

Feature Extraction from CAD Model for Milling Strategy Prediction

Jože Balič* - Simon Klančnik - Simon Brezovnik
University of Maribor, Faculty of Mechanical Engineering, Slovenia

In this paper we present a procedure of feature determination from a CAD model. From the model we extract information, which has the greatest influence on the technological parameters of treatment and then transform this information into appropriate input data for different intelligent processing strategy prediction systems (for example artificial neural network). With formally complex CAD models, different processing strategies are required on a single workpiece. For this reason we use segmentation as described in this paper, to partition the surface of the CAD model into regions, so that we treat each region as an independent model and determine its features.

© 2008 Journal of Mechanical Engineering. All rights reserved.

Keywords: CAD-CAM systems, milling strategies, feature extraction, CAD models, segmentation

0 INTRODUCTION

The rising complexity of the industrial production, the need for higher efficiency, better adaptability, higher quality and lower costs, has changed the production processes substantially in the last few years. Modern production science is interdisciplinary and often employs results of research from other fields of science such as: computer sciences, management, marketing and system theory [1] to [4]. The use of artificial intelligence in production systems [5] to [10] has increased greatly in the last two decades because of the adequate efficiency and availability of computers for a broader circle of researchers and industrial users.

In the commercially available CAD/CAM systems, a problem of converting a complex free surface of the product into a formal building block and later into a technological building block still appears. Most of the systems for recognition and connection of building blocks are based on basic geometrical solids, which do not allow satisfying cataloguing of complex free surfaces and their subsequent transformation into building blocks. CAD databases, from which building blocks are captured and identified, also do not contain elements, which would enable satisfying recognition of the free surface as a conglomerate of building blocks. With artificial neural network we can efficiently predict milling strategy from CAD model [11] to [15].

1 STATE OF THE ART

The procedures of automatic milling strategy prediction from surface CAD model can generally be divided in two areas; feature extraction from a 3D model, and feature recognition and classification [16] to [20]. In this paper we will limit to describe feature extraction from CAD model.

We have to take into account that 3D objects can be stored in different formats, such as triangle meshes, volumetric data, parametric or implicit equations, etc. There are different approaches to acquiring features from a 3D model.

2D methods are based on the fact that the 3D model is described with a series of 2D images acquired from different viewpoints. We retrieve useful data from these images with the help of methods used in digital image processing [21]. Different authors use different approaches, namely: Löffler [22] describes a method, which is based on comparison of shapes from images. Min [23] introduces a 2D procedure for detection of 3D objects on the image.

The second group of methods for determining features of a 3D model is comprised of methods based on mathematical description of 3D objects, the so-called histogram methods. Boyer and Srikantiah [24] present a procedure, which segments object in a network of cells and then calculates the so-called key value for each cell. Different values are used, such as: Gaussian

*Corr. Author's Address: University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia, joze.balic@uni-mb.si

curvature and variation of normals. Papers like [24] to [26] represent the use of differential equations for describing 3D objects, specifically - an estimation of Gaussian and mean curvature for describing 3D models. Then there are Kazhdan and others, who describe an object with the help of reflexive symmetry [27] and [28] or spherical harmonic representation [29].

In the next group of methods for extracting features from a 3D model, are methods based on description of topology of an object. Generally, the result describing an object is presented in form of a graph. Exact comparison of two graphs can be computationally very demanding. Hilaga [30] suggested a method, which uses the so-called Reeb graph. Topology of object is written in the graph, according to the geodetic distance calculated for all points on the surface of the body.

Novotni and Klein [31] present a method, which belongs to a group of methods based on measuring the errors between objects.

In general it is difficult to say which of the mentioned methods for determining features of a 3D object is better than others. Feature determination is a very important and interesting problem. We can conclude that each of the mentioned methods has its weaknesses and that an ideal method has still not been suggested yet.

2 FEATURE EXTRACTION

In classification procedure it is extremely important to describe the object with its essential characteristics, important for the given assignment.

