

A Genetic Algorithm based Optimization Method in 3D Solid Reconstruction from 2D Multi-View Engineering Drawings

Hamid Haghshenas Gorgani* and Alireza Jahantigh Pak

Engineering Graphics Center, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:

Received: 09 Jun. 2018

Accepted: 22 Mar. 2018

Keywords:

3D RECONSTRUCTION
ENGINEERING DRAWING
GENETIC ALGORITHM
B-REP

ABSTRACT

There are mainly two categories for a 3D reconstruction from 2D drawings: B-Rep and CSG that both these methods have serious weaknesses despite being useful. B-Rep method which has been older and have wider function range is problematic because of high volume of calculations and vagueness in answers and CSG method has problem in terms of very limited range of volumes and drawings that it can analyze. Proposed method in this paper is an innovative method based on B-Rep in which optimization of genetic algorithm has been used to identify the relationship among the components of various views in 2D drawings. Using genetic algorithm that is a stochastic algorithm contributes that high volume of calculation that is one of main weaknesses of B-Rep method is solved. Moreover, considering correspondence condition of one to one among response in this method has caused that vagueness problem which is another weakness of B-Rep method to be almost solved so it can be said in addition to having wide range, present method doesn't have common problems of B-Rep method and it even turns it to an effective method.

1. Introduction

As we know an image speaks a thousand words, on the other hand compared to 2D drawings, a 3D model is Realistic, easy, quick [1], better for working and project approval, easy remodeling and corrections, easy precision and control, easy scenario visualization, more effective in communication with suppliers, have ability to rendering and animations, test and validate our designs to reduce cost from quality problems and errors, reduce the need and cost of physical prototype, allow non-technical personals to understand our designs, have higher ideality degree [2] so the ability of converting engineering 2D drawings to 3D models is very important. These models cover a wide range of applications, for example in analysis of vibration and flexural sensitivity of atomic force microscopes [3].

The attempts that have been conducted in the field of converting 2D engineering drawings to 3D models can be divided into two main categories; B-Rep (Boundary Representation) and CSG (Constructive Solid Geometry). B-Rep method includes three stages; converting 2D vertices to 3D vertices, obtaining 3D edges through connecting these vertices and ultimately creating 3D plates from linking these Edges and formation of final volume. This method was first proposed by Idesawa in 1973 [4] and then was developed by Hamer and Woo [5] and Preiss [6]. After that it was turned to a comprehensive method by Markowsky and Wesley [7, 8]. The attempts of these two people were converted to more effective and accurate algorithm by Yan et al [9], Shin et al [10], Kou [11] and Liu et al [12].

On the contrary CSG method is on this basis that final 3D model is a combination of primitive volumes. These primitive volumes are identified in 2D drawings through recognizing their patterns and then extrusion will be implemented on them. In next stage obtained volumes will go under Boolean operations and the final answer will be obtained. This method was first discussed by Aldefeld in 1983 [13]. Shum made this method more complete in 1997 and 2001 [14, 15]. Cicek and Gulestin also introduced in two-step method for CSG in 2001 [16].

Both these methods have serious weaknesses despite being useful. Because of calculative problems and vagueness in responses and the possibility of absence of answer in some cases in B-Rep method, we face serious problems. Main problem in this method emerges when equivalent features are supposed to be identified in two various views or in another word the elements of two various views are supposed to be related to each other. CSG method although doesn't have mentioned problems and its answers are usually certain but the range of shapes which it includes is very limited [17].

On the other hand, genetic algorithm which is a stochastic optimization method can help us to achieve a good answer without requiring high memory and calculations. Genetic algorithm is specifically used for optimizing a particular fitness function using similar method of living organisms. The individuals of this algorithm are probable answers that can optimize desired function. Each individual becomes as a string that is called chromosome. All chromosomes that are experimented in one stage are called a population. Genetic algorithm was first proposed by John H.

* Corresponding author, Tel.: +98-21-66164823, Email Address: h_haghshenas@sharif.edu

Holland in 1975 [18]. Gradual evolutionary algorithms like genetic meet the principle of survival of the best fitness for making answers better and better. This characteristic encourages us to use GA in identifying the components of a 2D CAD model in which there are many elements and analytical detection process in it is less effective because of high number of variables that are used in each iteration. GA is very different from ordinary optimization methods as well as stochastic methods because the other answers only concentrate on one probable answer while GA investigates on one population of probable answers.

GA was first used to identify the shapes by Levine and Roth [19, 20] in 1993 and 1994 then in 1994, Lutton improved it somehow [21] and after that in 2000, it got more complete by Ma and Yuen [22]. Some new applications of Genetic algorithm in industry have done, for example in process parameters optimization [23, 24], Design of PID controller [25] and in internal and string stability of longitudinal platoons of vehicles [26]. Some attempts were conducted to identify shape as well as traffic lights and signs using GA by Mainzer in 2002 [27, 28]. Zhang in 2003 [29] and then Yao in 2004 [30] proposed some algorithms using GA to identify elliptic elements. After that in 2005, within a paper, Pralay Pal et al Discussed a relatively newer concept of introduction of Genetic Algorithm for Features Recognition (GAFR) from large CAD databases, which was significant in view of the growing product complexity across all manufacturing domains [31]. Some new applications of Genetic algorithm industrial are

The first attempt to use GA subjecting 3D reconstruction was also done by Zhang Chen et al in 2002 in which Based on holo extraction of information from paper drawings, this paper develops a systematic 3D reconstruction method, which simulates an experienced human designer's thinking mode in transforming inaccurate outlines with inaccurate projection relationships in 2D drawings into 3D image, with the aid of Genetic Algorithms [32].

Recently also in 2010, Siddique et al 2010 proposed a new method in this case [33] meanwhile Kabolizade et al proposed a method in 2012 in which they discussed Design and implementation of an algorithm for automatic 3D reconstruction of building models using genetic algorithm [34].

Proposed method in this paper is more based on B-Rep but using genetic algorithm that is a stochastic method and moreover operates on a population of answers, decreases the number of necessary equations that is considered as one of main weaknesses of B-Rep method to the great extent. Considering correspondence condition of one to one among response in this method has caused that vagueness problem to be almost solved.

