Control and Cybernetics

vol. **39** (2010) No. 1

Intelligent prediction of milling strategy using neural networks*

by

Simon Klancnik, Joze Balic and Franc Cus

University of Maribor, Faculty of Mechanical Engineering
Maribor, Slovenia
e-mail: {simon.klancnik; joze.balic; franc.cus}uni-mb.si

Abstract: This paper presents the prediction of milling tool-path strategy using Artificial Neural Network (ANN), by taking the predefined technological objectives into account. In the presented case, the best possible surface quality of a machined surface was taken as the primary technological aim. This paper shows how feature extraction from a 3D CAD model, and classification using a self-organizing neural network, are done. The experimental results presented in this paper suggest that the prediction of milling strategy using the self-organizing neural network (SOM) is effective.

Keywords: SOM neural networks, CAD/CAM system, feature extraction, milling strategy, CAD segmentation, STL model.

1. Introduction

The rising complexity of industrial production and the need for higher efficiency, better adaptability, higher quality, and lower costs have changed production processes substantially over the last few years. Modern production science is interdisciplinary and often employs the results of research from other fields of science, such as: computer sciences, management, marketing and system theory (see Buchmeister et al., 2008; Rahimic and Visekruna, 2007; Shishir, 2008; Tyagi and Jain, 2008). The usage of artificial intelligence in production systems has increased greatly over 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, the problem of converting a complex-free surface of the product into a formal building block, and later into a technological building block, is still present. Most systems for the recognition and connection of building blocks are based on basic geometrical solids, which do not allow for the satisfactory cataloguing of complex-free surfaces, and their

*Submitted: January 2009; Accepted: November 2009.

subsequent transformation into building blocks. CAD databases, from which building blocks are captured and identified, do, neither, contain elements, which would enable satisfactory recognition of the free surface as a conglomerate of building blocks. This article introduces the design and implementation of a system for selecting milling strategies by using a neural network (Tetickovic and Klancnik, 2005), which does not require using a system of building blocks when determining appropriate milling strategy, since it is replaced by a study model of a neural network.

2. State of the art

In practice, milling strategies are chosen on the basis of intuition, and later on the basis of experience. But experience can only be gained through years of practice, during which we make a lot of errors and do not always select optimal strategies. We wanted to create a model, which would be capable of choosing appropriate processing strategies and parameters for the production of a certain product based on a 3D model, without any human interference. For this purpose, many scientists and researchers have suggested artificial intelligence tools for automatic definition of machining phases, and very few have been discussed (the discussion being very limited). Among the methods of artificial intelligence, the most widely used in CAD/CAM systems are genetic algorithms (Colak, Kurbanoglu and Kayacan, 2005; Kovacic et al., 2005; Renner and Ekárt, 2003), fuzzy logic, neural networks (Azouzi and Guillot, 1997; Balic, 2004, 2007; Benardos and Vosniakos, 2002; Carpenter and Maropoulos, 2000; Tsai, Chen and Lou, 1999; Valenti, 1995), and expert systems.

The procedures of automatic milling strategy prediction from a surface CAD model can generally be divided into two areas: feature extraction from a 3D model (Boyer, Srikantiah and Flynn, 2002; Funkhouser et al., 2003; Hilaga et al., 2001; Kazhdan et al., 2002, 2003; Löffler, 2000; Min, Chen and Funkhouser, 2002; Mokhtarian, Khalili and Yuen, 2001; Novotni and Klein, 2001; Quek, Yarger and Kirbas, 2003), and feature recognition and classification (Potocnik, 2007).

3D objects can be stored in different formats, such as triangular meshes, volumetric data, parametric or implicit equations, etc. There are different approaches for 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. Useful data are retrieved from these images with the help of methods used in digital image processing (Klancnik, Balic and Planinsic, 2007). Different authors use different approaches, namely: Löffler (2000) describes a method, which is based on the comparison of shapes from images, while Min, Chen and Funkhouser (2002) introduced a 2D procedure for the detection of 3D objects on an image.

The second group of methods for determining the features of a 3D model is comprised of methods based on the mathematical descriptions of 3D objects, the so-called histogram methods. Boyer, Srikantiah and Flynn (2002) present a procedure, which segments objects in a network of cells and then calculates the so-called key value for each cell. Different values are used, such as: Gaussian curvature and variation of normals. Mokhtarian, Khalili and Yuen (2001) and Quek, Yarger and Kirbas (2003) represent the usage of differential equations for describing 3D objects, specifically - an estimation of Gaussian and mean curvature for describing 3D models. Then there are Kazhdan and associates (2002, 2003), who describe an object with the help of reflexive symmetry, or spherical harmonic representation (Funkhouser et al., 2003).

