Real-time Surface Meshing through HRBF Networks

N. Alberto Borghese[1], Stefano Ferrari[2] and Vincenzo Piuri[2]
[1]Department of Computer Science, University of Milano, Italy
Email: borghese@dsi.unimi.it
[2]Department of Information Technologies, University of Milano, Italy
Email: {ferrari, piuri}@dti.unimi.it

Abstract **– A procedure for real-time 3D meshing reconstruction from sparse data is presented. The approach is based on Hierarchical Radial Basis Functions Networks, which allow for an effective reconstruction of multi-scale surfaces. This model is extended to provide not only the continuous surface description, but also real-time operation. To this purpose the HRBF network differential properties have been exploited to produce a denser mesh in regions where geometry is more detailed.**

Index terms **– HRBF Neural Networks, Mesh, 3D scanners, Real-time approximation.**

I. Introduction

Digitization of 3D shapes of real objects is rapidly expanding to different fields, ranging from entertainment to medicine, from design to archeology and robotics. Digitization is performed by using 3D scanners, which execute the following basic steps: acquisition of range data (Fig. 1a), post-processing through filtering, surface reconstruction, and 3D mesh creation. Post-processing can be combined in different ways [1-2].

However, there no tool is yet available to provide the operator with a real-time good-quality feedback on how the acquired surface will appear. This is essential to plan a new acquisition session or particular acquisition sessions targeted to fill the incomplete parts in the 3D model. An approach in this direction is [3], which, however, does not produce the mesh in real-time, but only a set of data points on the surface.

The aim of this paper is to propose a real-time procedure, which allows for producing the 3D mesh in real-time from sparse noisy data, by using Hierarchical Radial Basis Function networks [4]. The obtained mesh is automatically adapted to the geometric differential properties of the surface, being denser in those regions where most the details are concentrated, by taking advantage of the analytical shape of the HRBF surface. This approach can be coupled with a variety of techniques described in the literature to sample range data, e.g., defocusing [6], stereo matching [7], silhouettes [8], structured light [3], and time of flight (commercial systems by 3DV and Perception). Range data consist of a cloud of 3D points that may be not equally spaced and whose position is affected by measurement noise.

Available meshing methods are post-processing tools – operating off-line and not in real time – in which the cloud of points is converted into a mesh; the mesh may be optimized according to geometry, color or other field attributes, by means of one of the techniques for mesh simplification [9] (some high-end software packages, like Softimage or Maya, even incorporate polygon decimation routines).

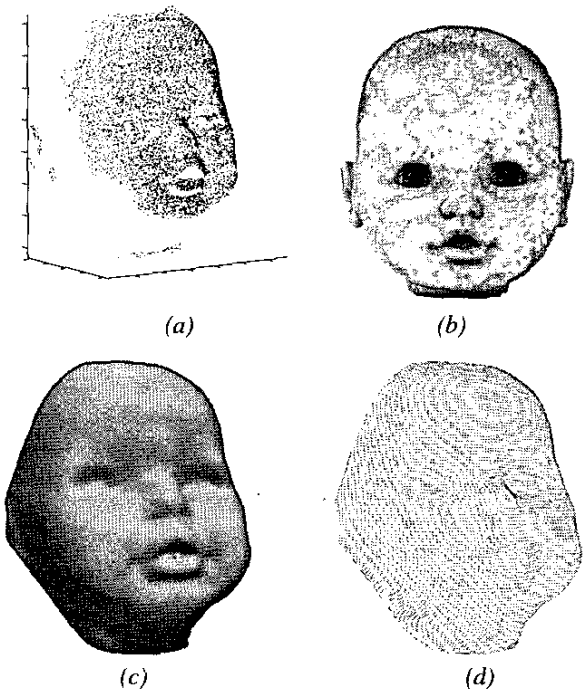

Fig. 1. 3D reconstruction: (a) range data sampled on the (b) "Ciccio" surface, (c) the reconstruction through HRBF. In (d) the HRBF are sampled with step equal to 0.682 mm to obtain a dense mesh of 30638 vertices and 61161 triangles.

