Springer

Dear Author:

Please find attached the pdf file of your contribution, which can be viewed using the Acrobat Reader, version 3.0 or higher. We would kindly like to draw your attention to the fact that copyright law is also valid for electronic products. This means especially that:

- You may not alter the pdf file, as changes to the published contribution are prohibited by copyright law.

- You may print the file and distribute it amongst your colleagues in the scientific community for scientific and/or personal use.

- You may make an article published by Springer-Verlag available on your personal home page provided the source of the published article is cited and Springer-Verlag is mentioned as copyright holder. You are requested to create a link to the published article in LINK, Springer's internet service. The link must be accompanied by the following text: The original publication is available on LINK **http://link.springer.de.** Please use the appropriate URL and/or DOI for the article in LINK. Articles disseminated via LINK are indexed, abstracted and referenced by many abstracting and information services, bibliographic networks, subscription agencies, library networks and consortia.

- You are not allowed to make the pdf file accessible to the general public, e.g. your institute/your company is not allowed to place this file on its homepage.

Yours sincerely,

Springer-Verlag Berlin Heidelberg

IJDAR (2000) 3: 117–124

International Journal on Document Analysis and Recognition

Content-based matching of line-drawing images using the Hough transform

Pasi Fränti[1], Alexey Mednonogov[2], Ville Kyrki[2], Heikki Kälviäinen[2]

[1] Department of Computer Science, University of Joensuu, P.O. Box 111, 80101 Joensuu, Finland
[2] Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland

Abstract. We introduce two novel methods for content-based matching of line-drawing images. The methods are based on the Hough transform (HT), which is used to extract global line features in an image. The parameter space of the HT is first thresholded in order to preserve only the most significant values. In the first method, a feature vector is constructed by summing up the significant coefficients in each column of the accumulator matrix. In this way, only the angular information is used. This approach enables simple implementation of scale, translation, and rotation invariant matching. The second variant also includes positional information of the lines and gives a more representative description of the images. Therefore, it achieves more accurate image matching at the cost of more running time.

Key words: Image matching – Graphics recognition – Line-drawing images – Hough transform – Content-based retrieval

1 Introduction

We consider a database of line-drawing images of complex engineering drawings, e.g., electrical circuits, cartographic maps, and architectural and urban plans. We assume that the drawings are binary (black-and-white) images, and that they consist mainly of line segments. This is a reasonable assumption for a variety of engineering drawings.

Flexible usage of the database requires that images and parts of images can be searched for and retrieved efficiently. The user should be able to find images containing objects similar to a given query image without human interaction or exhaustive computing resources. For example, in remote sensing [19] and in multimedia systems [8], it is important to analyze sets of images, not only as temporal sequences, but also as queries from an image database. This applies also to engineering drawings.

Correspondence to: P. Fränti

A common approach for the image retrieval problem is to use an indexing scheme in order to avoid full-scale image matching [18]. There are two main approaches for the indexing: *key words* and *content-based retrieval*. Key words have the advantage that the solutions for text retrieval are available for us [26]. This approach, however, is impractical because we must design key words and proper classification for the images in the database. Manual operations are thus needed for generating the indexing.

In content-based image retrieval, images are automatically indexed by generating a *feature vector* describing the global content of the image. The feature vector can be stored as an index in the database, or it can be generated on-line during the retrieval process. The similarity of the feature vectors of the query and database images is measured to retrieve the image.

In this paper, we address the problem of generating a feature vector to be used in image matching and object recognition. We introduce two novel methods based on the *Hough transform* (HT) [9, 13, 17, 21] for extracting the feature vectors. The HT is well suited for this task because it gives a global description of the spatial image content. It makes no assumptions on the image type and, in principle, it should be applicable to any type of binary images. The HT is essentially a transform of an image into a parameter domain, also called an *accumulator space*.

The first variant, presented in [6], uses only the angular information of the accumulator matrix. It gives a global description of the image with a very compact index size. This allows fast image retrieval with scale, translation, and rotation invariant matching. In the case of larger and more complex images, however, the angular information is not always sufficient to differentiate images from each other.

