Springer

Journal of Mechanical Science and Technology 23 (2009) 1234~1243

Journal of Mechanical Science and Technology

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-1014-4

Efficiency of grid representation and its algorithms for areal 3D scan data †

Minho Chang[1] and Yun Chan Chung[2,*]

[1]*Department of Mechanical Engineering, Korea University, Seongbuk-gu, Seoul 136-701, Korea*
[2]*Department of Die and Mold Engineering, Seoul National University of Technology Nowon-gu, Seoul 139-743, Korea*

(Manuscript Received August 20, 2007; Revised September 10, 2008; Accepted September 29, 2008)

Abstract

This paper describes the efficiency of a grid representation for an areal 3D scan data and the algorithms for managing measurement data captured by areal 3D scanners. Due to the measurement principles of areal 3D scanners, a measurement point is generated for each pixel of the imaging sensor inside the 3D scanner. Therefore, when the measurement points are perspectively projected on the image plane of the imaging sensor, each point has one-to-one correspondence to the imaging elements of the sensor that has a regular grid structure. By using this property, measurement points are represented by their depth values in a grid representation model. Compared to the conventional representation model, such as triangular mesh and cloud of points, the grid representation uses less memory and allows efficient algorithms for processing the measurement data captured by areal 3D scanners.

Keywords: Data structure; Reverse engineering; Scan data; 3D scanning

1. Introduction

3D scanners are widely used for computer graphics and in medical and manufacturing applications [1, 2]. A 3D scanner scans a three-dimensional subject and produces a digital description of geometric samples on the surface of the subject. Among various types of 3D scanners, the areal 3D scanner is one of the most widely used 3D scanners in industry. An areal 3D scanner projects two-dimensional patterns on a subject and observes the scene with a camera (s). It then calculates 3D coordinates using the principle of triangulation. While 3D scanners using a single fixed-laser slit beam generate measurement points along a curve irradiated by the laser-slit beam on the surface of an object, the areal 3D scanner generates measurement points on surfaces irradiated by two-dimensional patterns. In this paper, the term '3D scanner' refers to an areal 3D scanner.

Fig. 1 illustrates the typical sequence used in processing the scan data in 3D scanning systems. Multiple 3D range images are scanned from various viewpoints and aligned with each other. Various techniques have been developed for aligning multiple scan data sets [3-6]. Besl and McKay [7] proposed an iterative closest point (ICP) algorithm to align multiple range images. Later, Chen and Medioni [8] improved the converging speed of the ICP algorithm.

In many applications, such as rapid prototyping, a single triangular mesh model is preferred to a set of multiple range images [9, 10]. Two different methods are often used to merge multiple range images into a single triangular mesh. The marching cube algorithm presented by Lorensen and Cline [11] is a well-known method of forming a triangular mesh from an unorganized point set using constant density surfaces. The other method is the zippering method presented by Turk and Levoy [3]. Images in the overlapped region are averaged and removed, and then the images are zipped to each other to form a triangular mesh model.

† This paper was recommended for publication in revised form by Associate Editor Soon Hung Han

* Corresponding author. Tel.: +82 2 970 6395, Fax.: +82 2 976 5173
E-mail address: ychung@snut.ac.kr

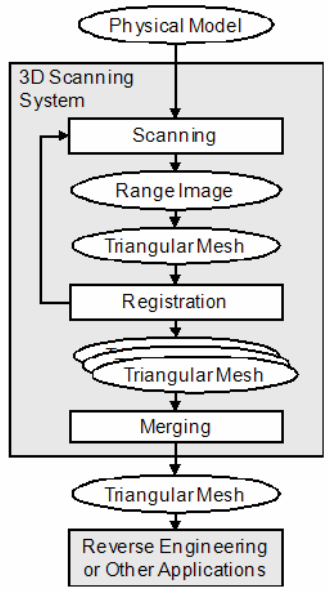


Fig. 1. Typical sequence of processing.

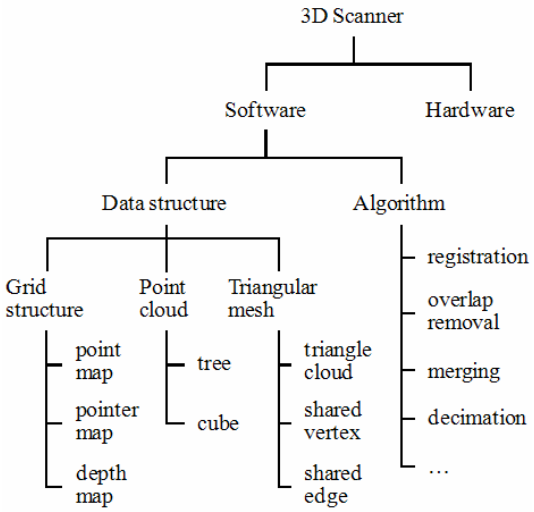


Fig. 2. Technology tree of software for 3D scanners.