In the approach presented here, mesh creation is obtained through an intermediate stage of continuous surface obtained by using HRBF networks, which allows for deriving a good quality estimate of the linear approximation offered by polygonal meshes. Thanks to the local properties of the computation, the method can work in real-time and can be implemented in strict real-time on parallel architectures.

In Section II, the HRBF configuration procedure is described. In Section III, the meshing algorithm and its assumption are reported. Section IV presents some results, which are discussed in Section V.

0-7803-7898-9/03/$17.00 ©2003 IEEE

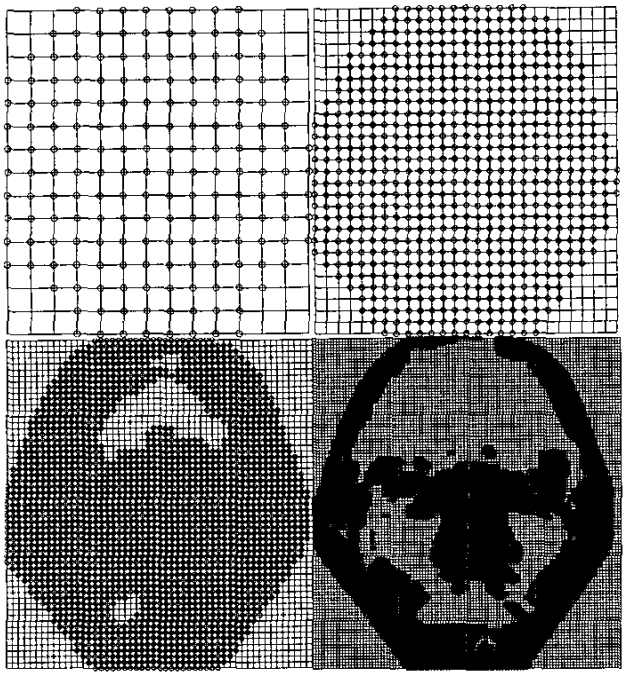

Fig. 2. The grids of the four layers of the HRBF networks that were used to reconstruct "Ciccio". The grid crossings in which Gaussians are inserted are indicated with a circle.

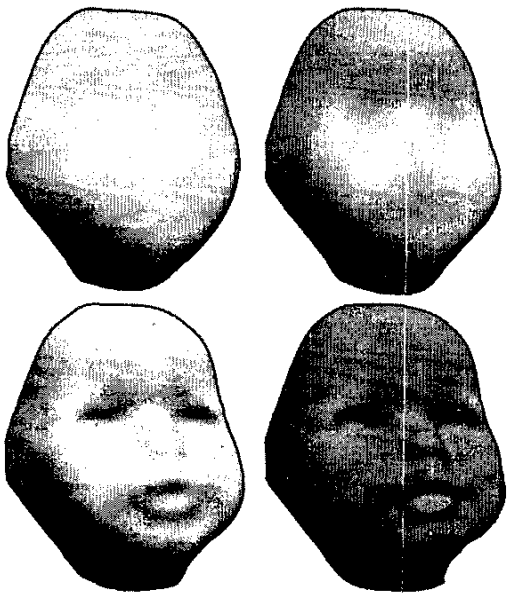

Fig. 3. The surfaces of "Ciccio" obtained by multi-layer HRBF reconstruction with one, two, three and four layers from left to right from top do bottom, respectively.

