Reconstructing the 3D thermal model of indoor environment from unorganized data set acquired by 3D laser scans and thermal imaging camera*

Dinko Osmanković[1] and Jamin Velagić[2]

***Abstract*—After recording data sets using a 3D laser scanner, the logical step would be to reconstruct 3D model from given points. This paper proposes one solution to the reconstruction of 3D model. It combines splatting methods with polygonalization to achieve most accurate 3D model of an indoor mobile robot environment with fast execution and rendering time. The main objective of the work, which was done as a part of ThermalMapper project, was to generate an accurate mesh of the indoor environment based on laser scans with added temperature scalars acquired by 6D SLAM method. This is useful for generating meshes of museums, buildings, tunnels, etc. which can be used for inspection and different analyses. A series of experiments demonstrate the usefulness and effectiveness of the proposed 3D surface reconstruction methods.**

***Index Terms*—3D model, reconstruction, splatting, marching cubes, slam, mesh**

I. Introduction

MAP building is itself a challenging task. The output data of methods for the map building is a set of points in Euclidean space [1], [2]. This can be basis for reconstructing a 3D model of the map built by an autonomous robotic platform using map-building algorithms.

Reconstructing the 3D model from point clouds, the logical continuation of mapping problem, is also a very hard problem to solve. Mostly, it is used for digitalization of cultural heritage [3], digitalization of architectural sites [4], and for visualization of force/energy fields [5]. For visualisation it is also used in MRI/CT scans [6], [7], [8] and in cosmology [9] for inspection and analytical purposes.

When it comes to reconstructing the model from point sets, methods are divided according to the structural organization of point sets and the spatial subdivision. Point sets can be divided into two main groups [10]:

- Unorganized points - points are not organized in any particular way, they are defined only by their position in Euclidean space, (x, y, z) coordinates,
- Structured points - points are organized with a regular topology and a regular geometry. The grid points (i, j, k) for $0 \leq i < nx, 0 \leq j < ny$, and $0 \leq k < nz$ located on a regular rectangular lattice with a given origin and fixed spacing in all dimensions.

Algorithms for reconstructing 3D models from unorganized point sets are based on coherence of the points in given data set. Because of this, reconstructing from unorganized point sets requires more time and is hardware consuming as it is impossible to *guess* the geometry of the model. Algorithms of this kind are much slower with higher hardware requirements for running.

These algorithms can be divided into two groups:

- Surface oriented[11]
- Volume oriented [12], [13]

Whichever method is used, the final result is a mesh, which represent a set of triangular (of quadrilateral) faces merged at their edges in order to create a 3D model, bit like a child playing with puzzles just extended to 3D.

The focus of this paper is to employ combined information captured by both laser scanner and thermal camera to build 3D model of indoor environment containing thermal information. In this sense we have decided to use Marching cubes algorithm based approach to reconstruct the model and fast tree traversal algorithm for thermal information mapping.

This paper is organized as follows. In section II mathematical formulations of the above mentioned algorithms are presented. Section III gives the results of three different case studies. In section IV the algorithm for mapping temperature scalars onto a newly reconstructed surface model is discussed and the final results are presented. Some concluding remarks and future directions are given in section V.

II. Methodology

In this paper the acquired scans of 3D laser Riegl VZ-400 mounted on the mobile platform are used (Fig. 1). As already mentioned in section I, points acquired are unorganized so it requires pre-processing and mapping onto structured grid.

The acquired data can be represented as a point cloud but for the human eye the mesh structure is a natural way to present the virtual model. The process of acquiring surface from the set of points is called surface reconstruction. This process will be further explained in this section. Through the next subsections the problematics of the MCA (Marching Cubes Algorithm) based approach is discussed as

*ThermalMapper is a SEE-ERA.NET project and has the project number ERA 14/01, website: *http://www.faculty.jacobs-university.de/anuechter/thermalmapper.html*

[1]D. Osmanković is with Faculty of Electrical Engineering Sarajevo, Department of Automatic Control and Electronics, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina (e-mail: dinko.osmankovic@etf.unsa.ba)

[2]J. Velagić is with Faculty of Electrical Engineering Sarajevo, Department of Automatic Control and Electronics, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina (e-mail: jasmin.velagic@etf.unsa.ba)