Fig. 2 shows a technology tree of software for 3D scanners. Algorithms for processing the scan data, such as registration, overlap removal, merging, and decimation as in Fig. 2, have been devised and developed by many researchers. Data structure has been rarely focused even though algorithm is based on a data structure. Three data structures are used in processing the scan data in general. The first one is the grid structure storing a range image. Because the range image has been considered as a raw data, it has been promptly processed and converted to other data structures in general. The second one is cloud of points [12], which is the simplest data structure. Because cloud of points is a set of points and has no topological information, it is inadequate in a complex computing process. The last one is triangular mesh. Triangular meshes are often used as a data structure of scan data for the aligning and merging process [4, 12]. Triangular meshes represent geometric and also topological information. A triangular mesh is denoted by vertex positions and their connectivity such as edges and faces [13]. As depicted in the Fig. 3, a face has the indexes of the vertices constituting the face and the indexes of the adjacent faces, while an edge has the indexes of the vertices constituting the edge and the indexes of faces sharing the edge [14]. Such topological information facilitates various polygon operations including neighbor searching, smoothing, and decimation. Although the triangular mesh structure is powerful for processing scan data, the required memory increases rapidly as the number of data points increases. Isenburg and Gumhold [15] proposed an out-of-core algorithm to handle extremely large triangular meshes.

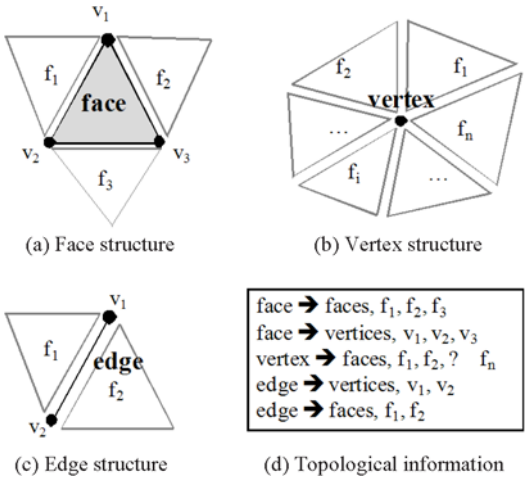


Fig. 3. A data structure for triangular mesh.

This study describes the efficiency of a grid representation for storing and processing the 3D images captured by areal 3D scanners. In the proposed sequence of scan data processing, a grid representation model is used for each single range image and maintained until it is merged. The proposed sequence differs from previous approaches in which images are converted to triangular meshes at the beginning of the data processing sequence. Several algorithms for the grid representation are proposed for the new sequence

of data processing. The described data structure and its algorithms enable the processing of hundreds of 3D images, even if a 3D image has a million coordinate values, because the gird representation model is more memory efficient compared to conventional representation models such as triangular mesh and cloud of points.

This paper is organized as follows: Section 2 presents a brief overview of areal 3D scanners. Section 3 describes and compares grid-type data structures used for storing and processing 3D image data. Section 4 describes the algorithms for grid-type data structures.

2. Overview of areal 3D scanners and grid representation

he data points of areal 3D scanners are ordered in rows and columns due to its measurement principle. When data points are ordered, a more efficient data structure can be designed and implemented. This section explains the measurement principle of areal 3D scanners, the characteristics of data points, and the operations necessary for processing the data points.

2.1 Scanning method

The phase-shifting optical triangulation method is one of the most widely used areal 3D scanning technologies in the manufacturing industry [16]. 3D scanners of this type consist of a projector and a camera, as shown in Fig. 4. The projector projects phase-shifting fringe patterns onto the surface of the object to be scanned, and the camera captures images of the projected patterns. The phase angle of each pixel can then be calculated from the images. For a phase angle of a pixel P of φ, Πφ is an imaginary plane of projected light that corresponds to the phase angle φ, and Pi is the point on the image plane that corresponds to P. The coordinates of the point on the object surface, Pw, corresponding to the pixel P are obtained by calculating the intersection point between Πφ and the line connecting Pi and the lens center O [17]. If this triangulation is repeated for each pixel, it is possible to determine the coordinates of points on the object surface within the FOV (field of view) of the projector and the camera. The set of points acquired from a single scanning process is known as a patch.

