Machine Vision and Applications (1997) 9: 240–250

Machine Vision and Applications

A protocol for performance evaluation of line detection algorithms

Liu Wenyin, Dov Dori

Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Haifa, 32000, Israel
e-mail: {liuwy,dori}@ie.technion.ac.il

February 27, 1997

Abstract. Accurate and efficient vectorization of line drawings is essential for any higher level processing in document analysis and recognition systems. In spite of the prevalence of vectorization and line detection methods, no standard for their performance evaluation protocol exists. We propose a protocol for evaluating both straight and circular line extraction to help compare, select, improve, and even design line detection algorithms to be incorporated into line drawing recognition and understanding systems. The protocol involves both positive and negative sets of indices, at pixel and vector levels. Time efficiency is also included in the protocol. The protocol may be extended to handle lines of any shape as well as other classes of graphic objects.

Key words: Performance evaluation – Vector detection quality – Vectorization - Line detection – Graphics recognition - Sparse pixel vectorization algorithm – Machine drawing understanding system

1 Introduction

Vectorization and other line detection techniques have been developed to convert images of line drawings in various domains from pixels to vector form (e.g., Kasturi et al. 1990; Nagasamy and Langrana 1990; Filipski and Flandrena 1992) and a number of methods and systems have been proposed and implemented (e.g., Boatto et al. 1992; Vaxiviere and Tombre 1992; Dori et al. 1993; Dori 1995). However, the performance of these methods and systems is known at best only from the reports of their developers, based on their own perceptual, subjective, and qualitative human vision evaluation. Objective evaluations and quantitative comparisons among the different line detection algorithms are not available. This is due to the lack of protocols that provide for quantitative measurements of their interesting metrics, a sound methodology acquiring appropriate ground truth data, and adequate methods for matching the ground truth data with the data describing the detected lines. To fully comprehend and reliably compare the performance of line detection method and to help select, improve, and even design new methods to be applied in further systems intended for some specific application strongly requires the establishment of objective evaluation protocols, matching methods, and a resulting performance evaluation methodology.

Performance evaluation is a very new research topic in the field of computer vision and image processing in general. Performance evaluation has been recognized as an important factor in advancing the research in this field. Most work to date has been carried out to evaluate performance of thinning algorithms. Haralick (1992) was the first to propose a general approach for performance evaluation of image analysis, with thinning taken as a case in point. Evaluation and comparison of thinning algorithms have also been performed by Lee et al. (1991), Lam and Suen (1993), Jaisimha et al. (1993), and Cordella and Marcelli (1996). Some of these evaluation and comparison works were carried out from the viewpoint of OCR, while the work of Jaisimha et al. (1993) is domain independent. Although thinning may also be employed as preprocessing of line detection, the latter has different characteristics and therefore requires different evaluation protocol. However, performance evaluation of vectorization and line detection has been reported only recently by Kong et al. (1996) and Hori and Doermann (1996).

Kong et al. (1996) have developed a protocol and a system for systematically evaluating the performance of line detection algorithms, mainly for dashed-line detection algorithms. They define the overlap criteria of the match between a ground truth and a detected linc based on the angle and the distance between them, and the partial overlap is also considered. The protocol includes several important evaluation aspects, such as endpoint detection and line style (pattern) detection. Although the evaluation aspects are selected primarily for dashed lines, they are suitable for any line style (pattern). Both the detection rate and the false alarm rate (misdetection rate) are used in the evaluation, yielding positive and negative evaluation, respectively. The geometry matching criteria are rather arbitrary and rigid, for example, the angle should be less than 3° and the distance between two lines less than 5 pixels. The protocol also excludes the evaluation of line width detection as well as detected line fragmentation and combination. Rather, they consider line

Correspondence to: D. Dori

fragmentation and combination as being simply wrong, and only the best match with the maximum overlap is chosen.

Hori and Doermann (1996) present a methodology for measuring the performance of algorithms for application-specific raster-to-vector conversion (also referred to as vectorization). They provide a set of basic metrics (evaluation contents) which is specifically geared to vectorization of mechanical engineering drawings but may be used in all applications. Using CAD representation, they specify the output (line) feature set to include the endpoints, the line thickness (width), and line type (pattern, which may be dashed, dotted, or solid) as well as a number of further feature points such as T junctions, crossing points, and corner points. Their point matching threshold is set as half the line width plus an allowable margin error. This threshold is adaptive and suits any line width. Another rational metric is the weighted total line matching value, which is the weighted sum of each line's matching value (1 for match, 0 for mismatch) with the line lengths being the weights. Hori and Doermann's performance evaluation protocol has also a number of disadvantages. First, it makes no distinction between detection rate and false alarm rate. Second, the metrics for line evaluation are given in several nonuniform units. It uses length ratio, deviation, and count ratio to evaluate the line length detection, line location detection, and line quantity detection, respectively. The evaluation of feature points is also given in four distinct forms: endpoints, T junctions, crossing points, and corner points. There is lack of an overall evaluation metric which provides an overall combined performance evaluation of the algorithm under consideration. Third, the location deviation metrics are not normalized with respect to the line widths; rather, the protocol uses absolute Euclidean distances. Finally, as in Kong et al.'s protocol, this protocol also neglects the evaluation of line width detection, detection fragmentation, and detection combination.

The problem of evaluating arcs and higher order curves has not been addressed. Kadonaga and Abe (1996) compare the methods for detecting corner points from digital curves. This may be related to curve detection, but their scheme is aimed at evaluating the detection of feature points rather than curves. The performance evaluation protocol of the detection of arcs and higher order curves should include other critical factors.

Time efficiency is an important factor of the performance of the vectorization and line detection algorithms, especially in industrial applications, where large drawing images must be processed in reasonable time to make the system practical. However, time performance is seldom evaluated. Only Lee at el. (1991) have compared the time efficiency of thinning algorithms.

We propose here an objective and comprehensive protocol for performance evaluation of vectorization and line detection algorithms which includes a reasonable line matching definition and its measurement degree (vector detection quality), a number of evaluation criteria, and a set of single and combined performance evaluation indices at both the pixel and vector levels. Both the detection and false alarm rates are considered. It is designed for straight lines as well as circular arcs of any style. Time efficiency is also included in this protocol. Experiments using this protocol in the performance evaluation of our sparse pixel vectorization algorithm, stepwise recovery arc segmentation algorithm, and dashed line detection algorithm with the machine drawing understanding system (MDUS) are presented. We compare and discuss the evaluation results between this protocol and those of Kong et al. (1996) and Hori and Doermann (1996). The proposed protocol may also be extended to cover the performance evaluation of text segmentation and other graphic object recognition algorithms.

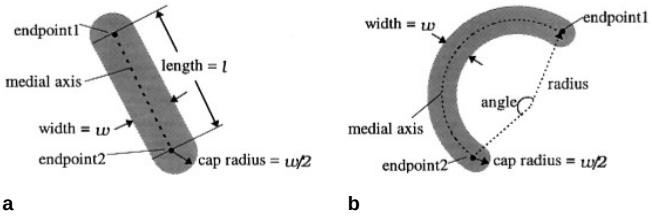


Fig. 1. a The ideal oval area occupied by a bar. **b** The ideal arc image

In Sect. 2 we first present and discuss the objectives, characteristics, and expected outputs of vectorization and line detection algorithms. The performance evaluation contents is determined and defined accordingly. Section 3 presents our matching methods for a pair of ground truth line and its corresponding detected line. In Sect. 4 we propose the general protocol for vectorization performance evaluation, which combines a set of indices for both detection and false alarms at the pixel and vector levels. Section 5 describes the experiments using the protocol in the performance evaluation of our sparse pixel vectorization algorithm, stepwise recovery arc segmentation algorithm, and the dashed line detection algorithm in the MDUS that we developed, and compares the experimental results of the proposed protocol with those of protocols proposed by Kong et al. (1996) and Hori and Doermann (1996). Discussion and summary appear in Sect. 6.