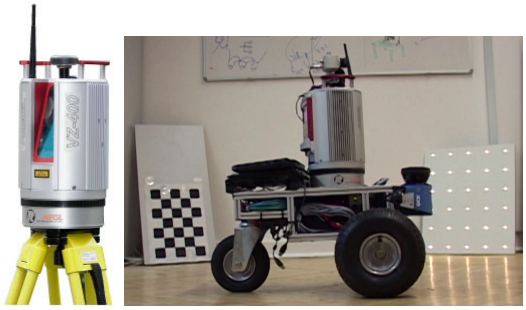

Figure 1. Riegl VZ-400 laser scanner (www.riegl.com) (left), Irma3D mobile platform (right)

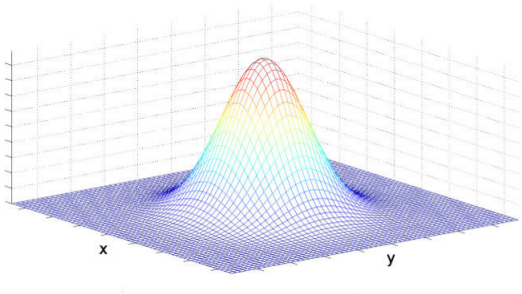

Figure 2. Gaussian distribution function for 2D case; This function represents iso–value spatial propagation around the point p (in this case located on the top of the peak)

we decided to organized data in cells since it is the best approach having in mind that application is for inspection purpose. This includes the estimation of iso-values, which is a very important step before the actual MCA-based iso-surface reconstruction and mesh generation. Also, mapping the scalar values acquired from the thermal imaging data will be discussed.

A. Iso - values estimation

To generate the iso surface more quickly (although not very precisely) a method different to the ones presented in [14], [15], [16] is used. One of the best estimation mechanism is a Gaussian filter. This is a probabilistic method that attempts to estimate the iso-value of each point in the data set by injecting the input point into structured geometry [10], [14]. This is accomplished by assigning a Gaussian distribution (graph in Fig. 2) to every input point, and then aggregating the final iso-surface by applying some aggregation operator (i.e. minimum, maximum or sum).

Given the point p, the Gaussian distribution function $F(p)$ is calculated as:

$$F(p) = s \cdot exp\left(f \cdot \left(\frac{r}{R}\right)^2\right) \tag{1}$$

where:

p - current voxel or sample point $p(x, y, z)$,

s - scale factor, can be multiplied by scalar value at the point,

f - decay factor,

r - Euclidean distance $|p_r - p|$where p_r is a voxel in the neighbourhood of p,

R - radius of influence.

If there are normals in data set (acquired or calculated) it is possible to extend the distribution from spherical to elliptical. The Gaussian distribution in this case is given by:

$$f(x) = s \cdot exp\left(f \cdot \frac{\left(\frac{r_{xy}}{E}\right)^2 + z^2}{R^2}\right) \tag{2}$$

where:

p - current voxel or sample point $p(x, y, z)$,

s - scale factor which expands distribution,

f - decay factor,

z - distance of the current voxel sample point along normal N,

r_{xy} - Euclidean distance between p_r and p, $|p_r - p|$, in direction perpendicular to the normal N,

E - eccentricity factor controlling the elliptical shape of the distribution.

This approach produces far better results. The problem of reconstructing the indoor environment with this iso-value estimation is that it is impossible to measure the normal at certain point in space with laser range scanner. Also, estimating the normals at every point is time and resource consuming it becomes nonsensical to employ it.

The main drawback of this method is that it distorts the mesh of the model. Outer vertices of the mesh, the visible ones, will not be in the same place in Euclidean space as the original point from the point cloud. This is due to the nature of the equation 2 which distributes iso–values in space around the point.

B. Marching Cubes

Marching cubes [17] is a algorithm for extracting polygonal mesh by computing iso-surfaces from discrete data which are basically scalar field, i.e. voxels.

The basic principle is that a contour (in 3D it's a iso-surface) can only pass through a cell in a finite number of ways. We enumerate all possible topological states of a cell, given the combinations of scalar value at the cell points. The number of topological states depends on the number of cell vertices, and the number of inside/outside relationships a vertex can have with respect to the contour value. A vertex is considered inside a contour if its iso-value is larger than the iso-value of the contour [10].