2.2 Characteristics of scan data

Due to the measurement principle of areal 3D scanners, measurement data has the following characteristics. First, a measurement point, Pw, exists along the line connecting Pi and the lens center of the camera, O. If the depth information of the measurement point is given, the coordinate of the point is uniquely determined. Second, Pi's have a regular grid structure. The imaging elements of the camera have the same size, and they are located at the image plane separated from the lens center by the focal length f. When Pw is perspectively projected on the image plane, Pi is then located at the center of the corresponding imaging element. Therefore, if two adjacent points are perspectively projected on the image plane, the distance between the projected points is equal to the size of the imaging element. Third, measurement points inherit the topological relations of the pixels. Although the points in a patch do not physically have a regular grid structure as imaging elements do, they topologically have a grid structure, implying that two measurement points corresponding to adjacent pixels do not have a point between them. Fig. 5 shows the characteristics of the scan data.

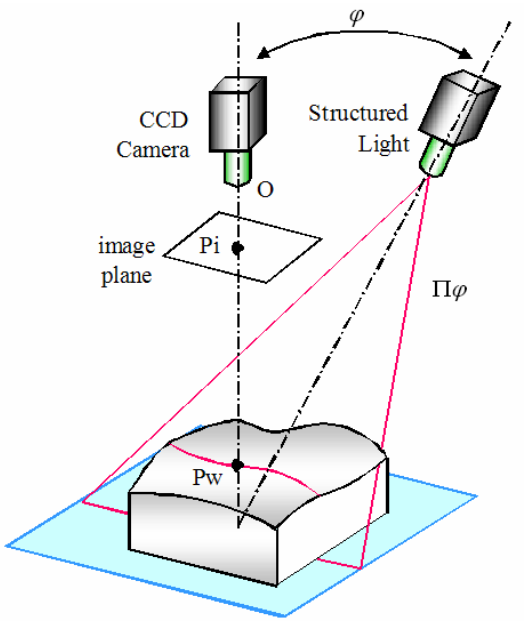


Fig. 4. Conceptual view of an areal 3D scanner.

3. Data structures for the grid representation model

Maintaining the grid structure in a computer data structure has an advantage from a memory point of view because no additional storage is necessary for

storing topological information in the data structure. Three data structures for the grid representation are proposed and evaluated: the point array, the pointer array, and the depth array. All three data structures use a two-dimensional array to implement the grid structure. When comparing the memory usage of three data structures, it is assumed that an image has a million (M) points at maximum and that each coordinate of a data point is stored as a double precision real value with using eight bytes.

3.1 Point Array

This data structure is the simplest among the three data structures. The coordinate values of the points in a patch are stored in a two-dimensional array. The two-dimensional array structure resembles the structure of imaging elements in the camera of a 3D scanner. The element in the i-th row and the j-th column of the array corresponds to the pixel in the i-th row and the j-th column of the camera. Each element of the array has x, y, and z coordinate values as depicted in Fig. 6.

Each element requires 24 bytes of memory to store three double precision real values. Therefore, 24 megabytes (MB) are required to store a patch having M points. The advantage of this data structure is that coordinates can be accessed directly by the indexes of the array without additional calculations or memory addressing. However, memory consumption is relatively high.

3.2 Pointer array

This data structure also uses a two-dimensional array. As shown in Fig. 7, however, the elements of the array have pointers to the data points and do not store the coordinates directly. Points are stored in a separate one-dimensional array. If a point is not generated at a pixel, its corresponding element of the array is set to NULL. In this data structure, the grid structure of the array represents the topological information, while coordinates are stored separately in a one-dimensional array.

When a memory address is stored in a double precision integer variable with 4 bytes of memory, a total of 4 MB of memory is required to store the memory addresses for a patch. If measurement data is generated for every pixel in a single patch, the memory required for a patch storing M points is 24 MB, and a total of 28 MB of memory is required. If 50% of the

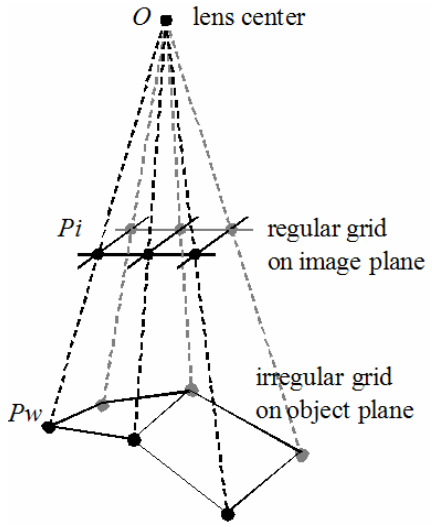


Fig. 5. Grid structure of scan data.