The requirements for equalizing two elements in two different views in terms of analytical geometry attitude have been first investigated and determined and a formula has been proposed due to calculate error level of assuming equivalent of two selected elements and its less amount obviously indicates more equivalent possibility. Then the problem of creating relationship between two different views has been formed as a standard optimization problem with fitness and extra conditions. Mentioned optimization problem is solved by genetic algorithm with all its characteristics in next step and the way of interpreting answers have been mentioned and after that obtaining 3D model from the points that in this step have been changed from 2D points to 3D points, formation of 3D edges and finally formation of answer volume have been discussed. To percept this issue better, a numerical case study derived from real engineering problems has been proposed, solved and its results have been finally discussed.

2. Equivalent components in different views:

One of the most difficult steps in reconstruction process is to find the image of a 3D component in an orthographic view and

then find the image of the same component in another view to put them in equivalence and determine that 3D component in space without having 3D volume. For this purpose, we use the surface analysis method.

In general, each face in a polygonal form, in an orthographic view, can be equivalent to a polygon or a line on another orthographic view (both are images of the same 3D component from different viewing angles). Consider Figure 1 and reverse the problem, with a 3D object, we can find images of different faces from A to H in front and top views. For example, let's take a look at face C. This face is in the form of a polygon in the front view, and in the top view it is also seen in the form of a polygon. What is noteworthy is that the corners of the two polygons are in a one-to-one correspondence. The same thing can be found for the polygons F and G. Now, look at the face H in the front view which is in the form of a polygon while in the top view it is seen in the form of a line. As the figure shows, for each corner of the polygon H, an intersection on the H line is seen, with which the corner is aligned. Of course, this correspondence is not necessarily one-to-one. The same is true for faces A, B & E. So we can make two important conclusions about the equivalent components in different orthographic views of an object:

- A polygon in an orthographic view of an object can be equivalent to a line in another orthographic view of the same object if, for all the corners of that polygon, there is an intersection point on the line, which of course is not necessarily one-to-one correspondence.
- Two polygons in two different orthographic views of an object can be equivalent if, there is a one-to-one correspondence between their corners.

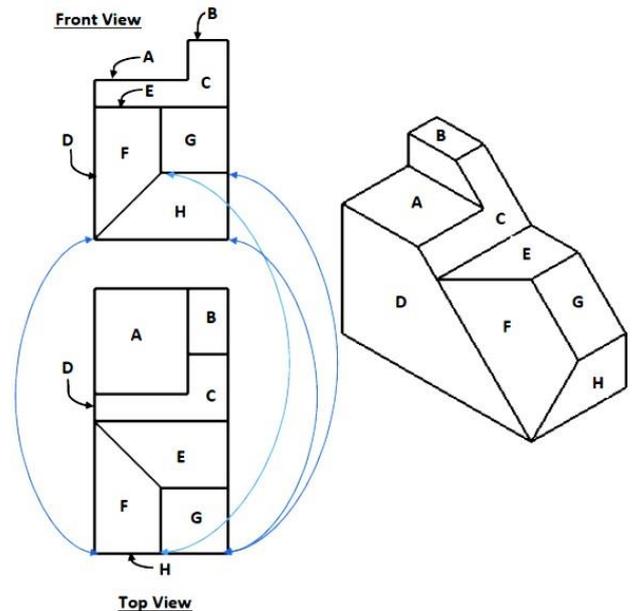


Figure 1: Front and top views for a 3D object and the status of the faces in them

Meanwhile, currently, the proposed method applies only to bodies with flat surfaces.

Formulating the problem above is as follows:

For all corners of desired polygon in first image, related components to common axis of two images are stored in a vector named A. then in second image if we face another image, related coordinate to the common axis for all corners and if we face one line, related coordinate to common axis of all intersection points will be stored in a vector named B. if polygon of the first image

has m corners and the polygon of the second image has k corners or the line of second image has k intersection points, it will be as followed:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}$$

It is defined that:

$$\text{Differ} = \begin{bmatrix} |a_1 - b_1| & \dots & |a_m - b_1| \\ \vdots & \ddots & \vdots \\ |a_1 - b_k| & \dots & |a_m - b_k| \end{bmatrix} \quad (1)$$

And then minimum amount of each matrix column that shows minimum distance is between which two corners or intersection points will be stored in matrix MnDfAB. About equalizing two surfaces, it has to be noted that if the minimum of first column happens in line i, existing component in i level cannot be selected as minimum in next column and minimum should be searched among other components not to damage correspondence one to one. About equalizing the surface with line, this one to one correspondence is ignored. The mode may have happened here is that all lines are used as minimum and there is no other line but there are still some columns for minimizing. In this mode, 2 times bigger than the amount of matrix Differ is recorded as minimum that is considered as a kind of penalty method.

After the steps above, variable SAB is defined as below:

$$SAB = \sum_{i=1}^n MnDfAB_i \quad (2)$$

In equalizing two surfaces, the reverse form of operations above is required to be iterated one time so SBA if also formed through the method above and only through changing A and B, this action isn't obviously needed for equalizing the surface of polygon with line. Variable f is now defined as below:

$$f = \frac{\text{Max}(SAB, SBA)}{\text{Max}\{(m \times \text{Max}(MnDfAB)), (k \times \text{Max}(MnDfBA))\}} \quad (3)$$

About equalizing line and surface, it is obviously MnDfBA=0. Also in both two modes if f denominator is zero, it will be f=0.

3. Problem statement:

3.1. Defining variables:

If all surfaces of first image polygon after naming are shown as a string and named it x₁ and locate some of possible and permitted lines and the surfaces of second image polygon after naming in a string with same size of x₁ and named it x₂, then it will be as:

$$X_1 = \begin{bmatrix} X_1(1) & X_1(2) & \dots & \dots & X_1(n) \end{bmatrix}$$

$$X_2 = \begin{bmatrix} X_2(1) & X_2(2) & \dots & \dots & X_2(n) \end{bmatrix}$$

This means that component 1 of X₁ is equal to X₂(1), component (2) of X₁ is equal to X₂(2) and X₁(n) is equal to X₂(n), of course this correspondence may be incorrect but it is in fact our aim to select the components of second string and the way of their locating order so that this correspondence is true. Each one of these strings is called as one chromosome and each one of their components is called as a gene.

