Vectorization in Recovering Engineering Drawings

Gomis, J.M.[1]; Company, P.[2] and Gil, M.A.[1]

[1] *Universidad Politécnica de Valencia (Dpto. de Expresión Gráfica en la Ingeniería)*
Camino de Vera S/N. E-46022, Valencia. SPAIN
E-mail: jmgomis@degi.upv.es
[2] *Universitat Jaume I (Technology Department)*
E-12071-Castellón. SPAIN
e-mail: pcompany@tec.uji.es

Abstract

The so-called “electronic documents” are becoming prevalent in Design process. Nevertheless, design documents on paper are still necessary to maintain those products that are now in the market, as well as they are the basis for the design and development of the next generation of products, because they contain the “know-how” on products and processes. Hence, the necessary use of documents on paper reduces the efficiency that could be achieved in this key aspect of design process by simply transferring information contained in paper documents to electronic ones. In order to solve this problem some transfer methods have been developed.

The most widely used method for transferring documents to electronic formats is scanning, which produces “raster images”. The document is broken down into a matrix of pixels. As a result, we may obtain a high quality electronic image of the document (depending on size and density of pixel). But the semantic information contained in the document is not transferred to the electronic file, i.e. the meaning of the symbols is not interpreted.

Therefore, a second step in the process known as “vectorization” is needed, by means of which the information necessary to define geometrical entities (such as straight lines, arcs, etc.) and symbols (such as letters) is obtained from a raster image. Then, this information is stored on an electronic file following the common methods used in CAD files.

The current work presents the state of the art of vectorization. Different vectorization methods are compared and the importance of advancing on different levels of intelligence (capacity of discrimination) is stated. Finally the paper presents the process of vectorization the authors are working on.

1. INTRODUCTION

For Engineering Drawings to become the link between users and CAD systems some present limitations must be overcome. First of all, Engineering Drawings convey implicit 3D information, while CAD systems need an explicit representation of 3D objects. *Geometrical Reconstruction* [1], will allow “explicitation” of 3D geometrical information contained in standard 2D representations. The part of the information that is contained in Technical Drawings but is not represented by means of geometrical projections (dimensions, tolerances...), must also be “understood” by computers. Finally, mismatches, errors and “complicities” present in all Technical Drawings ought to be filtered by computers.

In addition, it is very important to remember that already existing designs suppose an important “know-how”, and are specified in Engineering Drawings. This means that automatic solid-model generation from standardized drawings may be the “bridge” to recover the information related in the thousands of old designs filed in drafting rooms.

The final objective is to be prepared to accept the “worst” scenario in the transfer from drawings to CAD models. That is: sketches as input data and a fully automated process to obtain 3D geometrical models. This process is been traditionally decomposed in two main steps. The vectorization step is the one that converts sketches or line drawings, scanned as “raster” images, in two-dimensional graphs, containing geometrical and semantic information. In fact, two rough phases must be considered in this step: digitization (where documents are converted in raster images) and vectorization (or translation from raster to vector representations).

In a second step, a general-purpose reconstruction algorithm must accept the previously obtained 2D graphs as inputs, and must release a final 3D geometrical model. In a final phase of this second step, the complementary information contained in symbols (like dimensions, tolerances, etc.) must also be added in the corresponding Data Base, related to the 3D model.

In more detail, the process from sketches to 3D models can be decomposed in two main, sequentially ordered, steps:

- *To obtain vectorial format 2D drawings from engineering drawings on paper.*
- *To obtain 3D models from 2D drawings.*

The first objective can be decomposed in three principal aspects:

- *Identification of drawing entities.*
- *Discrimination of texts.*
- *Discrimination of shade patterns, symbols, complementary information, etc.*

This work analyzes the first of these three aspects, one of the first tasks being to evaluate some of the vectorizators available in the market [2] and to analyze some aspects such as the realm of application, integration of pre/postprocess editors, intelligence levels (capacity of discrimination). At the same time, an analysis of the state of the art about vectorization was done. Section 2 presents a summary of this. Section 3 presents the process of vectorization on which the authors of this paper are working at the moment.

