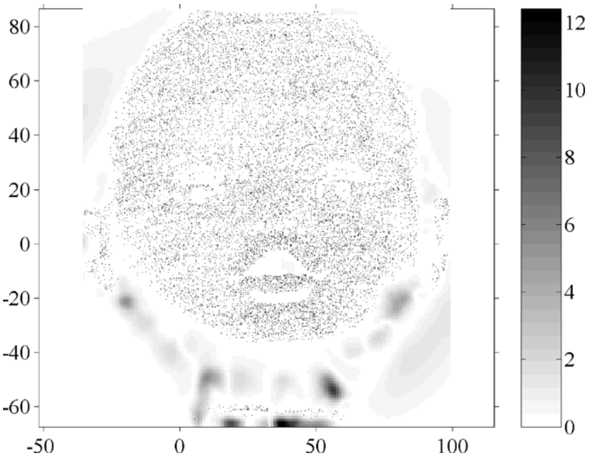

Fig. 6. Reconstruction error of the fast configuration schema.

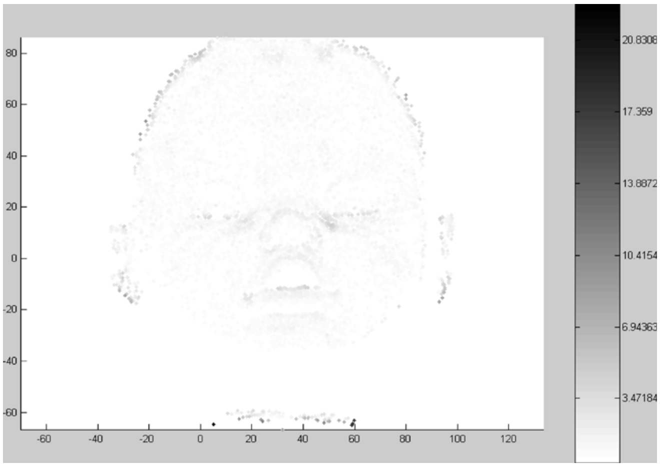

Fig. 7. Reconstruction error of the fast meshing schema.

for mesh optimization procedures, since it can be enriched with information on the differential properties of the surface. This information may be obtained with a little additional effort by the resampling procedure.

APPENDIX

The second-order Taylor's expansion of a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is

$$f(p+\Delta p) = f(p) + \frac{\partial f(p)}{\partial x}\Delta p_x + \frac{\partial f(p)}{\partial y}\Delta p_y + \frac{1}{2}\frac{\partial^2 f(p)}{\partial x^2}\Delta p_x^2 + \frac{1}{2}\frac{\partial^2 f(p)}{\partial y^2}\Delta p_y^2 + \frac{\partial^2 f(p)}{\partial x \partial y}\Delta p_x \Delta p_y \tag{10}$$

where $p \in \mathbb{R}^2$ and p_x and p_y are the components of p along the axes.

Reframing (1) as

$$S(P) = \sum_{l=1}^{L}\sum_{k=1}^{M_l} \frac{w_{l,k}}{\pi\sigma_l^2} \exp\left(-\frac{\|P-P_{l,k}\|^2}{\sigma_l^2}\right) = \sum_{l=1}^{L}\sum_{k=1}^{M_l} G_{l,k} \tag{11}$$

the derivatives of the HRBF surface (1) can be computed as

$$\frac{\partial S(P)}{\partial x} = \sum_{l=1}^{L}\sum_{k=1}^{M_l} -\frac{2}{\sigma_l^2} G_{l,k}(P_x - P_{k_x,l}) = \sum_{l=1}^{L}\sum_{k=1}^{M_l} H_{l,k}\Delta P_{k_x,l} \tag{12}$$

where $H_{l,k} = -(2/\sigma_l^2)G_{l,k}$ and $\Delta_x P_{k,l} = (P_x - P_{k_x,l})$, and

$$\frac{\partial S(P)}{\partial y} = \sum_{l=1}^{L}\sum_{k=1}^{M_l} H_{l,k}\Delta P_{k_y,l} \tag{13}$$

$$\frac{\partial^2 S(P)}{\partial x^2} = \sum_{l=1}^{L}\sum_{k=1}^{M_l} H_{l,k}\left(-\frac{2}{\sigma_l^2}\Delta P_{k_x,l}^2 + 1\right) \tag{14}$$

$$\frac{\partial^2 S(P)}{\partial y^2} = \sum_{l=1}^{L}\sum_{k=1}^{M_l} H_{l,k}\left(-\frac{2}{\sigma_l^2}\Delta P_{k_y,l}^2 + 1\right) \tag{15}$$

$$\frac{\partial^2 S(P)}{\partial x \partial y} = \sum_{l=1}^{L}\sum_{k=1}^{M_l} -\frac{2}{\sigma_l^2} H_{l,k}\Delta P_{k_x,l}\Delta P_{k_y,l}. \tag{16}$$

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: 3-D scanning of large statues," in *Proc. Computer Graphics (Siggraph 2000) (*Annual Conf. Series*)*, K. Akeley, Ed., Jul. 2000, pp. 131–144.