Lets consider a simple 2D case, so called *marching squares*. The cell has 4 vertices, and each vertex can be considered inside a contour, or outside which gives $2^4 = 16$ possible cases. They are given in Fig. 3.

After selecting a proper case, the location of the contour/cell edge intersection can be calculated using interpolation [10]. Then, algorithm moves (marches) to the next

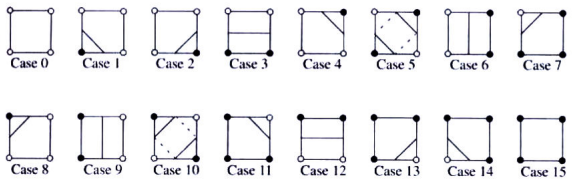

Figure 3. Sixteen cases of the marching squares vertex activation, dark vertices have scalar value above contour value

Algorithm 1 Marching squares pseudo - code

Require: Structured data set with calculated iso-values

1: **for** every point in the grid **do**
2: set the status of the point to inside or outside the iso-line
3: **end for**
4: **for** every square in the grid **do**
5: **for** every line segment that has a point outside and a point inside the figure **do**
6: Add a point in the list t placed in the middle between the two points
7: **end for**
8: **end for**
9: Draw sequential lines between the points found in the list t.

cell repeating the process. This can be written as given in Algorithm 1[10].

The 3D case, or *marching cubes*, is an extension of marching squares with the difference that the cell in this case is a voxel, which is a cube with 8 vertices. There are $2^8 = 256$ possible cases choosing the vertex inside the iso-surface. This will not be presented in the paper as there are too many of them (although they can be reduced to 15 using symmetry [10]).

C. Scalar Field Mapping

The Gaussian estimation of iso-values and subsequent MCA reconstruction do not deal with scalar values of temperature acquired by the thermal imaging camera. These values have to be *mapped* onto reconstructed surface. One problem that emerges is that it is not possible to establish one-to-one correspondence between the point cloud and the reconstructed model (i.e. point in the point cloud is not the same point in the reconstructed model). To overcome this problem, foremost, the K-d tree of the model is constructed, so every point is organized into a tree. This enables fast searching through the model (e.g. range searches and nearest neighbour searches). Now, it is possible to find corresponding point in the reconstructed model to every point in the point cloud. After finding such point, it gets the scalar value from its corresponding point in the point cloud. This can be accomplished since MCA with proposed iso-values estimation produces the model that is as *close* as possible to the original point cloud. The pseudo code for this procedure is given in Algorithm 2.

Algorithm 2 Scalar Field Mapping pseudo code

Require: Reconstructed iso-surface and point cloud with scalars, original point cloud

1: Build a k-D tree for the reconstructed iso-surface model
2: **for** every point p in the point cloud **do**
3: Find the nearest point p_m in the model to the point p
4: Assign the scalar value of the point p in the point cloud to point p_m from the model
5: **end for**

Figure 4. Model of teapot, subdivision 150 decimation 20% (left), subdivision 50, decimation 0% (right)

In the next two sections we present the resulting models obtained by employing MCA and mapping the scalar field onto the reconstructed model. We will also evaluate the model and analyse the performance of these algorithms.

III. Results (Model Reconstruction)

The described filter chain for reconstructing the 3D model of environment is tested with 3 data sets. The first data set is general case for reconstruction algorithms, a teapot. Second and third cases are real indoor environments at Jacobs University in Bremen, Germany. The second case is a hallway, and the third case is the acquired data set of one room connected to the hallway. Algorithms for reconstruction and temperature scalar field mapping are tested with different parameter sets. All of the data sets are unorganized points in Euclidean space formatted in the ASCII text file containing $(x \quad y \quad z)$ real values.

A. Case 1 - Teapot

The first case is a model of a teapot that is a standard test case for reconstruction algorithms. It consists of 30569 unorganized points in Euclidean space. Space subdivisions are 50, 100, and 150 without decimation (i.e. elimination of some triangles without destroying the geometry of the mesh). Also, for fixed space subdivision of 100, decimation change is tested for 10%, 20%, and 50% decimation.

