3D Surface Reconstruction of Stone Tools by Using Four-Directional Measurement Machine

Enkhbayar Altantsetseg [1], Yuta Muraki [2], Fumito Chiba [3] and Kouichi Konno [2]

[1] Graduate School of Engineering, Iwate University, Japan
[2] Faculty of Engineering, Iwate University, Japan
[3] Laboratory for Archaeology and Geoinformatics, LANG Co., Ltd, Japan

***Abstract*—In this paper, we present a whole procedure for constructing 3D models of stone tools including scanning, data acquisition and surface reconstruction with hole-filling. The process of scanning hundreds or thousands of small objects is time consuming. Our original 3D laser scanner optimizes the scanning process and reduces time significantly by four directional scanning of many small objects simultaneously. To reconstruct surface of stone tools, the scanned point clouds are processed with a new triangulation method that preserves the properties of sharp edges. Our approach is based on a projection based method in which points are distinguished into neighboring layers with a point cloud slicing method to be individually reconstructed. In addition, we introduce a simple hole-filling algorithm for mesh completion of models. The main advantages of our approach are speed and efficiency for reconstruction of many small objects.**

***Index Terms*—3D laser scanning, cultural heritage, hole-filling, stone tools, surface reconstruction.**

I. INTRODUCTION

In recent years 3D digitizing technology is widely used for conserving and studying cultural heritages [1, 2]. Laser scanning technology allows us to reconstruct highly accurate 3D models of the cultural artifacts. Depending on the size of the artifact, a lot of different devices, techniques and methods are studied for 3D model reconstruction. Generally, objects can be classified into three types: large size objects such as historical buildings and big statues [3, 4], medium ones like medium statues and bigger tools [5, 6] and small ones like stone tools and jewelries [7, 8].

Reconstruction of 3D stone tools with using a laser scanner is one of the integral parts for the cultural heritage conservation. Stone tools are one of the most common evidence of the prehistoric human culture. In the cultural heritage conservation as well as in the study of the stone tools, it is an important issue to construct accurate 3D model replicas. The study of stone tools allows us to understand the cultural and technological development of our far ancestors.

In the lithic analysis, the surfaces of stone tools contain specific properties such as sharp edge, flake scar and percussion point [9]. Due to this, during reconstruction process of stone tools, it is very important to preserve their geometric features.

The most commercial scanners can measure few stone tools at the same time with the multi point data acquisition. Thus, the scanning process of hundreds or thousands of stone tools is time consuming.

On the other hand, surface reconstruction from a large amount of point data is also time consuming and many existing surface reconstruction algorithms have difficulty to construct sharp edges.

In this paper, we have studied a whole procedure for stone tool reconstruction including scanning process and implementation algorithm to construct triangular meshes and fill the holes. The main advantage of our approach is fast and efficient for surface reconstruction of many small objects. Besides, separating points into neighboring layers before surface reconstruction of sharp edged objects brings a good result. Additionally, we present a simple method for filling the holes on the surfaces. Our hole-filling algorithm runs fast and efficient although the shape becomes C^0 continuity.

II. RELATED WORK

There are a lot of algorithms for surface reconstruction from unorganized point clouds. These algorithms are classified into a lot of different ways based on their techniques and methods. For example, Fausto Bernardini [8] classified the algorithms into Delaunay-based methods, surface-based methods, volumetric- based methods and deformable surface methods.

Based on the characteristics of our point data we adopted the projection-based algorithm proposed by Gopi et al. [10, 11]. Gopi's algorithm is effective for a large amount of unorganized point data due to its memory efficiency and operation speed.

Gopi's algorithm is a single pass and an efficient algorithm to reconstruct 3D surface. It requires only a set of unorganized points and no additional information like normal vectors is needed. The algorithm has the following six main steps:

(1) Find k-nearest neighbors for every sample point.
(2) Estimate the normal vector for each sample point **p** with minimizing the variance of the dot product normal vector at **p** and the vectors from **p** to its k-nearest neighbors.
(3) Select candidate points that might be possible neighbors to

Manuscript Received on November 10, 2010.
Email: bayar@lk.cis.iwate-u.ac.jp

a vertex in the final triangulation. This selection process uses certain sampling criteria and it has two stages.

