CZASOPISMO TECHNICZNE M
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ

z. 3-M/2008
ISSN 0011-4561
ISSN 1897-6328

MAREK CIĘŻOBKA *

CONVERSION OF RASTER IMAGES INTO VECTOR GRAPHICS

KONWERSJA OBRAZÓW RASTROWYCH NA OBRAZY WEKTOROWE

A b s t r a c t

This paper is focused on the conversion of raster images into vector graphics. File format conversion is a very well known problem in the area of computer graphics and there is a number of existing solutions, referenced in the literature. The article presents a novel solution of this particular problem, taking advantage of artificial intelligence, neural networks and fuzzy logic.

Keywords: *technical drawings*, *vector graphics*, *raster image*, *fuzzy logic*, *neural networks*, *regular expressions*

S t r e s z c z e n i e

Tematem artykułu jest przekształcanie rysunków rastrowych na rysunki wektorowe. Konwersja plików jest problemem znanym od dawna w grafice komputerowej i można znaleźć wiele rozwiązań w literaturze przedmiotu. W niniejszym artykule przedstawiono innowacyjne rozwiązanie tego problemu, wykorzystując algorytmy sztucznej inteligencji, sieci neuronowe i logikę rozmytą.

Słowa kluczowe: *rysunek techniczny*, *rysunek wektorowy*, *rysunek rastrowy*, *logika rozmyta*, *sieci neuronowe*, *wyrażenia regularne*

*Marek Ciężobka, V year student, Institute of Applied Informatics, Cracow University of Technology.

1. Introduction

Globalization and unification of standards has influenced virtually all branches of contemporary science and industry. It is possible thanks to the transition towards the information society. Digitization of our world, which is slowly becoming a global village, makes the standard unification process indispensable, literally necessary. The Autodesk company founders realized that fact very well, thus they created the *dxf* file format, allowing users to exchange data between the AutoCAD and other systems. With time, the *dxf* file format flourished and disseminated and an increasing number of companies started using it mainly because it comes with the open documentation and detailed description. The *dxf* format is based on a text file containing ASCII coded characters, thus facilitating its reading and writing on various hardware and software platforms. However, since in the end it is a text file, it has also a number of disadvantages, including excessive size and increased input/output times.

The problem of large file sizes for complex drawings results in an increasing demand for graphic file format conversion techniques. Unfortunately, there are serious problems with converting the raster files into vector graphics.

This particular problem becomes particularly serious when the electronic file format is lost and only the paper copy of the particular drawing prevails. In such cases, in order to retrieve the file, it is necessary to scan the given drawing and save it in the raster format. This paper presents the generic problem of file format conversion, including novel means for its resolution.

2. Analysis of commercially available solutions

The given problem of converting raster files into vector graphics has been targeted by many companies so far. However, the commercially available solutions are expensive and the obtained results are not always satisfactory. The conversion quality results from the low quality of the input, printed drawing. Figure 1 depicts part of a scanned printed drawing. The said image was saved in the bitmap format once the scanning process was completed. When analysed in more detail, the image features a number of noise patterns, resulting from paper creases and low scanning resolution.

The examined drawing was converted into vector graphics. All utilized programs had problems with the proper conversion of the analysed drawing. Figure 2 depicts the drawing previously shown in Fig. 1, though this time it is represented in the *dxf* format, as obtained using the "Img2Cad" software. It is visible that the figure contains a set of erroneous vectors, distorting the image. The said disturbances have various character, including deformed letters, circles and a number of vectors located in the former paper creases. Additionally, a number of drawing elements were not recognized at all.