3.2. Defining fitness function:

The function of Fit (X₁, X₂) should be defined in way that shows the error of selecting string x₂ as equivalent to x₁ and when this selection is completely correct, it shows error zero that is:

$$\text{Fit}(X_1, X_2) = 0$$

So if there is f function of defined error in section 1, it will be

$$\text{Fit}(X_1, X_2) = \sum_{i=1}^{\text{Size}(x_1)} f(X_1(i), X_2(i)) \quad (4)$$

While it is supposed to reach minimum amount for fit (X₁, X₂) through fixing x₁ and changing x₂. Ideal mode obviously happens when fit (X₁, X₂) = 0, so we face an optimization problem as below:

$$\begin{cases} \text{Min Fit}(X_1, X_2) \\ \text{Subject to: One-to-one correspondence between } X_1 \text{ and } X_2 \end{cases}$$

4. Solving constructed optimization problem using genetic algorithm:

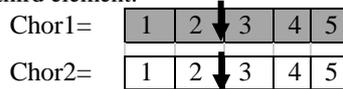
4.1. Constructing chromosomes and initial population:

The first step in solving optimization problems using genetic algorithm is creating chromosomes and as result initial population. Each one of numerical strings which have been explained in section 3 that is (X₁, X₂) is considered as a chromosome. Chromosome X₁ obviously includes a string that have all surfaces of polygon of first view. The order of locating the surfaces of first view inside this chromosome string isn't important.

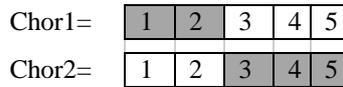
The number of initial population depends on the designer's idea and can be in different values but however it is necessary to calculate the value of Fit (X₁, X₂) for this initial population (various strings of X₂). This initial population is constructed using random selection of permitted components of second image and with their different arrangement.

4.2. Crossover step:

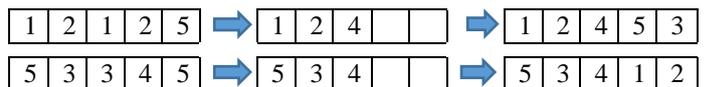
Among different existing methods for Crossover such as heuristic, arithmetic, mask, two points or one point, one-point crossover is selected here in which fracture is necessary to be conducted for two parent chromosomal two strings for example chor1 and chor2 and two strings of offspring will be constructed from them as chor3 and chor4. As an instance chromosomes are as below and the place of fracture is after second element and before third element:



As result offspring chromosomes will be as below:



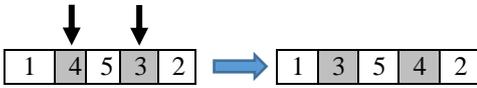
The problem which occur here is that repetitive elements might have been created while it isn't permitted in our problem. To solve this problem, repetitive elements are first eliminated from springs and then by randomizing, the first non-repetitive element will be added to the end of chromosomes, for example, for chor3 and chor4 it will be a below:



It has to be noticed that, which chromosomes should be used for crossover will be obtained based on Trolley Wheels method and selecting fracture place of chromosomes is obtained based on randomizing.

4.2. Mutation Step:

There are also many methods for mutation that considering the discrete problem, the most well-known methods are Swap, insert mutation, scramble and inversion that Swap is used among them because of simplicity. Based on this, two elements need to be selected for chromosome randomly and their place should be changes for example for the string below we have:



Choosing what chromosome needs to be mutated will be random.

4.2. Sorting step and the formation of offspring population:

Each one of formed population by crossover and mutation and initial population or elite has specific share in new population or offspring population, therefore each one of populations is sorted and according to the share of that population from the best fitness, it will be inserted in offspring population and then another time offspring population is sorted and the best fitness is selected as the answer of this step. New population is formed one more time using obtained population through mentioned method and the best fitness of that is selected as the answer of new step and continue this so far one of conditions of stopping is realized, the stopping conditions include one of three conditions below:

- Reaching maximum amount of iterations for forming new and newer populations
- Reaching fitness=0 that is absolute optimum answer
- The existence of best equal answers after formation of some consequent population and lack of improvement in it

After reaching one of realization conditions, the last population will be sorted and its best fitness will be selected as final answer. In case fitness reaches zero, the answer will be accurate and otherwise it will be approximate.

5. Obtaining 3D model from data:

In this step, the place of first and second image in optimization problems should be changed and the process should be iterated one more time to equalize those probable surfaces of polygon that remained in second image and their equivalent hasn't been identified through this method. Then considering that in two views, all three axes can be totally found, through fronting these two views and as result fronting corners or corners with intersection points, 3D coordinates of points and as result lines, surfaces and ultimately total volume can be obtained. Figure 2 shows the flowchart of solving problem through recommended method.

6. Case study: solving a problem with specific numerical data:

As it can be seen in figure 3, based on 1st Angle projection, front and left orthographic views of a shape have been shown. Each one of divisions of showed axis in this figure is one unit. The axes of front view include x and z and the axes of left view are z and y, as result common axis of these two views is z and therefore coordinate z of components will be particularly important in determining the relationship among the elements of these two views. Now the relationship among the components of these two views is supposed to be found and then through making the coordinates of points and surfaces as 3D, 3D volume will be reconstructed. The first step here is naming the components of views.

It is known that lines perpendicular to the common axis that is z as well as the closest lines to this axis cannot be equivalent to any

component in front view and therefore they aren't named. As it can be seen the components 1 to 9 in front view and the components 10 to 19 are located in left view. Now front view is considered as first view and left view as second view and optimization problem is implemented on them. Now the coordinates of all named components should be inserted in system as a matrix. Because the number of corners or intersection points is different for various elements the number of matrix lines will be located equal to the biggest element and consider zero for other elements in case of lack of coordinates. Obtained matrix that is called CT will be as figure 4.