The second variant, presented in [7], alleviates this problem by including positional information in the index. The feature vector is a binarized accumulator matrix, in which a number of most significant coefficients of the matrix are preserved. This gives a more representative description of the image and therefore allows more accurate image matching. The main problems of this ap-

proach are: (1) how to keep the feature vector compact and (2) how to keep the property of the matching translation and rotation invariant. We give solutions to both of these problems. Experiments show that this variant has better matching accuracy than the first variant, but at the cost of slower retrieval.

The methods are designed for matching full image similarity. We show with experiments that the methods are relatively fast and work well in the case of smaller image objects. The methods are therefore applicable to real-time matching of small images and object recognition. The results, however, are not very promising for large-scale images. The application to image retrieval will therefore require the method to be integrated with a suitable image segmentation technique in order to allow matching based on smaller image samples.

The feature extraction is also limited to line segments only; other image features such as complex curves and irregular shapes are not directly taken into consideration in the assessment of the similarities. However, variants of the HT could be used to extract any parametrizable features such as circles or elliptical arcs. In addition, the *generalized Hough transform* (GHT) could be applied for more general shapes [15]. Unfortunately, the use of higher-dimensional features would require sophisticated solutions for reducing the size of the parameter space.

The rest of the paper is organized as follows. The problem formulation and application area are briefly summarized in Sect. 2. The two variants of the new HT-based image matching are then introduced in Sects. 3 and 4. Practical experiments are described in Sect. 5, and finally conclusions are drawn in Sect. 6.

2 Content-based image retrieval

In image retrieval, a typical task is to find all database images similar to a given query image. The search can be limited to a given subdirectory defined by the user. The result of the query can be a list of similar images in descending order of similarity and the user may then repeat the search, using any of the result images as a new query image. This is the notion of *query by image example* [18].

Content-based image retrieval consists of three components: *feature extraction*, *multidimensional indexing*, and the *retrieval engine*. Feature extraction determines the representation of the visual features as multidimensional feature vectors. The images are indexed by their feature vectors, and the similarity of the feature vectors of the query image and the database images are measured to retrieve the images.

The selection of the features is the key part of the image retrieval. The index should be small enough to be stored compactly and processed efficiently, but it should also be representative so that the images can be differentiated from each other on the basis of their feature vectors. Fast matching of the feature vectors is also desirable. The design of the indexing scheme is a compromise between retrieval accuracy and speed.

The choice of the feature extraction method also depends on the application because different features represent different visual entities [18, 23]. Color and texture are commonly used features in the case of photographic images [14]. Unfortunately, there are typically no colors and texture in binary images and these features are, therefore, not likely to be useful.

Shape-based image retrieval, however, can be applied in the case of binary images [20]. Shape representation is usually required to be invariant to geometric transformations such as translation, rotation, and scaling [20]. Methods of shape representation can be divided into two categories: *boundary based* and *region based* [18, 23]. The methods in the first category utilize only the outer boundary of the shape, while the methods in the second category utilize the entire shape region. Boundary-based methods include *Fourier descriptors* [16, 22] and *polygonal approximation* [1]. *Moment invariants* are commonly used region-based shape features [10].

Since moment invariants are region based, they apply mainly to images with solid objects of filled regions. In contrast, most boundary-based features such as Fourier descriptors require that the boundary of the shape is closed in order to be rotation invariant. Therefore, these shape features are not very useful for comparing line-drawing images.

There has been a substantial amount of research on graphics recognition. Especially the steps required in the automatic conversion of images into CAD models (*raster-to-vector conversion*) have been studied extensively [2, 3, 5, 24, 25]. Structural image comparison can be built on these ideas and methods, as has been done with *graph matching* [4, 12]. Separate steps are required for the interpretation of the image for extracting the features, building the graph, and finally the graph matching itself. Unfortunately such systems can be rather complex and computationally expensive, which can be a serious limitation in a practical image retrieval system. Some work for inexact graph matching using histograms of pairwise attributes has been done to lower the cost of matching [11].