II. Configuration of the HRBF Networks

Reconstruction of a 3D surface from range data can be viewed as an instance of the problem of multi-variate approximation [5-10]. HRBF networks are one of the powerful tools made available by the connectionist community to solve this problem. They are particularly suitable in our case because they can be made operating in real-time [11]. The HRBF network is a self-organizing model, composed by a set of hierarchical RBF networks (*layers*). The HRBF networks produce a multi-scale reconstruction, where the first layer, $a_1(p)$, produces a rough approximation of the surface (high scale value), while the higher layers, $a_2(p)$, $a_3(p)$, ..., $a_M(p)$, provide details at decreasing scales. The surface down to the l-th scale is given by

$$s_l(p) = a_1(p) + \sum_{j=2}^{l} a_j(p) \tag{1}$$

where the larger is l, the more detailed is the reconstruction.

The neurons of each layer are distributed on a regular grid, with side Δx_l, and share the same scale parameter, σ_l, which determines the scale of the layer. In the case of Gaussian units, σ_l regulates their width and the output of each layer is a linear combination of Gaussian units of the same width:

$$\begin{aligned} a_l(p) &= \sum_{i=1}^{M_l} w_{i,l}\, g(\| p - c_{i,l} \|; \sigma_l) = \\ &= \sum_{i=1}^{M_l} \frac{w_{i,l}}{\pi\sigma_l^2} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^2} \right) \end{aligned} \tag{2}$$

where $\{w_{i,l}\}$ are the weights associated to the Gaussian positioned in the $\{c_{i,l}\}$ node of the grid. M_l is the number of Gaussians in the l-th layer. It has been shown that the value of the weights is equivalent to that of the surface sampled in the grid crossings. Since this value is not available in many applications, it is estimated as a weighted mean of the points, which belong to an appropriate neighborhood of $c_{i,l}$, called *receptive field*, $S_{i,l}$ [5] (cf. Eq. (5)).

The configuration algorithm proceeds as follows. Let us organize the range data into a data set $P = \{(p_i \in R^2; z_i \in R)\}$. For each layer, l, and for each neuron, i, the data points which belong to the receptive field of the neuron, $S_{i,l}$ are retrieved and form the subset $Q_{i,l}$. The difference between the data points in $Q_{i,l}$ and the surface height produced by the HRBF in the same points is computed as:

$$r_l(p_i) = z_i - \sum_{k=1}^{l} a_k(p_i) \tag{3}$$

For each grid crossing, $c_{i,l}$, the quality of the surface approximation in its neighborhood is evaluated through an integral metric to limit the impact of the outliers:

$$E_l(c_{i,l}) = \frac{\sum_{p_j \in S_{i,l}} |r_{l-1}(p_j)|}{|S_{i,l}|} \quad (4)$$

and constitutes the local residual error. If the residual is larger than the measurement error, the neuron is inserted in the grid crossing; otherwise no unit will be present there (Fig. 2).

The weight, $w_{i,l}$, associated to $r_{i,l}$ is computed by taking into account all the points inside the receptive field, $S_{i,l}$, with the rationale that the closer is a point, $p_{i,l}$, the larger is its weight:

$$w_{i,l} = \frac{\sum_{p_j \in S_{i,l}} r_{l-1}(p_j) g(\|p_j - c_{i,l}\|; \sigma_l)}{\sum_{p_j \in S_{i,l}} g(\|p_j - c_{i,l}\|; \sigma_l)} \quad (5)$$

Therefore, at each layer more details are added to the surface in those regions where the residual is large. The configuration procedure is stopped when the residual goes below a threshold over the entire input domain. In Fig. 3 we present the multi-scale reconstruction obtained from the range data in Fig. 1a, by a four-layers HRBF featuring σ = [16, 8, 4, 2] mm.

All the configuration operations are made on the points inside the receptive field. This locality property has been exploited to produce a real-time version of the configuration algorithm [9].