2 Performance contents of vectorization and line detection

Vectorization and line detection processes yield vector form lines from the pixel-based drawing image. We refer to vectorization specifically as the process which is responsible only for converting the image to raw fragment vectors (bars and polylines) and to line detection as the process which yields fine lines (with specific and accurate shapes and styles) from the raw fragmentary vectors.

A vector represents a bar which has, in addition to the two endpoints, a specified line width, an associated line style, and the round cap end style. Figure 1 shows examples of the area occupied by the image of a solid straight line segment and a solid circular arc. We define the following terms that are used throughout the paper:

- Bar: a solid straight line segment with nonzero width which occupies a (possibly slanted) black pixel area in the drawing image. The ideal area is an oval consisting of a rectangle and two semicircles, as shown in Fig. 1a, as if displayed using the round cap endpoint style.
- Arc: a solid circular line segment with nonzero width which occupies a (possibly closed) black pixel area in

the drawing. The ideal endpoints are also round capped, as shown in Fig. 1b.
- Polyline: a chain of two or more equal width bars linked end to end.
- Line: a generalization of bar, arc, and polyline to refer any vector, whose style may be solid, dashed, dash-dotted, or dash-dot-dotted, among others.
- Line detection algorithm: an algorithm that operates on raster image to detect lines. It is a generalization of vectorization which yields bars and polylines, arc segmentation, which yields arcs, and detection of other classes of lines.
- Basic line detection phase: the first phase of a line detection algorithm.
- Postprocessing phase: the second phase of a line detection algorithm, in which line fragments are manipulated to refine the output.

We expect the vectorized results to preserve to the greatest extent possible the original shape of all the lines in the image. To avoid cumulative errors and evaluate each major phase separately, the shape preservation is evaluated with respect to the results of basic line detection phase rather than on the final output after postprocessing phase.

We expected line detection algorithms to yield fine lines that are as accurate as possible. The attributes of a line in the drawing include style, width, endpoints and, for an arc, also center. The detection accuracy of each line is reflected by the detection quality of its attributes. The evaluation should consider the detected values of these attributes. Two additional factors affect the line detection accuracy. One is the detection fragmentation, i.e., the extent to which a ground truth line is detected as several line fragments. Fragmentation occurs not only in the detection of dashed style lines but also in the detection of solid lines. The other factor is the detection consolidation, i.e., the extent to which two or more ground truth lines are linked and detected as one line. This case occurs in situations where the endpoints of two collinear lines of (usually slightly) different widths are too close to be separated by the algorithm, or where two parallel lines are too close to each other. These two factors should also be included in the evaluation. *Vector detection quality* is a metric that incorporates the detection of the line's basic attributes with the detection fragmentation and detection consolidation factors.

Lines rather than feature points are used as the evaluated entities in our protocol. Feature points detection may be important in specific tasks, such as OCR. However, feature points alone do not directly reflect the detection capability of line detection algorithms. At any rate, important feature points are included directly in the line detection evaluation protocol since they are also attributes of the lines. For example, the feature points specified by Hori and Doermann (1996) are the endpoints, corners and T junctions. Corners and T junctions are also endpoints of lines.

3 Matching ground truth to detected output

Ground truth is the original data, which is the basis for comparison with the detection results. This is the essence of performance evaluation. For the pixel-level shape-preservation evaluation the ground truth is the original raster image on which the detection algorithms operate. To evaluate the processing capability (robustness) of image analysis algorithms on real-life, noisy images, image degradation models have been proposed by Kanungo et al. (1993) and used by Haralick (1992) and Hori and Doermann (1996). The ground truth image is the original clear image, and the actual input image is the degraded image.

For vector-level evaluation the ground truth is a set of lines whose attributes are known in advance and are more difficult to obtain. For real applications vector ground truth is usually not available. For experimental evaluations vector ground truth can be obtained by either manually measuring the image or automatically generating and using them to construct (with the optional degradation) the input image for the evaluation test.

The matching problem at the pixel level can be solved simply by considering the pixel coordinates since the ground truth pixels and the detected pixels are stored in the identical data structures.

The matching problem at the vector level is more complicated because a detected vector may have location offset, a different line style, and even a different geometry shape than its ground truth. Moreover, there may be no counterpart to either a ground truth or a detected line at all. The matching definition of Kong et al. (1996) is based on the angle, distance, and relative overlap between the two lines. This may yield a false matching result. For example, consider two equal, very long straight lines that intersect at their middle points at a 3° angle. The distances between the two pairs of corresponding endpoints of such two lines may be large. Nevertheless, these two lines are erroneously matched using the matching definition. Likewise, a short line cannot match a long line according to the relative overlap criterion. Hori and Doermann (1996) use endpoint matching in their line matching definition. This criterion is suitable only for straight lines and not for circular arcs and curves.

Our protocol defines the matching of a pair of lines based on both the area overlap and the endpoint matching between them. The area occupied by a bar is determined by its endpoints and width; therefore the area overlap between two bars can be calculated by their endpoints and width alone. For curves we require that the matched lines both overlap and have matching endpoints. We do not limit the matching definition to lines of the same geometric shape. Lines with different shapes may also be matched, and the matching degree (vector detection quality) is calculated with shape misdetection evaluation included. This option is needed because, for example, short arcs with a small open angle may be reasonably misdetected as bars.

Based on the above matching definition, we can calculate the overlapping segment of a detected line and its corresponding ground truth line. Figures. 2 and 3 illustrate the matching definition and the calculation of the overlapping segment.

For bars, consider a pair of a detected line (k) and a corresponding ground truth line (g). Let $c = k \cap g$ denote the line segment of k that overlaps g, i.e., c is the intersection of k and g. The method of calculating c is presented and illustrated in Fig. 2. A ground truth line and a detected line are defined to be overlapping if each one of at least two out

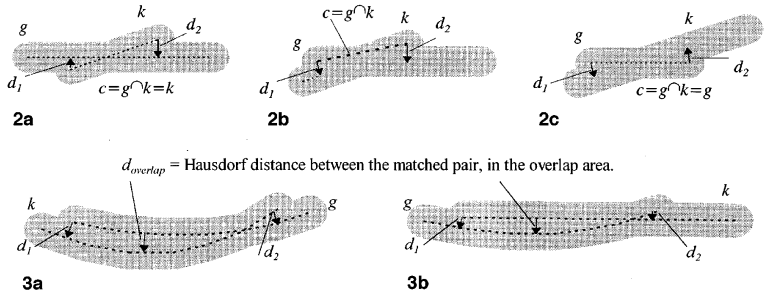


Fig. 2. Illustration of calculating the intersection part of two overlapping lines
Fig. 3a,b. Illustration of the matching of arcs. **a** Two arcs. **b** An arc and a bar

of the four endpoints of the two lines lies inside the image area of the other line, as shown in Fig. 2. A point lying inside the occupied area of a line also means that the distance of the point to the line is less than half the line width. This definition gives rise to the following three cases:

- Ground truth line overlap: both endpoints of the ground truth line lie inside the detected line's area.
- Detected line overlap: both endpoints of the detected line lie inside the ground truth line's area.
- Partial overlapping: exactly one endpoint of the ground truth line lies inside the detected line's area and exactly one endpoint of the detected line lies inside the ground truth line's area.

The overlapping segment is a virtual line whose endpoints are the two endpoints lying inside the area of the matching line, which are called touching points. The overlapping segment is evaluated, and the evaluation is part of the ground truth line and the detected line. The distances from these two touching points to the matching line are denoted d_1 and d_2.