[2] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3-D model acquisition," in *Proc. 29th Annu. Conf. Computer Graphics Interactive Techniques*, San Antonio, TX, 2002, pp. 438–446.

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

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

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

[6] S. Lee, G. Wolberg, and S. Y. Shin, "Scattered data interpolation with multilevel B-splines," *IEEE Trans. Vis. Comput. Graphics*, vol. 3, no. 3, pp. 228–244, Jul. 1997.

[7] N. A. Borghese, G. Ferrigno, G. Baroni, R. Savarè, S. Ferrari, and A. Pedotti, "Autoscan: A flexible and portable scanner of 3-D surfaces," *IEEE Comput. Graph. Appl.*, vol. 18, no. 3, pp. 38–41, May–Jun. 1998.

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

[9] N. A. Borghese, S. Ferrari, and V. Piuri, "Real-time surface reconstruction through HRBF networks," in *Proc. IEEE Int. Workshop Haptic Virtual Environments Their Applications (HAVE 2002)*, Nov. 2002, pp. 19–24.

[10] C. A. R. Hoare, "Algorithms 64: Quicksort," *Commun. ACM*, vol. 4, no. 7, p. 321, 1961.

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

Stefano Ferrari received the computer science degree from the University of Milano, Italy, in 1995 and the Ph.D. degree in computer and automation engineering from the Politecnico di Milano, Milan, Italy, in 2001.

Since 2003, he has been an Assistant Professor with the Department of Information Technologies, University of Milano, Crema, Italy.

His research interests are related mainly to neural networks and soft-computing paradigms and their application to the computer graphics.

Iuri Frosio is currently working toward the Ph.D. degree at the Bioengineering Department of Politecnico of Milan.

He is currently with the Information Science Department, University of Milano, Milan, Italy, where he works as a Research Fellow at the Applied Intelligent Systems (AIS) Laboratory. His research interests include three-dimensional reconstruction, motion capture systems, neural networks, digital radiography, and tomography.

Vincenzo Piuri (S'84–M'86–SM'96–F'01) received the Ph.D. degree in computer engineering from the Politecnico di Milano, Crema, Italy, in 1989.

From 1992 to September 2000, he was an Associate Professor of Operating Systems at the Politecnico di Milano. Since October 2000, he has been a Full Professor of Computer Engineering at the University of Milano, Milan, Italy. He was a Visiting Professor at the University of Texas at Austin during summers 1993 and 1999. His research interests include distributed and parallel computing systems; computer arithmetic; application-specific processing architectures; digital signal processing architectures; fault tolerance; neural network architectures; theory and industrial applications of neural techniques for identification, prediction, and control; and signal and image processing. His original results have been published in more than 150 papers in book chapters, international journals, and proceedings of international conferences.

Prof. Piuri is a Member of Association for Computing Machinery (ACM), the International Neural Network Society, and the Italian Electrotechnical and Electronic Association (AEI). He is Vice-President for Publications of the IEEE Instrumentation and Measurement Society, Vice-President for Member Activities of the IEEE Neural Networks Society, and a Member of the Administrative Committees of both the IEEE Instrumentation and Measurement Society and the IEEE Neural Networks Society. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the *Journal of Systems Architecture*.

N. Alberto Borghese (M'97) graduated with full honors in electrical engineering from the Politecnico of Milano, Milan, Italy in 1985.

He is Associate Professor at the Department of Computer Science of the University of Milano, Milan, Italy, where he teaches Intelligent Systems and Digital Animation and directs the Laboratory of Applied Intelligent Systems. He was Visiting Scholar at the Center for Neural Engineering of the University of Southern California (USC), Los Angeles, in 1991, at the Department of Electrical Engineering of the California Institute of Technology (Caltech), Pasadena, in 1992, and at the Department of Motion Capture of Electronic Arts, Vancouver, Canada, in 2000. His research interests include quantitative human motion analysis, modeling and synthesis in virtual reality, and artificial learning systems, areas in which he has authored more than 30 refereed journal papers.