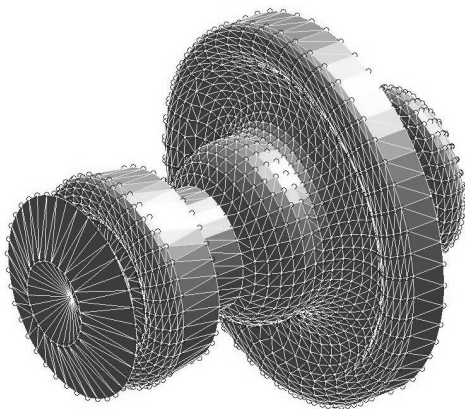


Fig. 1. CAD model example composed from STL triangles

In the recognition process, we arrange patterns in M classes, which means that from the perspective of pattern classification, the important characteristics of objects are only those, which emphasize the particularities of the individual pattern classes. We call such pattern characteristic feature. In our application triangle corners represent the pattern. In Figure 1, we can see a CAD model composed of triangles read from an STL file. STL file was stored in one of the commercial CAD packages (CATIA, SolidWorks etc.) Triangles describing face of the model are of different sizes (depending on the shape of the face).

Each triangle corner is a point, located on the surface of the model. Now we know the coordinates of the corners, and we also know that a between these points there lies a straight face (triangle) - we have described the whole surface of the body. To obtain useful and characteristic information of the pattern, this data has to be appropriately processed. We also need to adequately rearrange these data in order to enter it into the neural network.

Figure 2 presents the phases of data processing:

- In phase one, we have to use the appropriate algorithm to remove faces, which are not visible. First we define the plane from which we will project the network of points to the body. The normal to this plane represents the direction of our view.
- In the phase of segmentation, we partition the whole face representing the body into smaller sub-faces (surface patches) and then treat each of this sub-faces as an independent model.
- In the plane according to which we defined the direction of our view, we create a mesh of symmetrically arranged points and project it to the model at the right angle.
- In the last phase, we standardize the values obtained when projecting the network to the model, and write it in the form of a vector appropriate for input into our classifier.

2.1 Removing the Hidden Surfaces of a Body

The concealment of some parts of a body depends on the chosen viewpoint. If the body is shown with full faces, we use the hidden-surface removal algorithms. Although many hidden-surface removal algorithms exist today [32], we can't

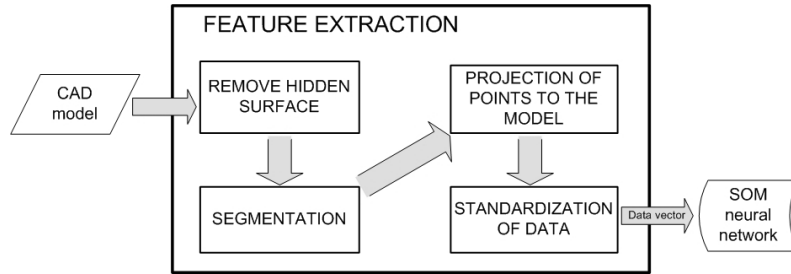


Fig. 2. Block scheme of data processing

determine which one is the best, since their results depend on the complexity of the scene.

In application we »shoot« symmetrically arranged points at the body. For the description of the model, we are interested only in the front surface therefore we use the appropriate algorithm to remove faces hidden to our view – faces hidden by the body itself, the so-called back-faces. For this we need normal vectors of all faces of the body, and they all have to be directed outward from the body or into the interior of the body. Each of the faces is a plane polygon, in our case a triangle. On the face we choose three adjacent corners (in our case these are corners of the triangle): p_{i-1} , p_i in p_{i+1} . We calculate normal face vector for each triangle with the following vector product:

$$n = (p_{i-1} - p_i) \times (p_{i+1} - p_i) = [a \quad b \quad c]^T \quad (1)$$

The equation of the plane in which the surface lies is defined with:

$$ax + by + cz + d = 0 \quad (2)$$

Coefficients a , b and c correspond to the components of normal vector n , coefficient d is calculated with the help of the following expression:

$$d = -(ax_i + by_i + cz_i) \quad (3)$$

where x_i , y_i and z_i are the coordinates of the corner p_i (otherwise we can select any point on the face). In this way we calculate equations of planes for all m faces of the object. Plane defined by the equation (2), divides the space into two half-spaces. Normal vector points to the half-space in which lies the point $p = [x \ y \ z]^T$, under the following conditions:

$$[a \ b \ c \ d] [x \ y \ z \ 1]^T > 0 \quad (4)$$

In the Equation (4), the positive elements in the product signify that the normal vectors of the corresponding planes are directed to the interior of the body, while negative elements signify that vectors are directed outwards from the body. Coefficients of planes, whose vectors are pointing outwards from the body, have to be multiplied by -1. In this way we achieve that all the normal face vectors are directed into the interior of the body.