- Use the axis-aligned bounding box that returns all points inside the box. Depth pixel array is used for data structure of point pruning.
- Next, use sphere of influence with its center at point **p**. Radius of the sphere is obtained by $m * s$, where s is the distance between sample point **p** and its closest point, and m is a constant value. Points inside the sphere of influence are considered as possible Delaunay neighbors.

(4) Map all candidate points onto the tangent plane at sample point **p** as they are rotated about a well-defined axis on the tangent plane.

(5) Compute the 2D Delaunay neighbors of every sample point **p**.

(6) Construct final surface triangles using Delaunay neighbors for each sample point **p**.

Gopi's algorithm is suitable for noiseless data and assumed that the point data of an object is as follows [12]:

a. Point data should be locally uniform, which means that the distance ratio between the farthest and nearest neighbors of a sample in the given sampling of the object is less than a constant value.
b. The points on two neighboring layers should be distinguishable
c. The object should be smooth, which means that the normal deviation between the any two triangles incident on a vertex is less than 90 degrees.

In addition, there are some researches related to this algorithm. For example, Gopi's algorithm is accelerated by using a GPU parallel computing [13], while the algorithm of [14] is improved for non-uniform data by adding new points on a surface. Both of the algorithms of [13] and [14], however, are based on the similar assumption as Gopi's algorithm.

III. 3D POINT CLOUD DATA ACQUISITION AND PROCESSING

In our study we used point cloud data scanned by our original 3D laser scanner, especially designed for a large number of small objects such as stone tools. The surfaces of stone tools are very complicated, and sometimes occlusion appears on the surfaces. To prevent such problems, our original range scanner has four laser parts to measure objects in four directions simultaneously.

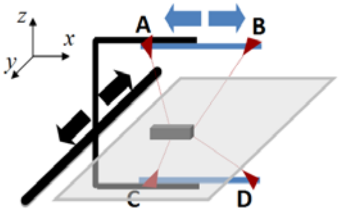

Fig. 1. There is Sensors of our scanner (A, B, C and D) are fixed on the movable robot arm. The robot arm move along the x and y axes.

Scanning process of a lot of small objects is very time consuming depending on accuracy and the number of objects.

Our scanner consists of three main parts as shown in Fig. 1:

(1) Four laser sensors
(2) Movable robot arm
(3) Glass board (1050 mm × 550 mm)

The glass board enables object scanning from four different viewpoints at the same time. Free and uniform movement along the x and y axes of the robot arm allows acquisition of uniform point data with high density. Sensors (A, B, C and D) shift together.

This chapter briefly describes major steps of the scanning process.

Before starting scanning, we place multiple small objects on the glass board so that the objects do not overlap each other. We collect point cloud data from four laser sensors fixed on the robot arm of the scanner.

The highest precision of our laser scanner is 0.1 mm. One of the advantages of our scanner is that it can capture data from four different sides at the same time. In addition, our scanner allows scanning of multiple small objects at the same time.

For example, the scanning with point accuracy of 0.2 mm for 300 small artifacts takes about ten hours (see Fig. 2). On the other hand commercial scanner such as PICZA LPX-600 can scan only one object simultaneously. This scanner takes about 60 minutes to scan the object with size of 50 mm × 50 mm ×50 mm with the point accuracy of 0.2 mm. It takes many hours if several hundred stone tools are scanned by using those kinds of scanners. It shows that performance of using our scanner is multiple times faster than that of those scanners while scanning many small objects.

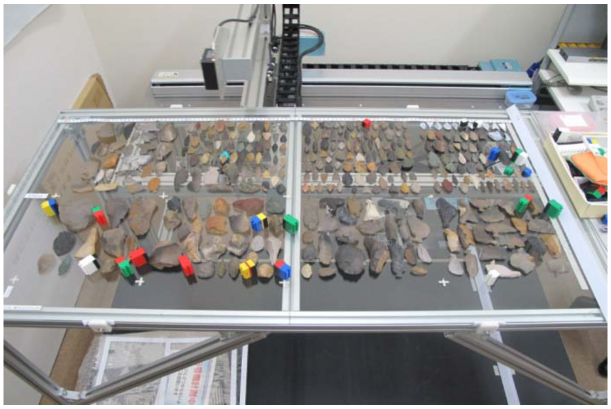

Fig. 2. Scanning process. More than hundred of small stone tools can be scanned at a time.