In the next group of methods for extracting features from a 3D model, there are methods based on describing the topology of an object. Generally, the result of describing an object is presented in the form of a graph. Exact comparison of two graphs can be computationally very demanding. Hilaga et al. (2001) suggested a method, which uses the so-called Reeb graph. The topology of the object is indicated on the graph, according to the geodetic distances calculated for all points on the surface of the body. Novotni and Klein (2001) present a method, which belongs to a group of methods based on measuring errors between objects.

In general, it is difficult to say, which of the here mentioned methods for determining the features of a 3D object is better than the rest. Feature determination is a very important and interesting problem. Each of these methods has its weaknesses but, as yet, an ideal method has still to be suggested.

After the data representing its characteristics are acquired from the 3D object, they have to be arranged in classes, which are composed of prearranged mutually-similar patterns. Many different methods of pattern classification can be found in literature. In the following passages only methods more frequently used in technical applications are mentioned.

Decision trees (Berikov and Litvinenko, 2003) are very suitable for pattern classification. The basis of a decision tree corresponds to the definition of a tree from graph theory. Bayesian classification (Domingos and Pazzani, 1997) belongs to the group of probability decision functions. The use of probability decision functions in classification has its advantages, since theory in the fields of statistical communication and decision theory is very well developed and tested. The next group consists of the so-called kernel methods (Cristianini and Shawe-Taylor, 2000). These methods are encountered, in particular, in the field of pattern classification when neural networks are used. The SVM-support vector machine (Fradkin and Muchnik, 2006) is the most often used methodology for pattern classification in this field.

Most of the pattern classification procedures, particularly decision trees, separate the classes among themselves according to linear limit, which can be an oversimplification for many real applications. Because of this, many real applications use artificial neural networks for pattern classification, which yields good results.

3. Feature Extraction

During the classification procedure, it is extremely important to describe the object using those of its essential characteristics, which are important for a given assignment (Balic and Korosec, 2008). In the recognition process, patterns are arranged in M classes, which means that from the perspective of pattern classification, the important characteristics of objects are only those that emphasize the particularities of the individual pattern classes, so that classification into M classes, rather than into any other number, arises. Such pattern characteristics are called features. In the described application it is the triangle corners that represent the pattern. In Fig. 1, a CAD model can be seen composed of triangular readings of an STL file. The STL file was stored in one of the commercial CAD packages (CATIA, SolidWorks,...). Triangles describing the face of the model are of different sizes (depending on the shape of the face).

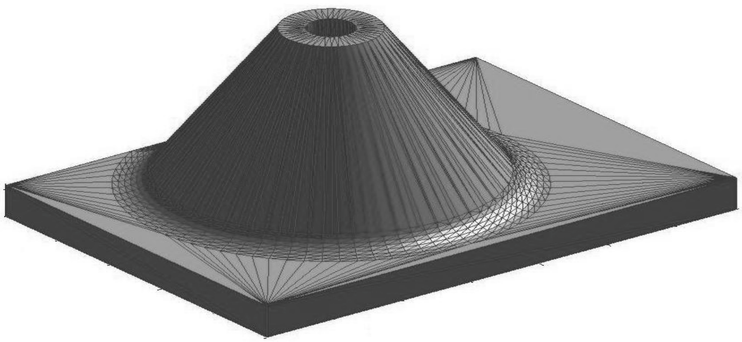

Figure 1. CAD model example composed of STL triangles.

Data processing is done during the following phases:

- In phase one, the appropriate algorithm is used for removing faces, which are invisible to us (Guid, 2001). The plane, from which the mesh of points to the body is projected is defined first. The normal to this plane represents the direction of our view.
- In the phase of segmentation (Lefebvre and Lauwers, 2004), the whole face representing the body is partitioned into smaller sub-faces (surface patches), and then each of these sub-faces is treated as an independent model.
- In the plane, which defines the direction of our view a mesh of symmetrically-arranged points is created and projected to the model perpendicularly.
- In the last phase, the values obtained when projecting the network to the model are standardized. They are written in the form of a vector appropriate for input into our classifier.

3.1. Removing the hidden surface of a body

Symmetrically arranged points are "shot" at the body in the application. Only the front surface is important for describing the model. Therefore, the algorithm appropriate for removing faces hidden from our view is used. The faces are hidden by the body itself. They are the so-called back-faces. The normal vectors of all faces of the body are needed for this, and they all have to be directed to the outward form of the body or into the interior of the body. Each of the faces is a plane polygon, in our case a triangle. The 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 Fig. 2, the direction of view is indicated 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. An example of a CAD model drawn with the help of our application is demonstrated in Fig. 7a. Fig. 7b represents the same model after it has been processed using the hidden-surface removal algorithm.