The matching of arcs (two arcs, or an arc and a bar) is similar to that of straight lines, except that the overlap distance ($d_{overlap}$) is added to the criteria. The overlap distance, as shown in Fig. 3, is defined as the Hausdorf distance: the maximum of all minimum distances between the points on the detected line (arc) and the ground truth line (arc) in the overlap area, including d_1 and d_2. Such matching requires that the overlap distance be less than half the ground truth line width. This criterion can be extended to the matching of polylines, which may be either the ground truth line or the detected line, or both.

4 Indices of performance evaluation

Customary current performance evaluation methods of computer vision, image processing, document image analysis, and recognition systems are based on perceptive measuring, i.e., relying on human vision. The results are subjective and qualitative. Different people may give different evaluations. In order to supply objective and quantitative measurements which are comparable, quantitative indices should be provided along with the evaluation protocol.

According to the performance evaluation criteria defined in Sect. 2, we define performance evaluation indices to measure each criterion. The shape comparison of complex graphic objects is easier in pixel form than in vector form. Therefore the shape-preservation capability is a pixel-level performance attribute of line detection algorithms. The vector attributes (such as style, width, and endpoints) are easier to compare at vector level. The line detection accuracy (vector detection quality) therefore lies in the accuracy of vector-level attributes.

The performance of detection and recognition algorithms is usually reflected by two rates: true positive and false positive (Nalwa 1993). The true positive rate is the rate of positive responses in the presence of instances of the feature, and in our case the detection (recognition) rate, which is the ratio of the number of correctly detected lines to the total number of ground truth lines. The false positive rate is the rate of positive responses in the absence of the feature, and in our case is the false alarm (misdetection) rate, which is the ratio of the number of those detected lines that have no matched ground truth lines to the total number of detected elements. These two rates are used together because we wish to increase the detection rate while decreasing the false alarm rate. The detection rate increase is often accompanied by the increase in the false alarm rate while adjusting the parameters within an algorithm. A tradeoff must be made between them. Therefore some single index combining the detection rate and the false alarm rate may be useful as an overall and parameter that is independent performance indicator.

Our protocol combines the measurement of two performance attributes of line detection algorithms. One is rate and the other is level. The two values of the rate attribute are detection rate (D) and false alarm rate (F). The two level attribute values are pixel (p) and vector (v). The cartesian product of the two rate values with the two level values yields four performance indices, which are listed in Table 1 and defined below.

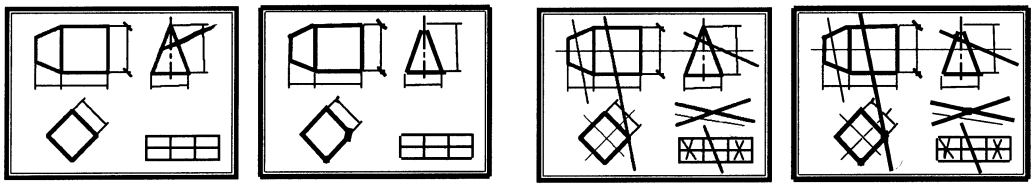
4.1 Pixel level performance indices

Pixel detection rate

Let P_g be the set of all the black pixels in the ground truth image, i.e., the original image to be vectorized; let $P_g(k)$ be the set of pixels belonging to the ground truth line k; and let P_d be the set of all the black pixels detected by the segmentation. The detection rate of the ground truth line k, $d(k)$, is defined in Eq. 1:

$$d(k) = \frac{|P_g(k) \cap P_d|}{|P_g(k)|} \tag{1}$$

Obviously, $0 \leq d(k) \leq 1$, because the nominator on the right side of Eq. 1 ranges from 0 to $|P_g(k)|$. The total pixel detection rate, D_p, is a weighted sum of $d(k)$:



4a **4b** **5a** **5b**

Fig. 4a,b. Synthetic drawing A (400 × 300 pixels. **a** Synthetic image. **b** Vectors

Fig. 5a,b. Synthetic drawing B (400 × 300 pixels. **a** Synthetic image. **b** Vectors

Table 1. The vectorization evaluation processes and the resulting performance indices

Evaluation subprocess	Level	Detection rate (D)	False alarm rate (F)	Perfomance index
Shape preservation evaluation	Pixel (p)	Pixel detection rate (D_p)	Pixel false alarm rate (F_p)	Pixel recovery index (PRI)
Vector detection quality evaluation	Vector (v)	Vector detection rate (D_v)	Vector false alarm rate (F_v)	Vector recovery index (VRI)

$$D_p = \sum_{k \in V_g} d(k) w_g(k) \tag{2}$$

where V_g is the set of lines in the ground truth image, and $w_g(k)$ is the weight of a ground truth line k, which is relative to its size (number of pixels). If no ground truth line in the drawing overlaps any other ground truth line, or if such overlap can be neglected, the weight of a ground truth line k can be defined as in Eq. 3:

$$w_g(k) = \frac{|P_g(k)|}{|P_g|} \tag{3}$$

Substituting Eqs. 1 and 3 into Eq. 2 yields:

$$D_p = \frac{1}{|P_g|} \sum_{k \in V_g} |P_g(k) P_d| \tag{4}$$

Note that the sum of $w_g(k)$ over all k in V_g is usually larger than 1, because the lines in an image often intersect each other, and their images therefore partially overlap, and the overlapping areas are counted more than once. To overcome this problem the total pixel detection rate D_p is calculated directly from the whole set of detected pixels rather than considering the individual ground truth lines, as follows:

$$D_p = \frac{|P_g \cap P_d|}{|P_g|} \tag{5}$$

Pixel false alarm rate

The pixel false alarm rate of the detected line k is defined as:

$$f(k) = \frac{|P_d(k) - P_d(k) \cap P_g|}{|P_d(k)|} \tag{6}$$

where $P_d(k)$ is the set of pixels of the detected line k. Similarly, the total pixel false alarm rate is a weighted sum of $f(k)$:

$$F_p = \sum_{k \in V_d} f(k) w_d(k) \tag{7}$$

where, as with $w_g(k), w_d(k)$ is the weight of the detected line k. If the detected lines do not overlap each other in the image, then:

$$w_d(k) = \frac{|P_d(k)|}{|P_d|} \tag{8}$$

and V_d is the set of lines detected from the ground truth image. Substituting Eqs. 6 and 8 into Eq. 7 yields:

$$F_p = \frac{1}{|P_d|} \sum_{k \in V_d} |P_d(k) - P_d(k) \cap P_g| \tag{9}$$

Again, since lines usually overlap in part, the total pixel false alarm rate is calculated directly from the whole set of detected pixels rather than through individual detected lines:

$$F_p = 1 - \frac{|P_g \cap P_d|}{|P_d|} \tag{10}$$

Using the pixel detection rate and pixel false alarm rate, the combined pixel recovery index is defined as:

$$PRI = \alpha D_p + (1 - \alpha)(1 - F_p) \tag{11}$$

where $0 \leq \alpha \leq 1$ is the relative importance of the detection and $1 - \alpha$ the relative importance of the false alarm.

4.2 Vector level performance indices

Vector detection quality measures the matching accuracy between a ground truth line and the corresponding detected line in terms of the line's attributes. We first define vector detection quality for the overlapping segment of a detected line and its corresponding ground truth line. The vector detection rate and the vector false alarm rate are on this basis.

Consider a matched pair of a detected line k and a corresponding ground truth line g, and their overlapping segment c, which are illustrated in Figs. 2 and 3. The distances from the two touching endpoints to the matching line, $d_1(c)$ and $d_2(c)$, and the overlap distance $d_{overlap}$ are measured relative to the ground truth line width. For ground truth lines whose width is an even number of pixels, $d_i(c)$ $(i = 1, 2)$ is taken as zero when their calculated value is 1 pixel, because

either one of the two pixels in the middle of the width can be considered correct.