Another important issue for the scanning process is to properly place objects on the glass board. Since our scanner could not scan some areas invisible to the sensors, point data of such areas is omitted among the obtained data.

Once the scanning process is completed, the point clouds obtained from the four directions can be registered.

We divide the obtained point clouds into groups based on each object according to given threshold criteria of the distance between the objects. After that, the obtained point clouds are aligned in a coordinate system of each object.

The coordinate system of glass board is used for aligning process because the glass board is fixed during scanning process. At the end we need to merge the point data of each object.

To produce a uniform point data of the objects, certain calculations are needed in the merging process since the data for some areas might overlap since the point data is obtained from the four laser sensors.

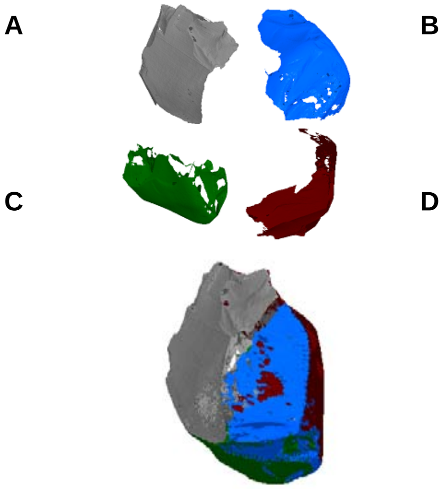

Fig. 3. Result of merging point clouds obtained from four different sensors. The point data A is chosen as a key set. The final result is shown in Fig. 8.

To merge data, an average value of the distance between adjacent points is found. Due to the fact that the point data captured from all four sensors should have the same density, first we choose one data set as the key set. The remaining point data will be added to the key set according to the distance criteria. If the distance between a new point and its nearest point of the key set is greater than the average distance, the new point will be added to the key set. In Fig. 3, we demonstrated the result of merging point clouds of a stone chopper measured by the four sensors. Before merging, the total number of the points was 240,478 and after merging, it was reduced to 160,801 points.

IV. SURFACE RECONSTRUCTION

Stone tools used by early humans were not of a formal form and have sharp edges in the blade sections as well as rough surfaces such as dents and cracks on the surfaces. Even though high density point clouds are obtained from our scanner, some points do not satisfy the assumptions of the algorithm [10]. Due to this, we consider additional arrangement for the point data.

Gopi's algorithm [10, 11] uses influence sphere for the sampling criteria. Possible Delaunay neighbors of point p will lie inside this sphere. If the angle between two sides of a blade is sharp, then the influence sphere of point p contains points on the other side of the blade as shown in Fig. 4. In this status, triangulation constructed with these points will be incorrect. Since point A on the sharp edge is not smooth (Fig. 4), ordering the candidate points of point A in the tangent plane will also be incorrect. Therefore, we need to increase density of the sampling points or distinguish the points on the two sides of the blade. By using medial axis of the object, we can easily separate the object into two layers, although medial axis construction is complicated to apply to our objects [15].

On the other hand it is possible to distinguish an object with simple complexity into two layers by using the method based on point cloud slicing technique. Fig. 5 illustrates the result of the point cloud slicing.

Here are the implementation steps of the point cloud slicing technique:

Step 1: Determine an appropriate direction for slicing a point cloud.

Step 2: Cut the point cloud into slices along a given direction (for example, along the x-axis), so that the point cloud is sliced by a given thickness. In our case, we choose the thickness value as a double of the average distance between the adjacent points. Let D_i (i=1, …, n) be point set of the i^{th} slice.

$$D_i = \{a \in S / | a^x - \mu_i | \le \delta\} \qquad (1)$$

where n is the slice number, δ is the thickness of the slice, a^x is the x coordinate of point a and μ_i is the slicing parameter of the i^{th} slice.

