www.icgst.com

AIML

# ICGST International Journal on Artificial Intelligence and Machine Learning (AIML)

## ICGST Original Manuscripts

**ICGST – SN:** P1120529002
**ISSN:** 1687-4846 Print, 1687-4854 Online, 1687-4862 CD-ROM
**Title:** "Neural Network Corner Detection of Vertex Chain Code"
**Publishing Date:** January 2006, Volume (6), Issue (1), Pages (37--43)
**Authors:** S. H. Subri, H. Haron, R. Sallehuddin
**Address:** Department of Modeling and Industrial Computing, Faculty of Computer Science & Information Systems, University Technology of Malaysia, 81310 UTM Skudai, Johor, Malaysia

**E-mail:** syhaniz@hotmail.com, habib@fsksm.utm.my, lina@fsksm.utm.my

www.icgst.com

# Neural Network Corner Detection of Vertex Chain Code

S. H. Subri, H. Haron, R. Sallehuddin
*Department of Modeling and Industrial Computing, Faculty of Computer Science & Information Systems, University Technology of Malaysia, 81310 UTM Skudai, Johor, Malaysia*
syhaniz@hotmail.com, habib@fsksm.utm.my, lina@fsksm.utm.my

**Abstract**
This paper presents a Neural Network Classifier to be implemented in corner detection of chain code series. The classifier directly uses chain code which is derived using Freeman chain code as training, testing and validation set. The steps of developing Neural Network Classifier are included in this paper. Comparison results between Neural Network Classifier corner detection and other computational corner detection are presented to show the reliability of the proposed classifier. This paper ends with the discussions on the implementation of proposed neural network in corner detection of chain code series. Experimental results have shown that the proposed network has good robustness and detection performance. This makes this method a great choice for machine vision.

***Keywords**: neural network, chain code, corner detection, line drawing*

## 1 Introduction

Corner detection is an important aspect in image processing and researchers find many practical applications in it. Corner that exists in any irregular line must be detected so that the irregular line can be interpreted to represent actual line. Corners serve to simplify the analysis of images by drastically reducing the amount of data to be processed [1].

Contours are commonly codified with the Freeman chain-code [2] where, assuming 8-connectivity, eight different values are given to the eight possible neighbours of a point. The Freeman chain code consists of eight different numbers, $d_i \in \{0,1,2,3,4,5,6,7\}, i = 1,2,3....,n$ where $d_i$ represents the position of point according to the eight possible neighbours. In this paper, contours or regular line drawings and irregular line drawings were presented by Freeman chain-code.

Many researchers' studies show that corner detection of chain-code use computational method as their main methodology. This computational method was used by Haron [3], Ji [4] and Lee [5]. Nevertheless, very few research is done on the corner detection of chain code series based on artificial intelligence approach such as neural network and fuzzy logic. Therefore, this paper discussed a biological system which used Artificial Neural Network technique as a methodology. The neural network applied Freeman chain code directly to the network and no computational method was used in this corner detector.

Artificial intelligence becomes more popular nowadays. This paper presents an Artificial Neural Network based approach to corner detection in two dimensional (2D) line drawing. The idea for initializing this neural network techniques in corner detection is based on past works which were done by Dias [6], Tsai [7] and Sanchiz [8]. However based on the research done, there was no latest further work done to enhance and improve this method. This paper is expected to lead other researchers to do research in this area.

The organization of this paper is as follows. It is divided into five sections. Section (1) gives introduction, several past works and application on neural network to corner detection using chain code series. Section (2) gives details of the proposed methodology are discussed. Section (3) presents experimental result and comparison of the result with computational method. Section (4) gives conclusion and finally Section (5) presents future works.