We present the model of the teapot for subdivision of 150, and decimation of 20% compared to the model with subdivision 50, and no decimation in Fig. 4. Test results of execution time and number of polygons are given in Tables I and II. Tests are run 20 times, and the average was taken. It is clear from Fig. 4 that space subdivision of 50 is not

Table I
RESULTS FOR CASE 1

Subdivision	50	100	150
Execution Time	3.4 seconds	13.8 seconds	31.3 seconds
Number of Polys	33780	141728	325176

Table II
RESULTS FOR CASE 1

Decimation	10%	20%	50%
Execution Time	13.8 seconds	13.7 seconds	13.8 seconds
Number of Polys	129543	114276	72771

sufficient for creating an iso-surface of the teapot. Also, higher the decimation, number of polygons in the model is lower, lowering the accuracy but increasing the interactivity, and inspection.

We compared the results of the teapot 3D model reconstruction with other methods [11], [15], [14] by running the algorithms 20 times on the same input set. The results in Table III clearly show that this algorithm is faster, but has lower precision compared to other algorithms. Accuracy is measured by calculating the maximal distance of the reconstructed model vertices to the original point cloud points.

All test were performed on a 2.2Ghz Intel Core2Duo PC with 3GB RAM. For next 2 cases, all four algorithms were tested and only Gaussian+MCA produced the resulting model in reasonable time.

B. Case 2 - Hallway

The second case is a scan of a hallway at Jacobs University in Bremen, Germany. This data set consists of 195629 unorganized points in Euclidean space. This model is divided in three parts, since the laser scans are acquired from three different positions. First, data set is registered using 6D Slam technique of iterative - closest - point algorithm, which gives transformation matrices for three parts of data sets [1], [2].

We present the model of the hallway for space subdivision of 150 with 20% decimation, and for space subdivision 50 with no decimation in Fig. 5.

As in previous case, we test the execution times and number of generated polygons for space subdivisions of 50, 100, and 150 without decimation, and for decimation of 10%, 20%, and 50% with fixed space subdivision of 100. Results are given in Tables IV, and V.

Table III
COMPARISON OF SEVERAL 3D MODEL RECONSTRUCTION ALGORITHMS

Algorithm	Error (cm)	Execution time (seconds)
Gaussian+MCA	< 5cm	13.7 seconds
Hoppe et al.	< 0.3cm	37.6 seconds
Ball pivotting	< 0.5cm	22.7 seconds
Poisson reconstruction	< 0.1cm	56.2 seconds

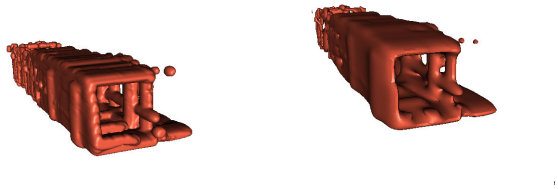

Figure 5. Model of the hallway, subdivision 150 decimation 20% (left),subdivision 50, decimation 0% (right)

Table IV
RESULTS FOR CASE 2

Subdivision	50	100	150
Execution Time	4.6 seconds	20.6 seconds	45.9 seconds
Number of Polys	37616	182164	437036

Table V
RESULTS FOR TEST 2

Decimation	10%	20%	50%
Execution Time	20.6 seconds	20.7 seconds	20.6 seconds
Number of Polys	164459	146335	93072

C. Case 3 - Room

The third case in this paper is a scan of one of the rooms at Jacobs University in Bremen, Germany. This case is the most complicated one as it contains 1003684 unorganized points in Euclidean space. The room, as any other, contains several tricky artifacts (tables, chairs, other furnitures, and also people) which really puts this designed filter chain to the test. Data set is again registered using 6D SLAM technique of iterative - closest - point algorithm, which gives transformation matrices for several parts of data sets.

Again, we present the model of the room for space subdivision of 150 with 20% decimation, and for space subdivision 50 without decimation in Fig. 6.

As mentioned before, this model contains some interesting features. We present some of them in Fig. 7.