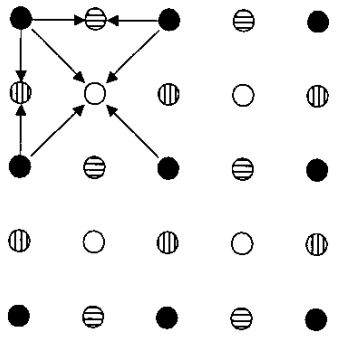

Fig. 4. Data sets. Base sampling set, $B \subset R^2$, (black circles). Testing set, $T \subset R^2$: testing points are located horizontally (points filled with horizontal lines), vertically (points filled with vertical lines), and diagonally (white circles). Arrows identify the base sampling points used to predict the surface height in the testing point.

III. From HRBF Surfaces to HRBF Meshes

A. Overview

A 3D mesh is a polygon approximation of a surface, usually constituted of triangles. To describe a mesh, the position of the polygon vertices and their connectivity is therefore required.

To get a 3D mesh from the surface in Fig. 1c (Eq. 1), one possibility is to densely sample it. As a result we obtain a very dense 3D mesh of the kind reported in Fig. 1d. This mesh has been obtained by sampling the surface by half the side of the last grid: Δx_s = 1/2 Δx_M = 0.682 mm, and it is constituted of 30638 vertices and 61161 polygons. This is actually much more than required to represent the object, and a more parsimonious representation like the one in Fig. 6 would be desirable to manage the model in real-time.

The rationale of the method presented here is to start with a low-density mesh and to make it denser where the deviation from the linearity has significant magnitude by inserting a new vertex in these positions.

In more mathematical terms a polygonal approximation of a continuous surfaces is equivalent to a piece-wise linear approximation. This approximation would produce an error outside the vertexes given by:

$$Err(p) = f(|s(p) - polyg(p)|) \quad (6)$$

where $f(\cdot)$ is an error evaluation function, for instance, L^∞ (max metric). The error is evaluated in few strategic points: a new vertex is created in those points where the error is above the threshold.

B. Meshing scheme

The meshing scheme operates as follows. First, a base sampling set, $B_1 \subset R^2$, is identified (Fig. 5a). This is composed by all the grid centers of the first layer. A mesh at the lowest resolution, K_1, is obtained by connecting the HRBF surface points evaluated in B_1 (Fig. 6a). From B_1 a *test set*, $T_1 \subset R$, is identified. This is constituted of the points in between adjacent points in B_1 (Fig. 4). Hence, they can be identified as the subset of the grid centers of the second layer, which do not belong to the grid centers of the first layer. Each point in T_1, t, can be related to the set of points in B_1, *neigh*(t), which surround it. This relation is shown in Fig. 4 as arrows that connect the elements of *neigh*(t) with t.

To evaluate the adequacy of K_1 in approximating the surface, we first estimate the surface height in T_1 by using a second order approximation of the HRBF surface (cf. Appendix): $q_1(t)$, $t \in T_1$. When the difference between $q_1(t)$ and the polygon approximation, $K_1(t)$, is above a given threshold, ε, a new vertex is generated in that position, t.

In more mathematical terms, let us define $\tilde{T}_1$, the set of test points where the error is above the threshold:

$$\tilde{T}_1 = \{t \in T_1 \mid Err(t) = |q_1(t) - K_1(t)| > \varepsilon\} \quad (7)$$

Since t is a midpoint of *neigh*(t), the height of the mesh in t, $K_1(t)$, can be computed as the mean value of the surface height in its neighbors, *neigh*(t). The surface height estimation, $q_1(t)$, is computed as the mean of the second order Taylor expansion centered in the element of *neigh*(t).

The updated base sampling set, $\tilde{B}_1$, is defined as the union of $\tilde{T}_1$ and B_1: $\tilde{B}_1 = B_1 \cup \tilde{T}_1$. The new mesh, $\tilde{K}_1$, is obtained by connecting the HRBF surface points evaluated in $\tilde{B}_1$.

This schema is iterated in the higher layers by assuming $B_l = \tilde{B}_{l-1}$ and $K_l = \tilde{K}_{l-1}$, until the approximation error goes below the threshold over the entire input domain. The test set for layer l, T_l, is composed by the grid centers of the $(l+1)$-th layer whose neighbors belong to B_{l+1}.