3 HT-based image matching

We apply the HT for feature extraction, as it gives a scale-independent global description of the spatial image content. This approach lies somewhere between the general image attributes (e.g., moment invariants) and structural matching because the features are extracted with global information of the images as is done with general features, but the feature itself represents the structure of the image, i.e., the relations between the individual points and lines.

Next, we introduce the variant where we use only the angular information. In this way, we limit the number of comparison operations needed in the matching. Furthermore, the angular information is independent of the spatial location of the lines, and therefore the method is translation and scaling invariant by its nature.

3.1 The Hough transform (HT)

In the HT, global features are sought as sets of individual points over the whole image. In the simplest case, the features are straight line segments. In the case of binary images, the line detection algorithm can be described as follows:

1. Create the set of the foreground pixels in the image.
2. Transform each pixel in the set into a parametric curve in the parameter space.
3. Increment the cells in the accumulator matrix A determined by the parametric curve.
4. Detect local maxima in the accumulator array. Each local maximum may correspond to a parametric curve in the image space.
5. Extract the curve segments using the knowledge of the maximum positions.

In Step 2, the transformation can be $\rho = x \cdot \cos\theta + y \cdot \sin\theta$, where (x, y) are the coordinates of the pixel to be transformed, and ρ and θ are the parameters of the corresponding line. Thus, every pixel (x, y) can be seen as a curve in the (ρ, θ) parameter space, where θ varies from the minimum to the maximum value, giving the corresponding ρ values. By transforming every point (x, y) in the image into the parameter space, the line parameters can be found in the intersections of the parameterized curves in the accumulator matrix.

3.2 Generating feature vectors

We denote the accumulator matrix by A, where each row corresponds to one value of ρ, and each column to one value of θ. The procedure for generating the feature vector from the accumulator matrix is described in Fig. 1. First we extract only the most significant information by thresholding the matrix using a threshold value T (step 1). Next, we shrink the thresholded accumulator matrix to a one-dimensional θ vector by summing up the remaining coefficients in each column (step 2). Thus, only the angular information of the matrix will be used. Finally, we normalize the feature vectors according to the mean value of the components of the vector (steps 3 and 4).

3.3 Translation- and scale-invariant matching

The usage of the angular information (θ vector) has several advantages. Firstly, the matching is independent of the spatial location of the lines, and therefore, the method is translation and scaling invariant by its nature. This is verified by the fact that every line has the same θ value independent of scaling and translation of the image. The angular information can be sufficient for separating different image types. For example, a drawing of buildings consists mainly of 45° and 90° angles.

We approximate the dissimilarity of the images by calculating the distance of their feature vectors. Let us assume that we have the feature vector R for a database image and the feature vector S for the sample query image (both of size N). Their distance is calculated as:

$$d(R,S) = \sum_{j=1}^{N} (R_j - S_j)^2 \; . \tag{1}$$

Here $d = 0$ coincides to absolutely similar images and $d = d_{\max}$ coincides to images with no similarities found. An important advantage of the simplicity of the formula is its speed. This is a highly desired property for searching a large image database.

1. Threshold the matrix:

$$A'_{ij} = \begin{cases} A_{ij}, & \text{if} A_{ij} > T \\ 0, & \text{if} A_{ij} \le T \end{cases} \quad \forall \; i = 1..M, \;\; j = 1..N$$

2. Calculate preliminary feature vector:

$$F_j^0 = \sum_{i=1}^{M} A'_{ij} \quad \forall \; j = 1..N$$

3. Calculate vector mean:

$$m = \frac{1}{N} \sum_{j=1}^{N} F_j^0$$

4. Normalize the feature vector:

$$F_j = \frac{F_j^0}{m} \quad \forall \quad j = 1..N$$

Fig. 1. Algorithm for generating the feature vector

3.4 Rotation-invariant matching

Rotation-invariant matching of the method can be obtained with minor modifications at the cost of increasing computation time if we consider the matching as a histogram-matching problem. Consider the two images and their rotated variants in Fig. 2. The feature vectors of the *Image A* and its rotated variant (see Fig. 3) have the same shape, but they have different locations along the θ-axis. The best match can be found by rotating the feature vector of the sample query image, and calculating the distance in all positions.