As in previous cases, we test the execution times and

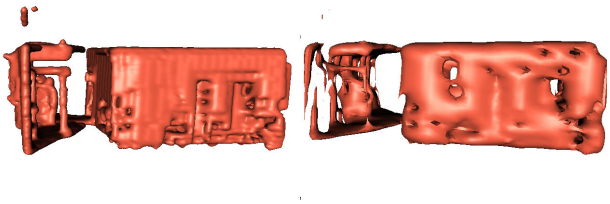

Figure 6. Model of room (external view), subdivision 150 decimation 20% (left),subdivision 50, decimation 0% (right)

Table VI
RESULTS FOR CASE 3

Subdivision	50	100	150
Execution Time	6.3 seconds	20.4 seconds	44.1 seconds
Number of Polys	37724	167072	389340

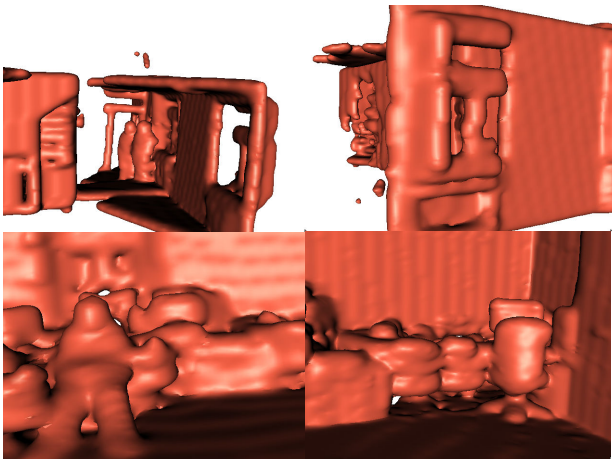

Figure 7. Features from room data set, 2 people standing (upper - left), window (upper - right), man sitting in the chair (bottom - left), chair, table, and computer monitor (bottom - right)

Table VII
RESULTS FOR CASE 3

Decimation	10%	20%	50%
Execution Time	20.5 seconds	20.5 seconds	20.6 seconds
Number of Polys	151106	133961	84258

number of generated polygons for space subdivisions of 50, 100, and 150 without decimation, and for decimation of 10%, 20%, and 50% with fixed space subdivision of 100. Results are given in Tables VI, and VII.

IV. RESULTS (SCALAR FIELD MAPPING)

In this section we present the final results of the reconstruction of the 3D model of indoor environment with added temperature data. In Fig. 8 4 views of the scanned indoor environment are shown.

The model contains close to 300,000 points for a volume of less than $20m^3$. Compared to the data sets presented in [14], [15], [16], where other reconstruction algorithms are applied, this data set is rather small. Therefore, the precision of the reconstruction leaves room for improvement. It is clear that geometry is fully preserved. The room dimensions

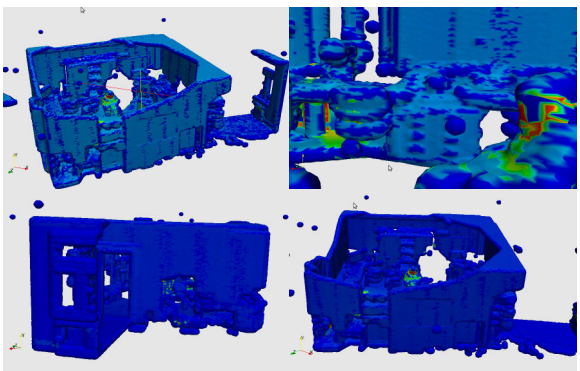

Figure 8. Images of reconstructed scene from scanned indoor environment

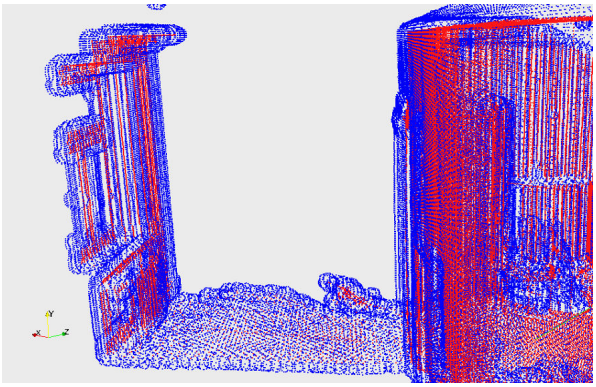

Figure 9. Comparison of the original point cloud (red) and the point cloud of the reconstructed mesh (blue)

are unchanged by the reconstruction algorithm, but it is visible that precision decreases if the number of points per unit volume is smaller (e.g. furniture and people looks very distorted).