Next step in this path is forming first chromosome that is arranging components as first view polygon surfaces in one string that considering figure 3, it will be as follows:

Chromosome 1=

1	2	3	4
---	---	---	---

Considering the existence of 4 surfaces of polygon in first view, the size of chromosome in our optimization problem will be 4.

Other parameters of genetic algorithm are set as below:

- Population size = 40 Chromosomes = npop
- Crossover percentage = 70% = pc
- Mutation percentage = 25% = pm
- Elitism percentage = 1-(pc+pm) = 5% = pe

The thing which has to be noticed is that considering the size of initial population, 70% of that might not be even number while we know that even number is needed for crossover. To eliminate this problem, it is acted as below:

$$N_{cross} = 2 \times \text{round} (npop \times pc / 2) \tag{5}$$

In which N_{cross} shows the number of offspring chromosomes for crossover. For number of mutation of offspring, it will be also done as below:

$$N_{mut} = \text{round} (npop \times pm) \tag{6}$$

In which N_{mut} represents obtained offspring from mutation. On the other hand, for the number of obtained offspring from elitism that is shown by N_e, we have:

$$N_e = \text{round} (npop \times pe) \tag{7}$$

After determining the number of each level above, their obtained population will be merged with determined percentage and then sorting operation will be implemented based on the best result that is the least amount of fitness. The best result of total in each iteration will be recorded as that iteration's best one.

Therefore, in each iteration, answer chromosome, iteration number, best value for fitness will be recorded and announced. By advancing the number of iteration, the answers will be obviously remained with same quality or will be improved and worsening of answer is not likely. In example above the maximum number of iteration is equal to 30. Setting all mentioned parameters definitely depends on the idea of designer and it isn't unique.

Stopping process can be due to one of three modes below:

- Reaching Max. Iteration.
- Reaching absolute best point that is fitness=0.
- Passing 20 different iterations and lack of improvement in best fitness.