Previously, some main similarities and differences between raster and vectorial representations must be highlighted, and a short historical reference is also introduced.

1.1 Vectorial versus raster data

Vectorial and raster formats are the two basic structures to store and handle graphical data. The most widely used CAD, GIS and Graphical Design packages are based on one of these two formats, although they may contain some other functions which permit the use of both.

Raster data describe images in terms of “pictorial elements”, or pixels. Raster data is given as an ordered collection (usually a rectangular matrix) of individual pixels, and each space position (or “resolution element”) has a biunivocal correspondence with one associated pixel. Usually, the numerical value of the pixel determines one or more attributes (like colour, elevation, or identification number). Data are normally input with an optical scanner, a digital

CCD camera or other “raster” input devices. Image resolution is determined by the resolution of the input device and the data source, as it can be a drawing on paper. As raster data files must have pixels for all positions they are limited by the size of the area they represent.

Vectorial data describe geometrical figures in terms of geometrical elements. Elements are given on its algebraic formulation, and the necessary parameters complete the particular description of every element. (i.e. the coordinates of the centre and the radius are the three parameters needed to define a circle contained in the reference plane). Attributes, like colour, thickness, etc., are also stored with the geometrical description [3].

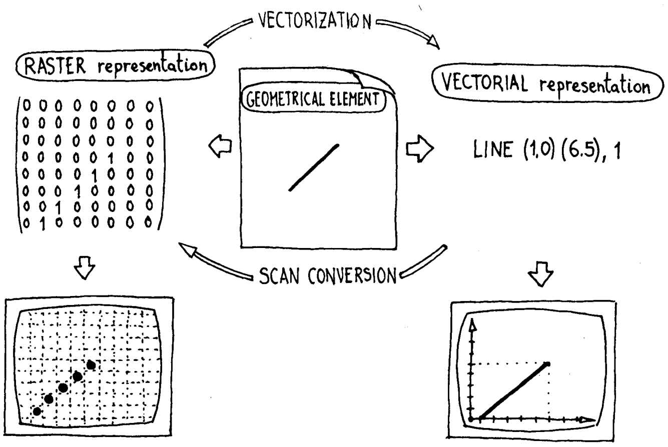


Figure 1. Raster and vectorial representation of a segment.

Raster formats can be considered closer to the image world, whereas vectorial formats are a geometrical abstraction. The latter are more flexible and efficient for geometrical representations, especially for engineering drawings and, in general, for CAD systems, and they need less requirements for handling and storing data.

Conversion from vectorial to raster format is a very well established methodology named “scan-conversion” [4]. On the contrary, obtaining vectorial representations from raster ones is the encouraging field of vectorization.

1.2 Historical background of vectorization

The first experiences on vectorization from scanned images started at the end of the sixties to convert raster data into vectorial information by using macro-computers. Later, this technology was applied to capture cartographic information, even to develop technology based on the capture of particle traces in nuclear physics [5].

In the seventies, some companies, like Mc Dowell Douglas, developed a semiautomatic vectorizer of line-tracing, used in sketches. It consisted mainly of real size sketches for pieces of aircraft. Also, during the seventies Computervision Corporation developed a system of automatic vectorization for cartographical representations and another system for vectorizing circuit diagrams.

At the beginning of the eighties different corporations of vectorization scanning emerged: Intergraph, ANATech, SkanteK, Optigraphics, Metagraphics, and Audre. During this period, FORMTEK studied the use of raster images as an alternative to CAD. During the second half of the 80s in the EEUU emerged companies like GTX Corportation, Information and Graphic Systems, Image Systems Technology (recently bought by Softdesk Inc) and Arbor Image, and in Germany, M.O.S.S. Gmbh and Softelec GmbH [5].