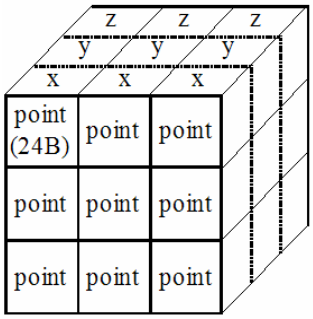


Fig. 6. Two-dimensional point array.

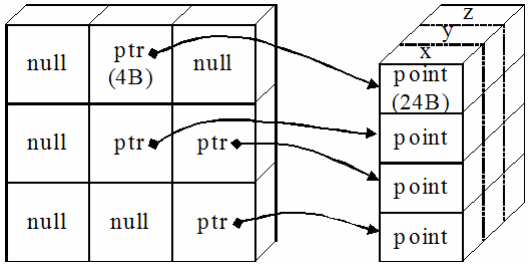


Fig. 7. Two-dimensional pointer array.

pixels do not generate data points as a result of occlusion or an out-of-FOV issue, the memory required for storing points is then dropped to 12 MB, and the total memory required is 16 MB. If the acquisition ratio of a data point in a patch is e, the memory space required to store the patch is calculated by Eq. 1.

$$4\text{MB} + 24\text{B} \times M \times e = (4+24e)\text{MB} \quad (1)$$

Table 1 shows the acquisition ratio of scanning results for the some typical examples as shown in Fig. 8. Many patches have an acquisition ratio smaller than 50%, and some of them are below 10%. If the acquisition ratio is 35% on average, this data structure requires 12 MB per patch. Performance may drop slightly because memory indexing is necessary when accessing the coordinates. However, this data structure can save memory when the acquisition ratio is not high.

3.3 Depth array

This data structure also uses a two-dimensional array, but it stores only the depth information of the point from the lens center of the 3D scanner in the array, as shown in Fig. 9. The depth value stored in the array is defined in a generalized form,

$$d = d(i, j) \tag{2}$$

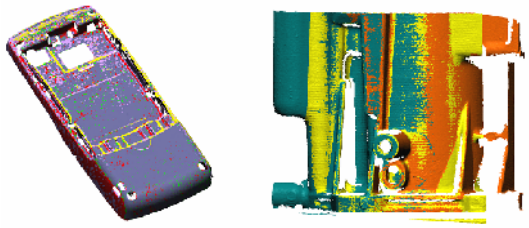
where (i, j) represent the indices of an element on the array, and depth d is the z coordinates of point Pw as shown in Fig. 10. The x and y coordinates in the image plane are calculated from the indexes. As the size of imaging elements is constant, the coordinates of Pi are found from the following equation. In the equation, (x0, y0) is the known coordinate value at the pixel (0, 0) and dx and dy are the size of imaging element in the x and y direction, respectively.

$$\begin{aligned} x_i &= x_i(i) = x_0 + dx \cdot i \\ y_i &= y_i(j) = y_0 + dy \cdot j \end{aligned} \tag{3}$$

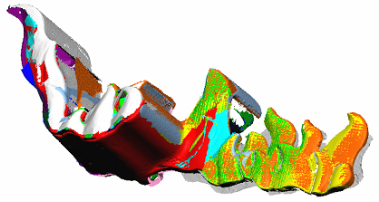
As the measurement point lies on the line connecting the lens center O and Pi, the coordinates of Pw are calculated from the following equation (Fig. 10). In the equation f is the focal length of the lens.

Table 1. Acquisition ratio of scan data.

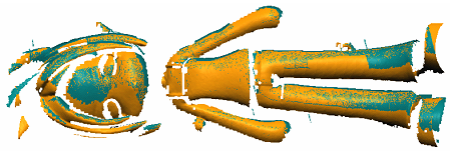
model	# of patches	pixel resolution	# of total vertices	average acquisition ratio (%)
cell phone	5	1280x1024	1,817,220	18.4
trans-mission	3	1280x1024	2,098,622	53.4
knob	12	1280x1024	4,617,608	29.4
puppet	2	1620x1220	507,934	12.9
total	22		9,041,384	33.8



(a) Cell phone (b) Transmission



(c) Knob



(d) Puppet

Fig. 8. Test examples.