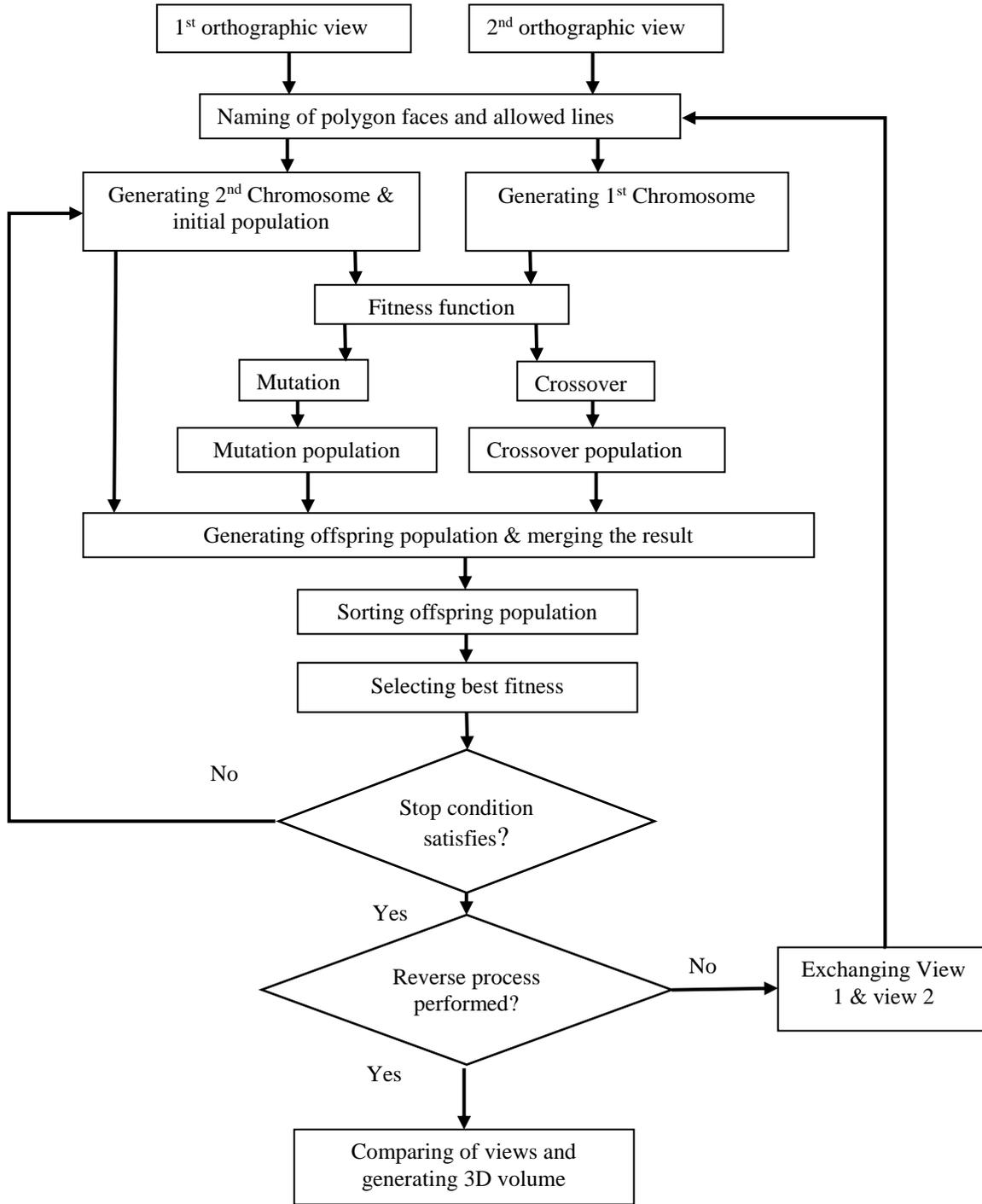


Figure 2: the flowchart of solving problem using proposed method

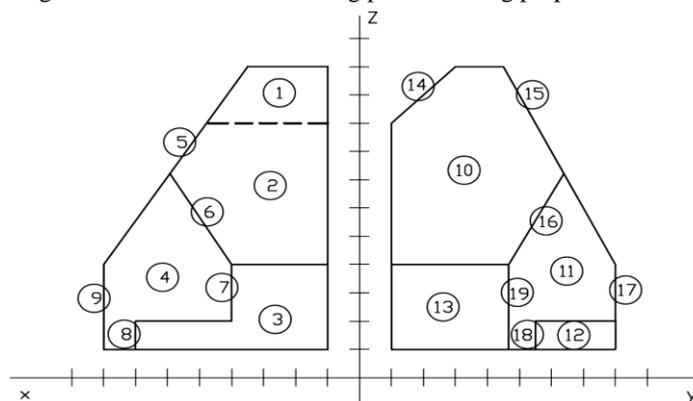


Figure 3: an example with specific numerical data

$$CT = \begin{bmatrix} 9 & 9 & 11 & 11 & 0 & 0 & 0 \\ 11 & 11 & 4 & 4 & 7 & 0 & 0 \\ 4 & 4 & 1 & 1 & 2 & 2 & 0 \\ 7 & 4 & 2 & 2 & 1 & 1 & 4 \\ 11 & 9 & 7 & 4 & 0 & 0 & 0 \\ 7 & 4 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 9 & 11 & 11 & 7 & 4 & 4 & 0 \\ 7 & 4 & 2 & 2 & 1 & 1 & 4 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 4 & 4 & 1 & 1 & 0 & 0 & 0 \\ 9 & 11 & 0 & 0 & 0 & 0 & 0 \\ 11 & 7 & 4 & 0 & 0 & 0 & 0 \\ 4 & 7 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4: the coordinates of all named components as a matrix

Executing this algorithm in different times may reaching answer in different iterations. Sometimes reaching answer is in initial iterations and sometimes in last ones. A sample of fitness diagram based on the number of iterations have been shown in figure 5 that as it can be seen the answer is obtained in fifth iteration. Another

diagram with the same characteristics has been shown in figure 6 that has reached to answer in iteration 25. In both these diagrams, element 1 is equal to 14, 2 equal to 15, 3 equal to 17 and 4 is equal to 11 that is chromosomes 1 and 2 are equivalent as below:

$$\text{Chromosome 1} = \begin{bmatrix} 14 & 17 & 3 & 11 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \text{Chromosome 2}$$

It is also likely that generally as it can be seen in figure 7, absolute and correct answer isn't obtained after passing this process and the number of best fitness remains in a value except zero (for example 0.27 for figure 7) that shows equalizing has been correct for some elements and for the other ones it has been incorrect. In this mode it can be seen that program has proposed equivalent chromosome as below:

14	10	17	11
----	----	----	----

As it was seen in figure 7, final error or best fitness is a number contrasting zero. Compared to best fitness=0, this mode is different in this respect that it has identified surface 2 as equivalent for element 10 that of course we know it isn't correct. In such mode, the process of genetic algorithm is necessary to be repeated till the answer is obtained with best fitness=0. Reaching the answer is obviously realized usually in maximum 2 or 3 times of implementation. Now if this number of searching is compared to the number of common and without optimization through genetic algorithm mode, it will be:

$$\text{Chromosome size}=4 \text{ then the number of possible modes} = 10 \times 9 \times 8 \times 7 = 5040 \text{ (Without GA)}$$

While in figure 5 only after 5 times of forming population we have reached to the answer that the number of its investigated modes includes:

$$5 \times 40 = 200 \ll 5040$$

And also for figure 6 we have:

$$25 \times 40 = 1000 < 5040$$

This issue shows the importance of using algorithm above in this problem clearly.

After this step, the places of views 1 and 2 should be changed and the process is needed to be repeated one more time same as previous one. The initial chromosome in this mode is obviously as 10-11-12-13 and as it can be seen in figure 8 for error diagram and final answer, the answer has been reached in iteration 22 and based on this, element 10 is equal to 5, 11 equal to 4, 12 equal to 8 and 13 is equal to 9.

Now through fronting views and equivalent elements of 2D coordinates, the 3D points will be obtained, as result table 1 will be obtained. Through joining the points of table, 3D lines and as result 3D surfaces will be obtained that summing them will led to formation and reconstruction of initial 3D volume. Obtained model has been shown in figure 9.

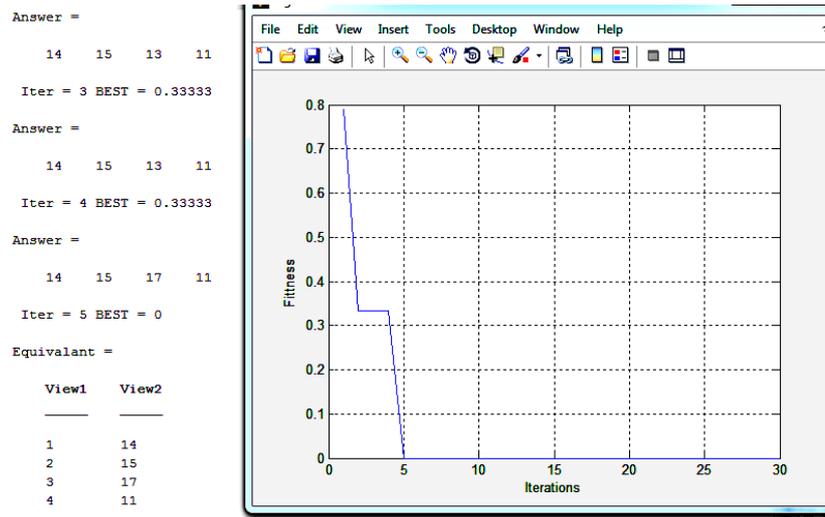


Figure 5: A sample of fitness diagram based on the number of iteration (program output) that answer has been obtained in fifth iteration

