Voting-Based Decision Framework for Optimum Selection of Inerpolation Technique for 3D Rendering Applications

Syed Altaf Ganihar, Shreyas Joshi, Nishant Patil, Uma Mudenagudi, Manish Okade
Department of ECE
BVBCET Hubli-580031
Email: <altafganihar, shreyasvj25, nishant.apatil3, uma.mudenagudi, manishokade>@gmail.com

***Abstract*—In this paper a novel decision framework for efficient selection of interpolation curve is proposed. The proposed framework is based on distance minimization for 3D rendering applications. The point clouds obtained from low resolution 3d scanners like Microsoft's Kinect or from sparse reconstruction algorithms usually fail to provide accurate information about the surface either due to occlusions during the scanning process or inability of the scanner to generate a dense model of the surface. The proposed decision framework selects the best interpolation technique on a local basis utilizing the voting parameters obtained from the original point cloud. This framework enables us to obtain the comparatively best fit interpolation curve for upsampling due to the decisive feature of the framework.**

I. Introduction

In this paper we propose a novel decision framework for efficient selection of interpolation technique based on distance voting for 3D rendering applications. The point cloud necessary for 3D reconstruction could either be obtained through a 3D scanning device like Microsoft's Kinect or through point cloud reconstructed by sparse model obtained from a bundler. The low resolution 3D scanners provide information about the shape of an object's surface by generating a point cloud of the object's surface using inexpensive rangefinding techniques. However, due to their low resolution these systems fail to capture all the details embodied in an object's surface resulting in an incomplete surface. In addition, due to the presence of occlusions during the scanning process the point cloud data obtained would be rendered as inaccurate. In some applications an incomplete point cloud data would serve the purpose where accurate geometrical properties and surface details of the object are not required. But in most of the applications a dense point cloud, accurately representing the object's geometric surface is required. Also the sparse reconstruction algorithms like bundler[1] which generate structure from motion fail to accurately portray the objects surface details. The point clouds obtained through such algorithms are less dense and hence surface reconstruction[2] from such point clouds would produce inaccurate results. Surface reconstruction from point clouds results in the production of accurate surfaces provided the point cloud rendered is dense enough. Hence there is a great need for the upsampling process to be implemented before the surface reconstruction process.

There have been many upsampling methods proposed like cubic spline interpolation, quadratic spline interpolation, polynomial interpolation etc. for the upsampling process[8][7][3][4][5][6]. Most of the upsampling methods implement interpolation algorithms to generate a dense data set by utilising the original point cloud data set. . All the non-linear interpolation techniques have their own set of advantages and disadvantages as the surfaces geometry might synchronize with the assumed interpolation curve or might contrast with it. Therefore producing results which are dependent on the surface properties of the point cloud is a requirement. For example consider the quadratic and cubic interpolation techniques. Quadratic interpolation would produce accurate results if the surface whose point cloud is obtained has a quadratic symmetry viz., a concave or a convex surface. On the other hand, if we are trying to upsample a point cloud which has a cubic symmetry embodied in it, the results produced by the quadratic interpolation technique would not be plausible. Here a cubic interpolation would be required. Similarly, for a cubic interpolation the results would depend on the surface properties of the object whose point cloud is considered. The geometric properties of a surface do not follow a particular symmetry throughout the surface, they usually comprise of variations in the geometric properties which vary from point to point and are even abrupt in certain cases. Hence a particular interpolation technique would fail to produce reasonable results due to the variations in the geometric properties of the surface under consideration. Keeping these objectives in perspective we make the following contributions:

1) A decision framework which decides the best interpolation technique to be selected based on the voting parameters computed on the basis of the distribution of the point cloud data.
2) A framework for fusing different interpolation techniques to upsample a given point cloud.

Experimental results are carried out using two interpolation techniques viz., quadratic spline interpolation and cubic spline interpolation technique to demonstrate the usefulness of such a decision framework for 3D point cloud data.

Rest of the paper is organized as follows. Section II gives the motivation for the voting-based decision fusion framework along with the explanation for computing the distance-based voting parameters invlolved in making this decision. Section III reviews the interpolation techniques employed for upsampling. Section IV discusses the proposed decision framework. Section V discusses the results obtained using two interpolation techniques viz., quadratic spline interpolation and cubic spline interpolation for the decision fusion framework using several distance-based voting parameters. Finally section VI gives the conclusions along with the scope for future work.