The vector detection quality $[Q_v(c)]$ of the overlapping segment c is the weighted product of the following five quality factors: endpoints quality, $Q_{pt}(c)$; overlap distance quality $Q_{od}(c)$; line width quality, $Q_w(c)$; line style quality, $Q_{st}(c)$; and line shape quality, $Q_{sh}(c)$. These are defined below. For the sake of simplicity the weights in the product are taken to be equal. Thus the total vector detection quality is the geometric mean of the five quality factors:

$$Q_{pt}(c) = e^{-\frac{d_1(c)+d_2(c)}{W(g)}} \tag{12}$$

$$Q_{od}(c) = e^{-\frac{2d_{overlap}(c)}{W(g)}} \tag{13}$$

$$Q_w(c) = e^{-\frac{|W(k)-W(g)|}{W(g)}} \tag{14}$$

$$Q_{st}(c) = e^{-|Style(k)-Style(g)|} \tag{15}$$

$$Q_{sh}(c) = e^{-|Shape(k)-Shape(g)|} \tag{16}$$

$$Q_v(c) = [Q_{pt}(c)Q_{od}(c)Q_w(c)Q_{st}(c)Q_{sh}(c)]^{1/5} \tag{17}$$

where $d_i(c)$ $(i = 1, 2)$ are the distances from the two endpoints of segment c to g; $d_{overlap}$ is the overlap distance, as shown in Figs. 2 and 3. They reflect the discrepancy between the locations of the ground truth line and the detected line. $W(g)$ is the line width of g, and $W(k)$ is the line width of k. The reason for dividing the distances by the width of the ground truth line in Eq. 12 and 13 is that the accuracy of location detection is related to the line width: a thick line allows greater location detection error in pixel units than a thin line. The width quality factor in Eq. 14 is also measured by the width difference relative to the width of the ground truth. $Style(l)$ is the value of the line style of l, which is assigned the value of 1 for solid, 2 for dashed, 3 for dash-dotted, and 4 for dash-dot-dotted. $Shape(l)$ is the value of the line shape of l, which is set to 1 for straight, 2 for circular, and 3 for polyline. The values of $Style(l)$ and $Shape(l)$ may also be extended to include other styles and shapes in the future.

For arcs the detection accuracy of the center and the radius may also be included; since these attributes are related to overlap distance quality, it is not included in our protocol.

Equations 12–17 can be used for any pair of detected line and ground truth line because the vector detection quality $Q_v(c)$ quickly vanishes as the location offset between the two lines increases, or the differences of other attributes increase. Hence $Q_v(c)$ reflects the extent to which a detected line matches a ground truth. However, for this to be useful one must know the ground truth vector data, which are normally not available for hand-made drawings unless they are measured manually. For synthetic or CAD-produced drawings the line attributes can be known in advance.

Vector detection rate

Since a single ground truth line may be detected as several lines, its vector detection quality is defined in terms of two elements. The first element is the line's basic quality $[Q_b(g)]$, which is the length-weighted sum of the vector detection qualities of the overlapping segments between the ground truth line and each of its detected lines. The second element is the fragmentation quality $[Q_{fr}(g)]$, which indicates how fragmented the detected line is with respect to the ground truth line. Denote by $D(g)$ the set of the detected lines that (fully or partially) overlap ground truth line g. Let k be an element of $D(g)$ and let $l(v)$ be the length of any line v. The basic quality of ground truth line g is defined as:

$$Q_b(g) = \frac{\sum_{k\in D(g)}(Q_v(k\cap g)l(k\cap g))}{\max(l(g), \sum_{k\in D(g)} l(k\cap g))} \tag{18}$$

where $(k \cap g)$ is the overlapping segment of k and g, and $l(k \cap g)$ is its length. $Q_{fr}(g)$ is defined in Eq. 19 as the average of the squared overlapping segment lengths, such that the more equally broken the segments are, the smaller is the fragmentation quality:

$$Q_{fr}(g) = \frac{\sqrt{\sum_{k\in D(g)} l(k\cap g)^2}}{\sum_{k\in D(g)} l(k\cap g)} \tag{19}$$

The total vector detection quality of g is defined as:

$$Q_v(g) = Q_b(g)Q_{fr}(g) \tag{20}$$

Note that if $|D(g)| = 1$ and $k \cap g = g$ then $Q_{fr}(g) = 1$, and $Q_v(g) = Q_b(g) = Q_v(k \cap g)$.

The image vector detection rate is the length-weighted sum of the vector detection qualities of all ground truth lines in the entire image:

$$D_v = \frac{\sum_{g\in V_g} Q_v(g)l(g)}{\sum_{k\in V_d} l(g)} \tag{21}$$

Vector false alarm rate

The vector detection quality of a detected line is defined in a way similar to the way in which $Q_v(g)$ is defined. Several ground truth lines may overlap k, which means that k erroneously links two or more distinct ground truth lines. To account for this type of false linking we use the fragmentation quality measure as follows. Suppose $G(k)$ is the set of the ground truth lines that (fully or partially) overlap k, then:

$$Q_{fr}(k) = \frac{\sqrt{\sum_{g\in G(k)} l(k\cap g)^2}}{\sum_{g\in G(k)} l(k\cap g)} \tag{22}$$

As before, $k \cap g$ is the overlapping vector segment of k and g, and $l(k \cap g)$ is its length.

The basic quality of k, $Q_b(k)$, is the length-weighted sum of the vector detection qualities of the overlapping segments between detected line k and each of its ground truth lines:

$$Q_b(k) = \frac{\sum_{g\in G(k)}(Q_v(k\cap g)l(k\cap g))}{\max(l(k), \sum_{g\in G(k)} l(k\cap g))} \tag{23}$$

The total vector detection quality of detected line k is defined as:

$$Q_v(k) = Q_b(k)Q_{fr}(k) \tag{24}$$

The false alarm factor $F_v(k)$ of the detected line k is defined in terms of its vector detection quality in Eq. 25:

Table 2. Examples of performance evaluation of a single bar detection

Detection case	Good	Short	Long	Skew	Narrow	Style error	Errors	Fragmentary	Merged
Ground truth image									
Detection discrepancy	(10,21)-(90,21)	(30,20)-(70,20)	(5,20)-(95,20)	(10,17)-(90,23)	Width=6	Dashed	(30,17)-(95,23) w=6	(10,20)-(35,20), (50,20)-(90,20)	(10,20), (90,20)
Gray display of the detection									
D_p—Eq. (5)	0.87	0.53	1	0.80	0.74	1	0.51	0.89	1
F_p—Eq. (10)	0.13	0	0.10	0.21	0	0	0.16	0	0.11
PRI—Eq. (11)	0.87	0.77	0.95	0.79	0.87	1	0.68	0.95	0.95
$Q_v(c)$—Eq. (17)	1	1	1	0.74	0.95	0.82	0.70	1 1	1 1
$Q_b(g)$—Eq. (18)	1	0.5	1	0.74	0.95	0.82	0.53	0.81	1 1
$Q_{fr}(g)$— Eq.(19)	1	1	1	1	1	1	1	0.73	1 1
$Q_v(g)$—Eq. (20)	1	0.5	1	0.74	0.95	0.82	0.53	0.59	1 1
D_v—Eq. (21)	1	0.5	1	0.74	0.95	0.82	0.53	0.59	1
F_v—Eq. (26)	0	0	0.11	0.26	0.05	0.18	0.35	0	0.41
VRI—Eq. (27)	1	0.75	0.94	0.74	0.95	0.82	0.59	0.80	0.71
CDI—Eq. (28)	0.94	0.76	0.95	0.77	0.91	0.91	0.64	0.88	0.83
Kong: match	1	0	1	0	1	1	0	0	0
Hori: count ratio	1	1	1	1	1	1	1	2	0.5
Hori: length ratio	1	0.5	1.13	1	1	1	0.81	0.81	1.23
Hori: deviation	1	0	0	6	0	0	6	0	0