The distance of the feature vectors is defined as the minimum distance of all possible fittings:

$$d(R,S) = \min_{k=1,2,\ldots,N} \sum_{j=1}^{N} (R_j - S_j^{(k)})^2 \; , \tag{2}$$

where $S^{(k)}$ is a vector obtained by cyclic rotation of the feature vector S by k steps to the right. Thus, we use an exhaustive search for finding the minimum distance. This requires $O(N^2)$ time, where N is the size of the feature vector.

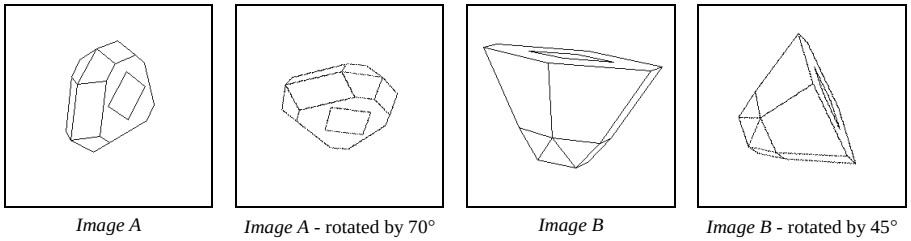

Fig. 2. Sample images and their rotated variants

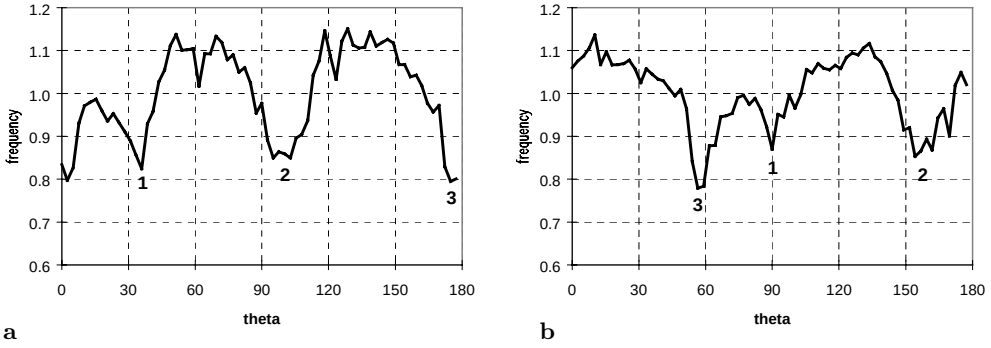

Fig. 3. a Graphical representation of the feature vectors; **b** its rotated variant

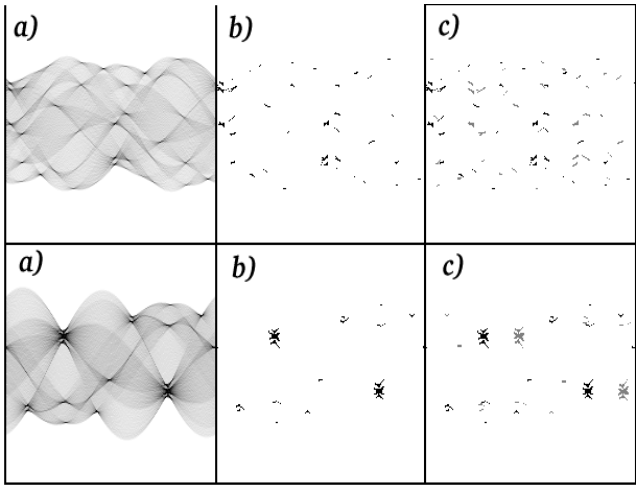

Fig. 4a–c. Example of the feature extraction for two sample images; *Image A* is on the top and *Image B* is on the bottom. The figures are: **a** an accumulator array ($\rho\theta$-matrix) of the sample image; **b** a thresholded and binarized array; **c** a combination of the feature vectors of the image and its rotated variant

4 Matching with the full matrix

The angular information can be sufficient for differentiating images with clearly different types. However, larger and more complex images can result in a practically uniform distribution of θ values, and in this case, the use of angular information may not be sufficient to differentiate such images. We therefore extend the approach presented in Sect. 3 by including positional information of the lines in the feature vector.