II. Motivation

This section gives the motivation of the proposed framework for fusion of different interpolation algorithms to obtain a refined and upsampled version of a point cloud. The input point cloud is obtained either through a 3D scanning device like Microsoft's Kinect or by running the sparse reconstruction algorithms like a Bundler which computes the structure from motion. In either of the cases the obtained point cloud does not accurately portray the inherent properties of the scanned surface thereby warranting the need for upsampling algorithms which enhances the density of the point cloud. Most of the upsampling algorithms are based on interpolation techniques whose performance depends upon the geometrical properties of the scanned surface. Hence a single interpolation technique will not produce plausible results as the scanned surfaces will not have even geometrical properties throughout the surface. This formed the motivation for a decision framework which decides on the best interpolation technique to be employed in a given local region of the surface, which is our novel contribution. This proposed decision framework relies on distance-based voting parameters which vote for the best fitting interpolated curve. The best fitting interpolated curve selected by the decision framework is used to interpolate the point cloud on a local region. The interpolation is carried out over the entire point cloud using the best fitting interpolated curve selected on a local basis. The details of the interpolation techniques and the proposed decision framework with respect to 3D rendering applications is described as under.

III. Review of Interpolation Techniques

The obtained point cloud is stored in a kd tree[9]. Kd tree is a k-dimensional binary search tree which stores the given input data in a tree data structure. Kd trees are very useful for range searching and nearest neighbour searching techniques. In the current case the point cloud is three dimensional, the value of k would be three. The nearest neighbour search technique is invoked for every point in the point cloud to obtain a given number of nearest neighbours to the referenced point in ascending order. In our case we obtain four nearest neighbours to every point in the point cloud. These four points are utilised to fit the interpolation curves using appropriate interpolation techniques based on the proposed decision framework. Quadratic spline interpolation and cubic spline interpolation are choosen for demonstrating the usefulness of having a decision framework instead of a single interpolation technique which is adopted currently. A brief overview of the qubic spline interpolation and cubic spline interpolation is given as under.

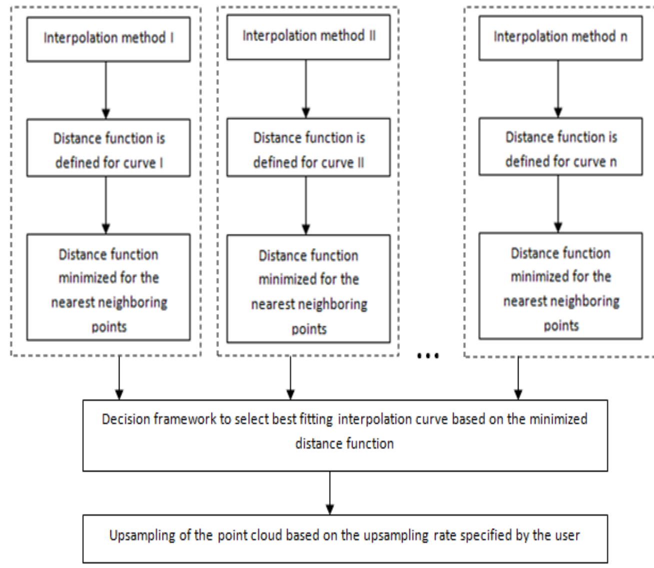


Fig. 1. Proposed Decision Framework

Quadratic spline interpolation assumes the surface point cloud to possess a quadratic symmetry. It fits a quadratic polynomial between every pair of points in a given set of n points. The quadratic spline between these n points has a great degree of smoothness due to the presence of n control points. In our case we carry out quadratic interpolation on a set of four points $P = \{p_0(x_0, y_0), p_1(x_1, y_1), p_2(x_2, y_2), p_3(x_3, y_3)\}$ to fit a quadratic spline curve between p_1 and p_2 as given below,

$$x(t) = a_1t^2 + b_1t + c_1 \quad (1)$$

$$y(t) = a_2t^2 + b_2t + c_2 \quad (2)$$

$$z(t) = a_3t^2 + b_3t + c_3 \quad (3)$$

where $a_1 = 2x_0 - 5x_1 + 4x_2 - x_3$, $b_1 = x_2 - x_0$, $c_1 = 2x_1$, $a_2 = 2y_0 - 5y_1 + 4y_2 - y_3$, $b_2 = y_2 - y_0$, $c_2 = 2y_1$, $a_3 = 2z_0 - 5z_1 + 4z_2 - z_3$, $b_3 = y_2 - y_0$, $c_3 = 2z_1$ and t is the parameter over which the curve is defined.

