

Hybrid Computing Algorithm in Representing Solid Model

Muhammad Matondang and Habibollah Haron

Department of Modeling and Industrial Computing, Universiti Teknologi Malaysia, Malaysia

Abstract: *This paper presents an algorithm, which is a hybrid-computing algorithm in representing solid model. The proposed algorithm contains two steps namely reconstruction and representation. In the reconstruction step, neural network with back propagation has been applied to derive the depth values of solid model that was represented by the given two-Dimensional (2D) line drawing. And then in the representation step, once the depth value was derived, the mathematical modeling was used to generate the mathematical models to represent the reconstructed solid model. The algorithm has been tested on a cube. Totally, there are eighty-three cubes has been used on the development of neural network model and six mathematical equations yielded to represent each one cube. The proposed algorithm successfully takes the advantages of neural network and mathematical modeling in representing solid model. Comparison analysis conducted between the algorithm and skewed symmetry model shows that the algorithm has more advantages in term of the ease of the uses and in simplifying the use of mathematical modeling in representing solid model.*

Keywords: *Solid model, reconstruction, representation, neural network, mathematical modeling, and hybrid computing.*

Received June 19, 2008; accepted February 25, 2009

1. Overview

Classical theories divide three Dimensional (3D) modeling into four types of model representation. They are wireframe, surface, solid and procedural model [2]. In the past several decades, a number of related researches on that issue have been developed. Focus on solid model representation, the discussion can be divided into two parts, namely the reconstruction and the representation. An approach for the automatic reconstruction of solid models from 2D engineering drawings was introduced [13]. The proposed algorithm [13] is a hybrid of the Boundary representation (Brep) and Constructive Solid Geometry (CSG) methods for reconstruction of engineering drawings. The solid model reconstruction algorithm successfully takes the advantage of both approaches to represent solids. An analytical heuristic method called skewed symmetry [7] has been used in reconstructing 3D object from given 2D line drawing. The proposed algorithm by them is seeking pairs of gradient estimates and then used those values to estimate the z-values of 3D object. Another approach in these issues also has been developed [9]. They applied soft computing approach, namely neural network in reconstructing and representing 3D object. The algorithm successfully takes the advantages of neural network in the both of process. This paper attempts to develop a system which is combined the reconstruction and the representation of the reconstructed solid model from given 2D line drawing. In terms of reconstruction some techniques introduced before are gradient space, linear

systems approach, interactive method, perceptual approach, minimum standard deviation, primitive identification, line labeling [6]. Unfortunately, not all these techniques are suitable for solid model reconstruction from given 2D line drawing. Then, in terms of representation, there are several established techniques, such as constructive solid geometry, boundary representation and sweep representation. However, in this paper mathematical modeling are replaced that techniques in representing solid model.

As a summary, the goal of this paper is to represent solid model from given 2D line drawing. Therefore, the work is divided into two parts: the reconstruction and the representation of the reconstructed solid model. At the end, an algorithm is achieved for both of discussions. Therefore, this paper called a hybrid computing algorithm in representing solid model.

2. Issues on Solid Model Reconstruction and Representation

As mentioned earlier, there are two steps involved in the proposed algorithm, namely the reconstruction of solid models from given 2D line drawing, and the representation of the reconstructed solid model. The first step derived the depth values of the solid model. Then, in the second step the values are employed to represent the solid model as mathematical models. This section presents a brief discussion on issues of them.

2.1. Solid Model Reconstruction

In the past several decades, most of established techniques for solid model reconstruction are related to mathematical modeling and geometrical manipulation. In some cases, those techniques are very complicated with the analytical problems. However, in the meantime the uses of soft computing as a modern approach have become a new tremendous approach in the area of solid reconstruction.