Nowadays the technology of automatic conversion of raster images to vectors allows a large variety of graphical information to be scanned and converted into vectorial formats. However from the point of view of information recovery from engineering drawings the problem has not been completely solved. The information recognition and discrimination capacities show the difference in performance among all vectorizers available in the market.

2. CONVERSION FROM RASTER TO VECTORIAL FORMATS

As said before, the conversion from raster to vectorial formats process can be decomposed in the two main aspects of digitization and vectorization. In turn this two main tasks can be studied in more detail if some more specific aspects are considered. This is the reason why the process of conversion from raster to vectorial formats is normally structured into the following tasks:

- Treatment of the image.
- Thinning.
- Vectorization.
- Entity recognition.

We are going to present the four processes in more detail. Nevertheless, it must be noticed that, traditionally, digitization software development is tightly tied to scanners hardware development, and consequently a detailed study is out of our present scope.

2.1 Treatment of the image

In order to obtain the appropriate vectorization, the raster image must pass trough a number of cleaning and restoration processes. This is necessary because the scanned images contain different kinds of defects, which may come from the original drawing or from the scanning process. The defects due to the conditions of the original drawing consist mainly of stains on the paper, wrinkles, holes and any other type of marks on the paper. The faults of the scanning process may be due to technical problems in the scanner, paper gloss or irregular laser beam reading of the scanner. All these problems give rise to a raster image with objects which shouldn’t appear in an ideal raster image. These strange objects consist mainly of:

- Isolated groups of pixels forming small patches
- Small dots in the background colour appearing inside the figures
- Lines, the ends of which are very close together in the original drawing but which in the raster image appear as continuous.
- Gaps in the lines where the original shows a continuous line

To solve these defects different morphological operations can be carried out [6], consisting of adding or removing pixels from the original drawing. The most commonly used are:

- *Stripping*, consisting of removing from the drawing the pixels which are in contact with any background pixel. This technique allows us to remove from the raster all the objects smaller than required.
- *Expansion*. Is the opposite process to stripping, and consists in to add to the drawing those background pixels that are in contact with the pixels of the drawing.

This process allows us to remove the background-colored pixels appearing inside the figures.

- *Opening*, which consists of a continuous stripping followed by expansion, thus removing isolated pixels without changing the dimensions of the object. In addition, this process allow us to separate lines whose ends are very near in the original drawing and which in the raster appear as one. However this has the problem of opening gaps between objects which originally were in contact.
- *Closing*, consists of the reverse process to the previous one, that is, expansion followed by erosion. Here the dimensions of the object are also maintained and the small gaps inside the objects are closed. In addition, it lets to remove, from the raster image, line discontinuities that didn't exist in the original drawing.

These operations, done by using filtres, can be implemented with the application of several parameters like: application thresholds (minimum number of neighbouring pixels to modify one pixel), depth (number of times one operation is repeated) or type of neighborhood. And they should be used in those regions of the image selected by the user so that the original image is not modified.

In the case of other defect types (like those produced by big stains or holes on the paper), they should be solved by using the related tools available in the image treatment programs.

2.2. Thinning

If we want to distinguish the different widths of the lines we have to scan the drawings at the appropriate resolution so as to allow us to observe the differences in the number of pixels necessary to show the width of the lines with different thickness. Therefore it is necessary for the line to be several pixels wide. On the other hand it is also advisable for the lines of the raster to be as simple as possible, to reduce the amount of information to be processed. Therefore, to get lines one pixel width is the most convenient thing.

In order to obtain a simplified image from the original raster image, where all lines are one pixel wide different thinning or skeletonization techniques can be used. With these techniques we obtain a skeleton one pixel wide which represents all the lines of the original drawing.

There are different thinning algorithms used for different applications: character recognition, printed circuits checking, biochemical imagery, quantitative typing, finger print analysis, etc. [7].