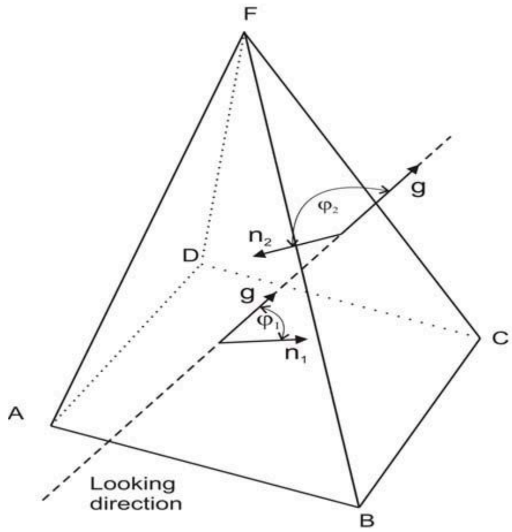

Figure 2. Surface Δ ABF is visible, because $\rho_1 < 90$ and surface ΔCDF is invisible, because $\rho_2 > 90$.

3.2. Segmentation of the CAD model

During this phase, the entire surface of the CAD model has to be partitioned into smaller sub-faces. Segmentation of the model has to be implemented according to the sizes of the angles between the normals of the triangles representing the surface of the body. For each triangle (the "basic" triangle) its adjacent triangles are searched for. These share two common corners with the "basic" triangle. In this way each triangle can have one, two or a maximum of three neighbours. Then the angle between the normal of the basic triangle and the normal of its neighbours is calculated. If the calculated angle is smaller than the threshold value, this triangle is added to 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 when the members of the next region are determined. Each new triangle added to the region has to be treated later as a basic triangle and examined whether its neighbours satisfy the conditions for being classified in this region. The algorithm is executed until each triangle belongs to a certain region. An example of the workpiece drawn with the help of our application can be seen in Fig. 7a. A segmented CAD model, in which the segmentation algorithm divides the model into five regions, can be seen in Fig. 7c. In the following procedure each of those regions obtained by the process of segmentation is treated as an independent CAD model. The appropriate treatment strategy is determined for each one separately.

3.3. "Scattering" of points over the model

Each region of the model is "scattered" with points in a certain raster. A window with the size of the largest possible drawn-in rectangle is created in the x-y plane. The symmetrically arranged points are identified in the window. The reliability of our system is improved by increasing the number of points. This, however, increases the amount of data representing the 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. During our tests 200 points in the direction x and 200 points in the direction y were used.

Fig. 3 shows a schematic representation of points in the plane $x-y$, which are projected to the body parallel to the z-axis. From the mathematical standpoint, this means that for each point $P(x, y)$, defined by the raster of points in the plane $x - y$, the value of coordinate z is calculated, occupied in that position by the surface of the model. Since the face-describing triangles are of different sizes, our interest lies in the values of component Z for fixed coordinates X and Y. Therefore, interpolation must be performed between these points. The triangle-based interpolation was used as presented in detail in Watson and Philip (1984).

The numbers of points and raster have to be identical for all the models in the learning base. Configuration of the model in the direction Z is the most influential for technological parameters. Only this information is used for input into the neural network. Values Z, which are calculated for each point $P(x, y)$, are written in the form of vector $S_N = \{x_1^*, x_2^*, \ldots, x_N^*\}$. In the test example, a raster of 200 points in direction x and 200 points in direction y were written, yielding a vector of 40,000 elements.

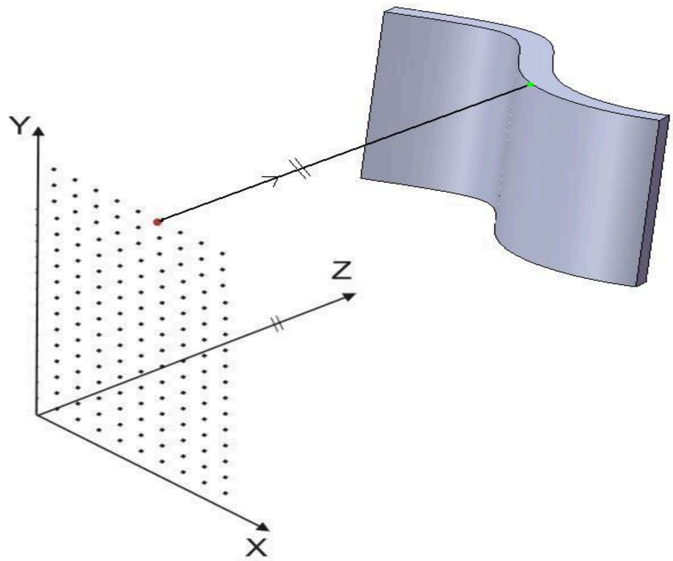

Figure 3. Schematic representation of points in the plane $x - y$, which are projected on to the body parallel to the z-axis.

