RESEARCH PAPERS
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

10.1515/rput-2016-0015 2016, Volume 24, Number 39

LINE WIDTH RECOVERY AFTER VECTORIZATION OF ENGINEERING DRAWINGS

Matúš GRAMBLIČKA, Jozef VASKÝ

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA,
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA,
INSTITUTE OF APPLIED INFORMATICS, AUTOMATION AND MECHATRONICS,
ULICA JÁNA BOTTU 2781/25, 917 24 TRNAVA, SLOVAK REPUBLIC
e-mail: matus.gramblicka@stuba.sk, jozef.vasky@stuba.sk

Abstract

Vectorization is the conversion process of a raster image representation into a vector representation. The contemporary commercial vectorization software applications do not provide sufficiently high quality outputs for such images as do mechanical engineering drawings. Line width preservation is one of the problems. There are applications which need to know the line width after vectorization because this line attribute carries the important semantic information for the next 3D model generation. This article describes the algorithm that is able to recover line width of individual lines in the vectorized engineering drawings. Two approaches are proposed, one examines the line width at three points, whereas the second uses a variable number of points depending on the line length. The algorithm is tested on real mechanical engineering drawings.

Key words

mechanical engineering drawing, vectorization, line width

INTRODUCTION

The electronic engineering documentation prevails today in the process of product design. However, paper based documentation is still required for maintenance of the products. The engineering drawing is the basis for design of the parts in industrial sectors. It can be seen as an information source for technical ideas and a communication tool of the techno-economic activities (1). The rules for the creation of engineering drawings are provided by the international ISO standards. The engineering drawings consist of various geometric elements and textual data. The lines as the basis of engineering drawings define the shape and dimensions of a part. On the mechanical engineering drawings there are used two main widths of lines; a thin line for dimensions, axes, annotation symbols and a thick line for the contour of the part. The width difference should not be less than 2:1.

The archiving of project documentation can be encountered in four forms. They are traditional paper-based drawings, electronic raster and vector drawings and 3D computer CAD models. The process of generating a 3D model from a paper-based engineering drawing can be divided into four main stages: scanning, preprocessing, vectorization and 3D computer model generating. The raster form of an engineering drawing created by the scanning process loses the semantic information of an original paper-based drawing. This information which is necessary to identify the geometric entities may be restored by vectorization.

Vectorization is the transformation of a raster image into a vector form consisting of an image change created by pixels to an image created by vectors (2). In this process it is gradually passed through all points of the image and it is observed what pixels could be described by the geometric element (3). Conversion from raster to vector representation consists of image altering formed of the points (pixels) to an image formed of the lines (vectors) (4). They may be the segments of lines, rectangles, triangles or polynomials describing the parts of circles or ellipses. Each of these elements is defined by the attributes as start point, end point, length, size and color etc. The accuracy is an important criterion of the vectorization, which reflects the extent to which the vector image corresponds to the raster model (3). More advanced vectorization includes an adaptation and an extension of the lines (5), which results in providing more accurate results (6), (7).

Most of the vectorization methods are based on the thinning (skeletonization) or the center-axis approaches. The thinning procedure is commonly used in vectorization. The skeleton of an object is identified by the Thinning process. The skeleton is a thinned version of the shape, in principle it is a central line extraction of an object (8). In many technical fields a skeleton is an important shape descriptor (9).

Image processing techniques were introduced more than 40 years ago and have been developed and established several vectorization methods. These methods can be roughly divided into seven classes (5), (8):

1. Methods based on Hough transform (HT).
2. Thinning methods.
3. Contour based methods.
4. Run-graph based methods.
5. Mesh pattern based methods.
6. Orthogonal Zig-Zag method (OZZ)
7. Sparse pixel vectorization (SPV).

The first two methods do not preserve the line width, others do but the best results in terms of line geometry are provided by thinning methods. As it is shown in this paper, the line width can be later recovered even using thinning vectorization methods.

HT is a known method for the recognition of geometric primitives on a raster image. The main advantage of the algorithm is the ability to extract desired graphic objects from the noisy environment (10) (11). As an example, it can be mentioned the work (12). The main disadvantage of methods based on Hough transformation is the memory inefficiencies. In addition, because the parameter space is sampled discretely, the location accuracy can be prevented (13).