Estimating and forecasting approaches using soft computing to define the optimum result are much preferred than exact techniques. Few previous works, explore the adaptive reconstruction of freeform objects with 3D Self-Organizing Maps (SOM) neural networks grid [1], explore the ability of neural networks in learning through experience when reconstructing an object by estimating its z -coordinate [9], use genetic algorithms on 3D shape reconstruction of template models [4], use neural networks and adaptive geometry meshes as a method for surface reconstruction [5]. The method is very successful in reconstructing forms with different geometry. A method based on neuro-fuzzy modeling for automatic 3D object recognition and reconstruction also has been developed [10]. So far, [14] carried out the latest research on reconstruction area in case the uses of neural network. They applied neural network with powerful property of approximation to reconstruct complex objects based on fringe projection. Other success in reconstruction of solid models gives strongest motivation for this research to use neural network approaches on solid model reconstruction. Therefore the proposed algorithm focuses on the development of neural network with back propagation to achieve the target on the first step of the proposed algorithm.

2.2. Mathematical Model for Solid Representation

Solid model is the unambiguous representation of the solid parts of an object. For representing it, there are several established techniques. They can be categorized into implicit scheme, enumerative scheme, boundary schemes [12], and deformational schemes [2]. The most techniques used for representing solid models are CSG, Boundary representation (B-rep) and sweep representation. However, in the proposed algorithm, none of the above is discussed or used because the reconstructed solid models are represented as mathematical models.

As an abstract model, a mathematical model uses mathematical language to describe the behavior of a system. It can take many forms, including but not limited to dynamical systems, statistical models, different equations, or game theoretic models. A mathematical model as a representation of the essential aspects of an existing system (or a system to be

constructed) which presents knowledge of that system in usable form is defined [3]. Generally, there are four steps of general approach to describe or represent a system or object to the mathematical modeling. The steps are:

- Identify the problem, define the terms in the problem, and draw diagrams where appropriate;
- Begin with a simple model, stating the assumptions that make the mathematical model as the focus on particular aspects of the phenomenon;
- Identify important variables and constants and determine how they relate to each other;
- Develop the equation(s) that express the relationship between the variables and constants.

Once the model has been developed and applied to the problem, it must be analyzed and interpreted with respect to the problem. The interpretations and conclusions should be checked for the accuracy. The proposed algorithm used mathematical equations to represent the reconstructed solid model. Based on the discussion on previous works and reports, the author found that there are no works or reports that use mathematical model in representing solid model from given 2D line drawing. Therefore, it can be said that this paper is the primarily research on that.

3. The Hybrid Computing Algorithm

This section explained the proposed algorithm into two parts. They are the framework and the hybrid process.

3.1. The Framework of Solid Model Reconstruction and Representation

There are few assumptions that have been made to simplify the implementation of this research work and the contributions. First, the solid models tested are assumed as an engineering sketch in the form of 2D line drawing that represent solid model on graph paper. Second, the 2D line drawing is assumed to represent a valid solid model where all unwanted junctions or lines have been removed and there are no unconnected junctions or lines. Third, the solid model is assumed as a 2D line drawing with all informative lines shown. Fourth, there is only one hidden junction in the backside of the solid model. Fifth, this research assumed that the (x, y, z) values of solid models are known for the comparison purpose. These assumptions make the proposed algorithms more logical or otherwise the engineering sketch is not seen as solid models because the projection is parallel to the other faces of the object. In this case, it is impossible to interpret, reconstruct and represent the sketches as solid model and hence the analysis of the accuracy of the results simpler. Figure 1 shows an engineering sketch of 2D line drawing that represent solid model.

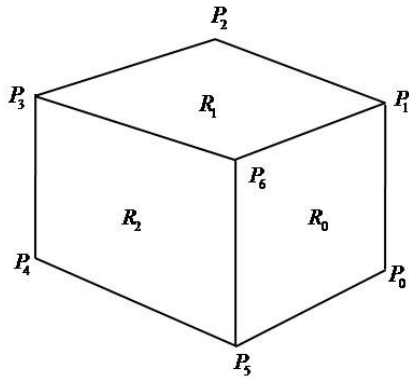


Figure 1. Cube, an engineering ketch of 2D line drawing.

In reconstruction process, neural network with back propagation is employed to derive the depth values of solid model from given 2D line drawing. For that purposed, neural network models should be trained in order to achieve the best of neural network model in deriving the depth. Once the best neural network model found and the depth derived, then the values are combined with mathematical models to represent the reconstructed solid model. Figure 2 shows the framework of the proposed algorithm.