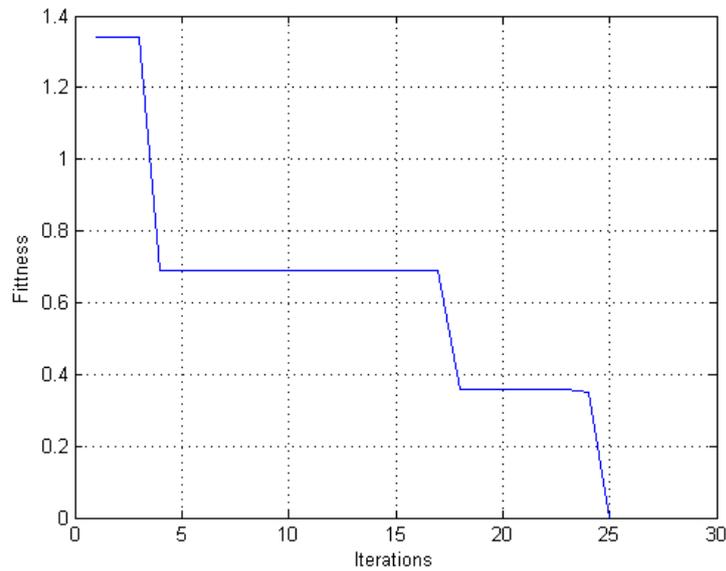


Figure 6: another example of program output in which the answer is obtained in 25th iteration

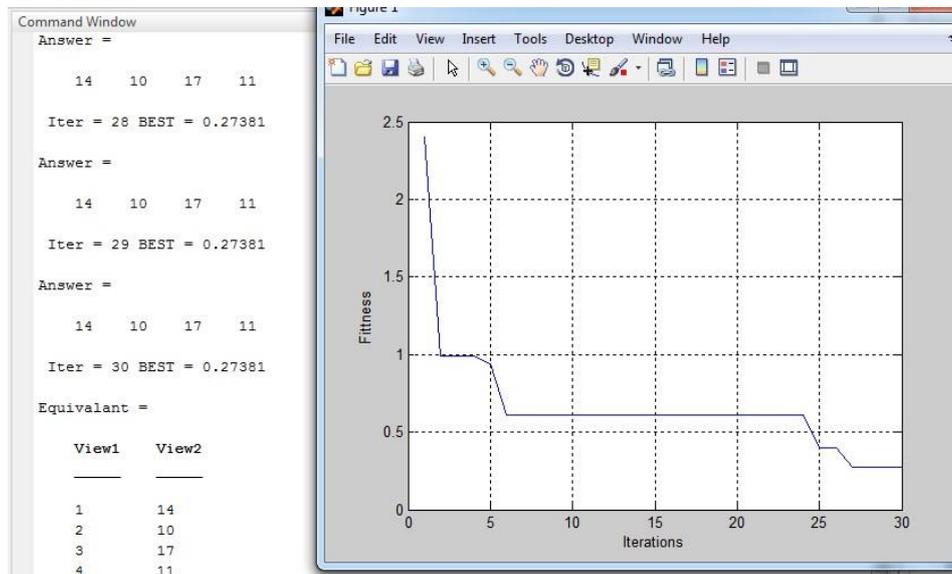


Figure 7: another sample of program output in which accurate answer hasn't been obtained and final fitness is non-zero

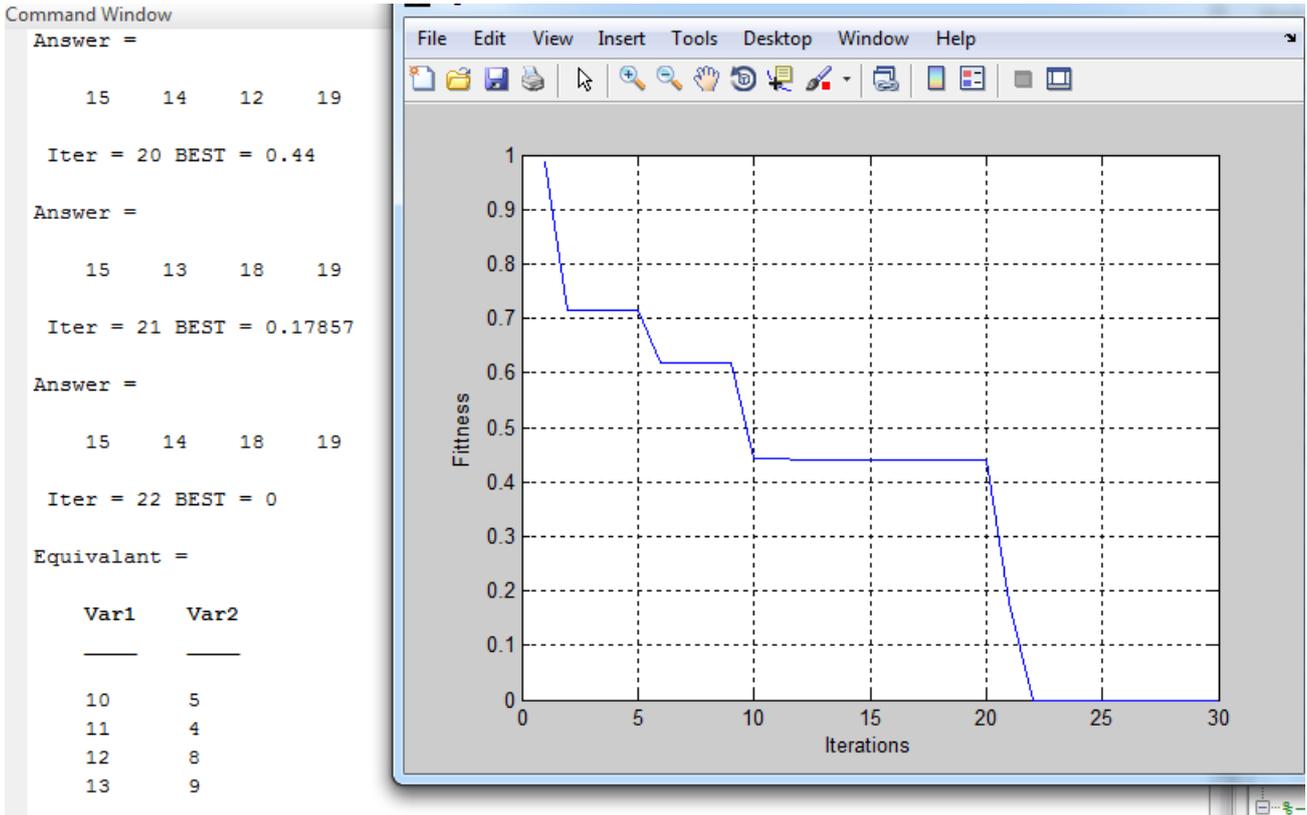


Figure 8: a sample of fitness diagram based on the number of iteration (program output) for the mode that places of 1 and 2 are changed