Hidden-surface removal algorithm uses the fact that the angle between the normal of the visible face and the direction of view is smaller than 90° , while the angle between the normal of the hidden face and the direction of view is larger or equal to 90° . In Figure 3, we indicate the direction of view by vector g . If the scalar product of the view direction vector and the normal of the face is positive, the face is visible, while in the opposite case it is not. In Figure 4 (left) we can see an

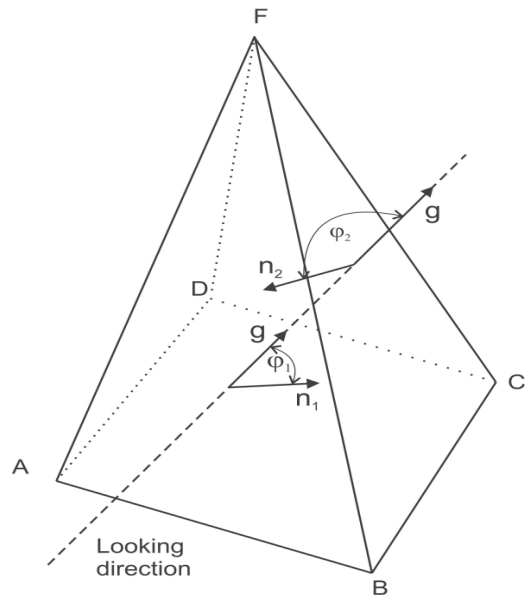


Fig. 3. Surface "ABF is visible, because $\varphi_1 < 90^\circ$ and surface "CDF is invisible, because $\varphi_2 > 90^\circ$

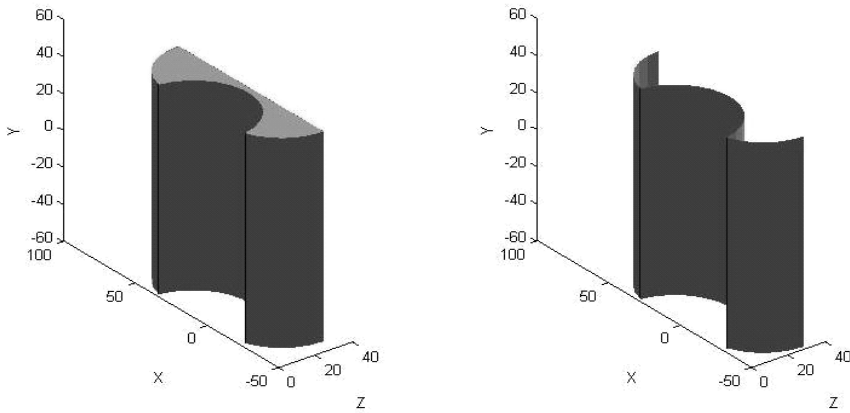


Fig. 4. Representation of hidden surface algorithm's functioning

example of CAD Model drawn with the help of our application. In the Figure 4 (right) we can see the same model after it as been processed with the hidden-surface removal algorithm.

2.2 Segmentation of CAD Model

In this phase we have to partition the entire surface of the CAD model into smaller sub-faces [33]. In other words, we have to segment the triangles describing the surface of the workpiece into regions, which are suitable for independent planning of milling strategy.

We have implemented segmentation of the model according to the size of the angle between the normal of triangles representing the surface of the body. The normal of triangle defined by the three corners p_{i-1} , p_i and p_{i+1} is calculated by Equation (1). For each triangle (the triangle for which we are trying to find its neighbours, will from this point on be called basic triangle) we search for adjacent triangles, which share two common

corners with the chosen triangle. In this way each triangle can have one, two or a maximum of three neighbours (Fig. 5). Then we calculate the angle between the normal of the basic triangle and the normal of its neighbours. If the calculated angle is smaller than the threshold value, we add this triangle into the region established by the basic triangle. If the angle is larger or equal to the threshold value, the triangle is not added to the region and is treated later when we determine the members of the next region. Each new triangle added to the region has to be treated later as basic triangle and examined whether its neighbours satisfy the conditions for classification into this region. The algorithm is executed until each triangle belongs to a certain region. In Figure 7 (left) we can see an example of workpiece drawn with the help of our application. In the right section of the Figure 7, we can see a segmented CAD model in which the segmentation algorithm has divided the model into three regions. In the following procedure we treat each of these regions, obtained