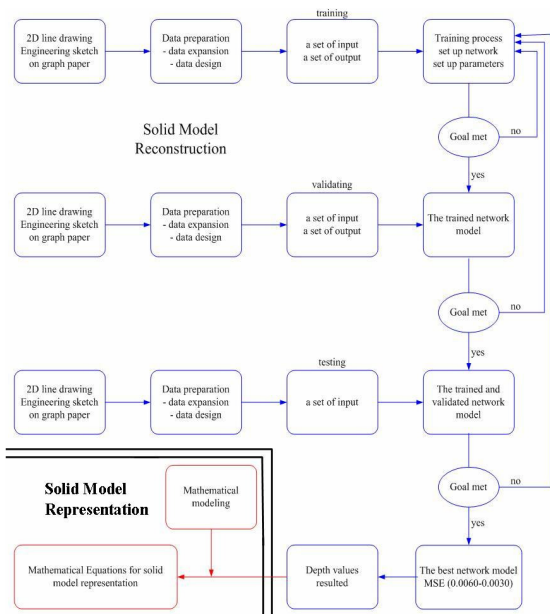


Figure 2. The framework.

3.2. The Hybrid Process

The solid model reconstruction process conducted with the development of neural network model. This process divides into training, validating, and testing. The training and validating process is performed to achieve the best of neural network model. The neural network models are trained with back propagation algorithm. See [8] for the general algorithm of back propagation (see also [9, 12]). Once the best neural network model achieved, then this model used to derive the depth values of the solid.

There are five steps in order to achieve the best neural network model. They are:

1. Obtain a set of data input and data output and design in a column.
2. Set up neural network model. In this research neural network toolbox in MATLAB 7.14 was used to train the neural network models.
3. Set the parameters involved.
4. Start training by applying input and output.
5. Defined the MSE value, if $MSE \leq Error_{max}$, then stop training, else repeat from the second step.

There are forty-two neural network models with the differences on the activation and training functions have been train to achieve the best model of the neural network models in deriving the depth values. Next the depth values of the solid model derived, and then the values are combined with general plane equation to represent the reconstructed solid models as mathematical models. In the proposed algorithm the standard convention of plane equation in form $Ax + By + Cz + D = 0$ is used to represent the solid model, with

$$A = \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{bmatrix}, B = \begin{bmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{bmatrix},$$

$$C = \begin{bmatrix} x_1 & y_1 & 1 \\ x_1 & y_2 & 1 \\ x_1 & y_3 & 1 \end{bmatrix}, D = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \tag{1}$$

and (x_1, y_1, z_1) , (x_2, y_2, z_2) and (x_3, y_3, z_3) are three random junctions of each plane. Using Gauss-Jordan method namely row reduced echelon forms, the value of A, B, C, and D also can be defined. Next section presents an experimental result of the proposed algorithm.

4. Experimental Result

A cube was drawn on graph paper and the coordinate defined as shown in Figure 3. For neural network training and validating purposed, affined transformation is employed and produced eighty-three cubes used the initial cube which is shown in Figure 3. Thirty-seven cubes for training process, thirty-six cubes for validating process and for testing purposes, there were ten cubes used. These data are design in column and used in training and validating process of the neural network models. Then the best neural network model used to derive the depth values of the testing data. Figure 4 shows the structure of the best neural network model and the parameters used in that model shown in Table 1.

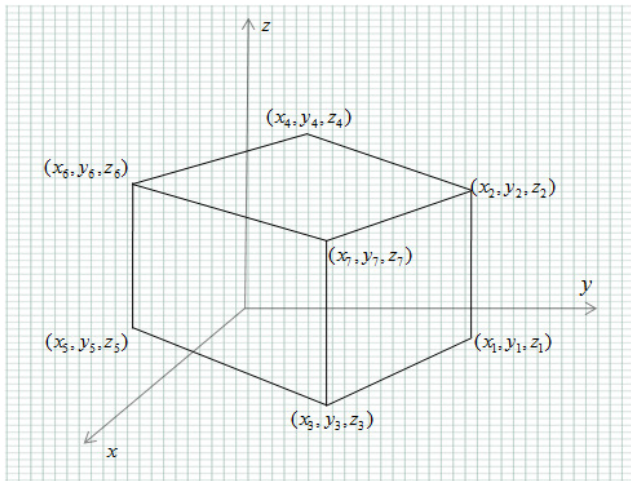


Figure 3. A cube on graph paper.