We propose to use the full matrix and, in this way, also use the spatial location of the lines with respect to each other. The main problem of this approach is that the matrix can be quite large, even after thresholding. This is because the lines are recognized not only by their orientation (θ), but also by their distance to the origin (ρ). This distributes the observations more widely in the feature space. At the same time, the storage and retrieval constraints force us to keep the feature vector relatively small.

Therefore, we utilize the full accumulator matrix, which is thresholded and binarized in order to keep the feature vector compact. The matching procedure is revised in order to preserve scale, translation, and rotation invariant matching.

We reduce the amount of information in the following ways. Firstly, we extract only the K most significant coefficients by thresholding the matrix, and then we binarize the thresholded values. Secondly, we optimize the size of the accumulator array by normalizing the coordinate space. This is done by calculating the centroid (μ_x, μ_y), i.e., the arithmetic average of the location of the object pixels:

$$\mu_x = \frac{1}{NM}\sum_{i=1}^{NM} x_i \qquad \mu_y = \frac{1}{NM}\sum_{i=1}^{NM} y_i \ . \tag{3}$$

The purpose of the normalization is to utilize the accuracy of the array more effectively by eliminating empty space from the matrix. We can now obtain a proper range for ρ by calculating the standard deviation of the spatial location of the object pixels in respect to their centroid:

$$\sigma = \sqrt{\frac{1}{NM}\sum_{i=1}^{NM}\left\{(x_i-\mu_x)^2 + (y_i-\mu_y)^2\right\}} \ . \tag{4}$$

The range can then be defined as:

$$[\rho_{\min},\ \ \rho_{\max}] = [0,\ \ 2.5\cdot\sigma] \ , \tag{5}$$

where 2.5 is a constant so that approximately 99% of the pixels belong to the range, assuming that the pixels are scattered around the centroid with a Gaussian distribution. Denote the binary matrix value of the database image as R_{ij}, and of the query image as S_{ij}. Scaling-

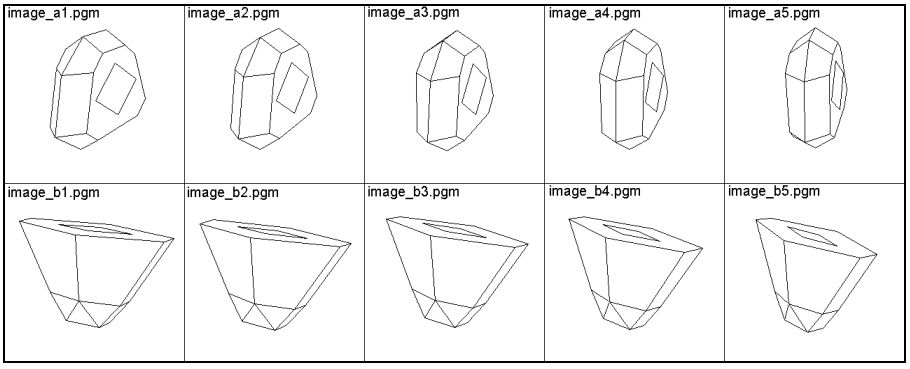

Fig. 5. Test database of ten images. The images have been generated by 3-D rotation of the two original images (*Image A* and *Image B*)