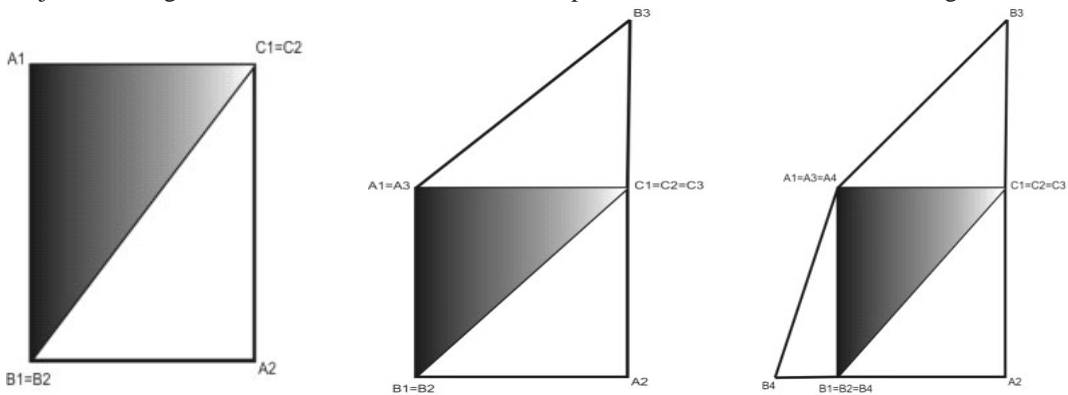


Fig. 5. Basic triangle and its neighbourhood

by the process of segmentation, as an independent CAD model and determine appropriate treatment strategy for each one separately.

2.3 “Scattering” Points Over the Model

Each region of the model is “scattered” with points in a certain mesh. In the x-y plane, we create a window the size of the largest possible drawn-in rectangle. In the window we determine points, which are symmetrically arranged. By increasing the number of points, we improve the reliability of our system but this also increases the number of data representing CAD model and causes more complex and slower processing. The number of points in directions x and y is a parameter which can be modified. In our testing we chose 200 points in direction x and 200 points in direction y.

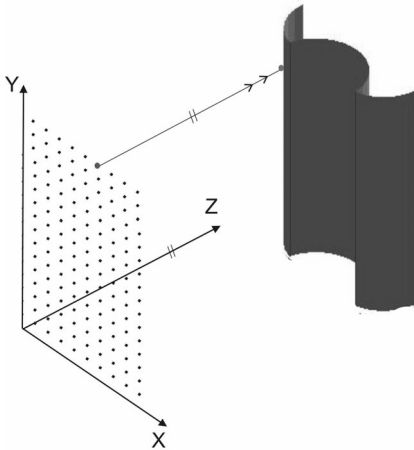


Fig. 6. Schematic representation of points in the plane x-y, which are projected to the body parallel to the z-axis

Figure 6 shows a schematic representation of points in the plane x-y, which are projected to the body parallel to the z-axis. From mathematical standpoint, this means that for each point $P(x,y)$, defined by the raster of points in the plane x-y, we calculate the value of coordinate z, occupied in that position by the surface of the model. Since the triangles describing the face are of different sizes, while we are interested in values of component Z for fixed coordinates X and Y, we have to perform interpolation between these points. We have used triangle-based interpolation presented in detail in [34].

Numbers of points and raster have to be identical for all the models in the learning base. Configuration of model in the direction Z has the greatest influence on the technological parameters; therefore we use only these data for input in the neural network. Values Z, which are calculated for each point $P(x,y)$, are written in the form of vector $S_N = \{x_1^*, x_2^*, \dots, x_N^*\}$. In the test example we have chosen mesh of 200 points in direction x and 200 points in direction y, thus receiving a vector the size of 40,000 elements.