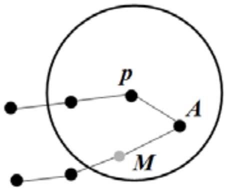

Fig. 4. Points on a sharp edge. Influence sphere of point **p** is contains point M on the other side of a blade surface.

Step 3: Order the points of each slice by the second coordinate (for example, by the y coordinate)

$$a_1^y \le a_2^y \le \ldots \le a_m^y$$

where a_i^y is the y coordinate of point a_i and m is the point number of set D_i.

Step 4: Divide the point set D_i equally into subsets and find the mass center point of each subset. The number of subsets depends on complexity of the object.

$$(a_1,\ldots,a_k), (a_{k+1},\ldots,a_{2k}), \ldots.., (a_{r*k+1},\ldots,a_m)$$

where k is the number of points of each set D_i (i = 1, …, n) and r+1 is the number of subsets of D_i

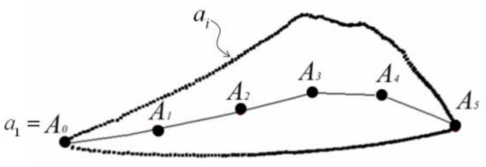

Fig. 5. A slice of the point data on an arrow head. A_0 A_1 ... A_5 configure a line segment separating the points into two regions.

Step 5: For each D_i (i=1, …, n), connect all the mass center points starting from a_1^y to reach the other end of slice (a_m^y). The obtained line segment is represented as

$$A_0A_1...A_{r+2} \tag{2}$$

where $A_0 = a_1$, $A_{r+2} = a_m$ and A_i (i = 1, …, r+1) are the mass center points of the i^{th} subset of D_i
Step 6: For the points of each slice, check the side of the line segment (2) where the point lies, so that all points are separated into two layers.
Step 7: After separating the points into two layers, we construct triangle meshes on one of the layers.
Step 8: Add the points on the border line of the layer to the point set of the next layer.
Step 9: Complete the triangulation

We can apply this method to simple objects with sharp edges such as arrow head or knife. The illustration of triangulation is shown in the figures 7-12.

V. HOLE FILLING

Filling holes is one of the important sections of our work. There are two kinds of hole-filling algorithms: volume-based algorithms [16, 17] and surface- based ones [18, 19].

The basic idea of volume-based methods is to convert a surface mesh into a volume representation. The volume-based methods are suitable for complex holes, although the time complexity of the methods is high.

The surface-based methods are used for filling holes locally. The methods do not change original surfaces during hole filling. It is not appropriate, however, for the case when a complex hole such as a hole with an island is filled.

In this chapter, we discuss on a surface-based simple algorithm. The algorithms mentioned above are good for filling holes with geometric continuity, although the time complexity of the methods is high.

In our case, there are two main reasons that the holes appear on the surface. First, holes are left in some invisible areas to the sensors in the obtained point cloud data. Invisible areas depend on the positions on the glass board and the shape of objects. In contrast, our triangulation algorithm leaves some small holes on the non-uniform sampling regions. In this paper, we introduce our original approach to fill holes. Our method fills holes with planar triangulation and does not provide a geometric continuity.

Main idea of our algorithm is as follows: to detect holes, add locally uniform points to a hole and construct triangle meshes with the added new points and the points on the border of the hole. Now the details of our algorithm are described below. We assume that contour curve of a hole is convex.

(1). Detect a hole by using boundary edges.

(2). Find the initial point of vertex coupling. First find the mass center point of the vertices on the boundary edges of the hole. Then, choose the most distant vertex of the hole from the mass center point and set it for the initial point.

(3). Perform vertex coupling. Number all the vertices of the hole starting from initial point. Create couples of vertices by following a simple rule. This process is described in Fig. 6. We will create two sets of vertex couples for vertices a_0, a_1, …, a_n:

a. 1st vertex coupling (a_i, a_{n-1-i})
   i=0, …, n/2-1
b. 2nd vertex coupling (a_i, a_{n+1-i})
   i=1, …, n/2

where a_0 is the initial point.