Table 3. Examples of performance evaluation of a single arc detection

Detection case	Center offset	Center offset	Radius error	Good	Center shift	Small	Big	Shape error
Ground truth image								
Detection discrepancy	Full circle c(52,50) r=40	Full circle c(54,50) r=40	Full circle c(51,50) r=39	Arc c(50,51) r=40 (30,16)-(70,16)	Arc c(53,50) r=40 (33,15)-(73,15)	Arc c(50,40) r=32 (34,12)-(66,12)	Arc c(50,58) r=70 (15,19)-(85,19)	Bar (30,12)-(70,12)
Gray display of the detection								
D_p—Eq. (5)	0.84	0.69	0.86	0.88	0.85	0.66	0.89	0.79
F_p—Eq. (10)	0.18	0.31	0.11	0.12	0.15	0.20	0.46	0.18
PRI—Eq. (11)	0.84	0.69	0.88	0.88	0.85	0.73	0.73	0.81
$Q_v(c)$—Eq. (17)	0.82	0.67	0.82	1.0	0.86	0.90	0.90	0.52
$Q_b(g)$—Eq. (18)	0.82	0.67	0.82	0.98	0.78	0.67	0.88	0.47
$Q_{fr}(g)$—Eq.(19)	1	1	1	1	1	1	1	1
$Q_v(g)$—Eq. (20)	0.82	0.67	0.82	0.98	0.78	0.67	0.88	0.47
D_v—Eq. (21)	0.82	0.67	0.82	0.98	0.78	0.67	0.88	0.47
F_v—Eq. (26)	0.18	0.33	0.16	0.02	0.22	0.15	0.49	0.50
VRI—Eq. (27)	0.82	0.67	0.83	0.98	0.78	0.76	0.70	0.48
CDI—Eq. (28)	0.83	0.68	0.86	0.93	0.85	0.75	0.72	0.65

$$F_v(k) = 1 - Q_v(k) \tag{25}$$

This is because $Q_v(k)$ reflects the degree of k being a true line. $1 - Q_v(k)$ therefore reflects the degree of it being a false alarm. An ideal detected line has a vector detection quality value of 1, and therefore its false alarm factor is 0. A bad line has quality value 0, meaning that it is definitely a false alarm.

The image vector false alarm rate is the length-weighted sum of the false alarm factors of all lines detected from the entire image:

$$F_v = \frac{\sum_{k \in V_d} F_v(k) l(k)}{\sum_{k \in V_d} l(k)} \tag{26}$$

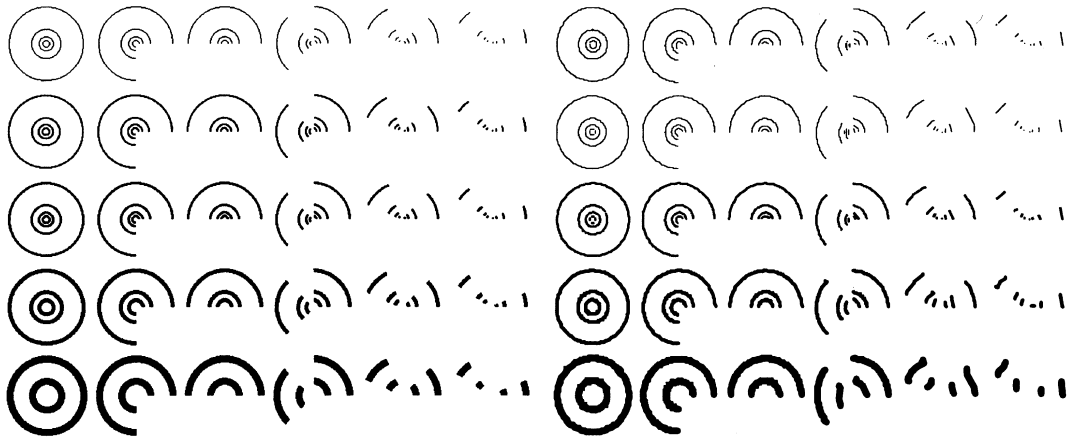
The combined vector recovery index (VRI) is:

$$VRI = \beta D_v + (1 - \beta)(1 - F_v) \tag{27}$$

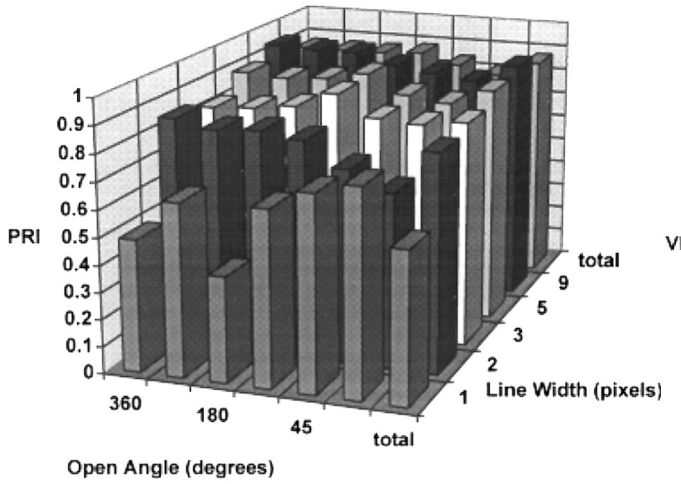
where β is the relative importance of detection, similar to α in Eq. 11. The terms α and β are originally set at 0.5 to assign equal importance to the detection and the false alarm, as used in the experiments in this paper. However, they may be set at other values in the performance evaluation of some task-specific systems. For instance, α and β can be set smaller to give greater importance to the false alarm if the reliability level rather than the level of detection is more critical. With appropriate values of α and β the combined indices give quantitative performance evaluation. The higher the combined indices, the better is the algorithm. In general, the pixel recovery index (PRI) is appropriate for evaluating the shape preservation capability of the basic vectorization, the output of which is raw bars and polylines, while the VRI is suitable for evaluating the final results of line detection algorithms.

Finally, PRI and VRI may be combined into a single measure, the combined detection index (CDI), where γ is originally set at 0.5 in this protocol:

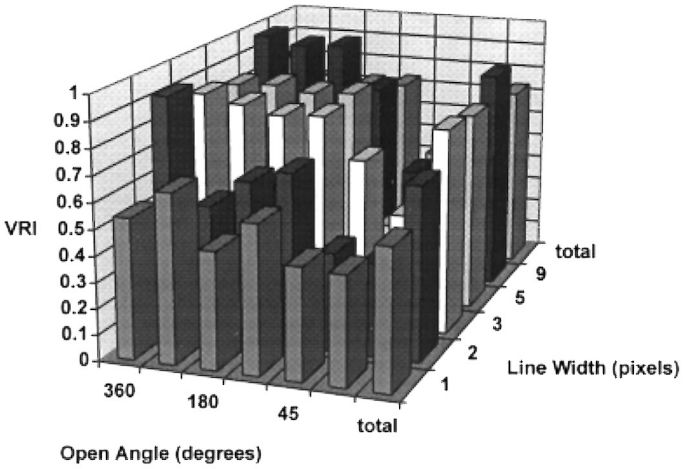
$$CDI = \gamma PRI + (1 - \gamma) VRI \tag{28}$$



6a 6b



7



8

Fig. 6a,b. Synthetic drawing (720 × 600 pixels) of arcs. **a** Synthetic image. **b** Vectors