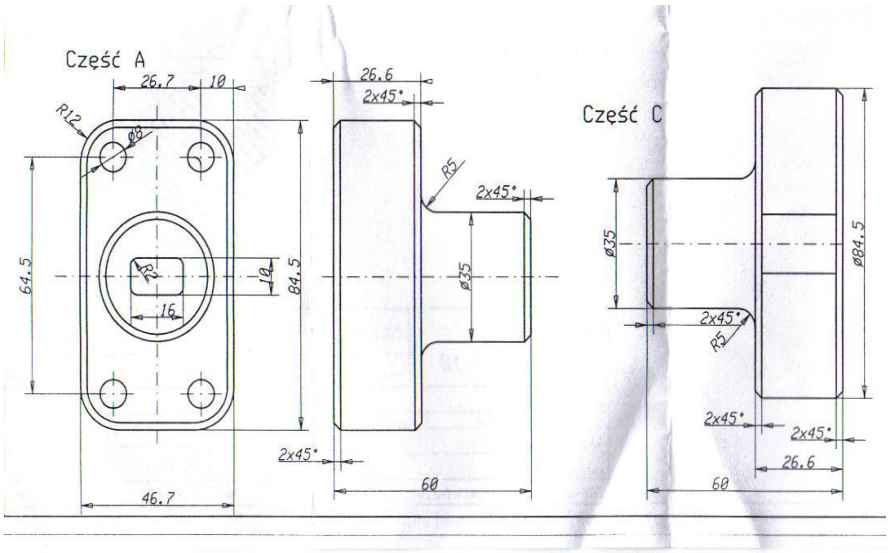

Fig. 1. Scanned drawing
Rys. 1. Zeskanowany rysunek

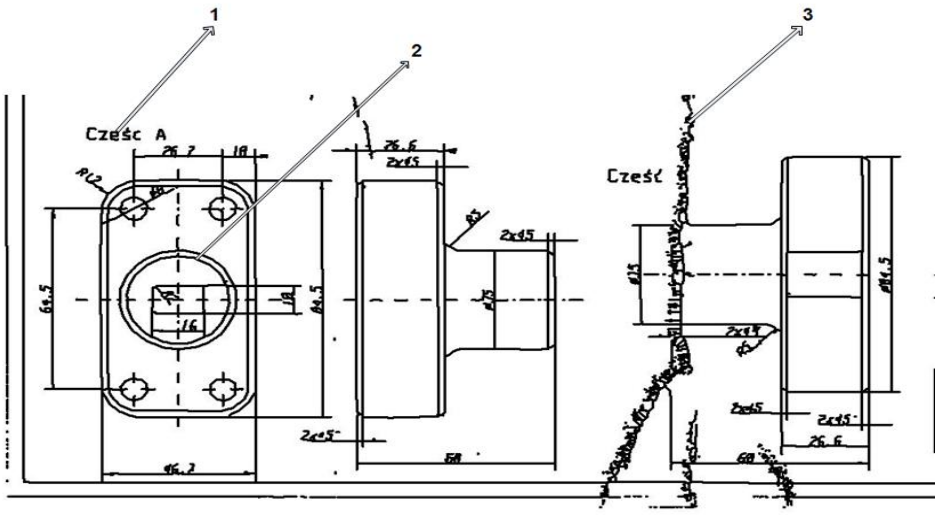

Fig. 2. Vector image obtained using the "Img2cad" software, including the basic distortions, namely: 1 – letter shape, 2 – distorted circles, 3 – vector noise generated by paper creases
Rys. 2. Obraz wektorowy uzyskany za pomocą programu „Img2cad", przedstawiający podstawowe zniekształcenia: 1 – kształtu liter, 2 – zniekształconych okręgów, 3 – szum wektorowy spowodowany zagięciami papieru

3. Intelligent file format conversion

Analyzing the most modern solutions in the area of computer graphics, in the present paper an original method for converting raster images into vector graphics is proposed, as depicted in Fig. 3.

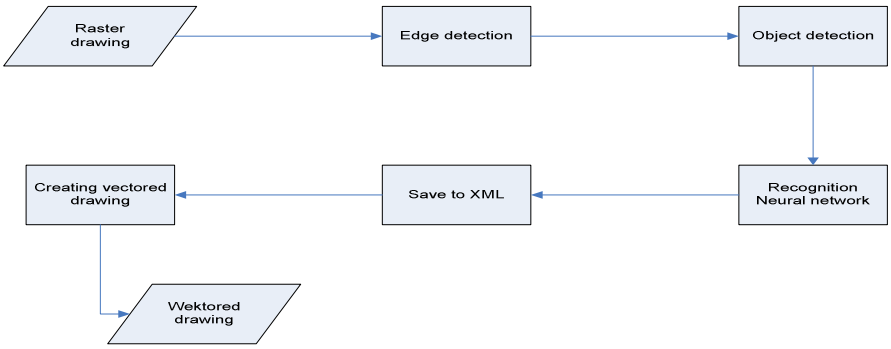

Fig. 3. A block diagram of the examined application
Rys. 3. Diagram blokowy testowanej aplikacji