## 2 Neural Network Classifier

The Neural Network (NN) Classifier in this paper identifies the corner detection of 2D line drawing. The line drawing was codified to chain code using Freeman chain code and was directly used as an input of the NN

# NEURAL NETWORK IN CORNER DETECTION OF VERTEX CHAIN CODE SERIES

S. H. SUBRI, H. HARON, R. SALLEHUDDIN
Department of Modeling and Industrial Computing,
Faculty of Computer Science & Information Systems
Universiti Technologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
syhaniz@hotmail.com, habib@fsksm.utm.my, lina@fsksm.utm.my

## Abstract

This paper presents a Neural Network Classifier to be implemented in corner detection of chain code series. The classifier directly uses chain code which is derived using Vertex chain code as training, testing and validation set. The steps of developing Neural Network Classifier are included in this paper. Comparison results between Vertex chain code Neural Network Classifier with other computational corner detection are presented to show the reliability of the proposed classifier. This paper ends with the discussions on the implementation of proposed neural network in corner detection of chain code series. Experimental results have shown that the proposed network has good robustness and detection performance. This makes this method a great choice for machine vision.

***Keywords:*** *Neural network, Chain code, Corner detection, Line drawing*

## 1. Introduction

Corner detection is an important aspect in image processing and researchers find many practical applications in it. Corner that exists in any irregular line must be detected so that the irregular line can be interpreted to represent actual line. Corners serve to simplify the analysis of images by drastically reducing the amount of data to be processed [1].

A different chain code named Vertex chain code (VCC) has been proposed by Bribiesca [2] in the end of 90's. VCC is a boundary chain code and it is based on the numbers of cell vertices which are in touch with the bounding contour of the shape. Shape can be computed directly from the VCC without going to Cartesian-coordinate representation. VCC has it own way to represent a shape. There are three different forms of cells to represent a shape which are triangular, rectangular and hexagonal.

Many researchers' studies show that corner detection of chain-code use computational method as their main methodology. This computational method was used by Haron [3], Ji [4] and Lee [5]. Previous paper by Haniz [6] used Freeman chain code (FCC) directly to neural network and developed FCC neural network corner detector. This gives motivation for this paper to apply VCC with neural network. This paper discussed a biological system which used Artificial Neural Network technique as a methodology. Neural network applied Vertex chain code directly to the network and no computational method was used in this corner detector.

Artificial intelligence becomes more popular nowadays. This paper presents an Artificial Neural Network based approach to corner detection in two dimensional (2D) line drawings. The idea for initializing this neural network techniques in corner detection is based on past works which were done by Dias [6], Tsai [7] and Sanchiz [8]. However based on the research done, there was no latest further work done to enhance and improve this method. This paper is expected to lead other researchers to do research in this area.

The organization of this paper is as follows. It is divided into five sections. Section (1) gives introduction, several past works and application on neural network to corner detection using chain code series. Section (2) gives details of the proposed methodology are discussed. Section (3) presents experimental result and comparison of the result with computational method. Section (4) gives conclusion and finally Section (5) presents future works.

## 2. Neural Network Classifier

The Neural Network (NN) Classifier in this paper identifies the corner detection of 2D line drawing. The line drawing was codified to chain code using Vertex chain code and was directly used as an input of the NN Classifier. The outputs of the NN Classifier are represented either by number 1 or 0. Number *1* represents *corner*. On the other hand *0* means *no corner*.

| No. | Training Function | Input | Hidden 1 | Hidden 2 | Output | α | β | Goal | Epochs | Accuracy (%) | MSE Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Traingdx | 9 | 36 Logsig | 9 Logsig | 1 Purelin | 0.1 | 0.1 | 0.015 | 10576 | 91.51 | 0.07468 |
| 2 | Traingd | 9 | 36 Logsig | 9 Logsig | 1 Purelin | 0.5 | nil | 0.015 | 8043 | 92.25 | 0.0725 |
| 3 | Traingdm | 9 | 27 Logsig | 9 Logsig | 1 Purelin | 0.3 | 0.25 | 0.015 | 15772 | 92.25 | 0.08502 |

Table 1: Best model for training function

Analysis is done to determine the best network architecture of NN Classifier. The analysis was based on trial and error. From the analysis done, the best network architecture for NN Classifier is a four-layer network model which consists of one input layer, two hidden layer and one output layer.

This analysis is done by training the network using variation of parameter, training function and differences of network structure. Three training functions were used in this analysis. The training functions are:

- Traingdx (batch gradient descent with momentum and adaptive learning rate). The function of Traingdx combines adaptive learning rate with momentum training. The performance of the algorithm is very sensitive to the proper setting of the learning rate.
- Traingd (batch gradient descent backpropagation) is the batch steepest descent training function. The weights and biases are updated in the direction of the negative gradient of the performance function.
- Traingdm (batch gradient descent with momentum). Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface.