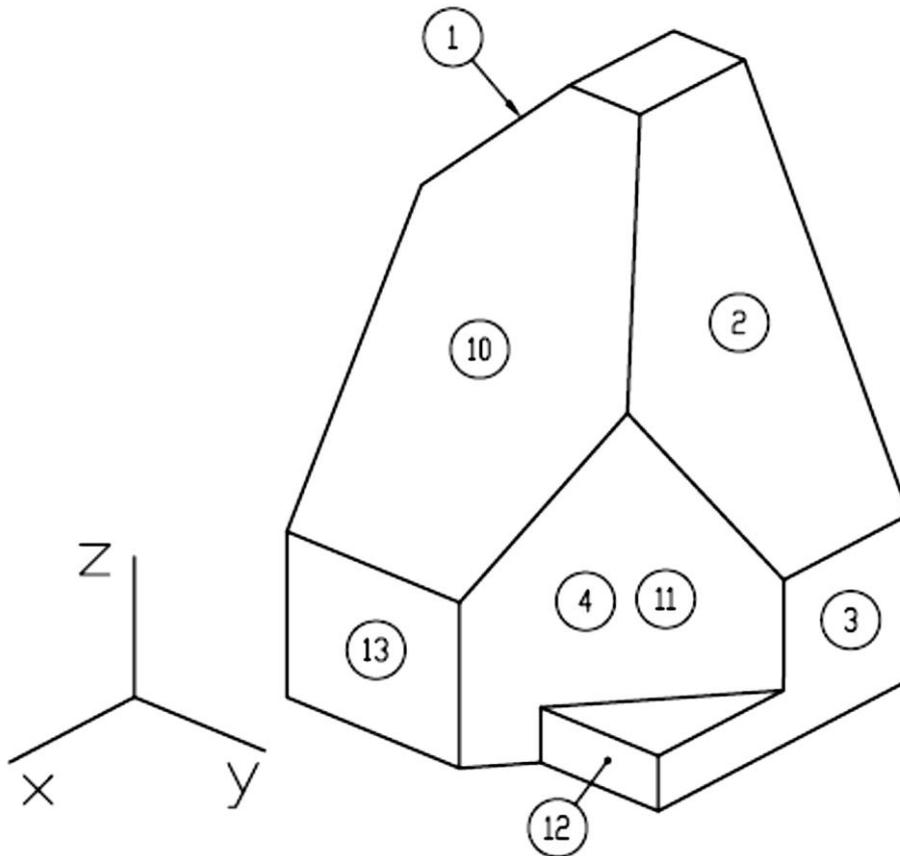


Figure 9: reconstructed 3D model resulted from proposed method

Table 1: obtained 3D coordinates for points, the result of fronting equivalent elements

Element	Equivalent	Coordinates	P1	P2	P3	P4	P5	P6	P7
1	14	X	1	1	3.5	4.8	NA	NA	NA
		Y	1	3	3	1	NA	NA	NA
		z	9	11	11	9	NA	NA	NA
2	15	X	1	3.4	6	4	1	NA	NA
		Y	4.6	4.6	6.5	8	8	NA	NA
		z	11	11	7	4	4	NA	NA
3	17	X	4	1	1	7	7	4	NA
		Y	8	8	8	8	8	8	NA
		z	4	4	1	1	2	2	NA
4	11	X	6	8	8	7	7	4	4
		Y	6.5	4.6	4.6	5.5	5.5	8	8
		z	7	4	1	1	2	2	4
10	5	X	3.5	3.5	4.8	8	8	6	NA
		Y	4.6	3	1	1	4.6	6.5	NA
		z	11	11	9	4	4	7	NA
12	8	X	7	7	7	7	NA	NA	NA
		Y	5.5	5.5	8	8	NA	NA	NA
		z	2	1	1	2	NA	NA	NA
13	9	X	8	8	8	8	NA	NA	NA
		Y	1	1	4.6	4.6	NA	NA	NA
		z	4	1	1	4	NA	NA	NA

7. Conclusion:

Through comparing different methods of 3D-reconstruction, it can be found that B-Rep method is very wider than CSG. However, the existence of main weaknesses of this method that mainly happen in the step of relating the components of different orthographic views, have made using this method difficult. The most important weaknesses include high volume of calculations, because of the number of proposed methods, this problem has been tried in this paper to be solved or decreased as far as possible. In such cases that we face many answers, naturally stochastic optimization algorithm patterns are more practical. Among these algorithms, genetic algorithm can also decrease the steps of reaching answer and volume of calculations to high extent because instead of working on one specific answer in each iteration, it operates on one population of answers. Moreover, considering correspondence one to one condition of answer elements can also decrease vagueness of problem to the great extent.

The weakness of proposed method might be that in genetic optimization algorithm, reaching to answer isn't guaranteed and an answer may be reached that isn't completely correct, of course repeating algorithm maximum for 2 or 3 times can overcome this problem that even with this iteration calculations volume is also very different from common mode. Meanwhile, parameters appropriate selection of optimization process with GA such as appropriate population size, crossover percentage, mutation percentage and appropriate number of iterations can contribute faster solving of the problem.

8. References

[1] H. H. Gorgani, Improvements in Teaching Projection Theory Using Failure Mode and Effects Analysis (FMEA), *Journal*

of Engineering and Applied Sciences, Vol. 100, No. 1, pp. 37-42, 2016.

[2] H. H. Gorgani, Innovative conceptual design on a tracked robot using TRIZ method for passing narrow obstacles, *Indian Journal of Science and Technology*, Vol. 9, No. 7, 2016.

[3] R. Javidi, H. Haghshenas Gorgani, M. Mahdavi Adeli, Size-dependent on vibration and flexural sensitivity of atomic force microscope, *Journal of Computational Applied Mechanics*, pp. -, 2018.

