

3D Surface Reconstruction from Scattered Data Using Moving Least Square Method

Soon-Jeong Ahn¹, Jaechil Yoo², Byung-Gook Lee³, and Joon-Jae Lee³

¹ Research Center for Advanced Science and Technology, Dongseo University,
Busan, 617-716, South Korea

² Department of Mathematics, Donggeui University,
Busan, 995, South Korea

³ Department of Computer Engineering, Dongseo University,
Busan, 617-716, South Korea

{sjahn, lbj, jjlee}@dongseo.ac.kr, yoo@deu.ac.kr

Abstract. This paper presents an efficient implementation of moving least square(MLS) approximation for 3D surface reconstruction. The smoothness of the MLS is mainly determined by the weight function where its support greatly affects the accuracy as well as the computational time in the mixed dense and scattered data. In a point-set, possibly acquired from a 3D scanning device, it is important to determine the support of the weight function adaptively depending on the distribution and shape of the given scatter data. Particular in case of face data including the very smooth parts, detail parts and some missing parts of hair due to low reflectance, preserving some details while filling the missing parts smoothly is needed. Therefore we present a fast algorithm to estimate the support parameter adaptively by a raster scan method from the quantized integer array of the given data. Some experimental results show that it guarantees the high accuracy and works to fill the missing parts very well.

1 Introduction to MLS Approximation

There are many methods to reconstruct the continuous 3D surface from discrete scattered data. The moving least square(MLS) method is introduced to interpolate the irregularly spaced data. The relationship between MLS and G.Backus and F. Gilbert [2] theory was found by Abramovici [1] for Shepard's method and for the general case by Bos and Salkauskas [3]. For scattered data $X = \{x_i\}_{i=1}^n$ in \mathbb{R}^d and data values $\{f(x_i)\}_{i=1}^n$, the MLS approximation of order m at a point $x \in \Omega \subset \mathbb{R}^d$ is the value $p^* \in \Pi_m$ is minimizing, among all $p \in \Pi_m$, the weighted least-square error

$$\sum_{i=1}^n (p(x_i) - f(x_i))^2 \theta(\|x - x_i\|), \quad (1)$$

where θ is a non-negative weight function and $\|\cdot\|$ is the Euclidian distance in \mathbb{R}^d . To get the local approximation $\theta(r)$ must be fast decreasing as $r \rightarrow \infty$. So D. Levin [4] suggested the weight function for approximation

$$\eta(\|x_i - x\|) = \exp\left(\frac{\|x_i - x\|^2}{h^2}\right) \tag{2}$$

and prove that (1) is equivalent to find the coefficient vector $\bar{a} = \{a_1, a_2, \dots, a_n\}$ for approximation $\hat{f}(x) = \sum_{i=1}^I a_i f(x_i)$ by minimizing the quadratic form

$$Q = \sum_{i=1}^n \eta(\|x_i - x\|) a_i^2, \tag{3}$$

subject to the linear constraints

$$\sum_{i=1}^n a_i p_j(x_i) = p_j(x), \quad j = 1, \dots, J = \binom{d+m}{m} \tag{4}$$

with $\eta(\|x_i - x\|) = \theta(\|x_i - x\|)^{-1}$ in (1). The coefficient vector \bar{a} is determined by

$$\bar{a} = D^{-1} E (E^t D^{-1} E)^{-1} \bar{c}, \tag{5}$$

where $D = 2\text{Diag}\{\eta(\|x_1 - x\|), \dots, \eta(\|x_n - x\|)\}$, $E_{ij} = p_j(x_i)$ and $\bar{c} = (p_1(x), \dots, p_J(x))^t$. Here h is an average distance between the data points and $p_j, j = 1, \dots, J$ are the fundamental polynomials in Π_m . This uses the whole given data, so that it takes too long time to process with the large numbers of data.

D. Levin and H.Wendland [5] suggested the parameter s to choose local data and keep the high approximation order but it is very slow and fit to real data because of the fixed

$$h = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|, \tag{6}$$

so that we suggest the new algorithm to determine the local parameter s and take h adaptively according to data in section 2. By calculating RMS error for 5 smooth test functions and comparing the results with Levin method, we demonstrate the accuracy of our method. The results show that the adaptiveness of h is very important and it works well for filling hole case in section 3.

2 A Local and Fast Algorithm

2.1 Fixed h Local MLS Algorithm

To generate C^2 surface, we let $m = 3$ and $d = 2$ which mean the degree of approximant and the dimension of domain, respectively. Let $P = \{(x_i, y_i, z_i) \in \mathbb{R}^3\}_{i=1}^n$ be the set of scattered sample data and $\Omega = \Omega_x \times \Omega_y$ be the rectangular domain which contains the given a set of its projected points to xy -plane. By dividing $\Omega_x = [a, b]$ and $\Omega_y = [c, d]$ uniformly with respect to any fixed resolution (n_x, n_y) , we get the evaluation points $W = \{\omega_{ij} \in \mathbb{R}^2 | 0 \leq i \leq n_x, 0 \leq j \leq n_y\}$, where $d_x = (b - a)/n_x$, $d_y = (d - c)/n_y$. To determine h , we do not calculate the distance between evaluation point and data points unlike the previous methods.