For a test point, t, such that $t \in T_l$, but $t \notin \tilde{T}_l$ (i.e., the mesh in t is sufficiently accurate) no more points will be added in between *neigh*(t) in the subsequent iterations.

The procedure can be repeated as long as none of the errors is below threshold for each test point, or a given resolution is achieved. For instance, the mesh in Fig. 6, has been obtained by considering a test set with points spaced by half the side of the grid of the last layer.

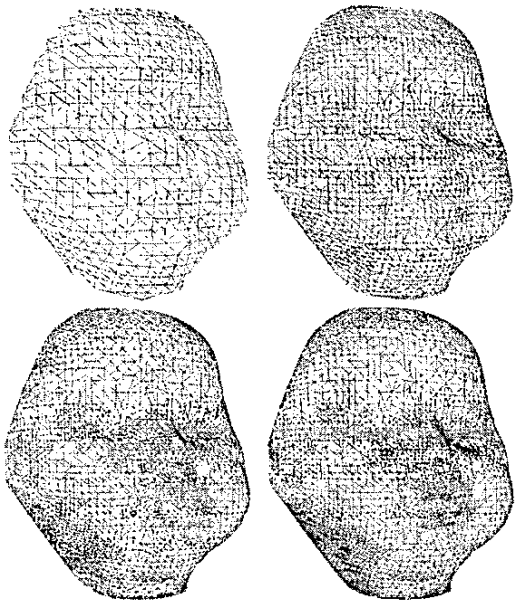

Fig. 5. The mesh is plotted for each of the four layers. In the higher layers the mesh is made denser in the most complex regions of the face.

IV. Experimental Results

The method has been extensively applied to generate various surfaces from range data. Results are reported here only for the "Ciccio" model (Fig. 1), whose surface was sampled in 16851 points.

The surface was reconstructed by using a four-layer HRBF network with the parameters reported in Table I and a final residual error of 0.779 mm (very close to the nominal measurement error equal to 0.5 mm).

For the four-layer HRBF mesh, a polygonal mesh was directly derived according to the method presented in Section III, by using ε = 1 mm (Eq. 7). The resulting meshes are shown in Fig. 5 and Fig. 6. This result should be compared with that reported in Fig. 3. As it can be seen, there is no appreciable difference. This has been also confirmed by a quantitative analysis of the approximation error. The polygonal approximation error was obtained by computing the difference in height between the HRBF surface and the polygonal approximation in 30,000 points uniformly distributed all over the domain. The resulting *mse* is 0.0802 mm, with an *std* of 0.119 mm.

The time figures reported in Table I refer to a PC Pentium 3, 1Ghz machine.

TABLE I
HRBF Surface Reconstruction Figures

layer	σ [mm]	grid size	used neurons	reconstruct std [mm]	computing time [ms]
1	16	14×15	177	5.19	551
2	8	27×29	635	2.53	482
3	4	53×57	2171	1.25	454
4	2	105×113	5104	0.779	297

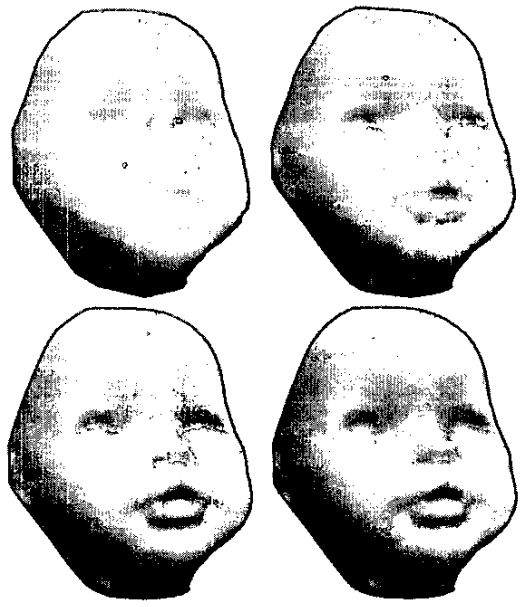

Fig. 6. The meshes obtained directly by the four layers of the HRBF network is plotted in the Gouraud shading.

