Proceedings of International Conference on Image
Processing and Character Recognition

A region-based vectorization method

John Y. Chiang[+]
[+]Department of Applied Mathematics
National Sun Yat-sen University
Kaohsiung, 80242, Taiwan

Y. C. Leu[++]
[++]Industrial Technology Research Institute
Computer & Communication Research
Laboratories
Hsinchu, 80810, Taiwan

Yun-Lung Chang[+++]
[+++] Chung-Shan Institute of Science and
Technology
(CSIST) P. O. Box 90008-8-11, Lungtan
Taoyuan,Taiwan

Abstract

Lines and junctions are principal features for a line image. The spatial relationships established among them are usually employed in applications such as OCR and architectural blueprint vectorization. The conventional thinning techniques often suffered the pitfall of spurious junction points that are crucial features to derive. The liability of the shape of the skeleton resulting from multiple fork points connected through several short branches will impede the further recognition stage. Also, different thinning algorithms often lead to different results and are very sensitive to the presence of noise. Since morphological thinning operation proceeds in a step-by-step peeling of outer-layer pixels down to a unit-pixel skeleton, the computation requirements are high and very time-consuming.

An algorithm which differentiates from the conventional thinning algorithms in vectorizing a raster line image called maximal inscribing circle (MIC) is used. This region-based vectorization algorithm takes the ensemble of pixels within the line segments collectively as legitimate candidates in deciding the vectorized representation. An iterative procedure is outlined and the criteria for merging short straight line segments into the corresponding curve representation are described. This method can not only segment the lines and junctions, construct their spatial relationships in a computation efficient manner, but also retain their line width.

Experimental studies comparing the MIC algorithm with a conventional thinning algorithm are performed using scanned raster diagrams. Engineering blueprints with various sizes and complexities are used as test raster images to demonstrate the versatility of our method. The results show promising performance in comparison with that of the traditional approaches. Finally, conclusions are made and future works are discussed.

1. Introduction

Line drawing hardcopies, *e.g.* engineering drawings, architectural blueprints and maps, are subject to damage, smear and take lots of shelf space. A raster representation of the paper drawing is difficult to reproduce, manipulate, and modify by using the computer and occupies a large amount of memory space. A conversion from raster representation to a vectorized form can render easier computer processing and require much less storage space. In vectorizing a bit-mapped graph, several options are available. An operator may use a large-format tablets to manually input the corresponding vector points. However, due to the disjunction between picking points on the tablet and seeing the results on the screen, it can be difficult to be sure each and every last detail from paper to computer are duplicated in the right place. Also, this method requires intensive man power and is not feasible for a reasonable amount of printing backlogs. Using automatic conversion software allows accurate duplication of the originals and supports various vector editing features. The process of converting the bit-mapped graphics into the corresponding vector representation is called raster-to-vector conversion.

Vectorization of the bit-mapped image into corresponding line or curve primitives, which represent significant features of the original image, plays an important role in a wide area of applications. For example, the primal constituents of engineering drawings, architectural blueprints and cartographic data are points, line segments, polygons and curves. If a scanned bit-mapped graphics can be vectorized into a collection of lines and curves, archival spaces can be saved significantly. Furthermore, if the vectorized image is converted into CAD file format, then the rich vector-editing features provided by the custom CAD software can be utilized to do whatever modification operations necessary. For character recognition applications, how to correctly map the input bit-mapped character shapes into connected segments of lines and curves without altering the crucial characteristics on the number and position of junction points is the key issue in determining the success of the postprocessing recognition stage.

Shape analysis plays an important role in the recognition of objects. Shape is a prime information carrier in the applications of machine vision and pattern recognition. The majority of shape analysis techniques can be categorized into two classes: boundary-based techniques and region-based techniques [1]. Boundary-based techniques analyze shapes by extracting features based on object's boundaries. For example, ploygons are fitted according to boundary edge points for the computation of area, moments, Fourier descriptors, *etc* [1,2]. In contrast, region-based techniques represent shapes based on the object's geometrical structure. For example, skeletons are used as "stick figures" to represent the abstract structure of objects for topological analysis and classification [1].

In analyzing line drawings, the hardcopies are usually scanned first to obtain digital versions. The scanned line drawing images usually are composed of multipixel width segments. Consequently, the traditional approaches usually apply morphological thinning techniques to reduce the complexity of data acquired. Skeletons are "stick figures" representing the abstract structure of objects for topological analysis and classification. The process of obtaining the "stick figures" from a binary image is called the thinning process. The purpose of this thinning operation is to reduce the original multi-pixel-width image into a one-pixel-width skeleton that preserves the significant features of the original line drawing. The skeleton of a digital object can be used in many applications to permit a simpler structural analysis and more intuitive design of recognition algorithms. Thin-line representations of elongated patterns would be more amenable to extraction of critical features such as end points, junction points, and connections among the components. Naturally, for a thinning algorithm to be effective, significant features of the pattern shall be preserved while eliminating local noises without introducing distortions of its own.