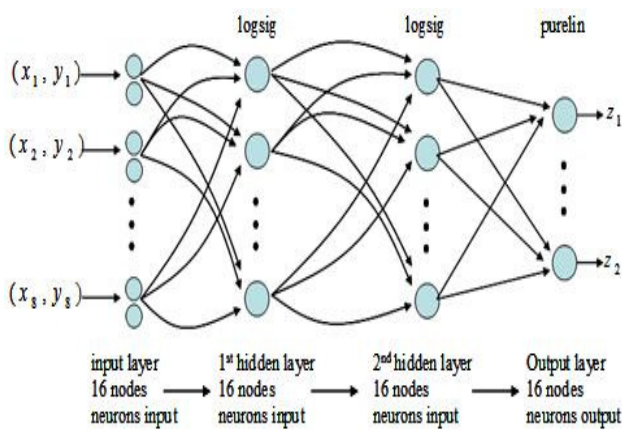


Figure 4. The best neural network structure.

Table 1. Parameters of the best neural network model for cube reconstruction.

Parameter	Value
epochs	200000 (goal met = 196)
goal	0.01
learning rate (lr)	0.6
learning dec (lr_dec)	--
learning inc (lr_inc)	--
max_fail	5 (default)
max_perf_inc	--
momentum constant (mc)	--
min_grad	1.0000e-010
show	100
time	Infinity
training function	Traingd

The value of MSE training, validating and testing process from the best neural network model is shown in Table 2. Followed by Table 3 which shows the target of the depth values of ten cubes and Table 4 shows the result derived by the best neural network model.

Figure 5 shows one of the visualization of the reconstructed cube. The left side is the expected and

the right side is the result. By human visual inspection, it can be shown that neural network with back propagation was successful in deriving depth values, and then reconstruct the cube from the given 2D line drawing. Table 5 shows the coordinates of the two cubes which shown in Figure 5 and the accuracy of the derivation are given.

Table 2. Neural network structure.

Item	Value / Character
Number of hidden layer	Two
Input node	16 nodes
Output node	8 nodes purelin
Hidden node	2 x 16 nodes logsig
MSE value (goal = 0.01)	
Training	0.0100
Validating	0.0101
Testing	0.0047

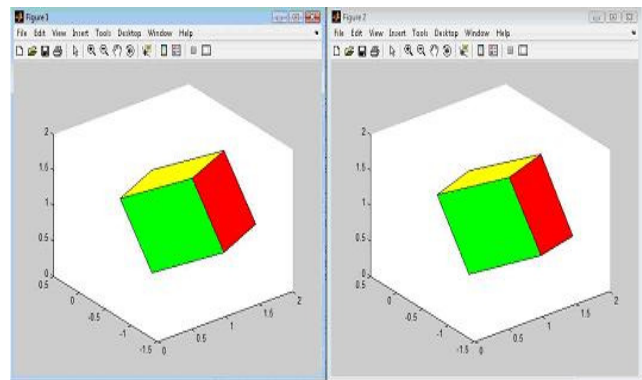


Figure 5. The expected and the reconstructed cube.

Once the depth values are derived, then the (x, y, z) coordinates of each junctions are defined, then the values are substituted to the mathematical models of each junctions in form of $Ax + By + Cz + D = 0$. Table 6 shows the results.

Next, the models shown in Table 6 are grouped consider to each plane. Then the plane equation for each plane is generated. Follow is an example to generate the plane equation for plane 1-5-8-4 (the front plane, as shown in Figure 6. The linear equations of each junction of the plane are:

- 0.9945A - 1.4863B + 0.8581C + D = 0: junct. 1.
- 0.8668A - 1.0386B + 1.7933C + D = 0: junct. 5.
- 0.3695A - 0.2954B + 1.3885C + D = 0: junct. 8.
- 0.4973A - 0.7431B + 0.4221C + D = 0: junct. 4.

Table 3. The target values of depth values of ten cubes.