The file format conversion task should be started by removing all the noise sources present on the rasterized image. Such distortions will always be there and since they are virtually impossible to eliminate on colour images, it is necessary to convert the RGB colour palette into 256 shades of grey.

$$Y = 0,3 \cdot R + 0,59 \cdot G + 0,11 \cdot B \tag{1}$$

where:

Y – the grey scale colour value,
R, G, B – colour components of the given image pixel.

The next file conversion stage includes downgrading the grey scale image into 1 bit colour map, resulting in a back & white image. A properly applied conversion routine allows automatic removal of any future image disturbances.

4. Edge detection

Edge detection allows removal of any superfluous vector noise as well as learning where it is located on the image. That is exactly where the algorithm designed by Lily Rui Liang and Carl G. Looney was applied (see: "Competitive fuzzy edge detection").

Algorithm description

Step 1: saving the user definable setting parameters, namely low and high.
Step 2: for each pixel, calculate the difference between its eight neighbours and store in a respective vector.
Step 3: for each vector, calculate the affiliation function value, defining which class the given pixel belongs to.
Step 4: based on the fuzzy rules, it is possible to determine whether the given pixel belongs to the background or an edge.

5. Object detection

The examined raster image has a 1 bit colour depth thus can be converted into a text file with no problems, where white pixels are replaced with a "0" character and black pixels – with "1". The resulting file can be then subject to pattern recognition routines. This particular process will be carried out using the finite search mechanism and regular expressions.

The finite machine M is defined as an ordered set of five parameters (Q, q_0n, A, Σ, δ), where:

Q – a finite set of the machine states,
$q_0 \in Q$ – the machine initial state,
$A \subseteq Q$ – the set of accepting states (finite states),
Σ – the input alphabet,
$\delta: Q \times \Sigma \rightarrow Q$ – the M machine transition function.

Regular expressions are the patterns describing string symbols. Therefore, in order to process correctly the text format images, it is necessary to search for specific text objects describing lines, ellipses, numbers, hatching etc. Since the finite state machine and the regular expressions will do the job of searching the text file line by line, it is expected that it is easy to identify the candidates for perpendicular lines. In such a case it is sufficient to employ the following regular expression: 1(1)+. The remaining objects will have the multi-line search heuristics applied, based on the neighbour matrix description.

For example, in order to locate a circle, it is necessary to find a line and then create a proper neighbour matrix and then in the line below find the following regular expression: 1+001+. The next step features reaching the circle radius and then a decrease in the number of zeros in the regular expression, until the next line is identified. The identified circle representation is depicted in Fig. 4, though it can represent a set of many lines, circle, eclipse as well as a closed broken line.

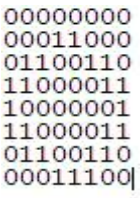

Fig. 4. A circle represented in a text file
Rys. 4. Reprezentacja okręgu w pliku tekstowym

For proper classification of the identified objects, it is necessary to employ the neural networks – specifically, the Hopfield net.

6. The Hopfield net

The Hopfield net is a set of many identical elements inter-connected with each other. In the case of the examined application, a recursive, single layer neural network was employed, where all the neuron outputs are fed back to the inputs of all the neurons, though with the appropriate weights. Obviously, each neuron has also a separate input with the test vector values being fed into the system.

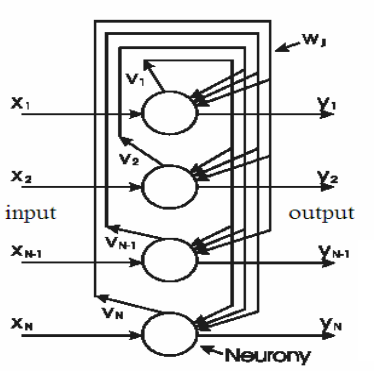

Fig. 5. The Hopfield neural network model
Rys. 5. Model sieci neuronowych Hopfielda

The Hopfield net can thus be used as a block of associative memory. The values for individual weights are selected in such a way that the network stores several vectors, which are treated as basic patterns. Providing that the network is fed with the basic vector or disturbed basic vector, the output should stabilize with the same, undisturbed basic vector. During the network initialization stage, it is fed with the input vector, containing e.g. an image in the form of a bitmap file with 1 bit colour depth. The network output state at the next step is obtained as a result of the operation of the iterative algorithm. The network terminates its operation when in two subsequent steps the output state does not change.