Thinning is one of the most important preprocessing steps for feature extraction in many pattern recognition systems. In the optical character recognition (OCR) application, the thinning process extracting the basic features of characters is generally followed by the stroke extracting process. In binary line drawings such as electrical schematic, map and architectural blueprint vectorization, the thinning process plays an important role to reduce the complexity of data acquired [4]. However, these rule-based thinning techniques often suffer the following major drawbacks. The application of a thinning procedure usually introduces spurious junction points that are crucial features to derive. The short branches connected through these junction points will distort the shape of the skeleton and impede the further recognition stage. Also, different thinning algorithms often lead to different results and are very sensitive to the presence of noise. Since morphological thinning operation proceeds in a step-by-step peeling of outer-layer pixels down to a unit-pixel skeleton, the pixel-based computation requirements are high and very time-consuming.

Owing to the above-mentioned drawbacks posed by the thinning algorithms, different approaches which deviate from the conventional thinning methods to obtain the spatial relationship among lines and junctions for a line image are developed [9, 10]. A procedure that differentiates from the conventional thinning algorithms in presenting a segregation approach based upon the line continuation and symmetry property to perform the segmentation and association among lines and junctions for a line image is outlined [9]. This approach gives a more precise segmentation of line segments and junctions than the conventional thinning algorithms do. But this method has high computation complexity and a single line will partite into a plural number of lines connected via multiple junction areas.

In order to analyze line drawings without introducing the above-mentioned problems, a new approach, called maximal inscribing circle (MIC) is presented in this paper [10,11,12]. This new method takes the ensemble of pixels within the line segments collectively as legitimate candidates in deciding the final vectorized representation. The MIC method based on digital circular disk that is defined first leads to the seeking of a maximal inscribing circle, and later is employed to derive the direction and width of a line segment. Since only line segments are used as an approximation to the original raster image, curve portions are characterized by a connected sequence of short straight line segments. The length of the line is thus an implication for the curvature of the specific segment. Further merging of these short line segments into a unified curve representation, e.g. Bézier curve, is followed. The results show that MIC method is computation efficient, robust, and may render optimal multi-pixel-width vectorized line representation at user's discretion.

In what follows, the MIC method is introduced next. An iterative procedure is outlined and the criteria for merging short straight line segments are described. The control points for the Bézier curve representation are also located. In Chapter 3, an experimental study comparing the results obtained by employing MIC and a conventional thinning method is performed. Engineering blueprints are used as test raster images to demonstrate the versatility of our method. Finally, conclusions are made and future works are discussed in Chapter 4.

2. Maximal Inscribing Circle

In analyzing line drawings, two principal features, namely, lines and junctions, must be obtained first. A line is characterized by its end points, slope, and possibly width. Junctions serve as linkages among intersecting line segments. Different line segments may associate through common junction areas. A correct construction of line segments and association among line segments via junctions are vital to the next recognition stage.

Theoretically, an ideal line primitive shall have zero width. However, a scan-converted primitive occupies a finite area in the hardcopy form. A raster line image is regarded as a rectangle of a certain thickness covering a portion of digital grids called the line segment. The MIC approach is based on the observation that for a maximal-radius circle inscribing a line segment, the direction of the diameter shall be perpendicular to that of the line length without presence of noise in the continuous domain. The derived slope, which is perpendicular to the diameter of maximal inscribing circle, approximates the slope of the real line segment. For example, In Fig. 2.1, P_1 and P_2 are the border points of a non-zero-width line segment L across the diameter of the maximal inscribing circle C. S_1, the slope of the line perpendicular to $\overline{P_1P_2}$, is an approximation to the slope of the line segment L.

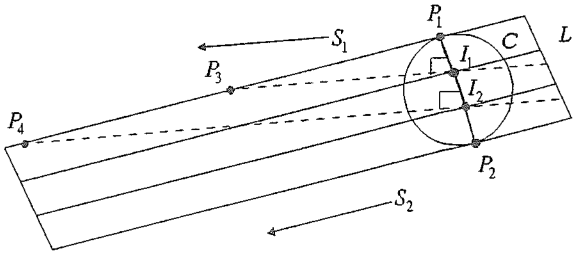

Fig. 2.1. C is the maximal inscribing circle of line segment L. P_1 and P_2 are border points in contact with the perimeter of the circle C across the diameter. S_1 is the slope of the dashed line perpendicular to $\overline{P_1P_2}$, found in the first pass. S_2 is the slope of $\overline{P_3P_4}$ found in the second iteration and a closer approximation to the real slope of the line image. Usually, points more than two shall appear across the width and along the border. Here, only P_1, P_2 and P_3, P_4 are shown for simplicity.