4. Pattern classification with the use of self organizing maps

From among all types of artificial neural networks, the self-organizing neural networks (Guid and Strnad, 2007) are the closest in structure and functioning to the real biological neural networks. A self-organizing neural network consists of K artificial neurons, which are arranged in one-dimensional, two-dimensional or multi-dimensional lattices. Fig. 4 shows an example of a two-dimensional self-organizing neural network composed of 3 x 3 neurons. Each neuron has two kinds of input: outer or external input, and inner or internal input. The outer input into the neuron is called the observation vector and is marked by x. In our case, this observation vector is represented by a pattern (feature vector). Besides external input, each neuron also has K-1 of internal input lines connected to the rest of the neurons in the self-organizing neural network. These internal lines are called lateral lines.

Self-organizing neural network was used for pattern classification by using the nearest neighbour method. The number of necessary neurons in the lattice depends on the number of samples in the training set of an individual class and the size of the samples input space (domain, range). A neural network the size of 2 x 2 neurons will suffice for small sets. Each self-organizing neural network i is trained by a corresponding set of samples S_i. If a neural network consists of K

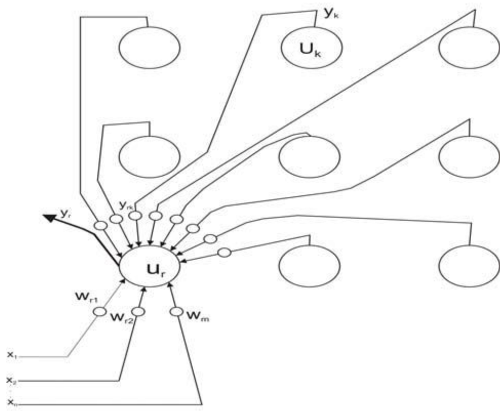

Figure 4. Schematic representation of a two-dimensional SOM neural network.

neurons, this means it also has K vectors of synaptic weights. An unidentified sample is classified by first determining the winning neuron of an individual self-organizing neural network. This is the neuron whose synaptic weights vector most closely resembles the input vector. This is written mathematically as: neuron $u_{i,s}$ is the s-th neuron from the network i, and is considered winning if:

$$d(x, w_{i,s}) = \min_{r=0,\ldots,K-1} d(x, w_{i,r}) \tag{1}$$

where $d(x, w_{i,r})$ denotes the distance between sample x and the synaptic weights vector of the r-th neuron in i-th neural network. As can be seen from (1), the minimum of this distance is searched for between all the neurons in the i-th network. For neural network i, this distance can be written as $d_i = d(x, w_{i,s})$. Based on distances d_i, the unidentified pattern x is classified by looking for the minimal distance d_s among all the distances d_i. This distance belongs to the neural network s or class C_s, into which our unidentified sample is classified. This rule is formalized as: sample x is classified in class C_s if:

$$d_s = \min_j d_j, \tag{2}$$

where index j runs through all neural networks.

5. Testing the system

In the following section, the functioning, training and testing of NN will be demonstrated on a practical example. This will be followed by a presentation of the results. The system was trained by those chosen training models required by the production of formally-demanding tools as mentioned by Balic and Korosec (2002).

5.1. Training phase

At the beginning, the representative 3D model best corresponding faces are chosen of the products we intend to model and process. The models have to be chosen so that they cover the most frequently-chosen shapes of faces used in manufacturing products in a certain environment (tool manufacturing, prototype workshop. . .). The developed application allows for a simple addition of new 3D models, and then performing the training of neural network according to the added models. The representative 3D models shown on the picture were chosen by referring to Balic and Korosec (2002).