Similarly we carry out cubic spline interpolation to fit a cubic spline curve between p_1 and p_2 as given below,

$$x(t) = a_1t^3 + b_1t^2 + c_1t + d_1 \quad (4)$$

$$y(t) = a_2t^3 + b_2t^2 + c_2t + d_2 \quad (5)$$

$$z(t) = a_3t^3 + b_3t^2 + c_3t + d_3 \quad (6)$$

where $a_1 = x_3 - 3x_2 + 3x_1 - x_0$, $b_1 = 2x_0 - 5x_1 + 4x_2 - x_3$, $c_1 = x_2 - x_0$, $d_1 = 2x_1$, $a_2 = y_3 - 3y_2 + 3y_1 - y_0$, $b_2 = 2y_0 - 5y_1 + 4y_2 - y_3$, $c_2 = y_2 - y_0$, $d_2 = 2y_1$, $a_3 = z_3 - 3z_2 + 3z_1 - z_0$, $b_3 = 2z_0 - 5z_1 + 4z_2 - z_3$, $c_3 = z_2 - z_0$, $d_3 = 2z_1$.

These resulting interpolated spline curves i.e., quadratic or

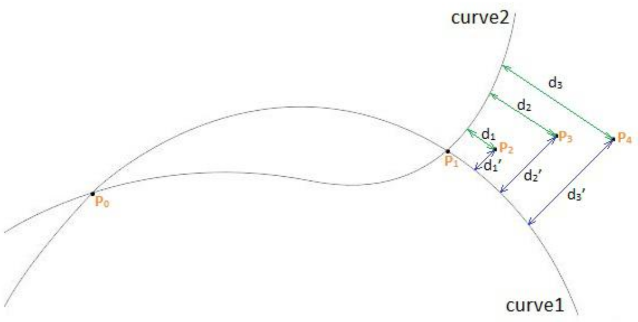


Fig. 2. Curve fitting based on distance minimization

cubic can be used to upsample the given point cloud as shown in Fig 2. But as already pointed out the performance of the individual interpolation techniques would depend upon the geometric properties of the surface. Hence the results produced due to them would not be plausible if the surface properties are in contrast with interpolation technique employed. In the next subsection we give the overview of the proposed decision framework which decides over the endorsement of a particular interpolation technique based on a distance-based voting method.

IV. Proposed Decision Framework

The proposed decision framework is as show in Fig 1. The obtained interpolated curves are passed through the distance-based voting method which decides on the interpolation technique to be employed for upsampling. Once the interpolation curves for a given pair of points are obtained, another set of three points are selected to vote for the curves. The next set of three points $Q = \{p_4(x_0, y_0), p_5(x_1, y_1), p_6(x_2, y_2)\}$ are selected based on the nearest neighbour search technique over the k-d tree employed to store the point cloud.

A distance function is defined from a random point (x, y) on the curve to another point $(x_i, y_i) \in Q$. The distance function can be defined either using Euclidean distance function, L_1 norm, city block distance formula or any other distance function. In our case we have generated the results by using Euclidean distance function. Using Euclidean distance function, the squared distance function for the quadratic spline interpolation curve is defined as,

$$B_{EQ}(x_i, y_i) = (x_q - x_i)^2 + (y_q - y_i)^2 + (z_q - z_i)^2 \quad (7)$$

where $x_q = a_1t^2 + b_1t + c_1$, $y_q = a_2t^2 + b_2t + c_2$, $z_q = a_3t^2 + b_3t + c_3$.

Similarly the Euclidean distance function defined for the cubic spline curve is as follows,

$$B_{EC}(x_i, y_i) = (x_c - x_i)^2 + (y_c - y_i)^2 + (z_c - z_i)^2 \quad (8)$$

where $x_c = a_1t^3 + b_1t^2 + c_1t + d_1$, $y_c = a_2t^3 + b_2t^2 + c_2t + d_2$, $z_c = a_3t^3 + b_3t^2 + c_3t + d_3$.