Traingdx and Traingdm training functions use momentum (β) for their training. The momentum is set to 0.1, 0.25, 0.5 or 0.9 while Traingd training function does not use momentum in its training. All these training function use learning rate (α) in their training. The value of this rate is set to 0.1, 0.25, 0.3, 0.5 or 0.75. The analysis was also done using variation of network structure. As shown in Table 1, 2 and 3 there are training model either with one hidden node or two hidden nodes. All hidden nodes in this analysis used Log-Sigmoid (Logsig) transfer function.

More than 363 models were trained during the analysis. Each training functions have their best model but for the NN Classifier the best model among the three models were chosen. Table 1 show the best training models of Traingdx, Traingd and Traingdm training function. Among these three models, one of the models has been identified as the best model with the highest percentage of accuracy and closest condition with exact output validation.

The best model is model number 2. This model uses Traingd training function. As a four-layer network, this model has two hidden nodes with 36 nodes for first hidden node and 9 nodes for the second hidden nodes. Hidden node used Log-Sigmoid transfer function while output node used Linear transfer function.

This model used feed-forward backpropagation as its network type. Mean square error (MSE) function was chosen to evaluate network performance. One value was set as a goal. All training network should be trained until the performance of the networks lower than a value of the goal. For this model, 0.015 has been chosen as a goal parameter. The other parameters of this model are learning rate (α) which was set for 0.5 and finally maximum epoch which was set for 200,000. For training models which had reached 200,000 epochs but the performance was still above the goal value, this means the network was failed. The step on how to train network and how NN Classifier is developed will be discussed in Section 2.1.

## 2.1 Training the Network

The NN Classifier uses supervised training technique. The process of training the network consists of feeding it with a set of training samples which is provided with input and output. The input sets are pieces of chain code which are 9 codes in length for every one output which is extracted from 2D line drawing. The teaching output is a value related to the result of the input set. Image in Figure 1 shows cube line drawing for training sessions while Figure 2 shows L-block line drawing for testing and Figure 3 is stair line drawing for validation. All of these line drawings are taken from Haron [3].

Figure 1: A Cube

Figure 2: A L-Block

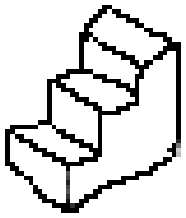

Figure 3: A stair

A total of 266 sets of input and output were involved in the training sessions while 271 sets of input and output for testing and 158 sets of input and output for validation session. A sample of 2D line drawing and its thinned binary image from Haron [10] which used computational method has been codified to chain code as an input and output set to train the network in the training session. Below are the steps taken to train the classifier.

**Step 1**: The input and output were arranged as an array. Figure 4 shows sets of half input and Figure 5 shows set of half output. Figures 4 and 5 also show the input and output arranged in column.

**Step 2**: Using Matlab, a network was trained using the value and parameter which have been discussed in section two.

**Step 3**: Trained network models are tested with a sample of 271 sets of input and output. In the testing stage, accuracy and MSE output are determined. The percentage of the accuracy is based on how many trained outputs are the same with the real output. All the trained output which are the same with real output will be divided by 271 to get the accuracy percentage. The model with the highest accuracy percentage is the best network model. This model is a neural network corner detector and is known as NN Classifier.

**Step 4**: The best network model is used as NN Classifier to detect corner and this corner is tested by using an image. The image has to be first codified to chain code. The chain code was arranged as an array and then it is tested with the classifier to detect corner. Experimental results are discussed in Section 3.