However, for the ease of computer processing, an image function $f(x,y)$ must digitize in both space and amplitude domains. Image sampling is referred to as digitization of the spatial coordinate (x,y), while amplitude digitization is called gray-level quantization. The sampling process may be viewed as partitioning the xy plane into a digital grid, with the coordinates of the center of each grid being a pair of elements from the Cartesian product Z× Z, which is the set of all ordered pairs of elements (x,y). Digital circular masks are used to obtain a maximal inscribing circle by successively enlarging the radius of the fitting circular mask. A 5×5 circular mask is shown below:

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

These inscribing circular masks are obtained by using the following equations:

$$x = \left\lceil \frac{d}{2}\cos\theta \right\rceil ,$$

$$y = \left\lceil \frac{d}{2}\sin\theta \right\rceil ,$$

where d is the diameter of the digital disk, $d \in N$, $0 \le \theta < 2\pi$. The circumference of the circular mask formed partitions areas into three disjoint connected areas corresponding separately to regions within the circle, on the circle, and outside the circle, *i.e.*,

$$R_1 : x < \left\lceil \frac{d}{2}\cos\theta \right\rceil \text{ and } y < \left\lceil \frac{d}{2}\sin\theta \right\rceil ,$$

$$R_2 : x = \left\lceil \frac{d}{2}\cos\theta \right\rceil \text{ and } y = \left\lceil \frac{d}{2}\sin\theta \right\rceil ,$$

$$R_3 : x > \left\lceil \frac{d}{2}\cos\theta \right\rceil \text{ and } y > \left\lceil \frac{d}{2}\sin\theta \right\rceil ,$$

where R_1 and R_3 are 4-connected components, R_2 is an 8-connected component and R_1, R_2 and R_3 are disjoint connected components.

For each interior pixel of the line segment under consideration, a circular mask whose circumference touches line border pixel is defined as an inscribing-circular-mask (ICM). For example, the ICM of the pixel (14,4) shown in Fig. 2.2 is 5. All the pixels confined by an ICM, including circumference, shall be completely contained within the line segment. Every interior point (*x*, *y*) of the line segment is mapped to one and only one $N \times N$ ICM, denoted as *ICM*(*x*, *y*)=*N*. For example, the ICM of pixel (14,4), shown in Fig. 2.2, is a 5×5 circular mask. Thus, *ICM*(14,4)=5.

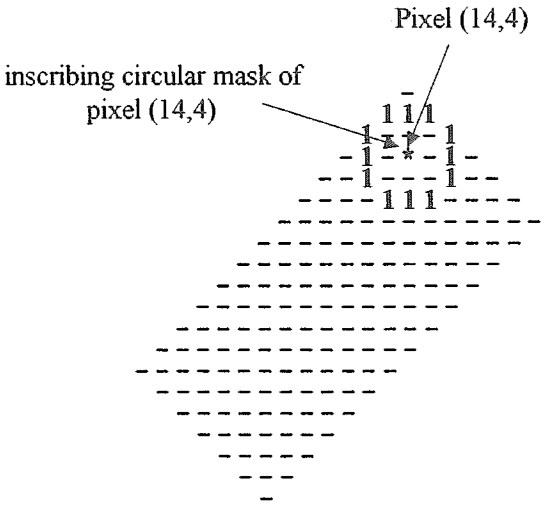

Fig. 2.2 The ICM of pixel (14,4) is a 5×5 circular mask, and denoted as *ICM*(14,4)=5.

As we process in a left-to-right, top-to-bottom fashion, pixels close to the medial axis of line image are corresponding to a larger-radius ICM. Therefore, a pixel with a larger ICM value indicates a direction closer to the medial axis. For example, in Fig. 2.3, the ICM of the pixel (*x*-2, *y*+2) is larger than the ICM of the pixel (*x*, *y*) and the pixel (*x*-2, *y*+2) is deemed closer to the medial axis of line image than that of the pixel (*x*, *y*). We employ an iterative comparison procedure to find a maximum of the inscribing circular masks as follows. The ICM value of current central pixel is compared with that of its eight neighbors. If the ICM of the central pixel is less than any of its eight neighbors, then the pixel with larger ICM value is chosen as the central pixel in the next iteration and the comparison procedure restarts again. Otherwise, the procedure stops and the pixel with the maximum ICM value is considered to be the center of a maximal inscribing circle. For example, in Fig. 2.4, *ICM*(14,4) is less than *ICM*(14,5). The MIC-finding procedure uses pixel (14,5) as a starting new central pixel to test ICM values of its eight neighbors. Since *ICM*(14,6) is larger than *ICM*(14,5), pixel (14,6) is chosen as the central pixel in the next iteration. Finally, the MIC-finding procedure stops when selecting pixel (14,7) as the central pixel, because all the ICM values of the pixel (14,7)'s eight neighbors are no greater than *ICM*(14,7). The ICM of pixel (14,7) is considered to be the maximal inscribing circle (MIC) of this line segment. The MIC-finding procedure is outlined below: Assume *Max* is a function that returns the maximum value of its parameters.