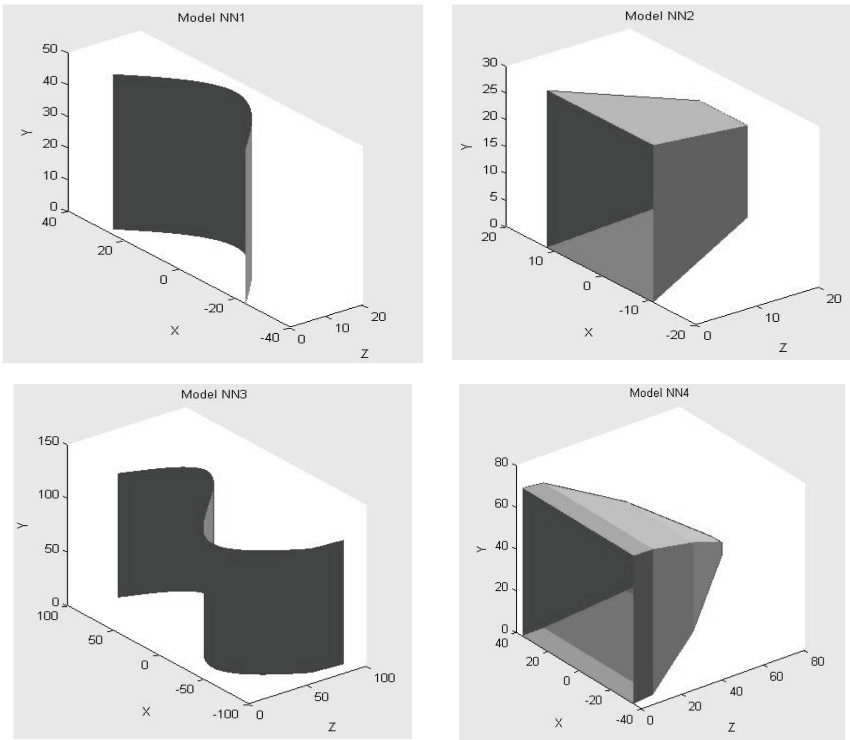

Figure 5. Four models representing basic milling strategies.

Milling strategies for the presented models are chosen on the basis of their adequacy for the selected technological aim. These strategies depend on the chosen CAM system. A combination of milling strategies has to be chosen in a manner that satisfies the chosen technological aim. The quality of surface (middle profile roughness Ra) was chosen as a primary technological aim in the test example. The chosen milling strategies for the presented training models are:

- Model NN1 representing ≫Profile finish≪ respectively ≫3D finish≪.
- Model NN2 representing Z-level finish.
- Model NN3 representing Profile finish (scallop height mode).
- Model NN4 representing a combination of Profile finishing and Z-finishing (slope mode option). First, the flat surfaces are machined in "profile finishing" mode, respectively those surfaces which have slope angles smaller than the boundary set angle, and then the rest of the surface is machined in "Z finishing" mode.

These strategies, although under different names and descriptions, are contained in almost all CAM systems. During the training phase, the model is duplicated into several new training models by rotating the basic model in different directions. Duplication of the model and the course of training phase are shown by the diagram in Fig. 6.

The initial and ending angles of rotation when duplicating the model, as well as the angle of the step, can be chosen arbitrarily. In the test examples, every basic CAD model was first rotated using a step of 3°, in the zone from -15° to +15°, around axes X and Y. Then, after 121 rotations, the model was turned around the Z-axis for 180° and another 121 rotations were performed around axes X and Y. In this way, 242 new training models were obtained from one basic training model. Duplication is performed for each of our four representative training models, after which, all the obtained models are processed by a hidden-surface removal algorithm. This is followed by scattering points over models in a certain raster. The training of SOM neural network is performed during the last phase. An ASCII file is created for each model and in this file the values for the synaptic weights of the trained SOM network for each individual model are written. A two-dimensional SOM neural network of 5x5 neurons was used. The training of the network was performed through 2000 training steps.

5.2. Automatic classification phase and pre-processing

Once the system has been trained, the milling strategy prediction can begin. The milling strategy will be predicted according to the best possible achieved surface quality (roughness). The functioning of the system will be demonstrated on a test example shown in Fig. 7. It has to be stressed that neural network has not yet seen the volume model and the face of the workpiece. This means that this model was not a training model and has never been entered into the neural network.