This is achieved by mapping sample data to grid point and calculating their distribution using simple raster scan method. When the sample data is mapped into grid point, one more points can be mapped into the same grid point. In this case we take their average value for each x, y, z component as representative and index it. For each $\omega \in W$, we can find the numbers of grids from ω_{ij} to nonzero grid point along right, left, up and down direction, denoting it by $q_\omega^r, q_\omega^l, q_\omega^u, q_\omega^d$, respectively. And then we set $q_\omega = \frac{1}{4}(q_\omega^r + q_\omega^l + q_\omega^u + q_\omega^d)$. By taking $q = \max_{\omega \in W}(q_\omega)$, we can set

$$h = q \cdot \max(d_x, d_y) \tag{7}$$

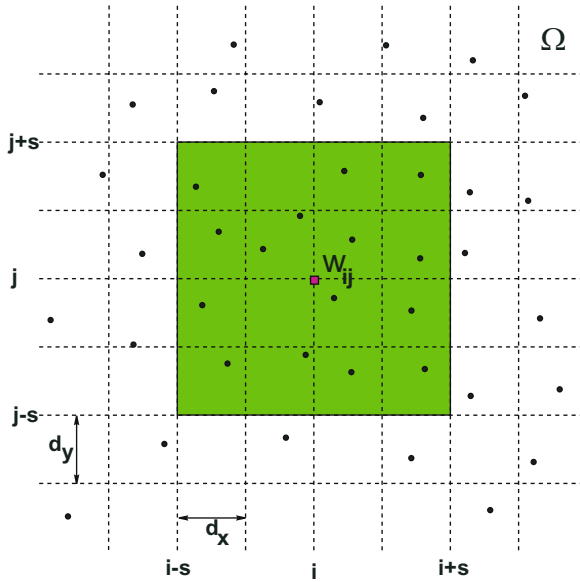


Fig. 1. Domain Ω and local extracted sample data domain

Next, sample data is chosen locally within window of size $2s \times 2s$, where

$$s = \lceil \frac{3}{\sqrt{2}} \cdot q \rceil, \tag{8}$$

centered at each evaluation point ω and $\lceil \cdot \rceil$ denotes the Gauss's symbol. This comes from the property of the weight function related to Gaussian function and the relation between h and q . Actually the standard deviation σ in Gaussian function is related to h , like $2\sigma^2 = h^2$. Because data within 6σ is contributed to the accuracy of the approximated function, we can get the above equation (8). Under these initial conditions, we extract the local subdata set consisted with the above representative data. Since the wanted degree of approximated

function is 3, we must increase the magnitude of s by one until the number of data in subdata set is greater than 10. Using this the final local subdata, we calculate the coefficient vector.

2.2 Adaptive h Local MLS for Data with Holes

If the sample data have big holes, the above fixed h algorithm is unsuitable to reconstruct the original image as shown in Fig. 4 (b),(c). Larger the size of holes is, bigger h is taken. Then the dense part, like nose, eyes and lips, are getting to lose their characteristic property. Therefore, we do not use global q but use local q_ω for each $\omega \in W$. Then the adaptive h_ω

$$h_\omega = q_\omega \cdot \max(d_x, d_y) \tag{9}$$

and the initial adaptive s_ω

$$s_\omega = \lceil \frac{3}{\sqrt{2}} \cdot q_\omega \rceil. \tag{10}$$

Under the adaptive initial conditions, we follow the same process as fixed algorithm. Refer the experimental results in section 3.4.

3 Experimental Results and Conclusions

3.1 Reconstruction Accuracy

In this section we demonstrate the accuracy of our proposed algorithm by the use of 5 test function as follows:

$$\begin{aligned} g_1(x, y) &= 0.75 \exp[-((9x - 2)^2 + (9y - 2)^2)/4] + 0.5 \exp[-((9x - 7)^2 + (9y - 3)^2)/4] \\ &\quad + 0.75 \exp[-(9x - 2)^2/49 - (9y - 2)^2/10] - 0.2 \exp[-(9x - 4)^2 - (9y - 2)^2] \\ g_2(x, y) &= (\tanh(9 - 9x - 9y) + 1)/9 \\ g_3(x, y) &= (1.25 + \cos(5.4y))/(6 + 6(3x - 1)^2) \\ g_4(x, y) &= (\exp[(-81/4)((x - 0.5)^2 + (y - 0.5)^2)])/3 \\ g_5(x, y) &= (\sqrt{64 - 81((x - 0.5)^2 + (y - 0.5)^2)})/9 - 0.5 \end{aligned}$$