The thinning is an important step of the preprocessing for an analysis and recognition of the different types of images (14). The purpose of the thinning method is a skeleton extraction of an object. There are two models for extracting the skeleton of an image, which are used for vectorization. The first is based on the iterative thinning of an image using the boundary erosion process. The iterative process removes pixels until one pixel wide sequence of pixels remains. In the second model, a skeleton may be calculated by determining the ridge lines created by

centers of the maximal disks contained in the image. Iterative thinning needs more passes to reaches the final skeleton, so the execution time can be high, and is also sensitive to noise.

The contour based methods detect line-like areas. These methods usually consist of four main steps: (i) extracting the contour vectors, (ii) pairing, (iii) creating the medial axis, (iv) processing of the joints (15), (16). A notable disadvantage of the method is a skipping of the contour pairs on the joints, resulting in gaps that violate the vectors. Maintaining the line width partially compensates this disadvantage (16). The contour based methods are better at positioning the joint points than the thinning methods.

The run-graph based methods examine raster image in each row or column to calculate the run-length encoding. The runs are then analyzed in order to generate a graph structure. These methods are not robust, if the image quality is degraded. In addition, the dependency on a scan direction leads to the unsatisfactory results for the diagonal lines. They work well for sparse line images, mainly consisting of horizontal and vertical lines (10), (16). The most common technique for the run-graph based methods is the run length encoding (RLE) (10).

The basic idea of the mesh pattern based methods is to split the entire image using a particular mesh and for the detection of characteristic patterns only to control the distribution of the black pixels on the border of each mesh unit. The extraction of long line segments is done by analyzing the control map. The image is then represented by a control map, in which each mesh unit in the original image is replaced with its characteristic pattern.

The OZZ method can be thought of as an analogy of the optical fiber when a light is reflected from the walls of the lines representing the object. The SPV method is an enhanced OZZ method proposed by Liu and Dori in 1999. The SPV approaches are time efficient because of the sparse sampling of the image data however, the main limitation is that they are susceptible to the double detection (13) and sensitivity to the noise (17).

MATERIALS AND METHODOLOGY OF EXPERIMENT

When testing vectorization applications it has been shown (18) that the results were inconclusive at certain objects. The main problem was that line width has not been preserved. The next section of the article comes with the solution that could address that problem.

The test samples for vectorization were obtained by scanning of the real paper-based engineering drawings. For creating vector representation from these scans the vectorization tool Ras2Vec (19) was used. Ras2Vec is available as full source code under GPL license for Windows. It takes 1 bit per pixel BMP or TIFF images and emits HPGL, DXF, EMF or TXT files. It uses the Zhang Suen parallel thinning algorithm (20) for thinning the image. The Zhang Suen algorithm utilizes two sub-iterations. The Zhang Suen algorithm is fast and should preserve the pixel connectivity (14) and the end points (21). The drawback of that algorithm is that the noises around the north-east and the south-west corners are extended instead of deleted (22).

For the purposes of this article this application was modified to produce a text file with the coordinates of individual vectors which are used by the proposed algorithm. The quality and the precision of the vectorization provided by Ras2Vec are comparable to other commercial vectorization applications. Ras2Vec does not preserve the line width therefore the proposed algorithm may be well demonstrated.

The preserving of line width

In the engineering drawings the lines that represent a part are thicker than the dimensions. The thinning of the lines appears to be a problem for the next processing. In the case of the same width of all lines, information about the type of the line is lost. This section describes an algorithm that can preserve the line width of the engineering drawings. In this article two

approaches of the algorithm are compared. The first approach takes into account only three points on the line to determine the line width and the second uses a variable number of points depending on a line length.

The algorithm is proposed so that for each line in the polyline, its actual width is calculated. Then the mean width of all lines is calculated as the resulting width of the entire polyline. The threshold that selects which lines are drawn thick and thin can also be determined. If two line widths are utilized in engineering drawings, this approach is satisfactory.

The first step of the algorithm is to determine the line length. If the first approach is used, it is necessary to determine three points on the line, the middle point, the quarter and the three-quarter. If the second approach is used, the number of points is calculated according to the line length. A preview of that arrangement is shown in fig. 1. These points can be called stationary points.

The line width is determined in the direction of a vertical to the direction vector of the line sequentially at these stationary points. Based on the coordinates of the stationary point it is found whether it lies on the part. In other words, it is determined whether the stationary point does not lie on the white color. The white color represents the background of the engineering drawing. If it does not lie on the white color, the width is determined on the particular stationary point. If it does lie on the white color, it is passed to the next of these stationary points.