Lee, Lam and Suen [8, 9], on one hand, and Zhang and Wang [10], on the other, compare the characteristics of these algorithms. One of the conclusions is that there is not one single thinning algorithm that can be applied to all the cases that may occur. Other conclusions are the unexpected results of many of these algorithms when applied to different problems. These problems can be:

- To try to obtain the *skeleton of a very wide area*. To solve this problem Kasturi et al [11] suggest to process the image previously in order to separate these areas from the image and to process the remainder with the thinning algorithm. This method should be used cautiously since some thick lines as well as relevant information, like symbols or arrow heads, can also be separated.
- At *line crossings*, mainly if they are thick. In these cases, in the area of intersection it is difficult to determine the precise pixel where the crossing takes place. To solve

these problems (known as *X* and *Y*) Hu and Li [12] propose a thinning algorithm specially designed to maintain line crossing. This algorithm presents the drawback of not guaranteeing image line connectivity and it also considers the lines to have similar width, something that is not always the case in engineering drawings.
- At the *corners*, as the thinning algorithm does not place adequately the node at the corner. To solve this problem, Janssen and Vossepoel [13] use one method in which after thinning and vectorizing, the position of the intersection points is corrected by using some criteria which have into account the area surrounding the intersection or corner. This method presents the inconvenience of causing the vectors thus generated to go out of the area of the original line, generating in such a way differences between the digitized and the vectorized images.

2.3. Vectorization

Once obtained the skeleton of pixels, the next step is their approximation to vectors. Vectorization should be done in such a way so as to minimize the differences between the skeleton lines and the vectors representing them.

Janssen and Vossepoel [13] have studied the different vectorization methods. It is worth noting that all of them, except for the method called "separate pixels", require the previous stage for their application. These methods are:

- *The Area* method. Used by Wall and Danielson [14]. The criterion used in this method is the maximum area skewness allowed per unit length comprised between vectorization and approximated curve. The last point of the curve not violating this criterion becomes the main node in the vectorization, the process being repeated up to the end of the curve. The main problem of this method is that the nodes sometimes do not correspond with the corners of the curve because a new vector is only defined when the criterion is violated.
- *Intersection Cone* method. Presented by Williams [15, 16] and Skalnsky and Gonzalez [17]. These methods is based on defining a circle around each point of the curve. The approximated vector should intersect all these circles.
- *The Minimax* method. Presented by Kurozumi and Davis [18]. This method is based on the minimization of the maximum distance between the original curve and the approximated vectors. Here the nodes of the approximated vectors are not necessarily a segment of the original curve
- *Tolerance Sector* method. Proposed by Dettori [19] and Leung and Yang [20]. This method adjusts an area around the points to be approximated either by maximizing the number of original points in this area or the area length, or by minimizing the area width. The resulting vector is the central axis of the area. When the starting point is a branching point, ambiguous situations may arise, mainly when the width of the area is large.
- *Run-length* method. Proposed by Pavlidis [21, 22]. In this method, after calculating the coding of the image run lengths, a graph of adjacent lines is done. This graph is used to determine the segments of vectorization. In this method the resulting vectorization can depend on the image orientation.
- *Marking-scanning* method. Proposed by Sirjani and Cross [23]. This method uses as input data the coordinates of the first point of a curve and the code of contour linkage and it consists of marking all those points that can be the end of a straight

line. The unnecessary marked points are neglected, and the nodes of the vectorization are obtained.