Fig. 7 shows individual phases of pre-processing (feature extraction phase) of the test model; first the hidden surfaces were removed, after which the segmentation of the model was performed. Fig. 7 shows that five regions were obtained by using segmentation, each of which was later treated as an individual CAD model which is scattered over by a network of raster 200 x 200 points. In this way three vectors were obtained: SN_1, SN_2 and SN_3, of the length of 40,000

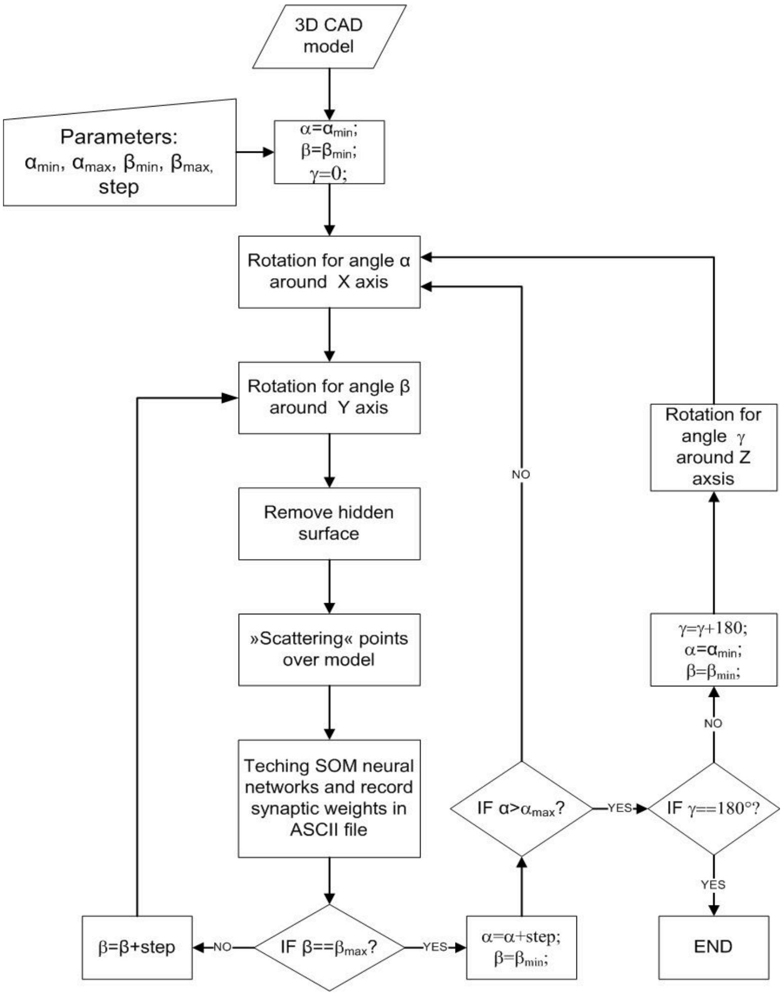

Figure 6. Flow chart of the system training phase.

elements and representing the features of individual regions. During the classification phase, vectors are classified by the nearest neighbour method, using the pre-trained SOM neural network.

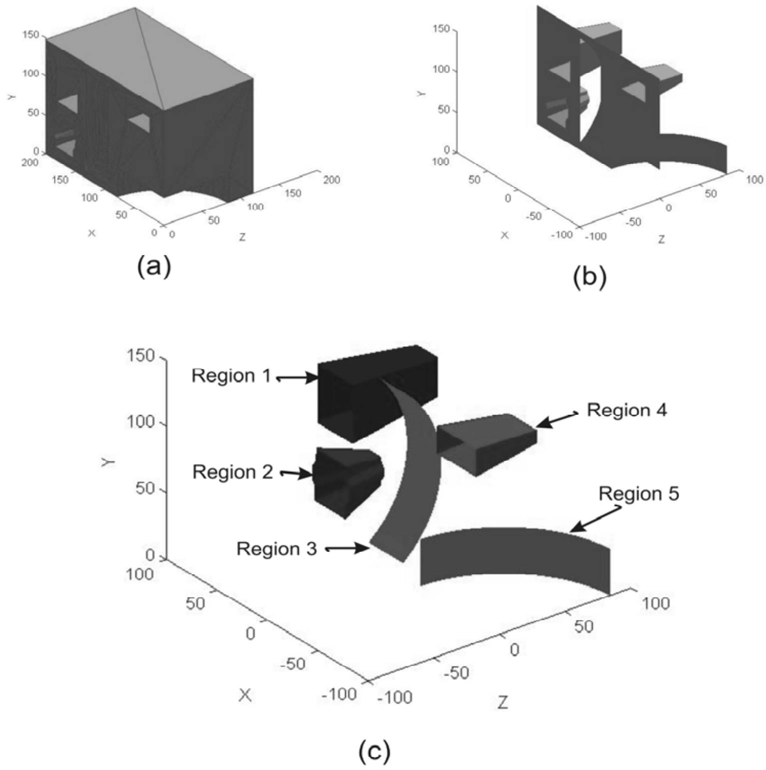

Figure 7. (a) Example of CAD model drawn with the help of our application. (b) The model after it has been processed with the hidden-surface removal algorithm. (c) Segmented CAD model.