(4). Choose candidate points. First, find the average value of the lengths of border edges of the hole:

$$l = \frac{\sum_{i=0}^{n} d_i}{n+1} \tag{4}$$

where n+1 is the number of edges of the hole, d_i is the length of the i^{th} border edge.

Second, find a dividing number k for each vertex couple:

$$k = \left[\frac{l_{i,j}}{l} \right] \tag{5}$$

where $l_{i,j}$ is the distance between vertices a_i and a_j , [.] is the function of integer part.

Third, find points that can divide the line segment (ai, aj) equally into k parts. These points will be considered as the candidate points for filling the hole.

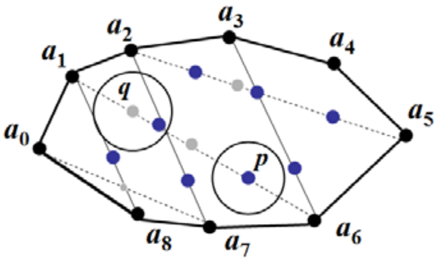

Fig. 6. The hole filling process.

(5). Create new point set for filling the hole. Let the initial points of the new point set be vertices on the border edges of the hole. By applying the rule below, we can add new points to the set from the candidate points. To add new points construct spheres with the centers at the candidate points that can be found on the each line segment. Radius of the spheres is l/2 where l is the average value of lengths of the border edges derived from equation (4). If the sphere does not contain any point of the new point set, the center point of the sphere will be added into the set of new points.

(6). Construct triangulation. After creating the new point set for filling the hole, we can construct triangle meshes by using Gopi's algorithm.

Fig. 6 shows the hole filling process. a_0 a1… a8 are the vertices on the border edge of the hole. Two sets of vertex coupling are represented with different lines. Blue dots, which are candidate points, are selected for filling the hole. Sphere with point p at the center does not contain any vertex and blue dot and it will be added to the set of new points. Since sphere with point q at the center contains the blue dot, point q cannot be added to the set of new points.

Due to the additional points in the hole, smaller holes might appear on the surface of the triangle mesh. We can use our simple algorithm repeatedly until all holes on the surface are filled. After new points are found in the hole, we smooth those new points by using Laplacian smoothing method [20]. If the contour curve of the holes is not convex, additional calculation is required. In this paper, we did not consider this case.

VI. RESULTS

This section presents some results of our approach to reconstructing surfaces and filling holes. Fig. 7 shows the result of creating a triangle mesh for an arrow head model.

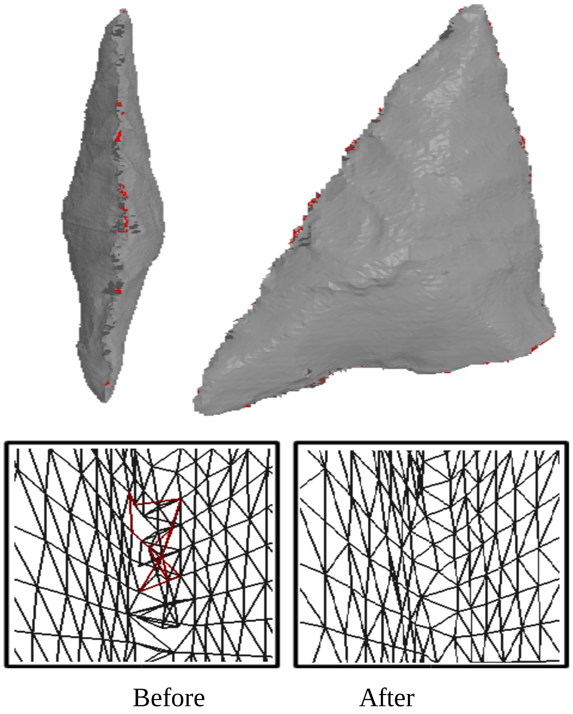

Fig. 7. 3D model of arrow head.

The original algorithm [10] cannot work for the sharp edged artifacts mentioned above. After our layer separation technique is applied, triangulation becomes correct.