1. $ICM(x',y') = Max(ICM(x+\Delta x, y+\Delta y))$ where $x' = x + \Delta x$, $y' = y + \Delta y$, $\Delta x, \Delta y = 1, 0, -1$ and $\Delta x \bullet \Delta y \ne 0$.
2. If $ICM(x,y) < ICM(x',y')$, then let $x = x', y = y'$ and go to step 1, otherwise go to step 3.
3. Pixel (x,y) is considered to be the center of a maximal inscribing circle and the procedure stops.

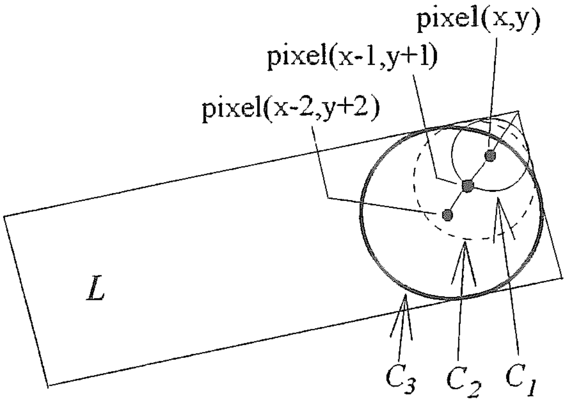

Fig. 2.3 An iterative procedure for finding MIC. The circle with thin border line C_1 represents the ICM of pixel(x,y). The dashed circle C_2 is the ICM of pixel($x-1, y+1$). The circle with bold circumference C_3 is the ICM of pixel ($x-2, y+2$), and corresponds to the MIC found.

Step 1 compares the ICM values of a central pixel $(x,y)'$s 8-neighbors. The coordinates corresponding to the maximum value are stored as (x',y'). In Step 2, The ICM value of the central pixel and the maximum of its 8-neighbor are compared. If the ICM of the central pixel is larger than those of its neighbors, then the central pixel associated with MIC is found. Otherwise, update the position of the central pixel and the whole procedure restarts.

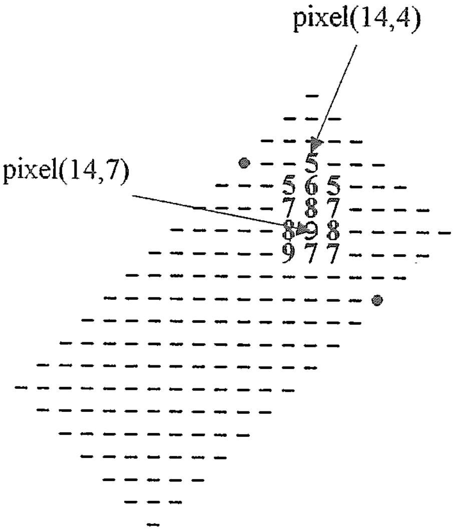

Fig. 2.4 *ICM*(14,4)=5, *ICM*(14,5)=6, *ICM*(14,6)=8 and *ICM*(14,7)=9

Once MIC is found, its diameter is an indication to the line width and the direction perpendicular to it can represent the direction of a line segment. Hereinafter, Fig. 2.1 is used as an illustration to explain our approach. The diameter pair P_1 and P_2 is regarded as the representation of the line width and the slope S_1, perpendicular to $\overline{P_1P_2}$, is an approximation of the line direction. Due to the digital sampling procedure mentioned above, S_1 may be skewed, since x and y are integers, not real numbers. We can correct this by reapply the slope S_2, which is derived from the lines with slope S_1 emanating from the internal pixels across the diameter $\overline{P_1P_2}$ of MIC and intersecting with the border of the line image at points P_3 and P_4. Since P_3 and P_4 are located on the edge of the line image and the direction connecting these two points can be served as a good inplication for the real line slope, slope S_2 now represents the new vectorized line slope. A criterion based on the area covered by the emanating lines within the original line image determines the final slope. A larger area represents a better fit for the raster image of the line segment and the corresponding slope is taken as a closer approximation to the real line slope.

If a single-pixel line is desired, further fitting criterion can be applied based on the line direction found for the best selection. Otherwise, a variable width line representation is implicit in the MIC approach by taking the fitting diameter into consideration. A labeling algorithm is then used to label different line segments and identify the junction areas. Since the line segment represents as an approximation to the raster image obtained, curve portions are characterized by a connected sequence of short straight lines. The length of a line is thus an implication for the curvature of the specific segment. Further merging of these short line segments into a curve representation, e.g. Bézier curve, is followed. Connected short line segments are merged next following the Bézier representation by locating relevant control points. The proposed MIC algorithm for vectorizing binary raster image is outlined below:

1. Use ICM to find a maximal inscribing digital disk that touches the border points through some circle diameter pair, P_1 and P_2.
2. Derive a line slope S_1 perpendicular to diameter $\overline{P_1P_2}$. From the interior points of the diameter, emanating lines with slope S_1. Record the area covered as A_1 and the intersection points with line border as P_j^1, $j = 1 \sim n$. The superscript 1 represents the first iteration and n the number of intersection points.
3. From the points P_j^i, where i is the iteration number and j the intersection points at the ith iteration, find a best-fit line with slope S_i intersecting with line border at points P_j^{i+1}. Record the area covered as A_i. Compare the area A_i with A_{i-1}. If A_i is less than or equal to A_{i-1}, the iteration stops and the slope of the line segment found is S_{i-1}. Otherwise, $i = i+1$, go to step 3.
4. A vectorization of the original line image can now be represented by the interior points of the diameter $\overline{P_1P_2}$ as the starting points and S_{i-1} the slope. Its length can be easily derived. If a line with unit-pixel width is desired, a further line fitting may apply later. Otherwise, the diameter of the MIC represents the line width for the corresponding line segment.
After a new line segment has been identified, the slope is recorded, and a labeling algorithm follows. The conventional labeling algorithm is used to label different line segments, identify the junction areas and construct spatial relationship among line segments and junctions. When recognizing and labeling procedure had finished, the important information, *e.g.* the slope of line segments, line width, junction areas and spatial relationship among line segments and junctions, are derived. If the vectorized image is converted into CAD file format, then the rich vector-editing features can be utilized to do whatever modifications needed. Archival spaces can be saved significantly in comparison with the raster counterpart.
5. Finally, the connected short line segments are identified, and further fitted by following a curve representation by locating corresponding control points within the junction area and endpoints.

In this paper, two short line segments are merged into one Bézier curve by taking the endpoints of each line segment as the control points in the Bézier representation. The intersection point of the connected line segments is counted as two control points instead of one. Fig. 2.5 (a) is composed of two connected short line segments, while Fig. 2.5 (b) is the corresponding Bézier curve by taking the endpoints and point of intersection as the fitting control points.

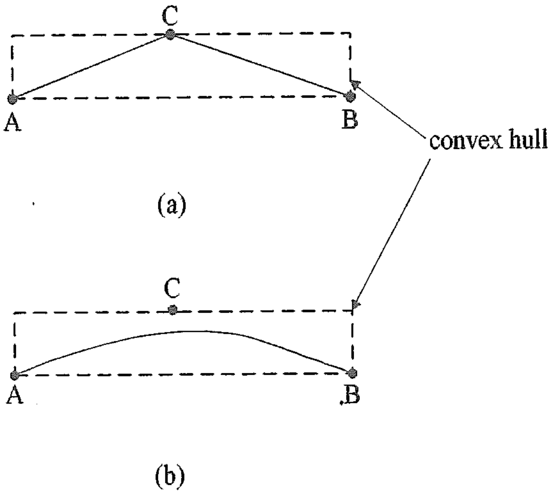

Fig. 2.5 Two short line segments are merged into one Bézier curve.(a) The original short connected line segment obtained from MIC . (b) The corresponding Bézier curve by taking the endpoints A, B and point of intersection C as the fitting control points. The intersection point C is regarded as two control points instead of one.

The Bézier curve representation is derived from the following equation

$$P(u)=\begin{bmatrix}u^3 & u^2 & u & 1\end{bmatrix}\cdot M_{Bez}\cdot\begin{bmatrix}A\\C\\C\\B\end{bmatrix},$$

$$M_{Bez}=\begin{bmatrix}-1 & 3 & -3 & 1\\3 & -6 & 3 & 0\\-3 & 3 & 0 & 0\\1 & 0 & 0 & 0\end{bmatrix},$$

where A,B are coordinates of the endpoints and C corresponds to the central control points.

Even though Bézier curve representation is employed specifically for this implementation, our approach is not limited to Bézier curve only. The Bézier curve is chosen for its simplicity. Once the control points are located, these vital information can be served as the basis for any curve representation desired.

The vectorization of the line image can be characterized by end points, the slope found, and possibly its line width at user's discretion. A line segment containing the maximum number of junction areas is vectorized first. If several line segments having the same number of junction areas, one of them is arbitrarily chosen. For two line segments passing through the same junction area, one junction point within the junction area can be determined. For example, in Fig. 2.6, L_1 through L_6 all contain three junction areas. The end points of the six line segments are ($V_{i,1}$, $V_{i,2}$) respectively, where i=1,2,...,6. However, if L_7 is processed last, it may be divided into three line segments whose end points are ($V_{7,1}$,JP_1), (JP_1, JP_2) and (JP_2, $V_{7,2}$) due to the previously determined intervening junction points JP_1 and JP_2. Therefore, three vectorized line segments, instead of one, may be required to represent L_7 .