The distance function computes the distance of the point (x_q, y_q) from any random point on the interpolation curve to the point (x_i, y_i) taken into consideration for voting as shown in Fig 2. The distance function is then minimized with respect to the parameter t to obtain the minimum distance of the point from the interpolation curve by taking its first derivative to obtain a fifth degree polynomial for the cubic spline interpolation curve using,

$$B'_{EC}(x_i, y_i) = At^5 + Bt^4 + Ct^3 + Dt^2 + Et^1 + F = 0 \quad (9)$$

where $A = 3a_1^2 + 3a_2^2 + 3a_3^2$, $B = 5a_1b_1 + 5a_2b_2 + 5a_3b_3$, $C = 2b_1^2 + 4a_1c_1 + 2b_2^2 + 4a_2c_2 + 2b_3^2 + 4a_3c_3$, $D = 3b_1c_1 + 3a_1d_1 - 3a_1x_i + 3b_2c_2 + 3a_2d_2 - 3a_2y_i + 3b_3c_3 + 3a_3d_3 - 3a_3z_i$, $E = c_1^2 + 2b_1d_1 - 2b_1x_i + c_2^2 + 2b_2d_2 - 2b_2y_i + c_3^2 + 2b_3d_3 - 2b_3z_i$, $F = c_1d_1 - c_1x_i + c_2d_2 - c_2y_i + c_3d_3 - c_3z_i$

Similarly, for the quadratic spline interpolation a 3rd degree polynomial is obtained using,

$$B'_{EQ}(x_i, y_i) = At^3 + Bt^2 + Ct + D = 0 \quad (10)$$

where $A = 2a_1^2 + 2a_2^2 + 2a_3^2$, $B = 3a_1b_1 + 3a_2b_2 + 3a_3b_3$, $C = b_1^2 + 2a_1c_1 - 2a_1x_i + b_2^2 + 2a_2c_2 - 2a_2y_i + b_3^2 + 2a_3c_3 - 2a_3z_i$, $D = b_1c_1 - b_1x_i + b_2c_2 - b_2y_i + b_3c_3 - b_3z_i$.

Equations (9) and (10) are solved to obtain the values of the parameter t for which the distance function is minimized by equating it to zero. The minimization of the function is checked by examining the second derivatives of the distance functions (7) and (8). The distance function is computed for all the points $(x_i, y_i) \in Q$ by minimizing the function for every point. The distance function values for the quadratic spline interpolation and cubic spline interpolation are compared and a vote V is registered by the point towards a particular interpolation technique using the relation,

$$V(x_i, y_i) = \begin{cases} 1 & \text{if } (B_{EQ}(x_i, y_i) < B_{EC}(x_i, y_i)) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The interpolation technique to be employed is then selected based upon the number of votes registered for a particular interpolation technique. Based on this selection the appropriate interpolation technique is choosen for upsampling the point cloud in a local region. The number of points to be interpolated is determined based on the upsampling rate specified by the user. The upsampling rate represents the number of points to be added for every pair of points. The proposed decision framework is applied for the entire point cloud.

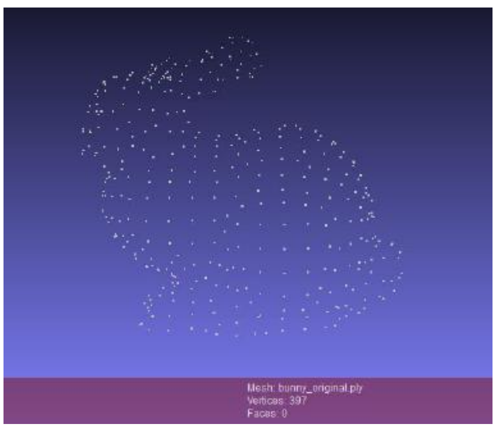
V. Results and Discussions

The effectiveness of the proposed framework is demonstrated using bunny point cloud and buddha point cloud. The original bunny point cloud and buddha point cloud are shown in Fig 3. Point cloud library on Intel(R) core i7 processor with 8 GB RAM and 2GB Nvidia GeForce 755M graphics is used to test the proposed decision framework. In addition