Cube 1	Cube 2	Cube 3	Cube 4	Cube 5	Cube 6	Cube 7	Cube 8	Cube 9	Cube 10
0.9477	0.8960	0.7957	0.6141	0.3200	0.0115	0.7111	0.6141	0.3649	0.0012
1.0027	1.0103	1.0378	1.1190	1.2050	1.0114	0.5554	1.1190	1.2069	1.0012
1.0522	1.1034	1.1989	1.3394	1.3327	1.0109	0.8011	1.3394	1.3595	1.0012
0.9973	0.9891	0.9568	0.8346	0.4477	0.0109	0.9568	0.8346	0.5174	0.0012
1.9450	1.8851	1.7525	1.4487	0.7677	0.0224	1.6679	1.4487	0.8823	0.0024
1.9999	1.9994	1.9946	1.9536	1.6527	1.0224	1.5122	1.9536	1.7243	1.0024
2.0495	2.0925	2.1557	2.1740	1.7805	1.0218	1.7578	2.1740	1.8769	1.0024
1.9945	1.9781	1.9135	1.6691	0.8955	0.0219	1.9135	1.6691	1.0349	0.0024

Table 4. The result values of depth values of ten cubes.

Cube 1	Cube 2	Cube 3	Cube 4	Cube 5	Cube 6	Cube 7	Cube 8	Cube 9	Cube 10
1.0293	1.0059	0.9446	0.7158	0.2469	0.0211	0.8697	0.7158	0.2904	-0.0705
1.0125	1.0138	1.0263	1.0645	1.1958	1.1057	0.6752	1.0645	1.1803	0.9425
0.9904	1.0341	1.1339	1.3393	1.3885	1.0703	0.8132	1.3393	1.4103	1.0513
1.0330	1.0267	0.9972	0.8816	0.4221	0.0013	0.9445	0.8816	0.4994	0.0176
2.0315	1.9697	1.8279	1.4511	0.6059	0.2153	1.7098	1.4511	0.7136	0.1501
1.9979	1.9815	1.9642	1.9486	1.5908	1.0877	1.4913	1.9486	1.6753	1.0841
1.9913	2.0056	2.0476	2.1947	1.7933	1.0244	1.6698	2.1947	1.9413	0.9526
1.9666	1.9558	1.9105	1.7139	0.8581	0.0238	1.9370	1.7139	1.0188	0.0666

Table 5. The coordinate of the two cubes.

Junction	The Expected Depth Values	The Derived Depth Values	Accuracy (%)
1	0.3200	0.2469	77.16
2	1.2050	1.1958	99.24
3	1.3327	1.3885	95.81
4	0.4477	0.4221	94.28
5	0.7677	0.6059	78.92
6	1.6527	1.5908	96.25
7	1.7805	1.7933	99.28
8	0.8955	0.8581	95.82

Table 6. Linear equation of each junction.

Junction	The Equation
1	0.9945A -1.4863B + 0.8581C + D = 0
2	1.8527A - 0.9890B + 0.6059C + D = 0
3	1.3554A -0.2459B + 0.2469C + D = 0
4	0.4973A - 0.7431B + 0.4221C + D = 0
5	0.8668A - 1.0386B + 1.7933C + D = 0
6	1.7249A - 0.5413B + 1.5908C + D = 0
7	1.2277A + 0.2019B + 1.1958C + D = 0
8	0.3695A - 0.2954B + 1.3885C + D = 0

Next, the values of *A*, *B*, *C* and *D* are calculated, it is used to generate the plane equation of each plane. In this paper, for simplicity Gauss-Jordan method is used to calculate the values. The way is: by random selection, select three junctions of 1-5-8-4 and then the linear equations are arranged into a matrix equation as follow. Let say 1-5-8 junction is selected, and then a linear system is generated as follow:

$$\begin{bmatrix} 0.9945 & -1.4863 & 0.8581 & 1 \\ 0.8668 & -1.0386 & 1.7933 & 1 \\ 0.3695 & -0.2954 & 1.3885 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

Execute an operation called row reduced echelon form (Gauss-Jordan method), the solution for equation 2 is given by

$$\begin{matrix} A & B & C & D \\ \left[\begin{array}{cccc} 1 & 0 & 0 & -140.6289 \\ 0 & 1 & 0 & -82.9340 \\ 0 & 0 & 1 & -20.4996 \end{array} \right] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{matrix} \quad (3)$$

From equation 3, assumed that *D* is equal to 1, then *A* = 140.6289, *B* = 82.9340, *C* = 20.4996, and *D* = 1. Substitute the values into the plane equation *Ax* + *By* + *Cz* + *D* = 0., then 140.63x + 82.93y + 20.50z + 1=0 is the plane equation for plane 1-5-8-4. By perform the same technique, the plane equation for the other planes could be generated. Table 7 shows the complete

mathematical models for the cube which is shown in Figure 6.