V. Discussion and Conclusions

The computation of the polygonal mesh does not introduce any significant increase in the computational time. The prediction of the surface height based on the second order approximation of the HRBF surface is in fact composed by two terms: the surface itself and a linear or quadratic difference.

As shown in Appendix, the terms that compose the derivatives of a HRBF surface are the same that have to be computed for the surface calculation.

As a result, when the five derivatives are computed at the same time of the surface their computation is quite cheaper: only a 25% time increase is required.

Moreover, since the value of the surface in *neigh(t)* will be in both $q_1(t)$ and $K_1(t)$, they will be simplified in the $Err(t)$ evaluation (Eq. 7).

The alternative approach would be to compute the effective height of the HRBF in the testing points; however, the computation of the HRBF function in those points would increase the computational cost.

Where the HRBF approximation considers a smooth function, the error is experimentally verified to be negligible. Since the prediction of the function height in the probing points is much cheaper than its direct computation, the meshing technique is efficient. As a consequence, a fast multi-scale meshing is obtained.

The polygonal error is computed in the middle point between two basic points, but it is not guaranteed that this is the maximum error available. Although other error schema can be adopted, for smooth function, the assumption to sample the error in the middle point is a reasonable assumption. On the other side, we do not claim to derive an optimal meshing (although it is indeed a good one), but the mesh derived is extremely powerful in deriving information about the quality of the scan in real-time.

Appendix

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

$$f(p+\Delta p) = f(p) + \frac{\partial f(p)}{\partial x}\Delta p_x + \frac{\partial f(p)}{\partial y}\Delta p_y + \\ + \frac{1}{2}\frac{\partial^2 f(p)}{\partial x^2}\Delta p_x^{\,2} + \frac{1}{2}\frac{\partial^2 f(p)}{\partial y^2}\Delta p_y^{\,2} + \frac{\partial^2 f(p)}{\partial x \partial y}\Delta p_x \Delta p_y \quad (8)$$

From the analytical shape of HRBF network (Eq. 1), the first derivatives of $s(p)$ are:

$$\frac{\partial s(p)}{\partial x} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2w_{i,l}}{\pi\sigma_l^{\,4}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right)(p_x - c_{i,l_x}) \quad (9a)$$

$$\frac{\partial s(p)}{\partial y} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2w_{i,l}}{\pi\sigma_l^{\,4}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right)(p_y - c_{i,l_y}) \quad (9b)$$

and the second order derivatives are:

$$\frac{\partial^2 s(p)}{\partial x^2} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2w_{i,l}}{\pi\sigma_l^{\,4}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right)\left(-\frac{2}{\sigma_l^{\,2}}(p_x - c_{i,l_x})^2 + 1\right) \quad (10a)$$

$$\frac{\partial^2 s(p)}{\partial y^2} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2w_{i,l}}{\pi\sigma_l^{\,4}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right)\left(-\frac{2}{\sigma_l^{\,2}}(p_y - c_{i,l_y})^2 + 1\right) \quad (10b)$$

$$\frac{\partial^2 s(p)}{\partial x \partial y} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} \frac{4w_{i,l}}{\pi\sigma_l^{\,6}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right)(p_x - c_{i,l_x})(p_y - c_{i,l_y}) \quad (10c)$$

It should be noticed that the formulas (9)-(10)'s share some terms with Eqs. (1) and (2). When derivatives are computed at the same time with the surface, this fact can be exploited to save computation time.