Certain extents of similarity between patterns are highly dependant on the criteria in which we present the features. If we wish for all the features in the pattern to have the same contribution to the calculation of the extent of similarity, we have to standardize pattern features appropriately. The vector obtained in this way, whose elements are standardized values, represents the input into a system for intelligent prediction of processing strategy (SOM neural network, feed-forward neural networks etc.).

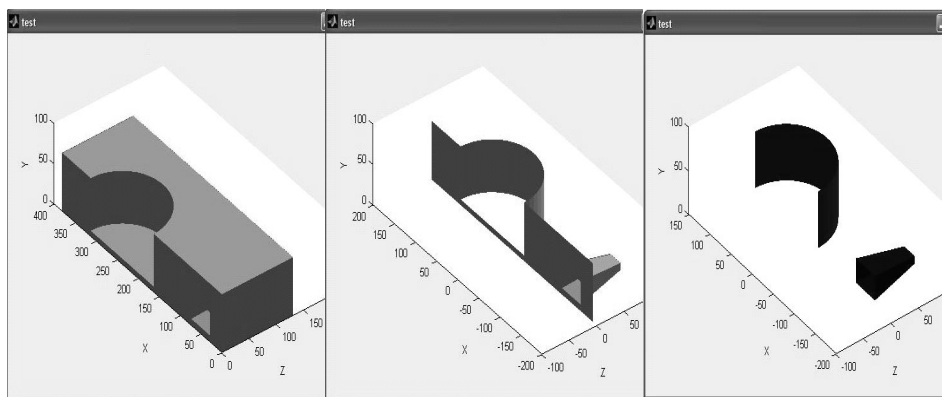


Fig. 7. Representation of processing results for the test model

3 RESULTS

Feature extraction system has been tested on different CAD models, which have been modelled in SolidWorks package. Figure 7 presents processing results of one of the test models. In the first phase we can see that the faces not visible from the x-y plane were removed from the model (middle figure). The right figure shows the model after the segmentation. We can see that we have obtained two independent regions, each of which has to be treated as an independent CAD model and analyzed according to its features for the appropriate processing strategy.

As seen from the Figure 7 and results of other tests performed with models of different shapes, the suggested procedure of feature extraction are very efficient and appropriate for use in prediction of milling strategy. Model segmentation was performed in a manner whose efficiency largely depends on the appropriate selection of threshold value, which varies from model to model. Therefore it has to be stressed that we can obtain good results only with the appropriate threshold value selection.

4 CONCLUSION

System testing has shown that with the help of the procedure presented in this paper, we can efficiently extract valuable information from a CAD model. Those informations are needed in determining strategy according to the shape of the workpiece. Such system would offer users a good supplement tool and help when working with modern CAM systems. Feature extraction program was implemented in the MATLAB integrated development environment. System has been implemented in such a way that its use is universal and allows processing of CAD models made in different commercial packages and stored in STL formats. The topics presented in this paper offers excellent foundation for additional research. In the future we intend to expand the system with intelligent system, which will be capable to predict appropriate processing strategy based on extracted features.

5 REFERENCES

[1] Shishir Bhat, B. N. Profits and reduce cycle time with manufacturing cells. *Advances in*

Production Engineering & Management, vol. 3, no. 1, p. 17-26, 2008.