Fig. 7. 3D plot of PRI in Fig. 6

Fig. 8. 3D plot of VRI in Fig. 6

Table 4. Automatic evaluation of pixel detection and vector detection of SPV algorithm

Fig. no.	Evaluation level	Detection rate	False alarm rate	Combined recovery index	Combined detection index
4	Pixel level	0.96	0.07	PRI=0.95	0.95
	Vector level	0.92	0.05	VRI=0.94	
5	Pixel level	0.96	0.08	PRI=0.94	0.93
	Vector level	0.90	0.08	VRI=0.91	

5 Experimental results of the protocol

We have applied the above definitions to devise an automatic evaluation protocol. The protocol is used to evaluate algorithms that are incorporated into the MDUS (Liu and Dori 1996b). The implementation is in C++ running on SGI Indy and Indigo2 workstations and SUN Sparcstations. The executable codes are available from the ftp address[1]. The pixel-level evaluation uses Eqs. 5, 10, and 11. The vector-level evaluation is carried out as follows. The first step is to match ground truth lines with detected lines. This is performed by finding, for each line in the ground truth set, the subset of detected lines that overlap it either fully or partially. The overlapping segments of every detected line and the ground truth line are measured using Eqs. 12–17. The vector detection qualities of these overlapping segments, as well as their lengths, are accumulated for both the ground truth lines and the corresponding detected lines. The basic quality (Q_b), fragmentation quality (Q_{fr}), and total quality (Q_v) for both the ground truth lines and their corresponding detected lines are calculated using Eqs. 18–20 and 22–24, respectively. After the quality of all ground truth lines and detected lines are calculated, the total vector detection rate, the vector false alarm rate, and the resulting VRI are calculated using Eqs. 21, 26, and 27, respectively.

[1] ftp.technion.ac.il/pub/supported/ie/dori/MDUS/sgim dus.gz, sunmdus.gz, and gtruth.tar.gz

Table 5. Automatic evaluation of the stepwise recovery arc segmentation algorithm

Row	Colum / level	1 D	1 F	1 RI	1 CDI	2 D	2 F	2 RI	2 CDI	3 D	3 F	3 RI	3 CDI	4 D	4 F	4 RI	4 CDI	5 D	5 F	5 RI	5 CDI	6 D	6 F	6 RI	6 CDI	Column total D	Column total F	Column total RI	Column total CDI
1	pixel	0.51	0.54	0.49	0.52	0.66	0.38	0.64	0.65	0.41	0.63	0.39	0.42	0.69	0.39	0.65	0.61	0.75	0.30	0.72	0.59	0.80	0.29	0.76	0.59	0.59	0.46	0.56	0.55
	vector	0.54	0.46	0.54		0.64	0.34	0.65		0.45	0.54	0.45		0.55	0.40	0.57		0.37	0.50	0.43		0.38	0.54	0.42		0.53	0.44	0.54	
2	pixel	0.86	0.17	0.85	0.89	0.83	0.18	0.82	0.67	0.83	0.18	0.83	0.73	0.80	0.19	0.81	0.74	0.75	0.31	0.72	0.55	0.66	0.36	0.65	0.51	0.82	0.20	0.81	0.84
	vector	0.91	0.07	0.92		0.51	0.49	0.51		0.61	0.37	0.62		0.65	0.30	0.67		0.34	0.57	0.38		0.35	0.63	0.36		0.66	0.32	0.67	
3	pixel	0.84	0.19	0.82	0.84	0.84	0.19	0.83	0.83	0.87	0.17	0.85	0.83	0.93	0.12	0.91	0.86	0.86	0.19	0.83	0.74	0.92	0.29	0.82	0.64	0.87	0.18	0.84	0.82
	vector	0.85	0.13	0.86		0.83	0.16	0.83		0.79	0.19	0.80		0.79	0.17	0.81		0.61	0.31	0.65		0.44	0.53	0.45		0.79	0.18	0.80	
4	pixel	0.97	0.19	0.89	0.86	0.97	0.20	0.88	0.86	0.98	0.20	0.89	0.86	0.95	0.12	0.92	0.88	0.89	0.20	0.85	0.58	0.92	0.25	0.83	0.63	0.96	0.19	0.89	0.84
	vector	0.84	0.17	0.83		0.84	0.16	0.84		0.82	0.18	0.82		0.82	0.15	0.83		0.26	0.65	0.30		0.39	0.55	0.42		0.77	0.22	0.78	
5	pixel	0.97	0.08	0.94	0.96	0.97	0.09	0.93	0.94	0.98	0.12	0.93	0.94	0.95	0.16	0.89	0.84	0.89	0.21	0.87	0.66	0.92	0.25	0.85	0.66	0.96	0.12	0.92	0.90
	vector	0.97	0.03	0.97		0.93	0.06	0.94		0.95	0.04	0.95		0.73	0.18	0.78		0.33	0.46	0.44		0.43	0.49	0.47		0.85	0.11	0.87	
Row total	pixel	0.91	0.16	0.87	0.85	0.91	0.16	0.87	0.81	0.91	0.18	0.87	0.80	0.92	0.16	0.88	0.81	0.89	0.22	0.84	0.64	0.89	0.27	0.81	0.62	0.91	0.18	0.87	0.80
	vector	0.82	0.18	0.82		0.74	0.25	0.75		0.71	0.28	0.72		0.70	0.24	0.73		0.38	0.50	0.44		0.40	0.55	0.42		0.71	0.26	0.73	
Global		D_p =0.91				F_p =0.18				*PRI*=0.87				D_v =0.71				F_v =0.26				*VRI*=0.73				*CDI*=0.80			

Table 6. Automatic evaluation of the dashed line detection algorithm

Ground truth line #	Endpt1		Endpt2		Detected line #	Endpt1		Endpt2		Location offset Endpt1		Endpt2		Kong's x	D	Fa	Hori's x	Lr	Dev	Cr	This paper Q_v	D_v	F_v	*VRI*
1	187	378	564	1	1	188	377	563	2	1	0	0	1	1	1	0	1	0.99	0	2	1.00	0.96	0	0.98
1	187	378	564	1	22[a]	492	73	503	60	305	-305	-61	59	0	0	1	1	0.03	1	2	0.73	0.96	0.27	0.85
2	12	673	483	202	13	24	661	481	203	12	-12	-2	1	1	1	0	1	0.97	1	1	0.89	0.86	0.11	0.88
3	34	792	533	293	15	47	779	531	294	13	-12	-2	1	1	1	0	1	0.97	1	1	0.89	0.86	0.11	0.88
4	388	576	962	2	2	389	575	961	3	1	0	0	1	1	1	0	1	1.00	0	1	1	1.00	0	1.00
5	203	946	973	176	12	204	945	971	177	1	0	-2	1	1	1	0	1	1.00	1	1	0.89	0.88	0.11	0.89
6	69	100	995	100	11	70	100	995	100	1	0	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
7	27	264	994	264	14	28	264	985	264	1	0	-9	0	1	1	0	1	0.99	0	1	1	0.99	0	1.00
8	55	400	958	400	16	56	400	958	400	1	0	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
9	60	509	982	509	17	62	509	981	509	2	0	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
10	39	607	970	607	18	41	607	949	607	2	0	-21	0	1	1	0	1	0.98	0	1	1	0.98	0	0.99
11	28	679	961	679	19	29	679	961	679	1	0	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
12	44	786	959	786	20	45	786	959	786	1	0	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
13	59	887	958	887	21	83	887	938	887	24	0	-20	0	1	1	0	1	0.95	0	1	1	0.95	0	0.98
14	152	18	152	998	6	152	20	152	996	0	2	0	-2	1	1	0	1	1.00	0	1	1	1.00	0	1.00
15	270	11	270	978	5	270	12	270	978	0	1	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
16	372	18	372	979	7	372	20	372	975	0	2	0	-4	1	1	0	1	0.99	0	1	1	0.99	0	1.00
17	508	3	508	981	3	508	4	508	981	0	1	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
18	647	25	647	992	8	647	26	647	978	0	1	0	-14	1	1	0	1	0.98	0	1	1	0.98	0	0.99
19	749	7	749	958	4	749	8	749	949	0	1	0	-9	1	1	0	1	0.99	0	1	1	0.99	0	1.00
20	824	32	824	978	9	824	33	824	978	0	1	0	0	1	1	0	1	1.00	0	1	1	1.00	0	1.00
21	952	4	952	964	10	952	33	952	964	0	29	0	0	1	1	0	1	0.97	0	1	1	0.97	0	0.99
For the entire drawing															1.00	0.05		0.99	0.18	1.05		0.97	0.01	0.98