- *Maximum and Minimum Curvature* methods. Used by Teh and Chin [24] and Ray and Ray [25]. These methods are based on the use of the maximum and minimum curvature points to vectorize. These points contain important information for the processing of human vision recognition systems.
- *Maximum perpendicular distance* method. Used by Douglas-Peuker [26], Ramer [27]and Janssen and Vossepoel [13]. This method uses the maximum perpendicular line in pixels D from the vectorized to the curve to approximate. The algorithm starts with two dominant points of the curve as nodes of vectorization. When the maximum perpendicular distance from the vectorization to the curve is greater than D, a new node is added to the vectorization in the region of maximum skewness. This process is repeated iteratively. This method tends to place all nodes on the corners of the original image, which is very interesting for vectorizing raster images of engineering drawings.
- *Separate Pixel* method. Developed by Dori et al [28] to vectorize engineering drawings. This method does not require the stage of previous thinning. Here the objects are processed by a zigzag algorithm, which allows the vectorization to be obtained. This method recognizes joins in the drawing and most of the arc segments are recognized as arc segments (and not as sequences of line segments) In addition this method allows arrows to be recognized in engineering drawings. A disadvantage of this method is that approximately 12 thresholds have to be fitted for the vectorization algorithm.

All the previous methods, except for the last one, show a number of problems due mainly to the use of the thinning techniques. These problems involve the slow speed of the algorithms, the generation of extra line segments, branching and distortions.

Han and Fan [29] propose an alternative approach to obtain the skeleton of engineering drawings. The starting skeleton that the algorithm generates is in vectorial form, thus avoiding the following step of vectorization, that is, the same results as with the method of “separate pixels” are obtained, though faster and with less storage requirements.

This method consists of four sections:

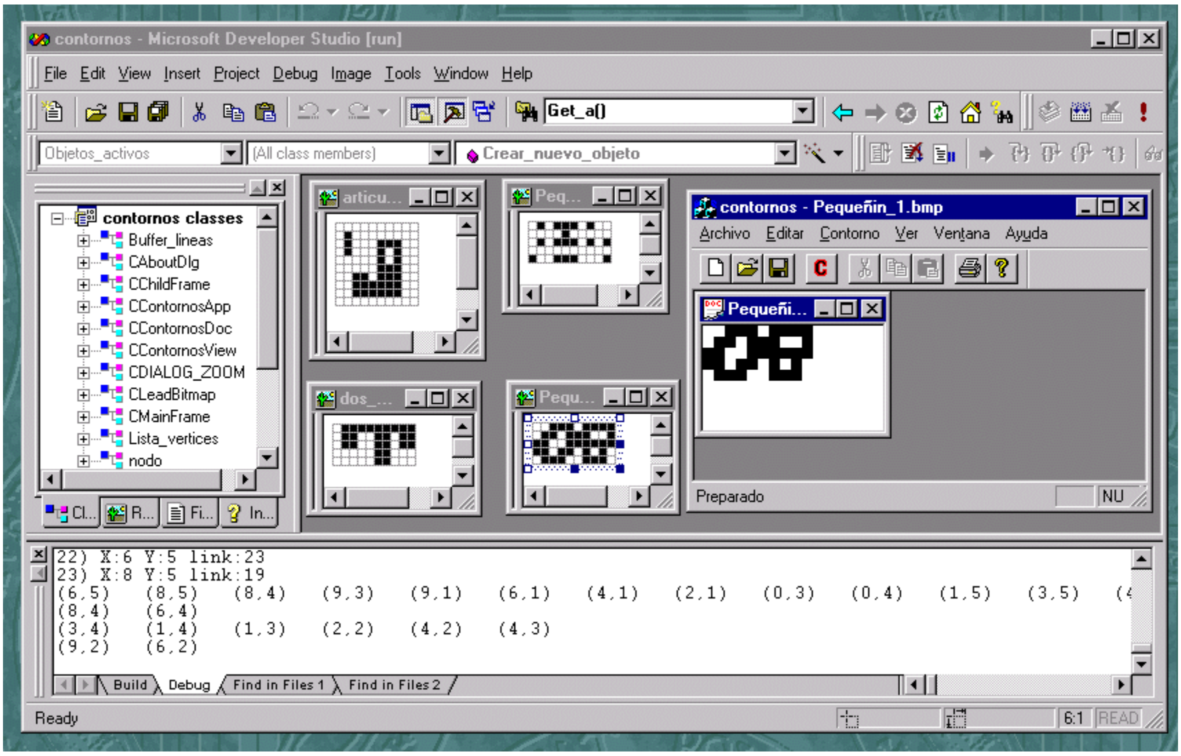
- *Contour extraction and vectorization*. The input image is transformed into a vectorized image of contours using the techniques based on slope variation [30] curve fitting [31] or polygonal approximation based on the image run-length compression [32].
- *Contour vector pairing*. Here the vector shape (if the contours are represented by circular arcs and curves besides straight lines), vector directions, overlapping of the image on the vectors and distance between vectors are obtained
- *Skeletonization of paired contour vectors*. Here the central line running between the paired sections of the opposite vectors of the original drawing is obtained. By generating these lines discontinuities appear at the joining points.
- *The skeleton vectors connection*. Here, the previously mentioned discontinuities are solved by joining the corresponding lines so as to fill the created gap. The techniques used are based on the graph theoretic technique.