where x, y are in the domain $[0, 1]$. We perform experiments with some sample data generated randomly in $[0, 1]$. Here M100 and M500 means 100 and 500 sample data with 49 and 625 uniform data respectively, while others are randomly sampled. On the other hand R500 is 500 totally random data. Fig. 2 is the feature of g_3 and its approximated image. The points on the surface are the sample data for M100 case. For each test function g_i , we can find the accuracy of the approximation f by comparing the normalized RMS(root mean square) error which is divided the RMS error by the difference if maximum and minimum values of g_i between the function values on a dense grid. That is,

$$\text{RMS} = \sqrt{\frac{\sum_{i=0}^{n_x} \sum_{j=0}^{n_y} (g_i(x_i, y_i) - f(x_i, y_j))^2}{(n_x + 1)(n_y + 1)}}$$

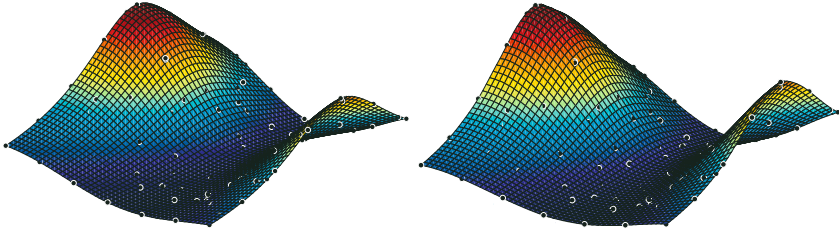


Fig. 2. Feature of g_3 (left) and its approximated image for M100(right)

Table 1. Comparison of RMS error between WD and LD for M100, M500 and R500

M100	g_1	g_2	g_3	g_4	g_5
WD	.01076	.00694	.00070	.00199	.00020
LD	.01085	.00699	.00070	.00201	.00020

M500	g_1	g_2	g_3	g_4	g_5
WD	.00079	.00047	.00004	.00013	.00001
LD	.00089	.00045	.00005	.00018	.00001

R500	g_1	g_2	g_3	g_4	g_5
WD	.00080	.00143	.00009	.00020	.00002
LD	.00089	.00152	.00009	.00022	.00004

where $x_i = i/n_x$, $y = j/n_y$ and $n_x = n_y = 50$. Under the same h , we compare the RMS values for each case. Table 1 shows that the fact that MLS approximation theory does not need to have the whole data(WD) but is enough only local data(LD).

3.2 Time Complexity

Table 2 shows that our proposed algorithms makes the processing time faster about 5 times than the use of WD and it is more efficient if the distribution of sample data is more dense and its number is larger.

Table 2. Time complexity between whole data and local data

Time (sec)	M100	M500
WD	10.64	46.22
LD	7.32	9.42

3.3 Filling Holes with Adaptive h Local MLS

Firstly we have experimented for g with additive random noise data of magnitude 20 and some data in original data set is removed to have one hole of the cross shape.

$$g(x, y) = \begin{cases} 50((\sin 2\pi x) + \sin(2\pi y)) & \text{if } x < y, \\ 32((\cos 2\pi x) - \sin(2\pi y)) & \text{o.w.} \end{cases}$$

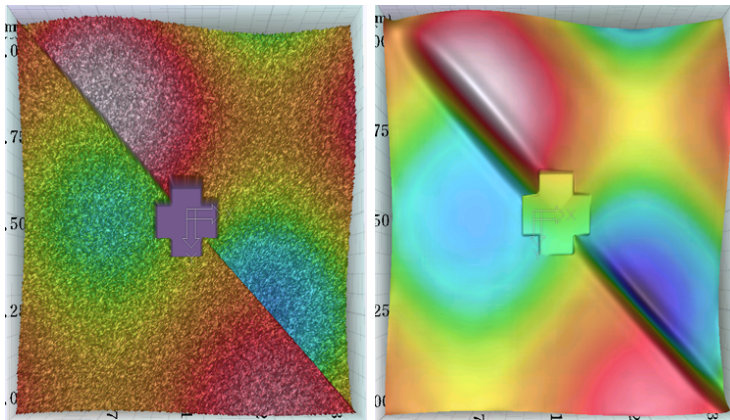


Fig. 3. g function with cross hole by generating noise(left) and its filling hole image(right)

Although the given surface is smooth, if it has missing data with the big sized hole, then fixed h local MLS algorithm is unsuitable. Next, we experiment with real face range data having many noisy data and very big holes. If we take the large fixed h from the largest hole, it can work well for filling holes but it cannot preserve the details. but if we use the adaptive h local algorithm, then we get the nice result like Fig. 4 (d).