5.3. Results

Milling strategy prediction for known models (used in the neural network training processes) was performed in order to examine the effectiveness of neural network training.

In Tables 1 and 2 the following symbols are used:

out1. . . . . . ≫3D finish≪
out2. . . . . . ≫Z-level finish≪
out3. . . . . . ≫Profile finish≪ (scallop height mode)
out4. . . . . . ≫Profile finishing + Z-finishing (slope mode option)≪

Table 1 presents the results of system testing obtained by using known training models and only with SOM networks, which were trained using the basic positions of the training models (without duplication). In pattern classification using the nearest neighbour method the smallest possible distance (Euclidean distance) between the sample and the winning neuron of all trained SOM networks is sought. The smallest numerical value for each line in Table 1 represents the most appropriate milling strategy, as suggested by the system. The results show that the neural networks have been efficiently trained, since an appropriate processing strategy has been suggested for all the learning models.

Table 1. Results of milling strategy prediction for teaching models.

	Milling strategy			
model	out1	out2	out3	out4
NN1	$3.345.10^{-6}$	124.481	$4.788.10^{3}$	135.791
NN2	124.482	$3.177.10^{-6}$	$4.788.10^{3}$	48.481
NN3	42.435	135.793	0.0032	124.482
NN4	135.791	48.481	$4.789.10^{3}$	0.0016

The appropriate milling strategy prediction has been performed for the model in Fig. 7 during the system-testing phase. The CAD model was first segmented (partitioned into five new models - regions) after which an appropriate milling strategy was determined for each of the newly obtained regions. The smallest numerical values in Table 2 also represent the suggested milling strategy for the test example. The system predicted the Z-level finish milling strategy for region 1 and region 4. For regions 3 and 5, the system predicted the 3D finish milling strategy, and for region 2 predicted the Profile finish (scallop height mode) milling strategy.

Table 2. Results for milling strategy prediction regarding the test model (Fig. 7).

	Milling strategy			
model	out1	out2	out3	out4
Region 1	210.453	0.000043	$1.579.10^{5}$	$0.324.10^{4}$
Region 2	112.982	$5.177.10^{2}$	$4.788.10^{-6}$	89.481
Region 3	$2.442.10^{-5}$	119.923	$2.733.10^{3}$	126.768
Region 4	135.791	0.0016	$4.789.10^{3}$	48.481
Region 5	0.0023	144.564	54.545	$3.654.10^{3}$

6. Conclusion

System testing has shown that milling strategy can effectively be predicted with the help of a SOM neural network. The described system provides NC programmers with an efficient tool, and a supplement, when working with modern CAM systems. In the presented application, segmentation of the model was performed in such a way that efficiency largely depends on appropriate threshold selection, which varies from model to model. The entire system was implemented in the

Matlab integrated development environment. The study, presented in this article offers excellent groundwork for additional research, especially in expanding the system for use with other processing technologies.

References

Azouzi, R. and Guillot, M. (1997) On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor fusion. *International Journal of Machine Tools & Manufacture* **37**, 1201-1217.

Balic, J. and Korosec, M. (2002) Intelligent tool path generation for milling of free surfaces using neural networks. *International Journal of Machine Tools & Manufacture* **42**, 1171-1179.

Balic, J. (2004) *Intelligent manufactory systems.* University of Maribor, Faculty of Mechanical Engineering, Maribor.

Balic, J. (2007) Intelligent CAD/CAM system for CNC programming. An Overview. *Advances in Production Engineering & Management* **1**, 13-21.

Balic, J., Klancnik, S. and Brezovnik, S. (2008) Feature extraction from CAD model for milling strategy prediction. *Journal of Mechanical Engineering* **54**, 301-307.

Benardos, P.G. and Vosniakos, G.C. (2002) Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments. *Robotics & Computer Integrated Manufacturing* **18**, 343-354.

Berikov, V. and Litvinenko, A. (2003) *Methods for Statistical Data Analysis With Decision Trees.* Novosibirsk, Sobolev Institute of Mathematics.

Boyer, K.L., Srikantiah, R. and Flynn, P.J. (2002) Salience sequential surface organization for free-form objects recognition. *Computer Vision and Image Understanding* **88**, 3.

Buchmeister, B., Pavlinjek, J., Palcic, I. and Polajnar, A. (2008) Bullwhip effect problem in supply chains. *Advances in Production Engineering & Management* **3** (1), 45-55.