TABLE I
THE CURVATURE VALUES FOR RANDOMLY SELECTED POINTS IN THE POINT CLOUD FOR QUADRATIC SPLINE INTERPOLATION, CUBIC SPLINE INTERPOLATION AND THE PROPOSED UPSAMPLING METHOD WITH PERCENTAGE DEVIATIONS FROM THE ORIGINAL POINT CLOUD

p_x	p_y	p_z	$R_{original}$	R_{cubic} *and* % deviation	$R_{quadratic}$ *and* % deviation	R_{fusion} *and* % deviation
-0.06145	**0.05916**	**0.01873**	**0.05239**	**0.03354** (35.98 %)	**0.045562** (13.03 %)	**0.0514** (1.73 %)
-0.05615	**0.050619**	**0.00929**	**0.17887**	**0.15870** (11.27 %)	**0.17947** (0.3 %)	**0.17917** (0.16 %)
-0.01904	**0.06773**	**0.05238**	**0.04027**	**0.03602** (10.55 %)	**0.04142** (2.85 %)	**0.03922** (2.6 %)
-0.02319	**0.15882**	**-0.00479**	**0.01517**	**0.01039** (31.5 %)	**0.01772** (16.8 %)	**0.01447** (4.61 %)

we also demonstrate the results using Microsoft Kinect's input obtained for Hampi's monuments.



(a) Bunny Original Point cloud

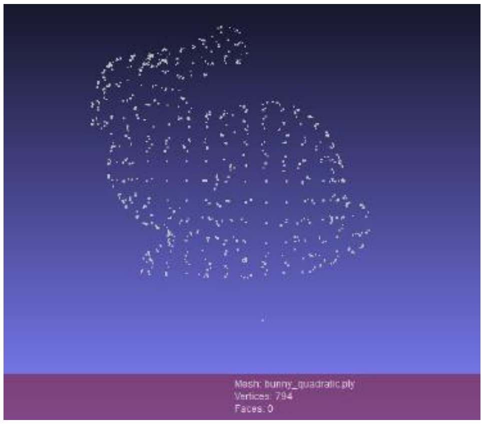
(b) Buddha original point cloud

Fig. 3. Orginal bunny and buddha point clouds

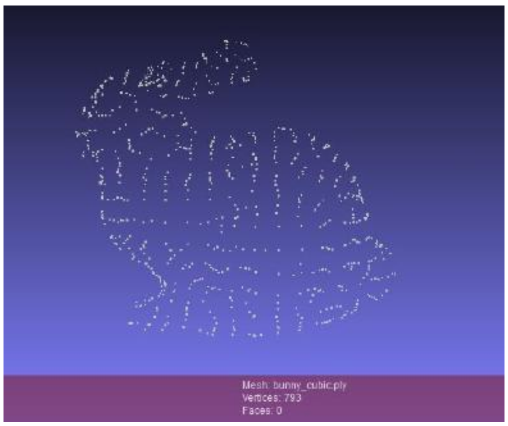
Fig 4 illustrates the implementation of quadratic and cubic spline interpolation based upsampling methods for the bunny point cloud which comprises of 397 points. The resulting point cloud comprises of 794 and 793 number of points for quadratic and cubic spline interpolations respectively. Fig 5 illustrates the implementation of the proposed framework with upsampling rate 2.0 which comprises of 794 points.

The curvature values are calculated at every point using Moving Least squares method[10][2] and they are as shown in Table I. As observed from the table, the curvature values in case of cubic and quadratic spline interpolation have a large deviation from the surface curvature of the original point cloud due to the deviations in the curvature values. This reveals that the interpolation curves would fit better if the scanned surface's geometrical properties would match with the geometrical assumptions made by the interpolation technique. As in case of quadratic spline interpolation we see that it assumes the surface to possess quadratic symmetry. Hence for surfaces possessing quadratic symmetry the results produced by quadratic spline interpolation would be plaussible as it would uphold the geometric property of the original surface. Similarly for cubic spline interpolation the results would be more plausiible if the scanned surface possesses cubic symmetry. If the geometric properties contrast with the interpolation technique's assumed symmetry then their would be deviations in the properties of the surface generated by the upsampling process. But if we fuse both the methods by the proposed decision framework we would be able to produce results which shall uphold the geometric properties inherent in the scanned surface. As we see from Table I the deviations in the curvature for proposed algorithm are quite less compared to cubic and quadratic spline interpolation techniques. Quadratic spline interpolation performs better for the bunny point cloud as we can clearly observe that it possesses quadradic symmetry over its surface due to the precense of concave parts on its surface.