2.4 Entity Recognition

After obtaining a representation of the image with vectors, the following step is to recognize the basic entities of the drawing, that is, straight lines, arcs and circles. At this stage techniques like those presented by Brusola [33] or methods based on the Hough transform [34, 35] can be used. A problem arising at this point is that the critical points of the lines do not coincide with the inflexion points at the transitions straight line-arc or arc-arc, having to recalcultate these points.

3. OUR VECTORIZATION ALGORITHM

The application was developed using Microsoft's Developer Studio, to run on PC platforms under Windows NT and Windows 95 operating systems. C++ was used to implement the calculations and data management. Graphical User Interaction was achieved by calling Leadtools, by Leadsoftware inc., and win32 operations through Visual C++. In figure 2, the general appearance of work environment desktop is shown.



```
22) X:6 Y:5 link:23
23) X:8 Y:5 link:19
(6,5)   (8,5)   (8,4)   (9,3)   (9,1)   (6,1)   (4,1)   (2,1)   (0,3)   (0,4)   (1,5)   (3,5)   (
(8,4)   (6,4)
(3,4)   (1,4)   (1,3)   (2,2)   (4,2)   (4,3)
(9,2)   (6,2)
```

Figure 2. General view of work environment desktop.

The vectorization method we have followed in this work is divided in the following four stages:

Stage 1. Previous treatment of the image

1.1 *Cleaning and Restoration Filters*: Stripping, Expansion, Opening and Closing, including the application threshold (depending on whether the type of contour is

8-connection or 4-connection) and depth (number of times that the operation is repeated).

1.2 *Filter for the discrimination of solid areas*: application of N strips folllowed by N expansions compared with the drawing.
1.3 *Filter for the determination of line width.*
1.4 *Filter for the determination of arrow heads.*
1.5 *Filter for the determination of shadowing.*

Stage 2. Generation of the skeleton by contour pairing

2.1. *Contour extraction*: The vectorized contours are extracted by using the technique of polygonal approximation based on image run-length compression
2.2. *Contour line pairing*: Pairing of the segments that define the contours of one line. Contour segments do not have to match all along its length. In such a case, the segment will be divided into two (paired and not paired) segments. Here the width of the paired lines is measured. The pairing conditions are:
    - Slope of approximately equal lines.
    - Ratio between the overlapping length of their projection on one axis and the length of the shorter of the two line segments must be greater than one preselected threshold. (90%)
    - At least two of the four distances from one terminal point to the other line and the lines must be smaller than a given threshold (maximum width of the lines).
2.3. *Identification of arrow heads*: analysing the existing unpairs permits to identify those caused by the presence of arrow heads.
2.4. *Generation of the skeleton vectors*: These vectors are obtained by joining the central points of the segments connecting the ends of paired contours.
2.5. *Connection of contour skeleton*: by using the graph theoretic technique.

Stage 3. Simplification of the skeleton

At this stage the size of the skeleton is reduced, in order to be more easily handled.