- [2] Tyagi, V., Jain, A. Assessing the effectiveness of flexible process plans for loading and part type selection in FMS. *Advances in Production Engineering & Management*, vol. 3, no. 1, p. 27-44, 2008.
- [3] Buchmeister, B., Pavlinjek, J., Palcic, I. Polajnar, A.: Bullwhip effect problem in supply chains. *Advances in Production Engineering & Management*, vol. 3, no. 1, p. 45-55, 2008.
- [4] Rahimić, S., Višekruna, V. Optimization of generative CAPP system with minimum cost per piece. *Advances in Production Engineering & Management*, vol. 2, no. 4, p. 177-185, 2007.
- [5] Carpenter, I. D., Maropoulos, P. G. Automatic tool selection for milling operations Part 1: cutting data generation. *Journal of Engineering Manufacture*, 214(4), p. 271-282, 2000.
- [6] Balic, J. Intelligent CAD/CAM system for CNC programming-an overview. *Advances in Production Engineering & Management*, vol. 1, p. 13-21, 2007.
- [7] Renner, G.; Ekárt, A. Genetic algorithms in computer Aided design. *Computer-Aided Design*, vol. 35, p. 709-726, 2003.
- [8] Colak, O., Kurbanoglu, C., Kayacan, M. C. Milling surface roughness prediction using evolutionary programming methods. *Materials and Design*, vol. 28, p. 657-666, 2005.
- [9] Kovacic, M., Brezocnik, M., Pahole, I., Balic, J., Kecelj, B. Evolutionary programming of CNC machines. *Journal of Materials Processing Technology*, vol. 164-165, p. 1379-1387, 2005.
- [10] Valenti, M. Machine tools get smarter. *Mech. Engrg.*, vol. 117, 1995.
- [11] Balic, J.; Korosec, M. Intelligent tool path generation for milling of free surfaces using neural networks. *International Journal of Machine Tools & Manufacture*. vol. 42, p. 1171-1179, 2002.
- [12] Balič, J. *Intelligent manufactory systems*. University of Maribor, Faculty of mechanical engineering, Maribor, 2004.
- [13] Benardos P. G., Vosniakos G. C. Prediction of surface roughness in CNC face milling

- using neural networks and Taguchi's design of experiments. *Robotics and Computer Integrated Manufacturing*, vol. 18, p. 343-354, 2002.
- [14] Azouzi, R., Guillot, M. On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor fusion. *Int J MachTools Manuf*, vol. 37, p. 1201-1217, 1997.
- [15] Yu-Hsuan, T., Chen J.C., Shi-Jer, L. An in process surface recognition system based on neural networks in end milling cutting operations. *Int J MachTools Manuf*, Vol. 39, p. 583-605.
- [16] Potočnik, B. *Pattern recognition with neural networks*. University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, 2007.
- [17] Berikov, V., Litvinenko, A. *Methods for statistical data analysis with decision trees*. 2003.
- [18] Domingos, P., Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, vol. 29, p. 103-137, 1997.
- [19] Cristianini, N., Shawe-Taylor, J. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.
- [20] Fradkin, D., Muchnik, I. Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 70, p. 13-20, 2006.
- [21] Klančnik, S., Balič, J., Planinšič, P. Obstacle detection with active laser triangulation. *Advances in Production Engineering & Management*, vol. 2, p. 79-90, 2007.
- [22] Löffler, J. Content-based Retrieval of 3D models in Distributed Web Database by Visual Shape Information. *Int. Conf. On Information Visualisation*, 2000.
- [23] Min, P., Chen, J., Funkhouser, T. *A 2D sketch interface for a 3D model search engine*. 2002.
- [24] Boyer, K. L., Srikantiah, R., Flynn, P.J. Saliency sequential surface organization for free-form object recognition. *Computer vision and image understanding*, vol. 88, 2002.
- [25] Mokhtarian, F., Khalili, N., Yuen, P. Multi-scale free-form 3D object recognition using 3D models. *Image and vision computing*, vol. 19, p. 271-281, 2001.
- [26] Quek, K. H., Yarger, R.W. I., Kirbas, C.: Surface parameterization in volumetric images for curvature-based feature classification. *IEEE transactions on system, Man and Cybernetics*, 2001.
- [27] Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A., Funkhouser, T. A reflective symmetry description. *European conf. on computer vision*, 2002.
- [28] Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A., Funkhouser, T. *A reflective symmetry description for 3D model*. Algorithmica, 2003.
- [29] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. A search engine for 3D models. *ACM transactions on graphics*, vol. 22, p. 83-105, 2003.
- [30] Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T. L. *Topology matching for fully automatic similarity estimation of 3D shapes*. SIGGRAPH, p. 203-212, 2001.
- [31] Novotni, M., Klein, R. A geometric approach to 3D object comparison. *Int. Conf. on shape modeling and applications*, p. 154-166, 2001.
- [32] Guid, N. *Computer graphics*. University of Maribor, Faculty of electrical engineering and computer science, Maribor, 2001.
- [33] Lefebvre, P., Lauwers, B. *STL Model Segmentation for Multi-Axis Machining Operations Planning*
- [34] Watson, D.F., Philip, G. M. *Triangle based interpolation*. *Mathematical Geology*. p. 779-795, 1984.