The proposed method is also demonstrated for the buddha



(a) Quadratic Spline Interpolation



(b) Cubic Spline Interpolation

Fig. 4. Results for Quadratic and cubic spline interpolation based upsampling for bunny point cloud

Fig. 5. Upsampling of the bunny model using proposed decision framework

point cloud and a hampi pillar model. The buddha point cloud comprises of 2591 points which after upsampling with an upsampling rate 2.0 produces 5181 points as shown in Fig 6. Fig 7 demonstrates the experimental results for the hampi pillar model obtained using kinect camera comprising of 3451 points which after upsampling. The upsampled point cloud comprises of 6902 points. As observed from both the

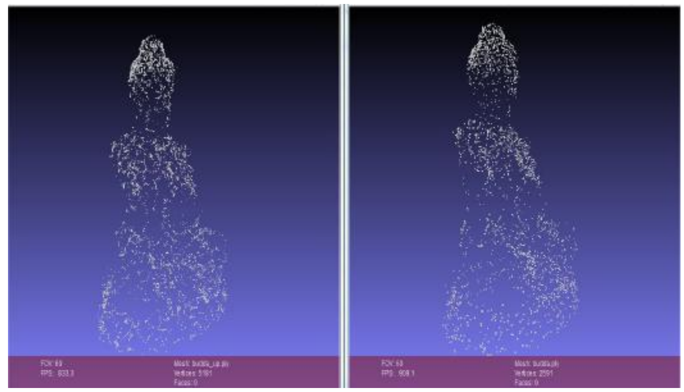


Fig. 6. Left:Upsampling of buddha model using proposed decision framework Right: original Buddha model

models the geometrical properties of the upsampled surface are enhanced when compared to the original point clouds.

The surface reconstruction for the buddha models is carried out using poisson surface reconstruction algorithm. The models after reconstruction are mapped with the appropriate textures and are as shown in Fig 8. The surface reconstruction is implemented with poisson recontruction algorithm with the following parameter values $Octree\ depth = 12$, $Solver\ \ divide = 10$, $Samplesper\ \ node = 2$, $Surface\ offsetting = 1$. As observed from the Fig 8 the surface reconstruction has produced better results for the upsampled point cloud when compared to the original point cloud. The upsampled surface contains greater amount of geometrical information and hence produces better results for 3D rendering applications.

VI. Conclusions

In this paper we have proposed a novel decision framework for efficient selection of interpolation technique based on distance voting for 3D rendering applications. Experimental results are carried out using two interpolation techniques viz., quadratic spline interpolation and cubic spline interpolation technique to demonstrate the usefulness of such a decision framework for 3D point cloud data. In future we would like to enhance the decision framework for surface reconstruction, hole filling and dense reconstruction algorithms.

References

[1] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: exploring photo collections in 3d,” in *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2006*, 2006, pp. 835–846.

[2] I.-K. Lee and K.-J. Kim, “Shrinking: another method for surface reconstruction,” in *Geometric Modeling and Processing, 2004. Proceedings*, 2004, pp. 259–266.

[3] Z. Jia, Y.-J. Chang, T.-H. Lin, and T. Chen, “Dense interpolation of 3d points based on surface and color,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 869–872.

[4] Y. Ohtake, A. Belyaev, and H. P. Seidel, “A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions,” in *Shape Modeling International, 2003*, 2003, pp. 153–161.

[5] Y. Wang, Z. Zhang, and B. Guo, “3d image interpolation based on directional coherence,” in *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, 2001, pp. 195–202.

(a) Hampi pillar point cloud

(b) Hampi pillar upsampled point cloud

Fig. 7. Results of upsampling for the hampi pillar model

[6] M. Ozturk and T. Kayikcioglu, "A curve fitting method for point clouds based on local line projections," in *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, 2011, pp. 218–221.

[7] S. Dyer and J. Dyer, "Cubic-spline interpolation. 1," *Instrumentation Measurement Magazine, IEEE*, vol. 4, no. 1, pp. 44–46, 2001.

[8] F. Tianxiang and L. Hongxia, "Computer realization of quadratic spline interpolation," in *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, 2011, pp. 1447–1450.

[9] L. Qiang, H. Hao, W. Yongmin, L. Xu, and G. Liqian, "The kd-tree-based nearest-neighbor search algorithm in grid interpolation," in *Image Analysis and Signal Processing (IASP), 2012 International Conference on*, 2012, pp. 1–6.

[10] B. Mederos, L. Velho, and L. de Figueiredo, "Moving least squares multiresolution surface approximation," in *Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on*, 2003, pp. 19–26.

(a) Surface reconstruction of buddha model

(b) Surface reconstruction of upsampled buddha model

Fig. 8. Results of Surface Reconstruction for the Buddha models