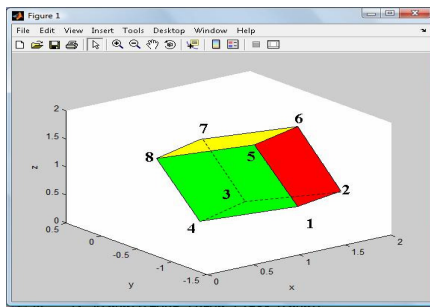


Figure 6. The reconstructed cube.

Table 7. Cube representation.

Plane	Plane Equation
1-5-8-4-1	$140.63x + 82.93y + 20.50z + 1 = 0$
2-3-7-6-2	$0.85x + 0.50y - 0.12z - 1 = 0$
1-2-6-5-1	$-0.28x + 0.37y - 0.20z + 1 = 0$
3-4-8-7-3	$-0.52x + 0.75y - 0.43z + 1 = 0$
1-2-3-4-1	$-0.07x - 3.50y - 7.14z + 1 = 0$
5-6-7-8-5	$0.04x - 0.40y - 0.81z + 1 = 0$

As a summary, Figure 7 shows the hybrid process in the experimental result.

5. Result Analysis

This section presents an analysis of the result of the research. To simplify the analysis, a comparison to a previous works by [7] is given. The conveniences of the implementation of the result between these two algorithms are compared. Based on previous work [7] which used skewed symmetry to derive the depth value of 3D object, there are some criteria that need to be comparing to the proposed algorithm. Involved to the ease and the difficulties, the proposed algorithm is prefer then skewed symmetry. The skewed symmetry has to define the best pair of gradient estimates before deriving the depth values. However, in the proposed algorithm does not. According to the process of derivation, skewed symmetry has to deal with many geometrical analyses; meanwhile the proposed algorithm minimized the analysis. One of the weaknesses of the proposed algorithm is impacted by the data that to be used during the development of the neural network model, where there are many data that have to be presents to train and validated the neural network models. Table 8 shows the summary of the comparison between the two algorithms.

Compare to a previous work by [9], the two techniques are different in case of the input and the object used to develop the neural network model. The difference is also involved to the experimental data design used, where in this research the data are design in column, while [9] designed the data in row.

Designing the data in column gives a contribution to the developed neural network models in deriving the depth values and the hidden junction to be more accurate, while it is not good enough if they are designed in row.

Table 8. Comparison of the proposed neural network model to an algorithm proposed by [7].

Criteria	The Proposed Neural Network Models	Previous Algorithm[7]
Input	<ul style="list-style-type: none"> solid model by given engineering sketch of 2D line drawing the (x,y)-coordinates of the sketch for deriving the depth value the (x,y,z)-coordinates of the sketch for deriving the hidden junction (invisible junction) 	<ul style="list-style-type: none"> 2D line drawing of 3D object the (x,y)-coordinates of 3D object for estimating the z-values
Method Used	<ul style="list-style-type: none"> neural network with back propagation 	<ul style="list-style-type: none"> skewed symmetry
Geometrical Analysis	<ul style="list-style-type: none"> it is easy to use the solid model which drawn on graph paper and extract all the information needed , such as the coordinates 	<ul style="list-style-type: none"> for extracting all the information needed, the 3D object should be drawn on graph paper with the actual size, then divided it into each regions or plans
The Ease of The Calculation	<ul style="list-style-type: none"> the calculations are done by computational method 	<ul style="list-style-type: none"> done by computational, but few parts should be done by manually
Data Used	<ul style="list-style-type: none"> the data are divided into three sets, set data for training, set data for validating and the testing data 	<ul style="list-style-type: none"> 1 for 1 testing
Accuracy	<ul style="list-style-type: none"> good and can be improved 	<ul style="list-style-type: none"> good
Other Tools Involved	<ul style="list-style-type: none"> graph paper affine transformation MATLAB 7.14 	<ul style="list-style-type: none"> graph paper MATLAB 7.14
Beautification	<ul style="list-style-type: none"> good by human visual inspection 	<ul style="list-style-type: none"> good by human visual inspection
Output	<ul style="list-style-type: none"> depth values 	<ul style="list-style-type: none"> z-values
Object Tested	<ul style="list-style-type: none"> cube 	<ul style="list-style-type: none"> cube