Columns 1 through 13

```
2   3   1   3   1   3   1   3   1   2   2   3   1
2   1   3   1   3   1   3   1   2   2   3   1   3
3   3   1   3   1   3   1   2   2   3   1   3   1
1   1   3   1   3   1   2   2   3   1   3   1   3
1   3   1   3   1   2   2   3   1   3   1   3   1
3   1   3   1   2   2   3   1   3   1   3   1   3
1   3   1   2   2   3   1   3   1   3   1   3   1
2   1   2   2   3   1   3   1   3   1   3   1   3
3   2   2   3   1   3   1   3   1   3   1   3   1
```

Columns 14 through 26

```
1   3   3   1   3   1   3   1   3   1   2   3   1
2   2   1   3   1   3   1   3   1   2   3   1   2
3   1   3   1   3   1   3   1   2   3   1   2   2
1   3   1   3   1   3   1   2   3   1   2   2   3
2   3   3   1   3   1   2   3   1   2   2   3   1
1   1   1   3   1   2   3   1   2   2   3   1   3
3   2   3   1   2   3   1   2   2   3   1   3   1
1   3   1   2   3   1   2   2   3   1   3   1   3
2   1   2   3   1   2   2   3   1   3   1   3   1
```

to

Columns 248 through 260

```
1   3   1   3   2   1   1   1   2   1   3   1   3
3   1   3   2   2   3   3   3   2   3   1   3   1
1   3   2   1   1   1   1   1   2   1   3   1   3
3   2   1   3   3   3   3   3   2   3   1   3   1
2   1   3   1   1   1   1   2   1   2   2   2   2
1   3   1   3   3   3   2   1   3   3   1   3   1
3   1   3   1   1   2   2   3   1   1   3   1   3
1   3   1   3   3   1   2   1   2   2   2   2   2
3   1   3   2   1   3   2   3   3   3   1   3   1
```

Columns 261 through 266

```
1   3   2   2   3   1
3   1   2   2   1   3
2   2   2   2   3   1
1   3   2   2   1   3
3   1   1   3   2   2
1   3   3   1   1   3
2   2   1   3   3   1
2   2   2   2   2   2
2   2   1   3   2   2
```

Figure 4: Sets of input

Columns 1 through 13

```
1   0   0   0   0   0   0   0   0   0   0   0   0
```

Columns 14 through 26

```
1   1   0   0   0   0   0   0   0   0   0   0   0
```

to

Columns 248 through 260

```
0   0   0   0   0   0   1   0   1   1   1   1   1
```

Columns 261 through 266

```
1   1   1   1   1   1
```

Figure 5: Sets of output

## 3. Experimental Results

The classifier is tested on line drawing in Figure 3. The line drawing is based on thinned binary which is shown in Figure 6. Thinned binary image from Figure 6 is used to get Vertex chain code and shape. The Vertex chain code and shape of the line drawing is shown in

Figure 7. The line drawing and thinned binary image is taken from Haron [3]. Line drawing in Figure 3 has been codified to Vertex chain code and shape. Then the chain-code was arranged as an array in length of nine codes every one column. The chain code of the line drawing is based on chain code of boundary list only.

After the chain code was arranged as an array, the boundary list in Figure 7 becomes 158 inputs. When the tested image was codified to Vertex chain code and shape, there are 11 corners found in the drawing and in the 158 inputs. Table 5 shows the real output and Table 6 shows output using NN Classifier to detect corner where *0* means it's not a corner and *1* means there is a corner in that column.