The overall evaluation of the reconstructed model can be done by visual expertise. Features like model dimensions compared to original point cloud dimension can be evaluated easily. This is shown in Fig. 9 where the reconstructed model is discretized to point cloud and compared to the original (or input) data.

Up to our knowledge no method for computational evaluation of the precision for reconstructed model is presented, but measuring the distance between the corresponding points of point cloud and reconstructed model we get errors ranging from $1mm$ to $5cm$.

V. CONCLUSIONS

In this paper we presented filter chain design for reconstructing the 3D model of indoor environment from unorganized set of points in Euclidean space acquired from 3D laser scans. Using the 6D slam technique, and iterative - closest - point algorithm data sets are registered to the global coordinate system.

After that, the data set is transformed with designed filter chain, and the 3D model is reconstructed using marching cubes algorithm.

Three cases of the filter chain test are performed using different data sets of unorganized points in Euclidean space. Tests clearly showed that it is possible to get a precise enough 3D model of environment in a reasonable time.

We found out that the quality of the model is not decreasing with decimation of polygons. Decimation value was 10, 20, and 50%, and the results showed that number of polygons really decreased in the model making it easier to manipulate, and inspect. Also, we observed the decrease in file size from 3.5 MB up to 1.8 MB for 50% decimation without using the quality of the model.

For the future work we intend to work on increasing the precision of the model to sub-centimeter range. This can be achieved by manually removing the points that are falsely

recorded now, but we will work on automatic removal of those points from the reconstruction. Also, the work on searching for heat leaks in reconstructed model based on gradient field analysis is in progress.

REFERENCES

[1] A. Nuechter, K. Lingemann, J. Hertzberg, and H. Surmann, "6d slam - 3d mapping outdoor environments: Research articles," *J. Field Robot.*, vol. 24, pp. 699–722, August 2007.

[2] A. Nuechter, *3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom*. Springer, 2009.

[3] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graph.*, vol. 22, pp. 463–470, July 2003.

[4] C. So, G. Baciu, and H. Sun, "Reconstruction of 3d virtual buildings from 2d architectural floor plans," in *Proceedings of the ACM symposium on Virtual reality software and technology*, ser. VRST '98. New York, NY, USA: ACM, 1998, pp. 17–23.

[5] A. J. F. Kok and R. van Liere, "A multimodal virtual reality interface for 3d interaction with vtk," *Knowl. Inf. Syst.*, vol. 13, pp. 197–219, October 2007.

[6] H. Wang, "Three-dimensional medical ct image reconstruction," in *Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation - Volume 01*, ser. ICMTMA '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 548–551.

[7] H. E. Cline, W. E. Lorensen, S. P. Souza, F. A. Jolesz, R. Kikinis, G. Gerig, and T. E. Kennedy, "3d surface rendered mr images of the brain and its vasculature." *Journal Of Computer Assisted Tomography*, vol. 15, no. 2, pp. 344–351, 1991.

[8] R. Motiur, U. Shorif, and H. Mosaddik, "3d segmentation and visualization of left coronary arteries of heart using ct images," *International Journal of Computer Applications*, no. 2, pp. 88–92, 2010.

[9] P. Navratil, J. Johnson, and V. Bromm, "Visualization of cosmological particle-based datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1712–1718, November 2007.

[10] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics (2nd Edition)*. Prentice Hall, 1997.

[11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, pp. 71–78, July 1992.

[12] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 303–312.

[13] T. Schreiber, F. Isselhard, and G. Brunnett, "Two approaches for Polyhedral Reconstruction of 3D Objects of Arbitrary Genus," *International Journal of Vehicle Design, Vol. 21, No. 2*, pp. 292–302, 1999.

[14] K. Inc., *The VTK User's Guide, Version 4.4*. Kitware Inc., 2004.

[15] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70.

[16] I. Mohamad and A. Bade, "Butterfly subdivision scheme used for the unorganized points reconstruction in virtual environment," in *Proceedings of the 2009 International Conference on Computer Technology and Development - Volume 02*, ser. ICCTD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 345–349.

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, pp. 163–169, August 1987.