Since there are no specific works in using mathematical modeling for representing solid model by given 2D line drawing, then it can be noted that this research is the first report on it. According to the situation, the analyses of the proposed algorithm only performed a comparison among the results to the expectation and the performance. However, the proposed algorithm has been tested to a cube, and as the results there are six equation models for representing it. Then, if the results are compared to the result, if the same things do with an establish methods or software, there are few disadvantages in using the proposed algorithm for representing solid models in terms of the simplicity and the convenience.

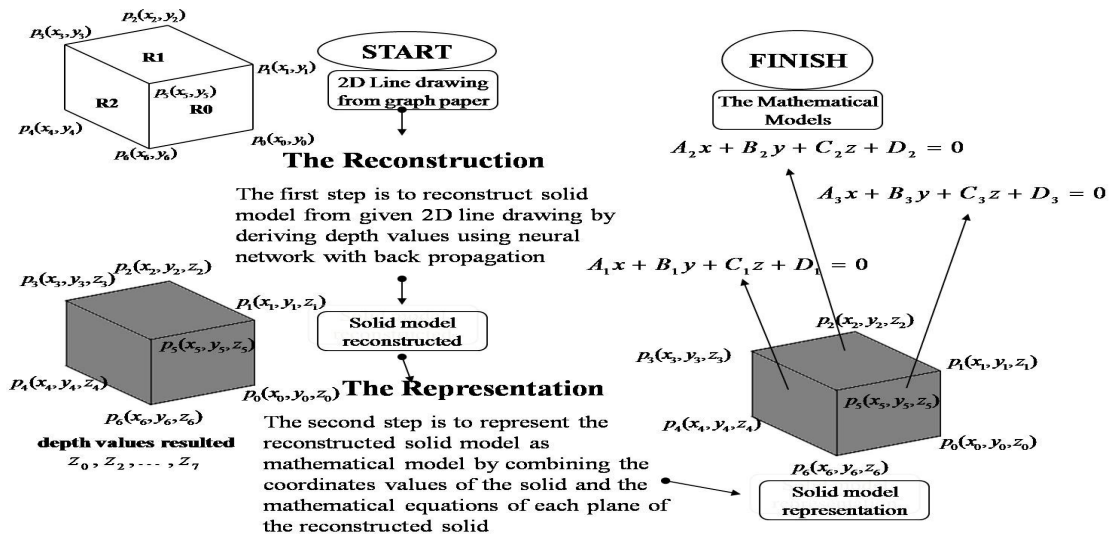


Figure 7. The steps of the hybrid process in the experimental result.

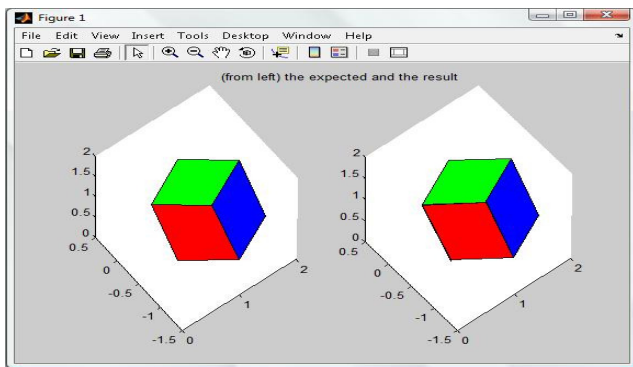


Figure 8. The visualization of the mathematical models for a cube from given 2D line drawing.