Carpenter, I.D. and Maropoulos, P.G. (2000) Automatic tool selection for milling operations Part 1: cutting data generation. *Journal of Engineering Manufacture* **214**, 271-282.

Colak, O., Kurbanoglu, C. and Kayacan, M.C. (2005) Milling surface roughness prediction using evolutionary programming methods. *Materials and Design* **28**, 657-666.

Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, Cambridge.

Domingos, P. and Pazzani, M. (1997) On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* **29**, 103-137.

Fradkin, D., and Muchnik, I. (2006) Support Vector Machines for Classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **70**, 13-20.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A. and Dobkin, D. (2003) A search engine for 3D models. *ACM Transactions on Graphics* **22**, 83-105.

Guid, N. (2001) *Computer Graphics*. University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor.

Guid, N. and Strnad, D. (2007) *Artificial intelligence*. University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor.

Hilaga, M., Shinagawa, Y., Kohmura, T. and Kunii, T.L. (2001) Topology matching for fully automatic similarity estimation of 3D shapes. *Proc. 28th annual conference on computer graphics and interactive techniques*. ACM, New York, 203-212.

Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A. and Funkhouser, T. (2002) A reflective symmetry description. *Proc. European Conference on Computer Vision (ECCV)*. Copenhagen, Denmark, 642-656.

Kazhdan, M., Chazelle, B., Dobkin, D., Finkelstein, A. and Funkhouser, T. (2003) A reflective symmetry description for 3D model. *Algorithmica*, **38**, 201-225.

Klancnik, S., Balic, J. and Planinsic, P. (2007) Obstacle detection with active laser triangulation. *Advances in Production Engineering & Management* **2**, 79-90.

Kovacic, M., Brezocnik, M., Pahole, I., Balic, J. and Kecelj, B. (2005) Evolutionary programming of CNC machines. *Journal of Materials Processing Technology* **164-165**, 1379-1387.

Lefebvre, P. and Lauwers, B. (2004) STL Model Segmentation for Multi-Axis Machining Operations Planning. *Computer-Aided Design and Applications* **1**, 277-284.

Löffler, J. (2000) Content-based Retrieval of 3D models in Distributed Web Database by Visual Shape Information. *Proc. IEEE International Conference on Information Visualization 2000*. London, UK, 82-87.

Min, P., Chen, J. and Funkhouser, T. (2002) *A 2D sketch interface for a 3D model search engine*. SIGGRAPH Technical Sketches.

Mokhtarian, F., Khalili, N. and Yuen, P. (2001) Multi-scale free-form 3D object recognition using 3D models. *Image and Vision Computing*, **19**, 271-281.

Novotni, M. and Klein, R. (2001) A geometric approach to 3D object comparison. *Proc. Int'l Conf. on Shape Modeling and Applications 2001*, Genova, Italy, 167-175.

Potocnik, B. (2007) *Pattern recognition with neural networks.* University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor.

Quek, K.H., Yarger, R.W.I. and Kirbas, C. (2003) Surface parameterization in volumetric images for curvature-based feature classification. *IEEE Trans. SMC*, **33**, 758-765.

RAHIMIC, S. and VISEKRUNA, V. (2007) Optimization of generative CAPP system with minimum cost per piece. *Advances in Production Engineering & Management* **2** (4), 177-185.

RENNER, G. and EKRT, A. (2003) Genetic algorithms in computer aided design. *Computer-Aided Design* **35**, 709-726.

SHISHIR BHAT, B.N. (2008) Profits and reduced cycle time with manufacturing cells. *Advances in Production Engineering & Management* **3** (1), 17-26.

TETICKOVIC, A. and KLANCNIK, S. (2005) Voice recognition with feed-forward neural network. In: B. Zajc, A. Trost, eds., *Proc. 14th International Electrotechnical and Computer Science Conference ERK' 05.* Portorose, Slovenia, 387-388.

TSAI Y.H., CHEN J.C. and LOU S.J. (1999) An in-process surface recognition system based on neural networks in end milling cutting operations. *International Journal of Machine Tools & Manufacture* **39**, 583-605.

TYAGI, V. and JAIN, A. (2008) Assessing the effectiveness of flexible process plans for loading and part type selection in FMS. *Advances in Production Engineering & Management* **3** (1), 27-44.

VALENTI, M. (1995) Machine tools get smarter. *Mechanical Engineering-CIME* **17**, 70-75.

WATSON, D.F. and PHILIP, G.M. (1984) Triangle based interpolation. Mathematical Geology **8**, 779-795.