[a] The detected line #22 is solid

Tables 2 and 3 are examples of our protocol's performance evaluation of bar detection and arc detection, respectively. Table 2 also compares the proposed protocol with the protocols proposed by Kong et al. (1996) and Hori and Doermann (1996). In Kong et al.'s protocol the match result is binary: 1 for a matched line and 0 for an unmatched line. In Hori and Doermann's protocol the count ratio is the ratio between the number of detected lines and the number of ground truth lines, the length ratio is the ratio between the sum of lengths of the detected lines and that of the ground truth lines, and the deviation is the average of euclidean distance between each pair of matched lines. In Table 2 the ground truth is an 8-pixel-wide bar ending at (10,20) and (90,20) in the first eight cases and is broken into two bars at (35,20)–(50,20) in the merged case. In Table 3 the ground truth in the first three cases is a full circle and in other cases is an arc ending at (30,15) and (70,15). The center is (50,50). The radius is 40. The width is 8. The width of the detected line is also 8.

To test the protocol on whole drawings we applied it to evaluate the performance of three line detection algorithms within MDUS: sparse pixel vectorization (SPV; Liu and Dori 1996a), stepwise recovery arc segmentation (SRAS), and dashed line detection (DLD; Dori et al. 1996). To test the SPV performance automatically, we manually generated the drawings in Figs. 4a, 5a (using Microsoft Paintbrush software). In this way we know all the attribute values of the ground truth lines. The vectorized results output by SPV are displayed in Figs. 4b, 5b, and the evaluation results are shown in Table 4.

Figure 6a is the ground truth image of arcs, generated in the same way, to evaluate the SRAS algorithm, and Fig. 6b shows the arcs recognized by SRAS. The arcs are drawn in a 5×6 matrix, the widths of lines in the rows numbering 1–5 from top are 1, 2, 3, 5, and 9 pixels, the open angles in

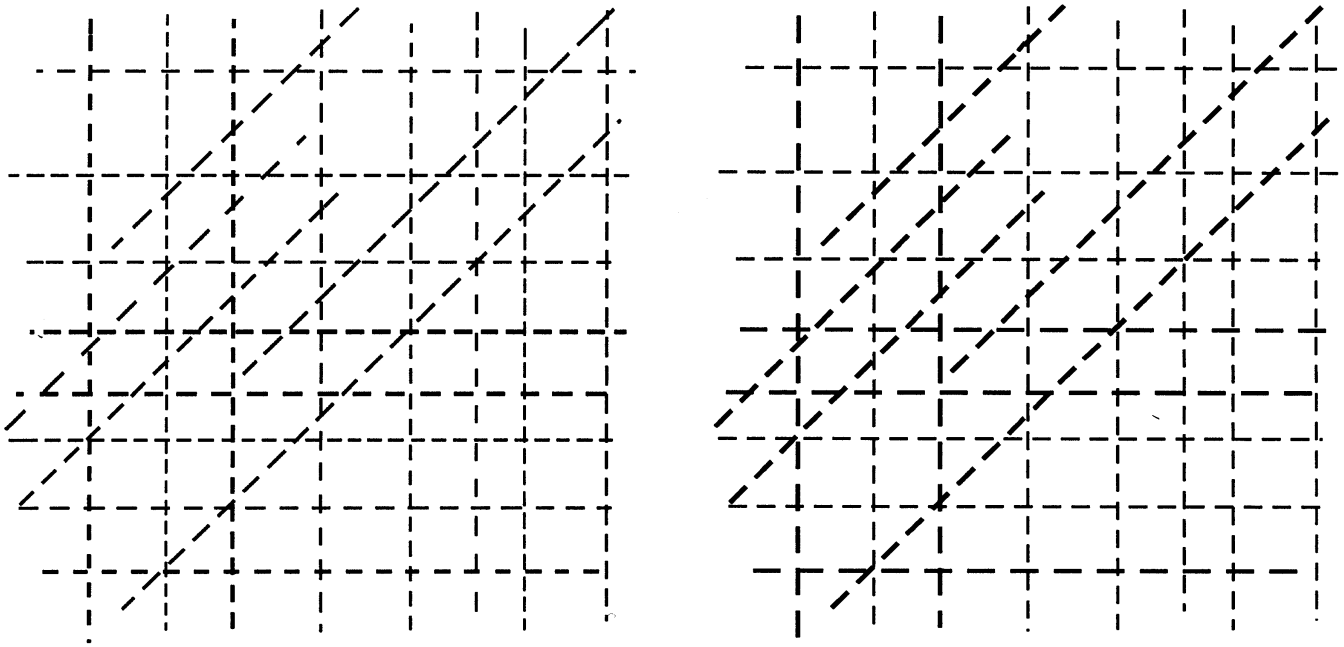


Fig. 9a,b. An automatically generated drawing (1000 × 1000 pixels) of dashed lines. **a** Image. **b** Vectors

Table 7. Time performance of MDUS (in seconds on SGI Indigo2)

Submodule	Fig. no.	Size (pixels)	Black pixels	Pixel density	Ground truth lines	Detected lines	Time (s)	Time/pixel (s)	Time/vector (s)
Vectorization	4	400×300	18379	0.153	50	60	1	54.4×10^{-6}	20.0×10^{-3}
Vectorization	5	400×300	22361	0.186	64	86	1	44.7×10^{-6}	15.6×10^{-3}
Arc segmentation	6	720×600	29510	0.068	153	241	4	135.5×10^{-6}	26.1×10^{-3}
Dashed line detection	9	1000×1000	53660	0.054	21	22	1	18.6×10^{-6}	47.6×10^{-3}

the columns numbering 1–6 from left are 2π, $3\pi/2$, $\pi/2$, $\pi/4$, and $\pi/8$, and the radii of the concentric arcs in each matrix cell are 5, 10, 20, and 50 pixels. The evaluation results are shown in Table 5, where D is the detection rate, F is the false alarm rate, RI is the recovery index at either pixel or vector level, and CDI is the combined detection rate defined in Eq. 28. Both PRI and VRI increase along with the open angle and the width of the arc, as shown in the 3D plots in Figs. 7 and 8. The highest PRI and VRI are 0.97 for a 9-pixel-wide circle, the worst PRIs are around 0.50 for 1-pixel-wide arcs, and the worst VRIs are around 0.40 for $\pi/8$ arcs. The ground truth images (in TIFF) and vectors (in IGES) are also available from the ftp address (see above).

Figure 9a is an automatically generated drawing of dashed lines (generated in the Dashed Line Detection Contest held at the Pennsylvania State University during the First International Workshop on Graphics Recognition, August, 1995). Figure 9b is the result generated by our DLD algorithm that won the contest (Dori et al. 1996). The automatic evaluation was performed by a program written especially for the contest by a team lead by Professor Haralick (Kong et al. 1996). The recognition rate was 100%, with only one false alarm. The automatic evaluation on the same drawing has also been performed using our protocol and Hori and Doermann's protocol. The results are listed and compared in Table 6, where x is match, D is the detection rate, Fa is the false alarm rate, Lr is the length ratio, Cr is the count ratio, and Dev is the deviation; Q_v, D_v, F_v, and VRI are as defined in Sect. 4.2.