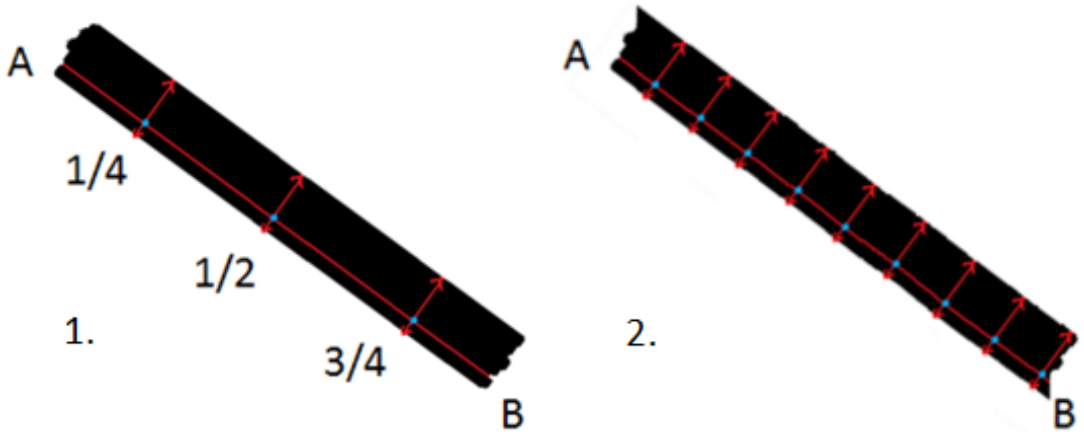

Fig. 1 *Line width is measured using: 1.) three points: in the middle, quarter and three-quarter. 2.) variable number of points*

In the subsequent steps, the directional and normal vector of the lines are calculated. A general linear equation [1] of the vertical is sought. On this vertical, the line width is located. The normal vector of the line passes in this direction of the vertical. Thus, for the vertical of the line, the normal vector is calculated. Subsequently, the coefficient c of the vertical is calculated. Next, the coordinates of the points that lie on this vertical are sought. These points determine the line width. In the first step, the x coordinate of the stationary point is incremented by one. Now it is required to calculate the y coordinate of the new point. Into the general linear equation are substituted normal vector components of the vertical and x coordinate of the stationary point. This identifies the new point at which it is sought whether it is located on the line of a part. Incremental increases continue until the new point lies on a part. If the next point does not lie on a part, the cycle starts all over again but in the opposite direction. Thus the x coordinate of the stationary point is decremented. In other words, the line width is calculated on both sides of the stationary point in the vertical direction. The sum of increment and decrement repetitions

represents the line width in the particular stationary point. The same procedure is repeated for the remaining stationary points.

$$Ax + By + C = 0 \quad [1]$$

The result of the line width is written as the sum of all calculated stationary points divided by the number of these points. This procedure is performed for all lines in polyline. Finally it is calculated the average width for the entire polyline. The procedure is repeated for all polylines of the engineering drawing.

RESULTS

The testing of the algorithm is carried out on the scanned engineering drawings. These scans are vectorized in Ras2Vec. The vectorized drawings are used for the comparison with the results of the algorithm. As mentioned above, Ras2Vec is adjusted to provide the coordinates of the polyline points. These coordinates are processed by the algorithm.

The algorithm meets the expectations and preserves the line width. But as it can be seen on figure 2, not all lines widths are maintained properly. The reason may be the presence of the lines in the polyline with the different widths. Vectorization application connects different types of lines together into the polyline because after thinning this information of the width is lost. Of course, this vectorization application does not have the ability to recognize this difference. Therefore, the resulting width can be different than the actual. On the basis of the tests between the first and the second approach, there is no big difference in the number of thin and thick lines but the first approach seems slightly more precise in indicating the line width. It was expected that the second approach that uses a variable number of points provides better results. The difference of the computing speed of both approaches is negligible.