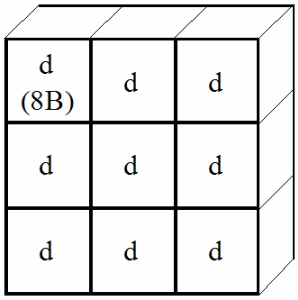


Fig. 9. Two-dimensional depth array.

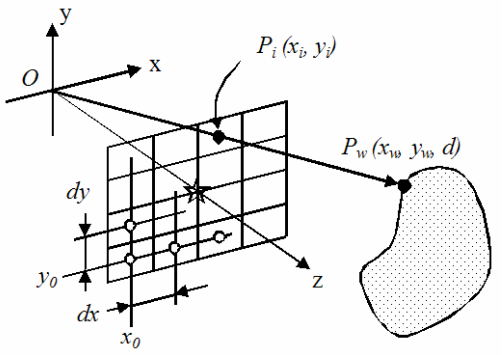


Fig. 10. Image point and world point.

Table 2. Comparison of representation models.

Type of data structure	Memory for a point (bytes)		Operations for a point
	e =1.0	e =0.35	
Point array	24	24	direct access
Pointer array	28	12.4	1 memory addressing
Depth array	8	8	5 multiplications
Cloud of points	*24*	*8.4*	direct access
Triangular mesh	*72*	*25.2*	1 memory addressing

e : acquisition ratio

$$x_w = x_i \cdot d / f$$
$$y_w = y_i \cdot d / f \quad (4)$$
$$z_w = d$$

This data structure stores one double precision value for each data point. Eight MB of memory is required to store M points. More computation is required for accessing coordinates, but this data structure requires much less memory.

3.4 Performance evaluation of data structures

Table 2 summarizes the memory usage and the number of operations required to retrieve a data point for the data structures described in the previous section. For comparison, the triangular mesh and cloud of points are also depicted in the table. From a memory point of view, the depth array is the most efficient data structure. The depth array uses only one-third of the memory that a point array needs. If the acquisition ratio is greater than 16.7%, which is typical, the depth array is more efficient than the pointer array.

In the table, the cloud of points [18] is assumed the most compact structure, as it has only x, y, and z coordinate values but does not have any topological information. It is meaningless to compare the computational efficiency for the cloud of points because of its characteristic, which is not directly applicable in the geometry processing. The triangular mesh is assumed as Lawson's model [19] without any additional information. A triangle in his model is represented as a tuple consisting of three vertex indices and three adjacent triangle indices, as shown in Fig. 11. The memory space required for a triangular mesh with M points is denoted by Eq. 5.

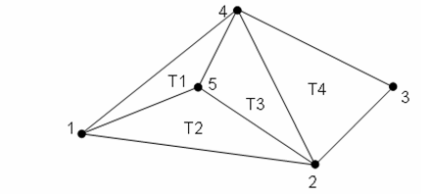
$$(24B/pt + 24B/tri \times 2\ tri/pt) \times M\ pt = 72\ MB \quad (5)$$

Theoretically, the point array and triangular mesh have the highest computational efficiency. The pointer array and triangular mesh require one additional memory addressing, while the depth array requires five floating-point multiplications to retrieve a data point. In contrast, the point array does not require memory addressing or floating-point operations to retrieve points. To compare the computational efficiency, a 3D data processing operation of removing redundant surfaces of images is implemented for the data structures. The operations were tested for two examples as shown in Fig. 8, and the result as shown in Table 3. This test shows that the triangular mesh has the highest computational efficiency. The pointer array and the depth array have similar computational efficiency, although the depth array is more memory efficient that the pointer array.

Table 3. Computational efficiency of data structures.

Data structure	Computing time in second		Ratio
	cell phone	transmission	
Point array	15.69	19.46	1.00
Pointer array	20.34	28.08	1.38
Depth array	20.84	26.27	1.34
Triangular mesh	14.25	16.10	0.86

* operation for removing overlapped surfaces of 3D images



vertex index	coordinate values		
1	*x1*	*y1*	*z1*
2	*x2*	*y2*	*z2*
3	*x3*	*y3*	*z3*
4	*x4*	*y4*	*z4*
5	*x5*	*y5*	*z5*

triangle index	vertex index			adjacent triangle index		
1	1	5	4	3	-	2
2	1	2	5	3	1	-
3	2	4	5	1	2	4
4	2	3	4	-	3	-

Fig. 11 Triangular mesh representation.