Fig. 9 (a) shows the result of filling a hole on a fragment of a pot. The surface of this artifact does not include any sharp edge, and there is no difference between the original point data and one arranged by using our distinguishing technique. This object, however, had a lot of small holes found after scanning. The holes were caused due to the form of the object. After surface reconstruction of this object about thirty holes were detected and filled using our method. Fig. 9 (b) shows a sample of hole-filling process that includes newly added points and the triangle mesh on the surface.

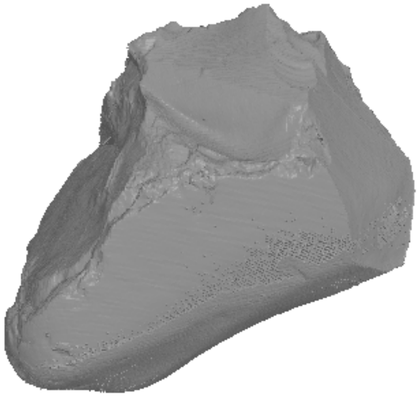

Fig. 8. 3D model of the stone chopper of Fig. 3.

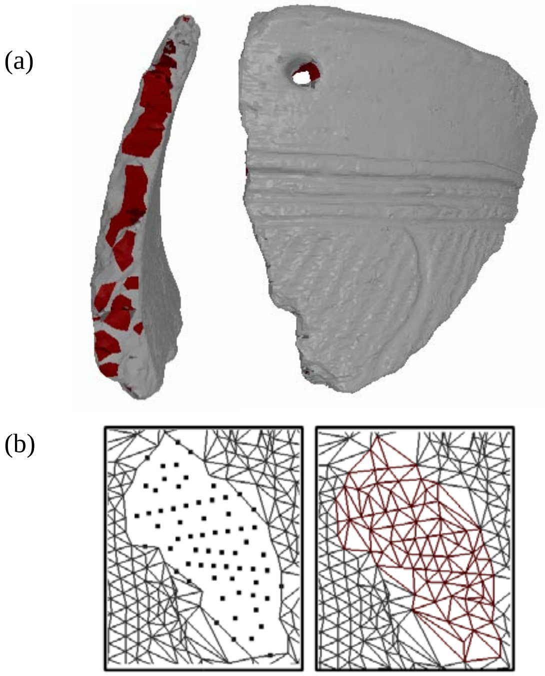

Fig. 9. 3D model of the fragment of a pot (a). Hole-filling algorithm includes added new points and mesh construction on the hole (b).

We demonstrated the final results of different sharp-edged stone tools in Fig. 8, 10 and 12. Fig. 11 shows the 3D model of the stone knife after applying the hole-filling algorithm. New points in the hole were smoothened.

Numerical results for Fig.7-12 are shown in TABLE 1 and TABLE 2. We illustrated the result of Gopi's algorithm including triangle number and computation time in upper part of the TABLE 1 and those of our approach in the lower of the table. A comparison results between our and Gopi's algorithm shows that our algorithm is slower than Gopi's algorithm because our algorithm includes point cloud separation process for improving quality of sharp edged objects. The result of hole-filling algorithm and its computation time are illustrated in TABLE 2. The hole-filling algorithm was implemented after our surface reconstruction algorithm.

TABLE 1: NUMERICAL RESULT OF THE SURFACE RECONSTRUCTION ALGORITHMS.

Features	**Fig. 7**	**Fig. 8**	**Fig. 9**	**Fig. 10**	**Fig. 12**
Number of points	126 046	160 801	642 576	100 197	235 991
Number of triangles (Gopi's algorithm)	248 891	318 256	1 269 909	199 811	470 321
Computing time of Gopi's algorithm	**4.37 sec**	**4.49 sec**	**22.44 sec**	**4.06 sec**	**8.27 sec**
Number of triangles (Our algorithm)	248 332	316 826	1 269 622	199 560	470 648
Computing time of our algorithm	**5.3 sec**	**5.47 sec**	**31.8 sec**	**3.2 sec**	**9.1 sec**

TABLE 2: NUMERICAL RESULT OF THE HOLE FILLING ALGORITHM.