[4] M. Idesawa, A system to generate a solid figure from three view, *Bulletin of JSME*, Vol. 16, No. 92, pp. 216-225, 1973.

[5] T. Woo, J. M. Hammer, Reconstruction of three-dimensional designs from orthographic projections, in *Proceeding of*, 247-255.

[6] K. Preiss, Algorithms for automatic conversion of a 3-view drawing of a plane-faced part to the 3-D representation, *Computers in Industry*, Vol. 2, No. 2, pp. 133-139, 1981.

[7] G. Markowsky, M. A. Wesley, Fleshing out wire frames, *IBM Journal of Research and Development*, Vol. 24, No. 5, pp. 582-597, 1980.

[8] M. A. Wesley, G. Markowsky, Fleshing out projections, *IBM Journal of Research and Development*, Vol. 25, No. 6, pp. 934-954, 1981.

[9] Q.-W. Yan, C. P. Chen, Z. Tang, Efficient algorithm for the reconstruction of 3D objects from orthographic projections, *Computer-Aided Design*, Vol. 26, No. 9, pp. 699-717, 1994.

[10] B.-S. Shin, Y. G. Shin, Fast 3D solid model reconstruction from orthographic views, *Computer-Aided Design*, Vol. 30, No. 1, pp. 63-76, 1998.

[11] M. Kuo, Reconstruction of quadric surface solids from three-view engineering drawings, *Computer-Aided Design*, Vol. 30, No. 7, pp. 517-527, 1998.

- [12] S.-X. Liu, S.-M. Hu, Y.-J. Chen, J.-G. Sun, Reconstruction of curved solids from engineering drawings, *Computer-Aided Design*, Vol. 33, No. 14, pp. 1059-1072, 2001.
- [13] B. Aldefeld, On automatic recognition of 3D structures from 2D representations, *Computer-Aided Design*, Vol. 15, No. 2, pp. 59-64, 1983.
- [14] S. S. Shum, W. Lau, M. M. Yuen, K.-M. Yu, Solid reconstruction from orthographic opaque views using incremental extrusion, *Computers & Graphics*, Vol. 21, No. 6, pp. 787-800, 1997.
- [15] S. S. Shum, W. Lau, M. M.-F. Yuen, K.-M. Yu, Solid reconstruction from orthographic views using 2-stage extrusion, *Computer-Aided Design*, Vol. 33, No. 1, pp. 91-102, 2001.
- [16] A. Çiçek, M. Gülesin, Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution, *Journal of materials processing technology*, Vol. 152, No. 3, pp. 291-298, 2004.
- [17] H. H. Gorgani, I. M. S. Neyestanaki, A. J. Pak, Solid Reconstruction from Two Orthographic Views Using Extrusion and Comparative Projections, *Journal of Engineering and Applied Sciences*, Vol. 12, No. 7, pp. 1938-1945, 2017.
- [18] J. H. Holland, 1992, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press,
- [19] G. Roth, M. D. Levine, Extracting geometric primitives, *CVGIP: Image Understanding*, Vol. 58, No. 1, pp. 1-22, 1993.
- [20] G. Roth, M. D. Levine, Geometric primitive extraction using a genetic algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, pp. 901-905, 1994.
- [21] E. Lutton, P. Martinez, A genetic algorithm for the detection of 2D geometric primitives in images, in *Proceeding of, IEEE*, pp. 526-528.
- [22] S. Y. Yuen, C. H. Ma, Genetic algorithm with competitive image labelling and least square, *Pattern Recognition*, Vol. 33, No. 12, pp. 1949-1966, 2000.
- [23] H. Mohammadi, M. Sharififar, A. A. Ataee, Numerical and Experimental Analysis and Optimization of Process Parameters of AA1050 Incremental Sheet Forming, *Journal of Computational Applied Mechanics*, Vol. 45, No. 1, pp. 35-45, 2014.
- [24] M. Sharififar, S. A. A. Akbari Mousavi, Numerical study and genetic algorithm optimization of hot extrusion process to produce rectangular waveguides, *Journal of Computational Applied Mechanics*, Vol. 47, No. 2, pp. 129-136, 2016.
- [25] M. Goharimanesh, A. Lashkaripour, A. Abouei Mehrizi, Fractional Order PID Controller for Diabetes Patients, *Journal of Computational Applied Mechanics*, Vol. 46, No. 1, pp. 69-76, 2015.
- [26] H. Chehardoli, M. R. Homaienezhad, A new virtual leader-following consensus protocol to internal and string stability analysis of longitudinal platoon of vehicles with generic network topology under communication and parasitic delays, *Journal of Computational Applied Mechanics*, Vol. 48, No. 2, pp. 345-356, 2017.
- [27] T. Mainzer, Genetic algorithm for shape detection, *Technical Report no. DCSE/TR-2002-06*, 2002.
- [28] T. Mainzer, Genetic Algorithm for traffic sign detection, *Applied Electronic*, Vol. 9, 2002.
- [29] X. Zhang, P. L. Rosin, Superellipse fitting to partial data, *Pattern Recognition*, Vol. 36, No. 3, pp. 743-752, 2003.
- [30] J. Yao, N. Kharna, P. Grogono, Fast robust GA-based ellipse detection, in *Proceeding of, IEEE*, pp. 859-862.
- [31] P. Pal, A. Tigga, A. Kumar, Feature extraction from large CAD databases using genetic algorithm, *Computer-Aided Design*, Vol. 37, No. 5, pp. 545-558, 2005.
- [32] K.-Z. Chen, X.-A. Feng, Solid model reconstruction from engineering paper drawings using Genetic Algorithms, *Computer-Aided Design*, Vol. 35, No. 13, pp. 1235-1248, 2003.
- [33] M. T. Siddique, M. N. Zakaria, 3D Reconstruction of geometry from 2D image using Genetic Algorithm, in *Proceeding of, IEEE*, pp. 1-5.
- [34] M. Kabolizadeh, H. Ebadi, A. Mohammadzadeh, Design and implementation of an algorithm for automatic 3D reconstruction of building models using genetic algorithm, *International Journal of Applied Earth Observation and Geoinformation*, Vol. 19, pp. 104-114, 2012.