The network teaching process features setting the weight vector for individual neural connections in accordance with the following algorithm

$$w_{ij} \sum_{k-1}^{N} x_k^i x_k^j$$

Fig. 6. Weight vector
Rys. 6. Wektor wagi

The system has a number of basic patterns stored in the associative memory, which it was taught to recognize correctly. Each fragment of the processed image is analysed and then the system selects the basic pattern which resembles most the analysed fragment. At the final stage, the examined image fragment is analysed in detail, resulting in an answer whether the fragment under scrutiny is sufficiently similar to the basic pattern to confirm its identity.

7. Input/output operations on the XML text file

The results of the operation of the aforementioned heuristics will be stored in an XML file. Thanks to that, the resulting file can be used on many system platforms. The next phase features the data reading process. It is performed using the created method, dubbed "ReadAndGo". This particular method calls appropriate methods from ClassLibrary_api.dll or ClassLibrary_dxf.dll, responsible for creation of appropriate files with the *dwg* and *dxf* extension. As far as converting the given text file into other text format is not a big problem, however creating a new *dwg* file will require the AutoCAD.

ClassLibrary_api.dll library

Api_autocad
+gbl_app -gbl_color -gbl_doc -gbl_modSpace
+Add_Text() : void +CloseDoc() : void +CreateAutoCadObject() : void +LineDraw() : void +CircleDraw() : void +TextDraw() : void +SaveDocument() : void

Fig. 7. Api_autocad () class
Rys. 7. Klasa Api _autocad ()

ClassLibrary_api.dll library is equipped with the Api_autocad class, which creates an AutoCAD file using the CreateAutoCADObject() method, and then calls other methods, e.g. DrawLine() to create the previously identified objects, which are automatically dimensioned.

In order to create the ClassLibrary_api library, we employed an already existing library Interop.Autocad(), which can be added to the references by searching for it in the Visual Studio 2005 environment in the COM library section.

Vector file format

Figure 8 depicts the drawing processed by my application. It can be seen that all the objects were identified correctly and properly saved into the AutoCAD file.

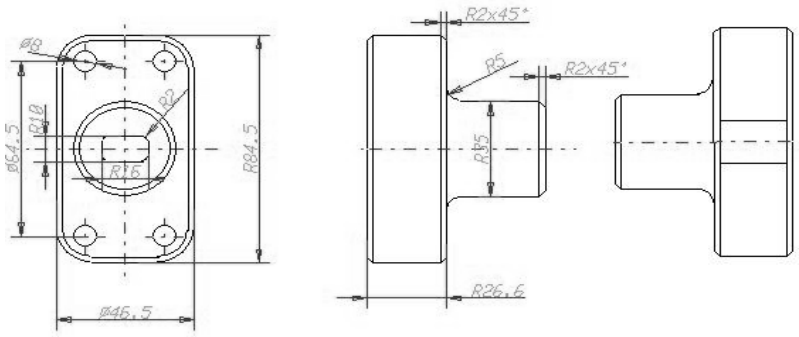

Fig. 8. Converted drawing
Rys. 8. Rysunek uzyskany w wyniku wektoryzacji

8. Conclusions

The proposed solution allows changing the file formats at will. Thanks to it, it is possible now to save time which was previously occupied with recreating the drawings. The application can also be developed further and utilized in other graphic programs.

References

[1] Tadeusiewicz R., *Sieci Neuronowe*, Warszawa 1993.
[2] Wojnar L., *Image analysis*. *Applications in Materials engineering*, CRC Press Boca Raton, Fl, 1999.
[3] Liang R.L., Looney C.G., *Competitive fuzzy edge detection*, Applied Soft Computing 3, 2003, 123-137.
[4] Jeffrey E., Friedl F., *Wyrażenia regularne*, Wydawnictwo „Helion", Warszawa 2001.