Features	**Fig. 7**	**Fig. 8**	**Fig. 9**	**Fig. 10**	**Fig. 12**
Number of points	126 046	160 801	642 576	100 197	235 991
Number of triangles (Our algorithm)	248 332	316 826	1 269 622	199 560	470 648
Number of new points in the holes	1 408	2 032	42 190	932	1 546
Number of new triangles in the holes	2 995	4 029	86 016	1 945	2 205
Computing time for filling the holes	**1.5 sec**	**8.64 sec**	**83.4 sec**	**2.0 sec**	**9.66 sec**

The implementation and performance measurements were obtained using a computer with Intel Core i5-650 (3.2 GHz) processor and 3GB of RAM.

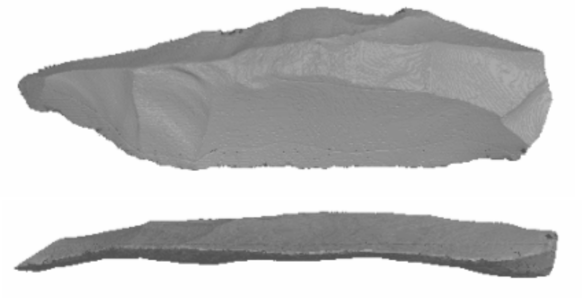

Fig. 10. 3D model of a stone knife.

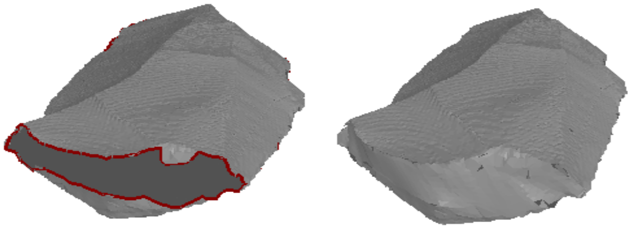

Fig. 11. 3D model of the stone knife. Result of the hole filling algorithm.

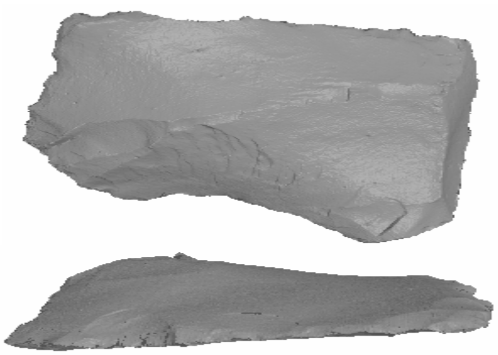

Fig. 12. 3D model of a stone scraper.

VII. CONCLUSTION

In this paper, we presented a full process of construction of 3D stone tool models including scanning of small artifacts, surface reconstruction and hole filling on the surface. Our original 3D scanner reduced a burden in scanning of a lot of small objects. Distinguishing point clouds into neighboring layers has improved the quality of surface reconstruction. In addition, we proposed simple hole-filling algorithm. Even though our hole-filling algorithm enables the actual shape with C^0 continuity, it is fast and efficient on simple holes.

REFERENCES

[1] M. Pieraccini, G. Guidi and C. Atzeni. 3D digitizing of cultural heritage. *Journal of Cultural Heritage*, vol. 2, no. 1, pp. 63–70, 2001.

[2] G.Pavlidis, A. Koutsoudis, F. Arnaoutologlou, V. Tsioukas and C. Chamzas. Methods for 3D digitization of cultural heritage, *Journal of Cultural Heritage*, vol. 8, pp. 93–98, 2007.

[3] G. Guidi, F. Remondino, M. Russo, F. Menna, A. Rizzi and S. Ercoli. A multi-resolution methodology for the 3D modeling of large and complex archaeological areas. *International Journal of Architectural Computing*, vol.7, no. 1, pp. 40–55, 2009.

[4] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura and Y. Okamoto. The great Buddha project: digitally archiving, restoring, and analyzing cultural heritage objects, *International Journal of Computer Vision*, vol. 75, no. 1, pp. 189–208, 2007.

[5] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. E. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues, in *Proceedings SIGGRAPH*, pp. 131–144, 2000.

[6] M. Tsakiri, C. Ioannidis and A. Carty. Laser scanning issues for the geometric recording of a complex statue, in *Proceedings of Optical 3D Measurement Techniques VI*, pp. 214–222, 2003.