Finally, time efficiency is evaluated by reporting the elapsed time of running MDUS for each drawing. Since drawings vary in sizes, density, and complexity, we use two drawing-independent time evaluation factors to evaluate the time efficiency of line detection algorithms: time per (black) pixel and time per (ground truth) line. The time performance evaluation of MDUS is shown in Table 7.

6 Conclusion

We have demonstrated a protocol for performance evaluation of line detection algorithms. A preliminary comparison between the proposed protocol and two other protocols is also presented. The proposed protocol is objective and comprehensive. It incorporates a set of single and combined performance evaluation indices at both the pixel and vector levels. Both the detection and false alarm rates are considered. The protocol is designed for straight lines, circular arcs, and polylines of any style. It may also be extended to other line shapes. The results presented in Tables 2 and 6 may indicate that the proposed protocol reflects line detection performance more accurately than the other two because it incorporates many aspects and factors. Comparing the results of the three evaluation protocols, we see that Kong's protocol is more binary and less accurate, and that Hori and Doermann's pro-

tocol yields several single indices and no overall index. To evaluate the performance of the evaluation protocols more work is required on a large sample of drawings.

Time efficiency is also included in the protocol. The time/vector factor is greater on high quality drawings in which the vectors are longer and their number is small. This factor is not very suitable for evaluating the time efficiency of the vectorization algorithms because the vectors' attributes (e.g., length and width) vary much from drawing to drawing. However, the time/pixel does not vary significantly from drawing to drawing. In general, the higher the pixel density, the higher is the time/pixel factor because high density pixels have higher coupling.

Based on the experiments presented in this work, an algorithm with a combined detection (recovery) index of 0.8 or higher may be considered good with respect to human vision evaluation. However, more work should employ this protocol on a series of algorithms and degraded drawings to obtain an objective assessment on commonly accepted criteria.

Acknowlegements. This research was supported by the fund for the promotion of research at the Technion.

References

1. Boatto L et al (1992) An interpretation system for land register maps. IEEE Computer 25(7):25—32
2. Cordella LP, Marcelli A (1996) An alternative approach to the performance evaluation of thinning algorithms for document processing applications. In: Kasturi R, Tombre K (eds) Graphics recognition – methods and applications (Lecture Notes in Computer Science, vol 1072). Springer, Berlin Heidelberg New York, pp 13–22
3. Dori D (1995) Vector-based arc segmentation in the machine drawing understanding system environment. IEEE Trans Pattern Anal Machine Intell 17:959–971
4. Dori D, Liang Y, Dowell J, Chai I (1993) Spare pixel recognition of primitives in engineering drawings. Machine Vision Appl 6:79–82
5. Dori D, Liu W, Peleg M (1996) How to win a dashed line detection contest. In: Kasturi R, Tombre K (eds) Graphics recognition – methods and applications (Lecture Notes in Computer Science, vol 1072). Springer, Berlin Heidelberg New York
6. Filipski AJ, Flandrena R (1992) Automated conversion of engineering drawings to CAD form. Proc IEEE, 80:1195–1209
7. Haralick RM (1992) Performance characterization in image analysis – thinning, a case in point. Pattern Recogn Lett 13:5–12
8. Hori O, Doermann DS (1996) Quantitative measurement of the performance of raster-to-vector conversion algorithms. In: Kasturi R, Tombre K (eds) Graphics recognition – methods and applications (Lecture Notes in Computer Science vol 1072). Springer, Berlin Heidelberg New York, pp 57–68
9. Jaisimha MY, Haralick RM, Dori D (1993) A methodology for the characterization of the performance of thinning algorithms. In: Proceedings of the Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, pp 282–286
10. Kadonaga T, Abe K (1996) Comparison of methods for detecting corner points from digital curves. In: Kasturi R, Tombre K (eds) Graphics recognition – methods and applications (Lecture Notes in Computer Science, vol 1072). Springer, Berlin Heidelberg New York, pp 23–34
11. Kanungo J. Haralick RM, Phillips IT (1993) Global and local degradation models. In: Proceedings of the Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, pp 730–732
12. Kasturi R, Bow ST, El-Masri W, Shah J. Gattiker JR, Mokate UB (1990) A system for interpretation of line drawings. IEEE Trans Pattern Anal Machine Intell 17:978–992
13. Kong B, Phillips IT, Haralick RM, Prasad A, Kasturi R (1996) A benchmark: performance evaluation of dashed-line detection algorithms. In: Kasturi R, Tombre K (eds) Graphics recognition – methods and applications (Lecture Notes in Computer Science vol 1072). Springer, Berlin Heidelberg New York, pp 270–285
14. Lam L, Suen CY (1993) Evaluation of thinning algorithms from an OCR viewpoint. In: Proceedings of the Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, pp 287–290
15. Lee S, Lam L, Suen CY (1991) Performance evaluation of skeletonization algorithms for document image processing. In: Proceedings of the First International Conference on Document Analysis and Recognition, Saint-Malo, France, pp 260–271
16. Liu W, Dori D, Tang L, Tang Z (1995) Object recognition in engineering drawings using planar indexing. In: Proceedings of the First International Workshop on Graphics Recognition, Pennsylvania State University, USA, pp 53–61
17. Liu W, Dori D (1996a) Sparse pixel tracking: a fast vectorization algorithm applied to engineering drawings. Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, vol III (Robotics and Applications), pp 808–811
18. Liu W, Dori D (1996b) Automated CAD conversion with the machine drawing understanding system. In: Proceedings of Document Analysis Systems, Malvern, Penn, USA, pp 241–259
19. Nagasamy V, Langrana N (1 990) Engineering drawing processing and vectorization system. Comput Vision Graphics Image Process 49:379–397
20. Nalwa VS (1993) A guided tour of computer vision. Addison-Wesley, New York
21. Vaxiviere P, Tombre K (1992) Celestin: CAD conversion of mechanical drawings. IEEE Comput 25:46–54

Liu Wenyin received his BEng (1988) and MEng (1992) degrees from the Department of Computer Science and Technology, the Tsinghua University, Beijing, People's Republic of China. Currently he is a DSc candidate at the Technion, Israel Institute of Technology. His research interests include automated engineering drawing interpretation, software engineering, object-oriented programming, and artificial intelligence. He was a senior member of the team that won first place in the Dashed Line Recognition Contest held during the IAPR International Workshop for Graphics Recognition at Pennsylvania State University, 1995.

Dov Dori is Alexander Goldberg Senior Lecturer of Industrial Engineering and Management at the Technion, Israel Institute of Technology. In 1995–1996 he was head of Information Systems Engineering Area in the Faculty of Industrial Engineering and Management. Between 1987 and 1990 he was assistant professor at the Department of Computer Science, University of Kansas. Between 1975 and 1984 he was an officer of the Israel Defense Force (last rank Major) and served as chief industrial engineer of the Merkava Tank Production Plant and head of the Armored Vehicle Rehabilitation Computer Unit. He was head of the team who won first place in the Dashed Line Recognition Contest held during the IAPR International Workshop for Graphics Recognition, Pennsylvania State University, 1995. Dov Dori has developed the object-process methodology for systems development and is associate editor of *International Journal of Pattern Recognition and Artificial Intelligence*, co-editor of "Shape Structure and Pattern Recognition" (World Scientific 1995), member of IEEE, IEEE Computer Society, Association for Computing Machinery, and International Association for Pattern Recognition.