Reframing Eqs. (1) and (2) as:

$$s(p) = \sum_{l=1}^{L}\sum_{i=1}^{M_l} \frac{w_{i,l}}{\pi\sigma_l^{\,2}} \exp\left(\frac{\| p - c_{i,l} \|^2}{\sigma_l^{\,2}}\right) = \sum_{l=1}^{L}\sum_{i=1}^{M_l} G_{i,l} \quad (11)$$

The derivatives (9)'s and (10)'s can be computed as in (12)'s and (13)'s, respectively:

$$\frac{\partial s(p)}{\partial x} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2}{\sigma_l^{\,2}} G_{i,l}(p_x - c_{i,l_x}) = \sum_{l=1}^{L}\sum_{i=1}^{M_l} H_{i,l}(p_x - c_{i,l_x}) \quad (12a)$$

$$\frac{\partial s(p)}{\partial y} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} H_{i,l}(p_y - c_{i,l_y}) \quad (12b)$$

$$\frac{\partial^2 s(p)}{\partial x^2} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} \frac{H_{i,l}}{\sigma_l^{\,2}}\left(-\frac{2}{\sigma_l^{\,2}}(p_x - c_{i,l_x})^2 + 1\right) \quad (13a)$$

$$\frac{\partial^2 s(p)}{\partial y^2} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} \frac{H_{i,l}}{\sigma_l^{\,2}}\left(-\frac{2}{\sigma_l^{\,2}}(p_y - c_{i,l_y})^2 + 1\right) \quad (13b)$$

$$\frac{\partial^2 s(p)}{\partial x \partial y} = \sum_{l=1}^{L}\sum_{i=1}^{M_l} -\frac{2}{\sigma_l^{\,2}} H_{i,l}(p_x - c_{i,l_x})(p_y - c_{i,l_y}) \quad (13c)$$

The computation of the derivatives requires only very simple computation since the more expensive terms previously computed for the surface calculation can be reused.

Besides, as most of the Taylor expansions used in the surface height estimation (Eq. 7) are computed along the axis, they do not require all derivatives. This contributes to make the surface estimation more preferable than its direct computation.

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," *Proc. ACM SIGGRAPH 2000*, 2000.

[2] M. Proesmans, L. Van Gool, and F. Defort, "Reading Between the Lines – A Method for Extracting Dynamic 3D with Texture," *Proc. ICCV 1998*, 1998.

[3] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, "3D real-time model acquisition," *Proc. ACM SIGGRAPH 2002*, 2002, pp. 438-446.

[4] N.A. Borghese and S. Ferrari, "A portable modular system for automatic acquisition soft 3D objects," IEEE Trans. on Instrumentation and Measurements, vol. 49, 2000, pp. 1128-1136.

[5] N.A. Borghese, M. Maggioni, and S. Ferrari, "Multi-Scale Approximation with Hierarchical Radial Basis Functions Networks," IEEE Trans. on Neural Networks, in press.

[6] S.K. Nayar, M. Watanabe, and M. Noguchi, "Real-Time Focus Range Sensor," *Trans. on PAMI*, vol. 18, No. 12, 1996.

[7] N. D'Apuzzo, "Modeling human faces with multi-image photogrammetry," *SPIE Proc. 3-Dimensional Image Capture and Applications, V*, vol. 4661, 2002, pp. 191-197.

[8] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image-Based Visual Hulls," *Proc. ACM SIGGRAPH 2000*, 2000.

[9] R. Heckbert and M. Garland, *Survey of Polygonal Surface Simplification Algorithms*, Technical Report, Carnegie Mellon University, 1997.

[10] F. Girosi, M. Jones and T. Poggio, "Regularization theory and Neural Networks Architectures," *Neural Computation* 7, 1995, pp. 219-269.

[11] N.A. Borghese, S. Ferrari, and V. Piuri, "Real-time surface reconstruction through HRBF networks," *Invited paper, Proc. of IEEE International Workshop on Haptic Virtual Environments and Their Applications HAVE 2002*, 2002, pp. 19 –24.