4. Algorithms for the grid representation

The grid representation has the following advantages over the triangular mesh model. In the grid representation, neighbors of a data point can be easily found with simple index operations. Triangulation is straightforward. A one-to-one relationship between the pixel and the point is maintained. This can be used

to evaluate the reliability of the points.

4.1 Triangulation

Triangulation in the grid representation described in the previous section is straightforward because the data is a two-dimensional rectangular grid. Triangulation in 3D is a problem that has received much attention, and many sophisticated algorithms, such as the Delaunay triangulation algorithm [20], have been formulated. They are well suited for an irregular distribution of the points in 3D. However, the points on the data structures described in the previous section are a two-dimensional rectangular grid topologically, as they inherit this feature of an image plane, as it is also a rectangular grid. Triangles can be generated by connecting four adjacent points in the rectangular grid, as shown in Fig. 12. The diagonal edge is generated by connecting the pair of opposite points with the smaller difference in the depth value. If the difference in the depth value is too large, the two points are considered as disconnected. Only one triangle can be generated when one of the points is invalid. No triangle can be generated when two or more points are invalid. The triangulation process is much faster than the Delaunay triangulation algorithm, which is often used for the triangulation of unordered point sets.

4.2 Normal vector evaluation

The normal vector for a triangle can be efficiently calculated in the grid representation. The normal vector of a point is often calculated by averaging the normal vectors of the adjacent triangles. The normal vector of a triangle is calculated by the cross product of two edges of the triangle. If H (dx, 0, h) and V (0, dy, v) are the edges of a triangle in a grid in the x and y directions, respectively, in which dx and dy are the grid intervals in the x and y directions, respectively, and h and v are the depth differences (see Fig. 13), then the normal vector of a triangle can be calculated by Eq. 6. If it is assumed that dx and dy are constant, the normal vector of a triangle can then be calculated by two multiplications.

$$\mathbf{H}\times\mathbf{V}=\begin{vmatrix}\mathbf{i} & \mathbf{j} & \mathbf{k}\\ dx & 0 & h\\ 0 & dy & v\end{vmatrix}=(-h\cdot dy)\mathbf{i}-(v\cdot dx)\mathbf{j}+(dx\cdot dy)\mathbf{k} \quad (6)$$

The normal vector of a point can also be calculated efficiently in the grid-type data structure. Instead of calculating the normal vectors of adjacent triangles and averaging them, one can calculate the normal vector of a point by adjacent four points. Once depth differences in the x and y directions are obtained from four points around the point P, the normal vector of point P can be calculated by two multiplications, as in Eq. 6.

4.3 Nearest point searching

Searching for the nearest point is computationally inexpensive in grid representation. The alignment, the remove overlap, and the zippering are very typical operations in processing areal 3D scan data. Those operations extensively execute this searching procedure. As shown in Fig. 14, the projected point Pw of P is calculated directly by the camera geometry, and the nearest point of P can be found easily by searching among the neighbors in the two-dimensional grid.

5. Strategy of data processing and results

A data processing strategy using the grid representation is presented in Fig. 15. Multiple patches are scanned from various viewpoints and they aligned to each other. Points in the overlapped region are aver aged and removed. Averaging and removing points in the overlapped surfaces are typical operations to make

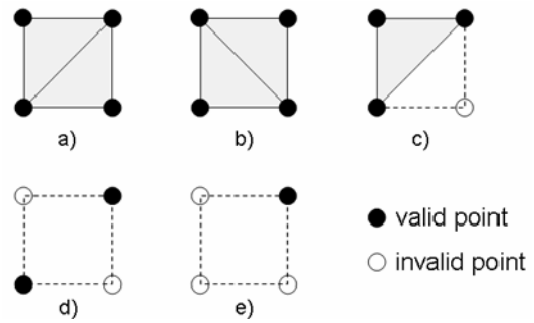


Fig. 12. Possible triangulations.

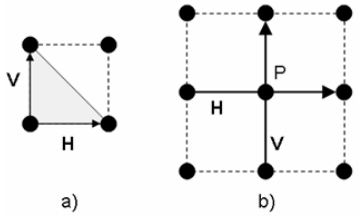


Fig. 13. Normal vector evaluation.