Anyway, the proposed algorithm could be used as an alternative or new method for represent solid model by given 2D line drawing since there are no works focused on it. Any improvements for future works will producing a good result as like as the expectation. Figure 8 shows the visualization of the mathematical models for solid model from given 2D line drawing which is shown in Table 8 compare to the expectation result (the left side is the expected result and the right side is the result).

6. Conclusions

This paper has presented a hybrid-computing algorithm in representing solid model that employed two techniques those are neural network with back propagation, and mathematical modeling. The neural network with back propagation technique is used to reconstruct solid model from given 2D line drawing by deriving the depth values. Once the solid model is reconstructed, then the algorithm is extended to represent it as mathematical model.

The analysis of the result also shows that the proposed algorithm is much better than the algorithm proposed [7] in terms of reconstruction but not in term

of representation. For future works, the proposed algorithm should be developed to enhance solid model representation from given 2D irregular line drawing.

Acknowledgement

The authors would like to thank Ministry of Science, Technology, and Innovation (MOSTI) for the e-Science grant and Research Management Center (RMC), Universiti Teknologi Malaysia (UTM) for the support in making this projects success.

References

- [1] Barhak J. and Fischer A., “Adaptive Reconstruction of Freeform Objects with 3D SOM Neural Network Grid,” in *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, Japan, pp. 97-195, 2001.
- [2] Cretu M., “3D Object Modeling-Issues and Technique,” *PhD Thesis*, University of Ottawa, 2003.
- [3] Eykhoff P., *System Identification*, John Wiley, 1974.
- [4] Fayolle A., Rosenberger C., and Toinard C., “3D Shape Reconstruction of Template Models Using Genetic Algorithms,” in *Proceedings of 17th International Conference on Pattern Recognition*, UK, pp. 269-272, 2004.
- [5] Junior B., Neto D., and De Melo D., “Surface Reconstruction Using Neural Networks and Adaptive Geometry Meshes,” in *Proceedings of IEEE International Joint Conference on Neural Networks*, USA, pp. 803-807, 2004.
- [6] Lipson H. and Shpitalni M., “Correlation-Based Reconstruction of a 3D Object from a Single Freehand Sketch,” *International Proceedings of AAAI Spring Symposium Series-Sketch Understanding*, USA, pp. 99-104, 1996.

- [7] Matondang Z., Haron H., and Talib S., "Three-Dimensional Visualization of Two-Dimensional Data: The Mathematical Modeling," in *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics Statistics and Applications*, Malaysia, pp. 79-87, 2006.
- [8] Negnevitsky M., *Artificial Intelligence: A Guide to Intelligence Systems*, Addison-Wesley, 2005.
- [9] Peng W. and Shamsuddin M., "3D Object Reconstruction and Representation Using Neural Networks," in *Proceedings of Association for Computing Machinery*, Redwood City, pp. 139-147, 2004.
- [10] Samadzadegan F., Azizi A., Hahn M., and Lucas C., "Automatic 3D Object Recognition and Reconstruction Based on Neuro-Fuzzy Modeling," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 6, no. 3, pp. 255-257, 2005.
- [11] Shapiro V., *Handbook of Computer Aided Geometric Design*, Elsevier Science Publishers, 2002.
- [12] Subri H., "Aplikasi Rangkaian Neural Dalam Pengesan Simpang Bagi Penterjemah Lakaran Pintar," *Master Thesis*, Universiti Teknologi Malaysia, 2006.
- [13] Wang Z. and Latif M., "Reconstruction of a 3D Solid Model from Orthographic Projections," in *Proceedings of the International Conference on Geometric Modeling on Graphics*, Japan, pp. 75-82, 2003.
- [14] Yan T., Wen-jing C., Xian-yu S., and Li-qun X., "Neural Network Applied to Reconstruction of Complex Objects Based on Fringe Projection," *Computer Journal of Optics Communication*, vol. 12, no. 6, pp. 274-278, 2007.



Muhammad Matondang is master student in computer science and information system at the Universiti Teknologi Malaysia, Johor, Malaysia. He holds Bachelor degree in mathematics from the University of Sumatera Utara, Medan, Indonesia in 2005, and now still continuing his study in master degree.



Habibollah Haron is a senior lecturer and head of Department of Modeling and Industrial Computing, Faculty of Computer Science and Information Systems, UTM. he obtained his PhD from UTM in 2004.