3.1. Discontinuous lines recognition.
3.2. Replacement of discontinuous lines by continuous lines with their corresponding attributes.
3.3. Simplification of straight lines checking the quasi-collinearity of the segments.
3.4. Removal of the lines non-exceeding a certain length.

Stage 4. Identification

At this stage the arcs and circles formed by straight lines included in the skeleton are identified. In this case the methods used are based on the Hough transform.

4. CONCLUSIONS

In this paper we have highlighted the importance of vectorization in the processing of the information of engineering drawings and in the reconstruction of 3D models from drawings on paper.

Likewise we have presented the state of the art of the different methods of vectorization. And the method developed by the authors has been explained in some detail. The authors classify their work in the field of the reconstruction of drawings, by identifying the entities of the drawing from the starting image. To try to develop text discrimination, elements definition, symbols detection, etc. are the intended following stages.

5. ACKNOWLEDGEMENTS

This work is framed within the research program “Reconstruction of 3D objects from 2D representations”, sponsored by the local authorities of the Comunidad Valenciana (project GV97-TI-04-39). The authors are also indebted to undergraduate student M.A. Herreras who carried out the algorithm implementation.

6. REFERENCES

[1] J.M. Gomis and P. Company. "Reconstrucción geométrica tridimensional". *Anales de Ingeniería Gráfica*, 1998 (Not yet published).

[2] J.M. Gomis. M. Alcañiz, C. Diaz del Rio, M.A. Herreras y M.A. Landete. “La vectorización y la gestión de documentos en los sistemas CAD”. *X Congreso internacional de Expresión Gráfica en la Ingeniería*. Pendiente de publicación. 1998.

[3] Y. Wu. “Raster, Vector and automated digitizing”. 1996. *http://www.ablesw.com/rasvect.html.*

[4] Foley J.D., Van Dam A., Feiner S.K. and Hughes J.F. *Computer Graphics. Principles and Practice.* Ed. Addison Wesley, 1990.

[5] C. Machover. “The CAD/CAM Handbook”. *McGraww-Hill, New York.*

[6] J.F. Harris, J. Kittler, B. Llewellyn and G. Preston. “A modular system for interpreting binary pixel representations of line-structured data”. *Pattern Recognition Theory and Applications, NATO Advanced Study Institutes Series*, Vol. C81. 311-351. 1981.

[7] VV.AA. *Internat. Journal of Pattern Recognition and Artificial Intelligence*. Vol. 7. 1993.

[8] S.W. Lee, L. Lam and C. Y. Suen. “A systematic evaluation of eskeletonization algorithms”. *International Journal of Pattern Recognition and Artificial Intelligence.* Vol. 7. 1203-1225, 1993.

[9] L. Lam, S.W. Lee and C.Y. Suen. “Thinning methodologies –A comprehensive survey”. *IEEE transactions on pattern analysis and machine intelligence.* Vol. 14. 869-885, 1992.

[10] Y.Y. Zhang and P.S.P. Wang. “Analytical comparison of thinning algorithms”. *Internat. Journal of Pattern Recognition and Artificial Intelligence.* Vol. 7. 1227-1246, 1993.

[11] R. Kasturi, S.T. Bow, W. El-Masri, J. Shah, J.R. Gattiker and U.B. Mokate. “A system for interpretation of line drawings”. *IEEE transactions on pattern analysis and machine intelligence.* Vol. 12. 978-991, 1990.

[12] G. Hu and Z.N. Li. "An X-crossing preserving skeletonization algorithm". *Pattern recognition letters*. 1031-1053.

[13] R.D.T. Janssen and A.M. Vossepoel. "Adaptative vectorization of line drawing images". *Computer vision and image understanding.* Vol. 65. 38-56, 1997.

[14] K. Wall and P.E. Danielsson. "A fast sequential method for polygonal approximation of digitized curves". *Compt. Vision Graphics Image Process*. Vol. 28. 220-227, 1984.