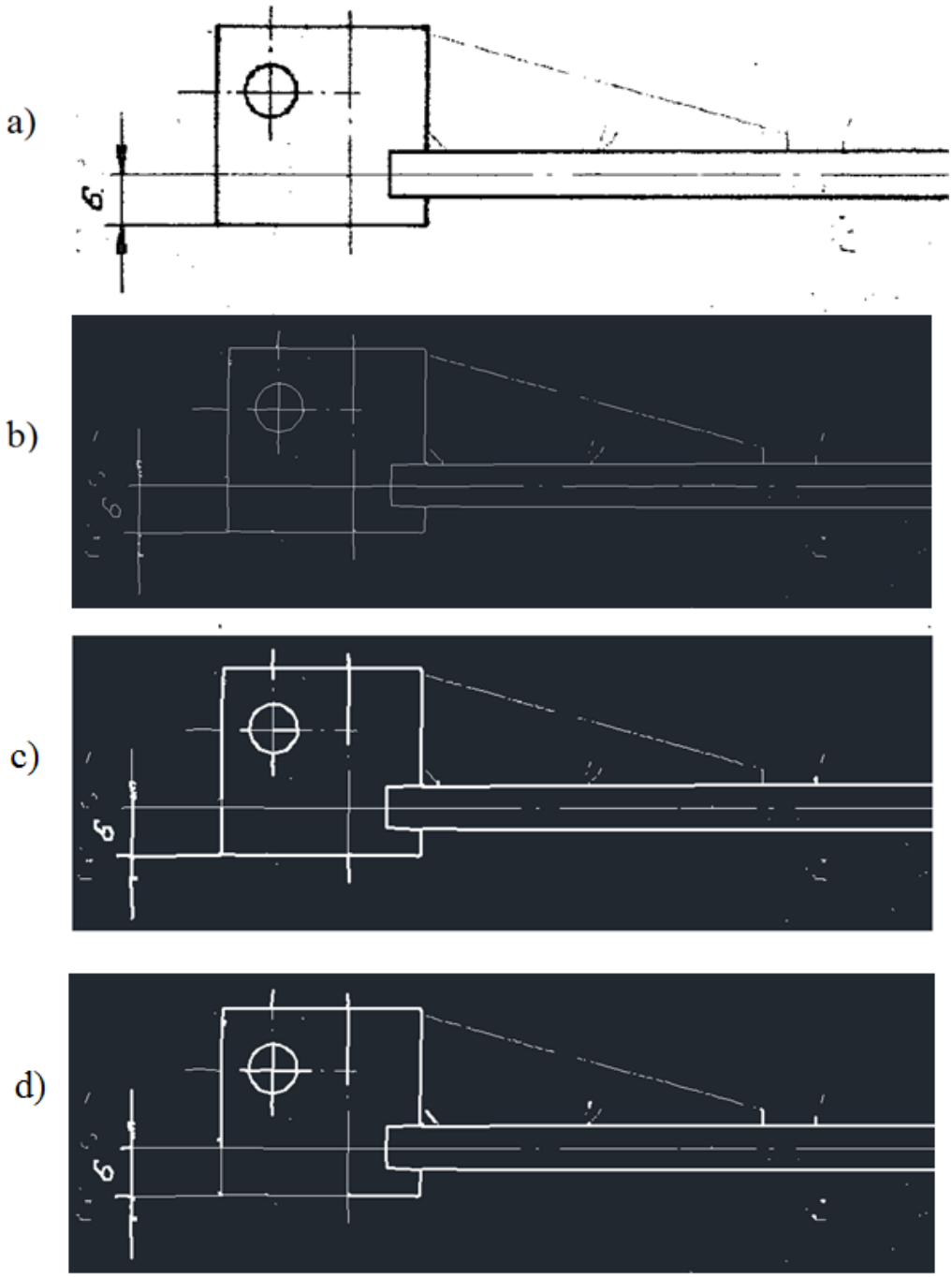

Fig. 2 *a) Original raster image; b) vector drawing without line width preserving; c) vector drawing with the first approach of line width preserving; d) vector drawing with the second approach of line width preserving*

DISCUSSION

The algorithm fulfills the purpose for which it was created. It relatively successfully preserves line width. It was expected that the second approach which uses a variable number of points to determine line width increases the accuracy. This objective was not met. Both approaches are capable to recovery the line width. There would be better results if the individual lines would not be linked into the polyline first but after the line width recovery process. After the indication of unique line width, the individual lines should be connected into polylines based

on line width as one of the criterion. This approach would require another vectorization application which would be a scenario for future work.

CONCLUSION

After the vectorization of an engineering drawing using the vectorization method based on thinning, the information of the line width is lost. The line width is important information about the line in the engineering drawing. In this paper, the algorithm capable of recovering the line width after vectorization is proposed. The algorithm together with the application can be found in the software repository https://github.com/firidion/LINE-WIDTH-RECOVERY.

Acknowledgement

This publication is the result of implementation of the project: "UNIVERSITY SCIENTIFIC PARK: CAMPUS MTF STU - CAMBO" (ITMS: 26220220179) supported by the Research & Development Operational Program funded by the EFRR.