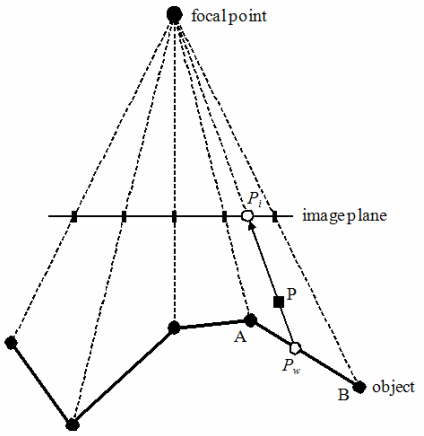


Fig. 14. Nearest point searching.

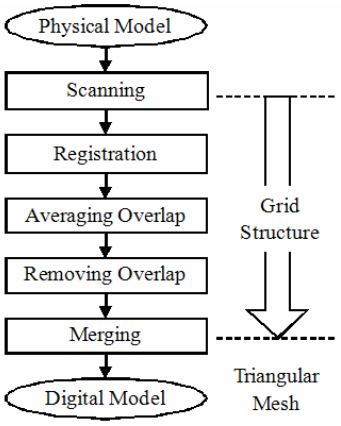


Fig. 15. Proposed strategy of data processing.

make better quality of the measurement data. Then, the patches are integrated to form a single triangular mesh. At the merging operation, the grid structure of the scan data is converted into triangular mesh structure.

For comparison with a strategy using only triangular mesh as the data structure, some operations for depth array as a grid structure are implemented for handling the measurement data of an areal 3D scanner. The first implemented algorithm is a rendering operation. The rendering operation uses triangulation and normal vector evaluation algorithms presented in the section 4 of this paper. Other implemented operations are averaging and removing overlapped surfaces. Those two operations use the nearest point searching algorithm presented in the section 4.3.

The example shown in Fig. 16 was used to validate the presented strategy. The pixel resolution of the

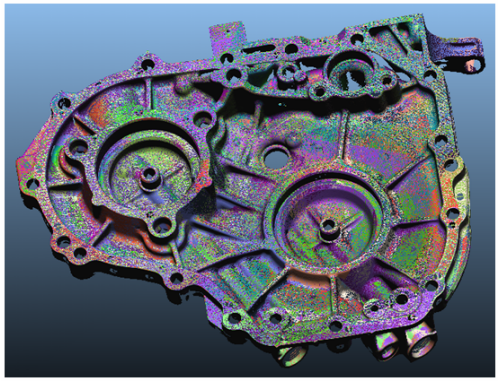


Fig. 16. Example: 42 patches with 1,620 x 1,220 resolutions.

scanner was 1,620 x 1,220, and 42 patches were scanned and aligned. To render the 42 patches as one image, less than 1 gigabyte of memory was required. Averaging and removing overlapped surfaces were also processed with less than 1 gigabyte of memory. The tests were executed on Windows XP. Similar operations were applied at a commercial reverse engineering system. The system could not read all of the patches in 3 gigabytes of main memory. After reducing data points by less than 50% the commercial system could handle the all patches.

6. Conclusion

This paper describes the efficiency of a grid representation for handling the measurement data of an areal 3D scanner and its algorithms. Based on an investigation of the measurement data of an areal 3D scanner and the operations required for the data, the benefit of the grid representation was described. Three data structures for the grid representation, the point array, the pointer array, and the depth array that each uses a two-dimensional array were suggested and evaluated. Among the three data structures, the depth array consumes the least memory. When the depth array is used, over 200 patches with a resolution of one million points can be processed simultaneously with 2 gigabytes memory. In terms of the speed of the retrieval of data points, both the depth array and the pointer array showed no significant differences. When the grid representation model is used, many operations, especially a neighbor searching operation, gain advantages from index operations. It is recommended to switch to the triangular mesh model after the merging operation because a more flexible representation model is necessary and the

memory burden is alleviated in the downstream process.

Acknowledgment

This research was funded by the Korean Ministry of Commerce, Industry, and Energy.

Nomenclature

d : Depth of the point from the lens center
dx : Size of imaging element in the x direction
dy : Size of imaging element in the y direction
e : Acquisition ratio of data points
f : Focal length
O : Lens center of a camera
P_w : Point on the object surface
P_i : Point on the image plane

References

[1] T. Varady, R. R. Martin and J. Cox, Reverse engineering of geometric models - and introduction, *Computer-Aided Design*, 29 (1997) 255-268.