[15] C.M. Williams. "An efficient algorithm for the piecewise linear approximation of planar curves". *Compt. Vision Graphics Image Process*. Vol. 8. 286-293, 1974.

[16] C.M. Williams. "Bounded straight-line approximation of digitized planar curves and lines". *Compt. Vision Graphics Image Process*. Vol. 16. 370-381, 1981.

[17] J. Sklansky and V. Gonzalez. "Fast polygonal approximation of digitized curves". *Pattern Recognition*. Vol. 12. 327-331, 1980.

[18] Y. Kurozumi and W.A. Davis. "Polygonal approximation by the minimax method". *Compt. Vision Graphics Image Process*. Vol. 19. 248-264, 1982.

[19] G. Dettori. "An on-line algorithm for polygonal approximation of digitized plane curves". *Proc. 6th Int. Conf. Patt. Rec.* Vol. 2. 739-741, 1982.

[20] M.K. Leung and Y.H. Yang. "Dynamic two-stip algorithm in curve fitting". *Pattern recognition*. Vol. 23. 69-79, 1990.

[21] T. Pavlidis. "A Hybrid vectorization algorithm". *Proc. 7th Int. Conf. Patt. Rec*. 490-492, 1982.

[22] T. Pavlidis. "A vectorizer and feature extractor for document recognition*". Compt. Vision Graphics Image Process*. Vol. 35. 111-127, 1986.

[23] A. Sirjani and G.R. Cross. "An algorithm for polygonal approximation of digitized object". *Pattern recognition*. Vol. 7. 299-303, 1988.

[24] C.H. Teh and R.T. Chin. "On the detection of dominant points on digital curves". *IEEE Trans. Pattern Anal. Machine Intell*. Vol. 11. 859-872, 1989.

[25] B.K. Ray and K.S. Ray. "Detection of significant points and polygonal approximation of digitized curves". *Pattern recognition*. Vol. 13. 443-452, 1992.

[26] D.H.Douglas and T.K.Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". *Can. Cartographer.* Vol. 10. 112-122, 1973.

[27] U. Ramer. "An iterative procedure for the polygonal approximation of plane curves". *Compt. Graphics Image Process*. Vol. 1. 244-256, 1972.

[28] D. Dori Y. Liang. J. Dowell and I. Chai. "Sparse-pixel recognition of primitives in endineering drawings". *Machine Vision Appl*. Vol. 6(2-3). 69-82, 1993.

[29] C.C. Han and K.C. Fan. "Skeleton generation of en¡gineering drawings via contour matching". *Pattern recognition.* Vol. 27 nº2. 261-275, 1994.

[30] P.P. Nahin. "The theory and measurement of a silhouette descriptor for image pre-procesing and recognition". *Pattern Recognition.* Vol. 6. 85-95, 1974.

[31] M. Plass and M. Stone. "Curve-fitting with piecewise parametric cubics". *Computer Graphics.* Vol. 17 nº 3. 1983.

[32] D. W. Capson. "An improved algorithm for the secuential extraction of boundaries from a raster scan". *Computer Vision, graphics and image procesing* Vol. 28. 109-125, 1984.

[33] F. Brusola. “Contribuciones al desarrollo de un sistema de regeneración de dibujos a mano alzada por ordenador”. *Tesis Doctoral. Departamento de expresión gráfica en la ingeniería. Universidad Politecnica de Valencia*. 1989.

[34] R.K.K. Yip, P.K.S. Tam and D.N.K. Leung. “Parallel algorithm for circle detection in images”. *Pattern Recognition.* Vol. 25 nº9. 1007-1022, 1992.

[35] S. Kumar, N. Ranganathan, D. Goldgof. “A new curve detection method: randomized Hough transform”. (RHT). *Pattern Recognition.* Vol. 27 nº 8. 1019-1028, 1994.