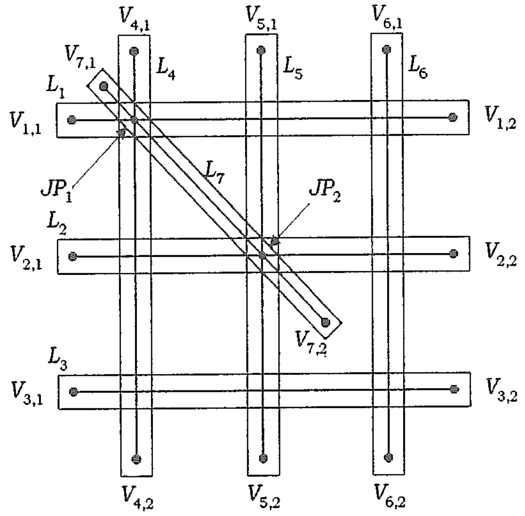

Fig. 2.6 The end points of the L_1, L_2,..., L_6 are ($V_{i,1}$,$V_{i,2}$), where i=1,2,...,6. L_7 is divided into three line segments, if processed last.

An experimental study comparing the performance of MIC approach with that of a conventional thinning method is performed in the next chapter. The results show that our approach is computation efficient, robust, and may render optimal multi-pixel-width vectorized line and curve representation at user's discretion.

3. Experimental Study

In this section, engineering blueprints, freehand drawings and a flowchart diagram are used as test raster images to compare the performance between the proposed MIC algorithm and a chosen conventional thinning method by Chen [5]. The major components present in these test images are lines with different width, along with some curve segments to show the versatility and feasibility of the MIC approach.

Three sets of different image sources are considered. The water faucet design is scanned first at 480 d.p.i resolution as shown in Fig. 3.1 (a). The scanned line image is mainly composed of lines and curves with different width, and contains jagged edges and scattered noises. Fig. 3.1 (b) shows the raster image after thinning operation. The vectorization result employing only straight line segments by using the MIC algorithm is shown in Fig. 3.1 (c). The MIC vectorization results by incorporating Bézier curve representation is displayed in Fig. 3.1 (d). It is obvious the resulting skeleton in Fig. 3.1 (b) is highly susceptible to the influence of noise. Isolated noise points and spurious junction points make further merging of short branches and noise removal heuristically a necessity. Moreover, the original line drawing with varied line width has been uniformly reduced to unit-pixel-width line segment to facilitate the later segmentation and association stages. These latter processes may have to merge short branches, prune junction points, and delete isolated noises due to loss of useful information cues removed by the thinning process. Also, the thinning operation is very time-consuming even for a medium-size hardcopy in comparison with our approach.

Next comes the cramp blueprint, which contains circles with different radius, straight line segment, dash lines, curve segments, numbers and noises. Fig. 3.2 (b) is the result after thinning operation. The line segments have been uniformly peeled down to single-pixel width and noises remain intact. The result of applying MIC method to another cramp design shown in Fig. 3.2(c). Instead of reducing the raster image into a unit-pixel skeleton first, the MIC method takes the ensemble of pixels within the line segments collectively in deciding the final vectorization representation. In contrast with the thinning results in Fig. 3.2(b), the MIC method is not affected by the presence of jagged edges and noises. The feature of the original multi-width line segment is preserved by using our method and no further merging process is needed. Also, the execution speed of the region-based MIC approach is much faster than that of the Chen's pixel-based thinning algorithm. The line segments and junction areas found are then vectorized into CAD file format as shown in Fig. 3.2(d). Comparing Fig 3.2(b) with Fig. 3.2(d), apparent improvements over the number and location of junctions, branches, and noises are reached.

4. Conclusions

Raster to vector conversion technique can be applied to a wide range of applications such as optical character recognition, engineering drawings and architectural blueprints vectorization. A line drawing analysis must segment the line segments from the junction areas, effectively associate the line component through common junction areas and construct their spatial relationships.

In this paper, a maximal inscribing circle concept is used to derive the direction and line width of the line segment. An iterative procedure is developed to identify the slope of each line segments, put in the correct label, and unify the line segments through common junction area. After associating different line segments through common junction areas, a vectorized representation can be obtained. Two short line segments are merged into one Bézier curve by taking the endpoints of each line segment as the control points in the Bézier representation. The intersection point of the connected line segments is counted as two control points instead of one.

The conventional thinning algorithms often suffered the pitfalls of spurious junction points that are crucial features to derive. The liability of the shape of the skeleton resulting from multiple fork points connected through several short branches will impede the following recognition stage. Furthermore, different thinning algorithms will lead to different results and are very sensitive to the presence of noise. In contract with our approach, the thinning methods strip down the useful pixels to obtain one-pixel-width skeleton. This width reduction procedure may result in a loss of information and impede the correctness of the following recognition stage. In our approach, no further processing procedure is required to prune the short branches and merge spurious junction points.

References

[1] B. K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Trans. Syst. Man Cybern.*, vol. 12, no. 6, pp. 541-551, 1990.