[7] C. Georgiadis, P. Patias and G. Stylianou. Capturing the past and present: tools and methodologies for 3D modeling of small objects, *22nd CIPA Symposium*, Japan, 2009.

[8] R. Bernardini and H. Rushmeier. The 3d model acquisition pipeline, *Computer Graphics Forum*, vol. 21, pp. 149–172, 2002.

[9] B. P. Kooyman. Understanding stone tools and archaeological sites, University of Calgary Press, Calgary, 2000.

[10] M. Gopi, S. Krishnan and C.T. Silva. Surface reconstruction based on lower dimensional localized Delaunay triangulation, in *Proceedings of Eurographics*, vol. 19, no. 3, pp. 467–478, 2000.

[11] M. Gopi. Theory and practice of sampling and reconstruction for manifolds with boundaries, PhD thesis, Department of Computer Science, University of North Carolina Chapel Hill, 2001.

[12] M. Gopi and S. Krishnan. A fast and efficient projection-based approach for surface reconstruction, *International Journal of High Performance Computer Graphics, Multimedia and Visualization*, vol. 1, no.1, pp. 1–12, 2000.

[13] C. Buchart, D. Borro, and A. Amundarain. GPU Local Triangulation: An interpolating surface reconstruction algorithm, *Computer Graphics Forum*, vol. 27, no 3, 2008.

[14] B. S. Jong, P. F. Lee, J .L. Tseng. Raising local density for object reconstruction using auxiliary points. *1st IEEE/ACIS International Workshop on Component-Based Software Engineering*, Software Architecture and Reuse (ICIS-COMSAR'06), 2006.

[15] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction, *Graphical Models and Image Processing*, vol. 60, no. 2, pp.125–135, 1998.

[16] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion, in *Proceeding of the 1st International Symposium on 3D Data Processing Visualization and Transmission*, pp. 428-438, 2002.

[17] T. Ju. Robust repair of polygonal models, *ACM Trans. Graph*, vol. 23, no 3, pp. 888-895, 2004.

[18] J. Wang and M. M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point cloud, *Image Vision Computing*, vol. 25, no 1, pp. 103–113, 2007.

[19] Y. Jun. A piecewise hole filling algorithm in reverse engineering, *Computer-Aided Design*, vol. 37, no 2, pp. 263–270, 2005.

[20] G. Taubin. A signal processing approach to fair surface design, in *Proceedings of SIGGRAPH'95*, pp. 315–358, 1995.

Enkhbayar Altantsetseg received the BS and MS in mathematics from National University of Mongolia in 1995 and 1997, respectively. He is currently working toward the PhD degree in computer science at Iwate University. His research interests include computer graphics, geometric modeling, illustrative visualization, medical visualization and simulation.

Yuta Muraki is currently a Post-doctoral Fellow in Faculty of Engineering at Iwate University. His research interests include geometric modeling, CG, CAD and 3D scanning. He received the B.E. and M.E. degrees in Computer and Information Sciences, and the D.E. degree in Electronic Information Science from IWATE University in 2005, 2007, and 2010, respectively. He is a member of JSPE.

Fumito Chiba is a director of LANG Co., LTD. He received a bachelor of engineering from Iwate University in 1996. He earned his Dr. Eng. in Electrical Engineering and Computer Science from Iwate University in 2000. He worked on an assistant professor of Faculty of Engineering at Iwate University from 2000 to 2005. His research interests include 3D measurement systems and image processing. He is a member of Japan Society for Archaeological Information.

Kouichi Konno is a professor of Faculty of Engineering at Iwate University. He received a BS in Information Science in 1985 from the University of Tsukuba. He earned his Dr. Eng. in precision machinery engineering from the University of Tokyo in 1996. He joined the solid modeling project at RICOH from 1985 to 1999 and the XVL project at Lattice Technology in 2000. He worked on an associate professor of Faculty of Engineering at Iwate University from 2001 to 2009. His research interests include virtual reality, geometric modeling, 3D measurement systems, and computer graphics. He is a member of IEEE CS.