3.4 Conclusions and Further Studies

Some experimental results announce us that the proposed algorithm is very simple and efficient with reasonable accuracy for real range data approximation. By introducing h adaptively in every evaluation points, we get the smooth surface that preserves the detail parts such as nose, eyes and lips and fills holes nicely. However, this algorithm occurs some discontinuity on the boundary of each hole due to abrupt change of h near it. So we are researching about multilevel method for getting more smooth results. Some experimental results give us clues that it is very reasonable guess.

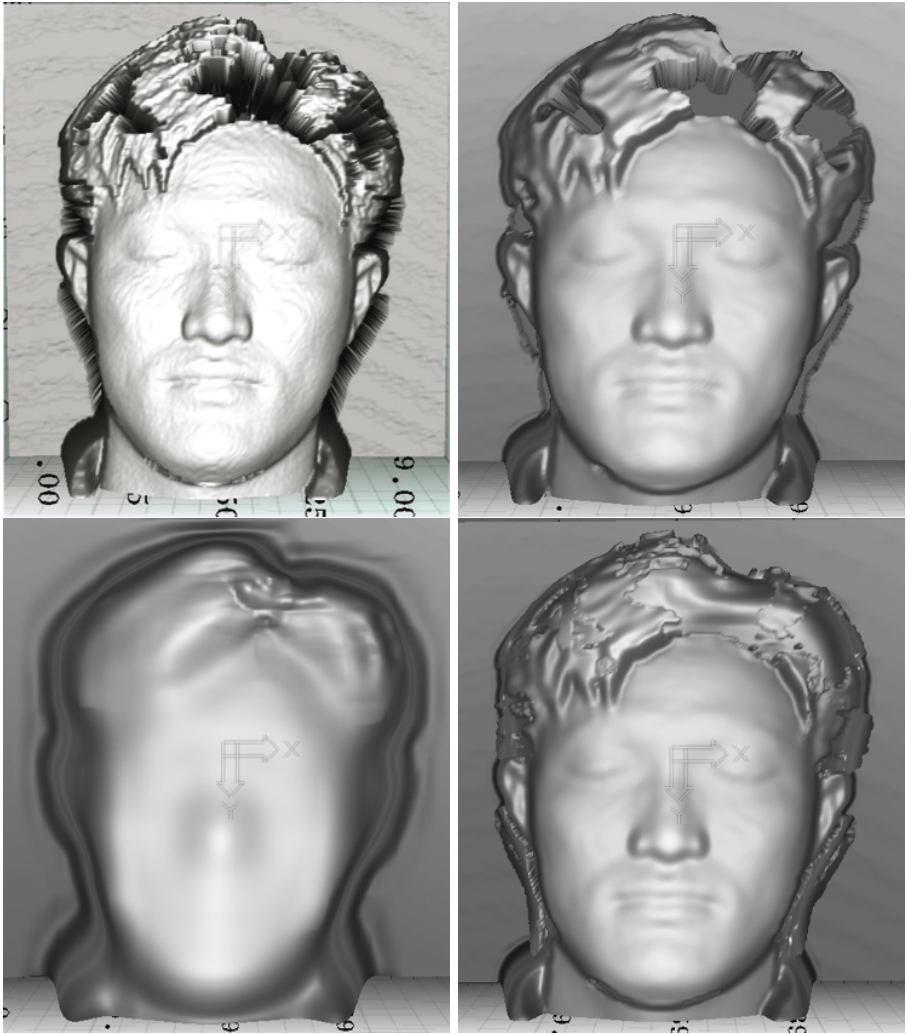


Fig. 4. (a) original face range image, (b) approximated image with fixed $h = 3$, (c) $h = 15$ (d) approximated image with adaptive h

Acknowledgements

1. This work was supported by grant No.R01-2004-000-10851-0,R05-2004-000-10968-0 from Ministry of Science and Technology
2. This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korea Government

References

1. F. Abramovici, The Shepard interpolation as the best average of a set of data, Technical Report, Tel-Aviv University, 1984.
2. G. Backus and F. Gilbert, The resolving power of gross earth data, *Geophys.J.R. Astr. Soc.* **16** 169-205, 1968.
3. L. Bos and K. Salkauskas, Moving least-squares are Backus-Gilbert optimal, *J. Approx. Theory* **59**(3) 267-275. 1989.
4. D. Levin, The approximation power of moving least-squares, *Mathematics of computation*, vol 67, **224**, 1517-1531, 1998.
5. Wendland, H., Local polynomial reproduction and moving least squares approximation. *IMA J. Numer. Anal.* **21**, 285-300. 2001.