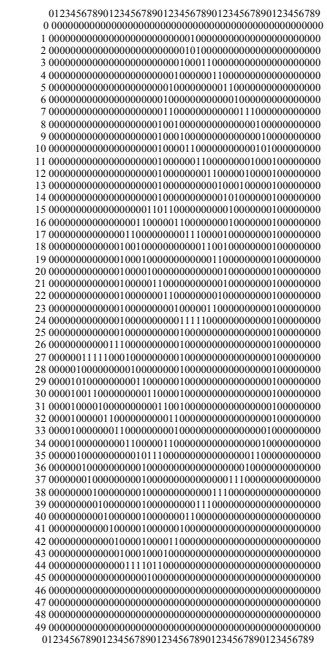

Figure 6 : Thinned binary image

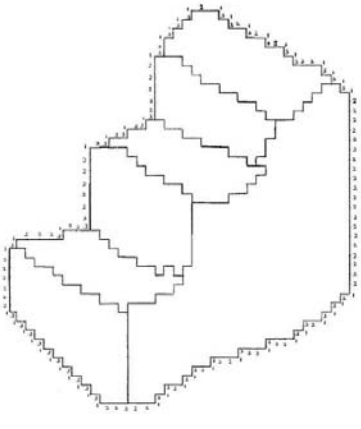

Figure 7: Vertex chain code and shape

Table 5: Real outputs

Columns 1 through 15

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Columns 16 through 30

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 31 through 45

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 46 through 60

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 61 through 75

0 0 0 1 0 0 0 0 0 1 0 0 0 0 0

Columns 76 through 90

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Columns 91 through 105

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

Columns 106 through 120

0 1 0 0 0 0 0 0 0 0 1 0 0 0 0

Columns 121 through 135

0 0 0 0 0 1 0 0 0 0 0 0 1 0 0

Columns 136 through 150

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

Columns 151 through 158

0 0 0 0 0 0 0 0

As shown in Table 6, there are columns which are not the same as real output in Table 5. Column 125 and column 142 show that corner exists in each column. In the real output there was no corner in that column. However in column 107,126 and 137, result shows that corner does not exist but there should be a corner in that column. It shows that NN Classifier detected 10 corners. All 10 corners were in the range of the corner column and 2 corners were not detected by NN Classifier.

Table 6: Neural Network classifier outputs

Columns 1 through 15

0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

Columns 16 through 30

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 31 through 45

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 46 through 60

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 61 through 75

0 0 0 1 0 0 0 0 0 1 0 0 0 0 0

Columns 76 through 90

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Columns 91 through 105

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

Columns 106 through 120

0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

Columns 121 through 135

0 0 0 0 1 0 0 0 0 0 0 0 0 0 0

Columns 136 through 150

0 0 0 0 0 0 1 1 0 0 0 0 0 0 0

Columns 151 through 158

0 0 0 0 0 0 0 0

## 3.1 Comparison of Results

In order to test the performance of the NN Classifier, the experimental results are compared to the computational method done by Haron [3]. Since the boundary line chain code is used to test the classifier, by looking at the sketch, there are 10 corners that exist along the boundary line as shown in Figure 8. How ever there were 11 corners exist when the tested image was codified to Vertex chain code and shape. It is shown in Figure 9

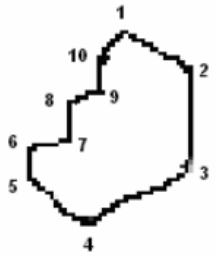


Figure 8: Corner from Haron [3]

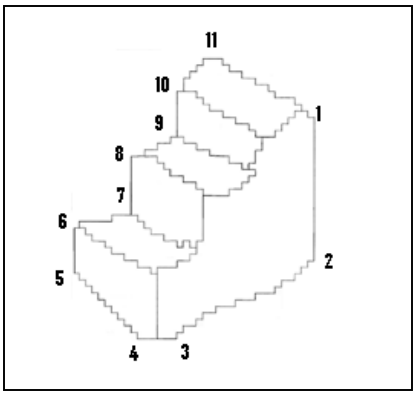


Figure 9: Corner for VCC

Comparison results between proposed Artificial Neural Network method and computational method is shown in Table 7. Out of all 10 corners, the computational method detects 9 corners and the proposed NN Classifier detects 9 corners out of 11. The corner at location 5 has not been detected by computational method while location 7 and 10 has not been detected by NN Classifier. Comparison results between NN Classifier and computational method shows that NN Classifier performance using VCC is average with computational method in terms of the number of corners detected. How ever performance of NN Classifier can be better by give more sample of training set to train the network classifier.

| Method | No. of Corner Detected | Corner Location |
|---|---|---|
| The Proposed NN Classifier | 9 | 1, 2, 3, 4, 5, 6, 8, 9 and 11 |
| Computational Method (Haron [3]) | 9 | 1, 2, 3, 4, 6, 7, 8, 9 and 10 |

Table 7: Comparison table

## 4. Conclusion

The results shows that the strength of applying neural network in corner detection is it makes corner detector more sensitive in detecting a corner. That is why more corner points are detected using this method in one corner. Corner detection in neural network is based on pattern training sample which trained the network. Corner is detected when there is a similarity between corner chain-code trained pattern and chain code of the line drawing. The proposed method used chain code series directly without any calculation to fit it with the network. It makes this method easy to be used and applied. The chain code series just need to be arranged as an array to make it an input.