[2] S. Azernikov and A. Fischer, Efficient surface reconstruction method for distributed CAD, *Computer-Aided Design*, 36 (9) (2004) 799-808.

[3] G. Turk and M. Levoy, Zippered polygon meshes from range images, *SIGGRAPH 94 Conference Proceedings*, (1994) 311-318.

[4] Y. Sun, Mesh-based integration of range and color images, *Proceedings of SPIE Conference on Sensor Fusion: Architectures, Algorithms, and Applications IV*, 4051 (2000) 110-117.

[5] X. Ju, T. Boyling, J. P. Siebert, N. McFarlane, J. Wu and R. Tillett, Integration of range images in a multi-view stereo system, *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, (2004) 280-283.

[6] M. Hirose, H. Furuhashi and K. Araki, Automatic Registration of Multi-view Range Images Measured under Free Condition, *Proceedings of the 7th International Conference on Virtual Systems and Multimedia (VSMM'01)*, (2001) 738-746.

[7] P. J. Besl and N. D. McKay, A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 (1992) 239-256.

[8] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, *Image and Vision Computing*, 10 (1992) 145-155.

[9] L. M. Galantucci, G. Percoco, G. Angelelli, C. Lopez, F. Introna, C. Liuzzi and A. De Donno, Reverse engineering techniques applied to a human skull, for CAD 3D reconstruction and physical replication by rapid prototyping, *Journal of Medical Engineering and Technology*, 30 (2) (2006) 102-111.

[10] J. Hradek, M. Kuchar and V. Skala, Hash functions and triangular mesh reconstruction, *Computers and Geosciences*, 29 (6) (2003) 741-751.

[11] W. E. Lorensen and H. E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, *SIGGRAPH 87 Conference Proceedings*, (1987) 163-169.

[12] W. Sun, C. Bradley, Y. F. Zhang and H. T. Loh, Cloud data modelling employing a unified, non-redundant triangular net, *Computer-Aided Design*, 33 (2001) 183-193.

[13] J. Peng, C. S. Kim and C. J. Kuo, Technologies for 3D mesh compression: A survey, *Journal of Visual Communication and Image Representation*, 16 (6) (2005) 688-733.

[14] S. K. Park and S. H. Lee, A compact and efficient polygonal mesh representation, *Transactions of the Society of CAD/CAM Engineers*, 9 (4) (2004) 294-305 (in Korean).

[15] M. Isenburg and S. Gumhold, Out-of-core compression for gigantic polygon meshes, *SIGGRAPH 2003 Conference Proceedings*, (2003) 935-942.

[16] L. Kovacs, G. Brockmann, A. Zimmermann, H. Baurecht, K. Udosic, M. Gühring, K. Schwenzer, N. A. Papadopulos, E. Biemer, R. Sader and H. F. Zeilhofer, Precision and accuracy by scanning of the facial region with the Minolta-Vivid 910® 3D Scanner, *International Congress Series*, 1281 (2005) 1288.

[17] A. Srivastava, X. Liu and C. Hesher, Face recognition using optimal linear components of range images, *Image and Vision Computing*, 24 (3) (2006) 291-299.

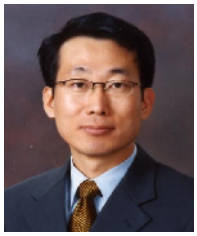
[18] Y. Ohtake, A. Belyaev and H. P. Seidel, 3D scattered data interpolation and approximation with multilevel compactly supported RBFs, *Graphical Models*, 67 (3) (2005) 150-165.

[19] C. L. Lawson, Software for C1 surface interpolation, *Mathematical Software III, Rice ed., Academic Press, New York*, (1977) 161-194.

[20] C. C. Kuo and H. T. Yau, A Delaunay-based re-

gion-growing approach to surface reconstruction from unorganized points, *Computer-Aided Design*, 37 (8) (2005) 825-835.

Minho Chang is a Professor at the department of mechanical engineering at Korea University in Seoul Korea. He received a PhD degree in Mechanical Engineering from MIT in 1996. He worked for Korea Institute of Science and Technology. His research interests include mechanical design, three-dimensional measurement, and CAD.



Yun Chan Chung is a Professor in the department of die and mold engineering at Seoul National University of Technology, Korea. He worked for Cubictek and DaimlerChrysler, developing CAD/CAM systems mainly in die and mold making. He received PhD in Industrial Engineering from KAIST in 1996. His research interests include digital manufacturing, tool-path generation and verification, and software engineering.