References:

1. VESELOVSKÝ, J., 2002. *Technická dokumentácia a CAD.* (Technical documentation and CAD). Bratislava: STU.
2. HILAIRE, X., TOMBRE, K., 2002. Improving the Accuracy of Skeleton - Based Vectorization, Graphics Recognition - Algorithms and Applications.
3. GOMIS, J. M., COMPANY, P., GIL, M. A. Vectorization in Recovering Engineering Drawings, http://www.regeo.uji.es/publicaciones/N98upv.PDF [Accessed: 20.1.2016].
4. SHEREEN, A. T. et al., 2001. A New Model for Automatic Raster-to-Vector Conversion. *International Journal of Engineering and Technology*, **3**(3), 182-190.
5. ČOMAJ, P., SYROVÁ, L., 2007. *Vektorizácia rastrových obrazov (Vectorization of raster images)*. Bratislava: STU, Fakulta elektrotechniky a informatiky.
6. ZHENG, Y., LI, H., DOERMANN, D., 2005. A Parallel-Line Detection Algorithm Based on HMM Decoding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(5), pp. 777-792.
7. TONG LU, CHIEW-LAN TAI, HUAFEI YANG, SHIJIE CAI, 2009. A Novel Knowledge-Based System for Interpreting Complex Engineering Drawings: Theory, Representation, and Implementation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, **31**(8).
8. GIRIJA DHARMARAJ, 2005. *Algorithms for Automatic Vectorization of Scanned Maps*. Calgary: University of Calgary. Department of geomatics engineering.
9. LAKSHMI, J. K., PUNITHAVALLI, M., 2009. A Survey on Skeletons in Digital Image Processing. In: *Proceeding ICDIP '09 Proceedings of the International Conference on Digital Image Processing*, IEEE Computer Society Washington, DC, USA, pp. 260-269. ISBN: 978-0-7695-3565-4, doi 10.1109/ICDIP.2009.21.
10. SONG, J., CAI, M., LYU, M. R., CAI, S., 2002. Graphics Recognition From Binary Images: One Step or Two Step. In: *16th International Conference on Pattern Recognition (ICPR'02)*. Quebec City.
11. XU, X. W., BAI, Y. B., 2000. Computerising Scanned Engineering Documents. *Computers In Industry*, 42, p. 59-71.
12. SONG. J., LYU, M. R., 2005. A Hough transform based line recognition method utilizing both parameter space and image space. *Pattern Recognition*, **38**(4), pp. 539–552.

13. LLADOS, J., RUSINOL, M., 2014. Handbook of Document Image Processing and Recognition. London: Springer-Verlag Editors: Doermann, D., Tombre, K. kapitola Graphics Recognition Techniques. p. 489-521. ISBN 978-0-85729-858-4. DOI 10.1007/978-0-85729-859-1.
14. KUMAR, H. KAUR, P., 2011. A Comparative Study of Iterative Thinning Algorithms for BMP Images / (IJCSIT). *International Journal of Computer Science and Information Technologies*, **2**(5), pp. 2375-2379.
15. TOMBRE, K., TABBONE, S., 2000. Vectorization in Graphics Recognition: To Thin or not to Thin. In: *International Conference on Pattern Recognition (ICPR'00)- 2*. Barcelona.
16. SONG, J., SU, F., TAI, C.L., CAI, S., 2002. An Object-Oriented Progressive-Simplification-Based Vectorization System For Engineering Drawings: Model, Algorithm, and Performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(8), pp. 1048-1060.
17. HILAIRE, X., TOMBRE, K., 2006. Robust and Accurate Vectorization of Line Drawings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **28**(6), pp. 890-904.
18. VASKÝ, J., GRAMBLIČKA, M., 2014. Vectorization of scanned paper-based engineering drawings – contemporary software abilities. *Applied Mechanics and Materials*, Vol. 693, Trans Tech Publications, pp. 457-462.
19. Davide Libenzi, http://www.xmailserver.org/davide.html [Accessed: 1.2.2016].
20. ZHANG, T. Y., SUEN, C. Y., 1984. A fast parallel algorithm for thinning digital patterns. *Comm. ACM*, 27(3), pp. 236-239.
21. SUBASHINI, P., JANSI, S., 2011. Optimal Thinning Algorithm for detection of FCD in MRI Images. *International Journal of Scientific & Engineering Research*, **2**(9). ISSN 2229-5518.
22. JAGNA, A., 2012. *Some algorithms for image thinning using spatial domain processing.* Jawaharlal Nehru Technological University, India, 2012. URI: http://hdl.handle.net/10603/3466.