The drawbacks of the this method is it can be classified as tedious and trial and error process. It is tedious because it involves training samples that have to pass through three stages while the trial and error process will sometime lead to no result.

This proposed method is limited for 2D line drawings only. However, this method can be applied in line drawing interpretations. It is not possible to implement this method to sketch interpreter like SILK which was developed by Landay [11] and made the sketch interpreter faster and more efficiently.

The experiment shows that the optimal parameter of the classifier are alpha is equal to 0.5, and beta is equal to 0.015, and finally maximum epoch is equal to 200,000. The parameters is considered the optimal parameter after the training is conducted.

## 5. Future Works

This proposed method is a 2D line drawing corner detector. The corner detection by neural network Classifier is based on chain code series by Bribiesca [2]. An improvement can be done to this proposed method. The lists of the improvement are given below:

- Give more samples of training set to train the network classifier so the NN Classifier can detect a corner more accurate and precise.
- Chain code techniques are widely used because they preserve information and allow considerable data reduction. In this proposed method, we use the proposed Vertex chain code to represent 2D drawing by Bribiesca [2]. Bribiesca [12] also proposed chain code to represent 3D curve. For detecting corner of these curves using NN classifer, the 3D chain code proposed by Bribiesca [12] can be used to detect a corner of 3D drawing.
- Besides neural network, fuzzy logic is another one of the artificial intelligence techniques. A research of fuzzy in corner detection done by Pahor [13] can be applied to detect corner of chain code series.

## References


[1] H.C. Liu and M.D Srinath, Corner Detection from Chain-Code. *Patt. Recognition Lett*., vol. 23, pp. 51-68, 1990.

[2] E. Bribiesca, A New Chain Code, *Pattern Recognition,* vol. 32, pp. 235-251, 1999.

[3] H. Haron, S. M Shamsuddin and D. Mohamed, A New Corner Detection Algorithm for Chain-Code Representation of Thinned Binary Image, *International Journal of Computer Mathematics, U.K*., vol. 81 no. 3/4, 2004.

[4] Q. Ji and R.M. Haralick, Corner Detection of Covariance Propagation, *Computer Vision and Pattern Recognition,* pp. 362-367, 1997.

[5] J. Lee, Y-N Sun and C-H Chen, Boundary-Based corner Detection Using Wavelet Transform, *Systems, Man and Cybernetics,* vol. 4, pp. 513-516, 1993.

[6] Haniz, S., Haron, H and Roselina, S., Neural Network in Corner Detection of Chain Code Series. *Proceedings of the International Arab Conf. IT ACIT 2004 Algeria*, Vol. 1, 426-431, 2004.

[7] P.G.T. Dias, A.A. Kassim and V. Srinivasan, Neural Network Classifier for detecting Corners in 2D Images, *Systems, Man and Cybernetics.'Intelligent for the 21st Century,* vol. 1, pp. 661-666, 1995.

[8] D-M Tsai, Boundary-Based Corner Detection Using Neural Networks, *Pattern Recognition,* vol. 30 no. 1, 1997, pp. 85-97, 1997.

[9] J.M. Sanchiz, J.M. Inesta and F. Pla, A Neural Network-Based Algorithm to Detect Dominant Points From the Chain-Code of a Contour, *Pattern Recognition Proceeding of ICPR,* vol. 4, pp. 325-329, 1996.

[10] H. Haron, *Enhanced Algorithms for Three-Dimensional Object Interpreter*, PhD Thesis, University Technology of Malaysia, 2004.

[11] J.A. Landay and B.A. Myers, Sketching Interfaces: Toward more human Interface, *Computer*, vol. 34, pp. 56-64, 2001.

[12] E. Bribiesca, A Chain Code for Representing 3D Curves, *Pattern Recognition,* vol. 33, pp. 755-765, 2000.

[13] V. Pahor and S. Carrato, A Fuzzy Approach to Mouth Corner Detection, *Image Processing, ICIP 99*, vol. 1, pp. 667-671, 1999.