[2] R. C. Gonzalez and R. E. Woods, "Boundary descrpitors," in *Digital Image Processing*. Reading, MA:Addison-Wesley, 1992, pp. 494-503.

[3] R. C. Gonzalez and R. E. Woods, "Morphology," in *Digital Image Processing*. Reading, MA:Addison-Wesley, 1992, pp. 518-560.

[4] V. Nagasamy and N. A. Langrana, "Engineering drawing processing and vectorization system," *Comput. Graphics Image Processing*, vol. 49, pp. 379-397, 1990.

[5] Y. S. Chen and W. H. Hsu," A modified fast parallel algorithm for thinning digital patterns," *Pattern Recogn. Lett.*, vol. 7, no. 2, pp. 99-106, 1988.

[6] H. Freeman. "On the encoding of arbitrary geometric configurations," *IEEE Trans. Electron. Computers*, vol. EC-10, pp. 260-268, 1961.

[7] D. P. Fairney and P. T. Fairney, "On the accuracy of point curvature estimators in a discrete environment," *Image Vision Comput.*, vol. 12, no. 5, pp. 259-265, 1994.

[8] M. Tsai and M. F. Chen, "Curve fitting approach for tangent angle and curvature measurments," *Pattern Recogn.*, vol. 27, no. 2, pp. 261-275. 1994.

[9] Y. S. Chen, "Segmentation and association among lines and junctions for a line image," *Pattern Recogn.*, vol. 27, no. 9, pp. 1135-1156, 1994.

[10] C. C. Han and K. C. Fan, "Skeleton generation of engineering drawings via contour matching," *Pattern Recogn.*, vol. 27, no. 2, pp. 261-275, 1994.

[11] Shih-Chun Tu and Yi-Wu Chiang, "MIC (Maximal Inscribing circle)-Based line-fitting Method," *Proc. 1995 Workshop on Computer Applications*, pp. 65-69, Taiwan, April 1995.

[12] S. J. Tue and John Y. Chiang, "The Archiving of line Image Drawing Images," *Proceedings of SPIE's Photonics East'95 Symposium-Conf. On Digital Image Storage and Archiving systems*, Philadelphia, PA, USA, Oct. 1995.

[13] John Y. Chiang, "A New Approach For Binary Line Drawing Images," *Proc. 1995 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1489-1494, Vancouver, BC, Canada, Oct. 1995.

[14] C. Arcelli and G. Sanniti di Baja, "On the sequential approach to medial line transformation," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-8, no. 2, pp. 139-144, 1978.

[15] C. Arcelli and G. Sanniti di Baja, "A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 4, pp. 411-414, 1989.

[16] C. Arcelli, L. P. Cordella, L. P. and S. Levialdi, "From local maxima to connected skeletons," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-3, no. 2, pp. 134-143, 1981.

[17] A. R. Dill, M. D. Levine and P. B. Nobel, "Multiple resolution skeletons," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-9, no. 4, pp. 495-504, 1987.

[18] L. Lam, S. W. Lee and C. Y. Suen, "Thinning methodologies - A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869-885, 1992.

[19] Q. Z. Ye and P. E. Danielsson, "Inspection of printed circuit boards by connectivity preserving shrinking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 737-742, 1988.

[20] Y. Xia, "Skeletonization via realization of the fire front's propagation and extinction in digital binary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 1076-1086, 1989.

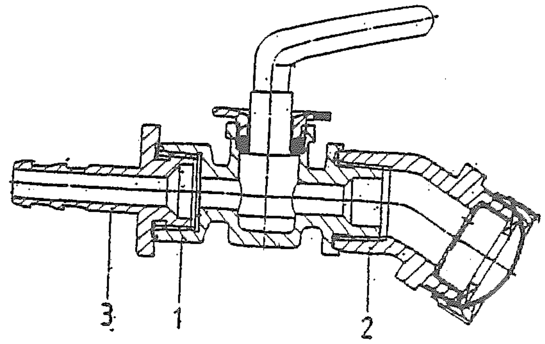

Fig. 3.1(a) Scanned raster image(Original size 18cm×12cm).

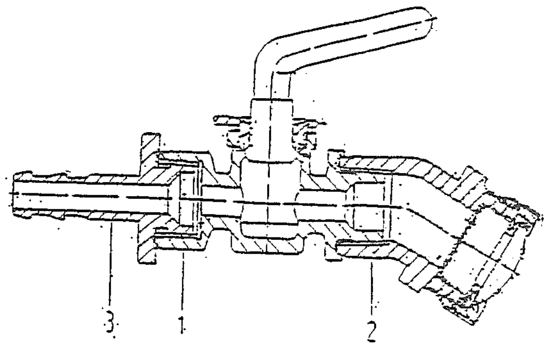

Fig. 3.1(b) Raster image after thinning operation.

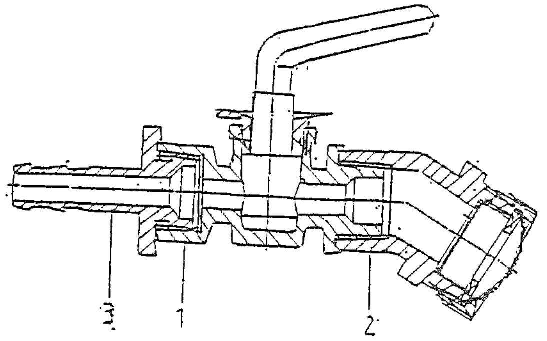

Fig. 3.1(c) Vectorized graph employing line fitting only.

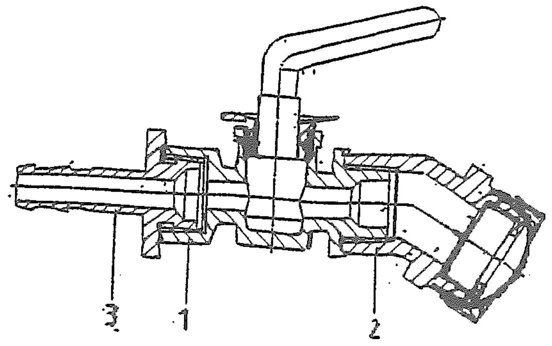

Fig. 3.1(d) Vectorized graph incorporated both line and curve fitting.

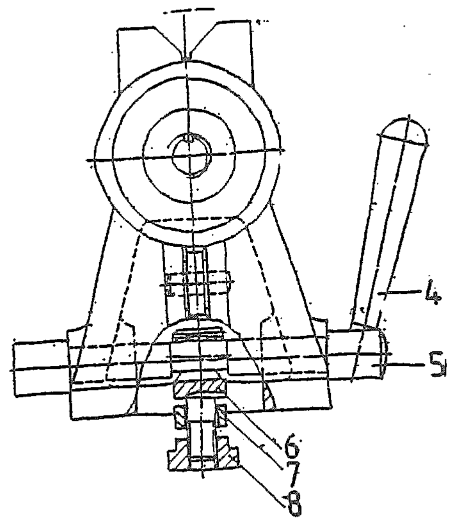

Fig. 3.2(a) Scanned raster image(Original size 12cm×13cm) .

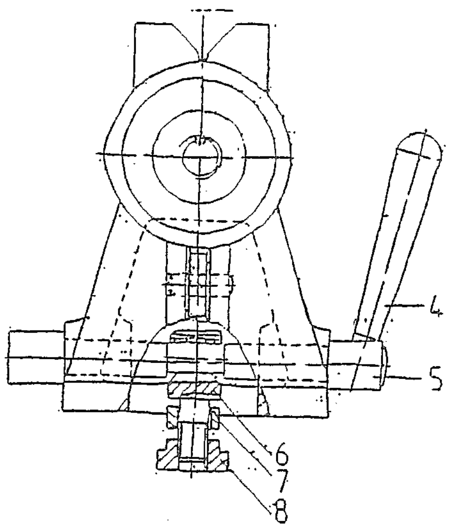

Fig. 3.2(b) Raster image after thinning operation.

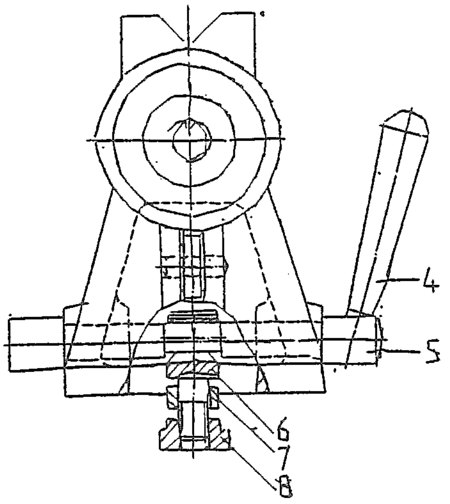

Fig. 3.2(c) Vectorized graph employing line fitting only.

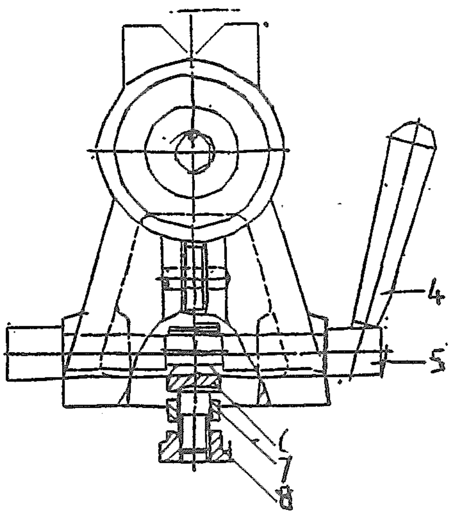

Fig. 3.2(d) Vectorized graph incorporates line and curve fitting.