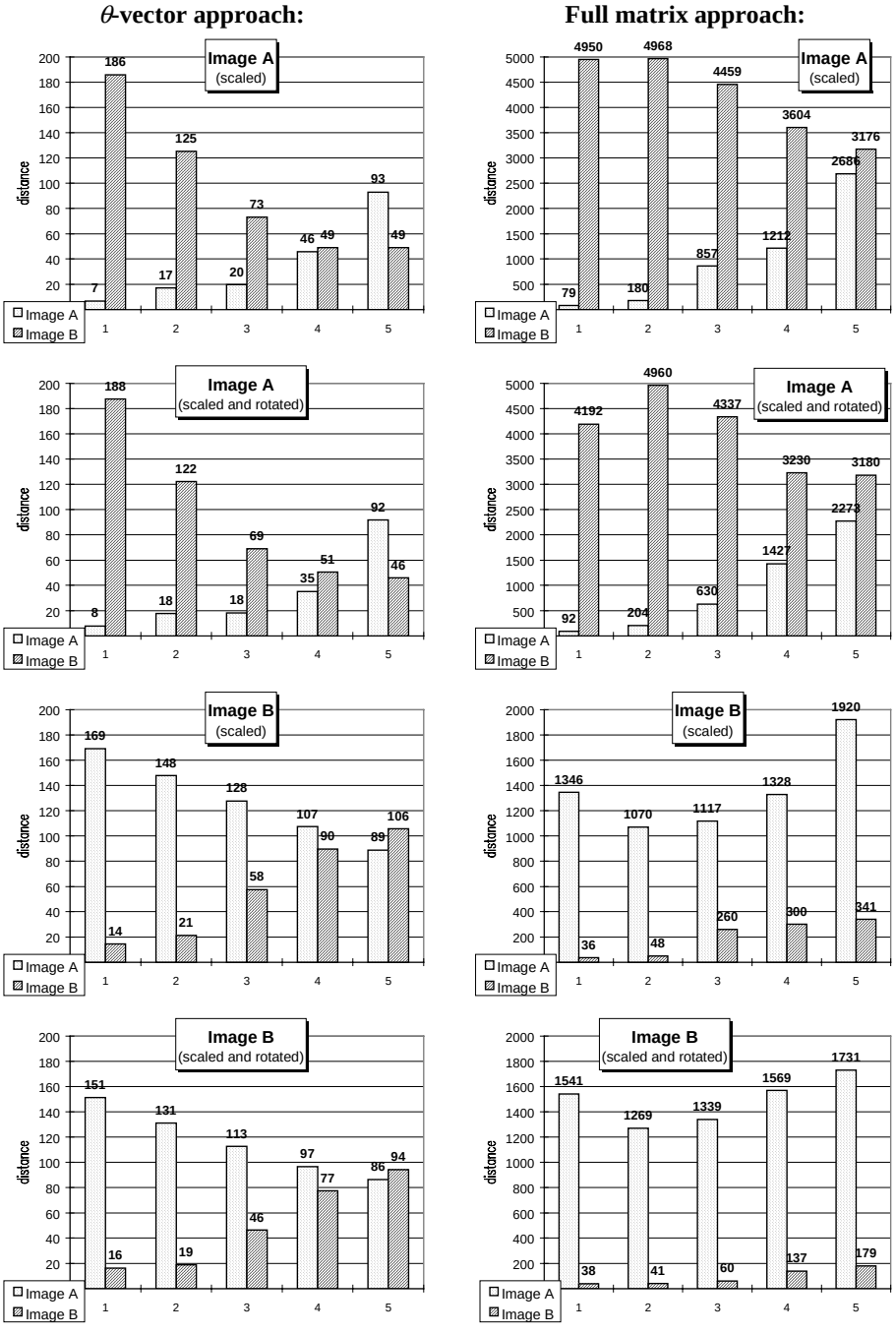

Fig. 6. Matching results of the query images A and B (scaled $a1s$, $b1s$; scaled and rotated $a1sr$, $b1sr$). The diagrams to the left show the results for the θ-based approach according to Eq. 2, and the diagrams to the right for the full matrix approach according to Eq. 7

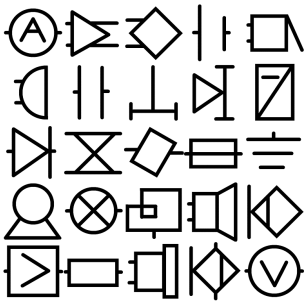

Fig. 7. A symbol library of 25 line-drawing objects

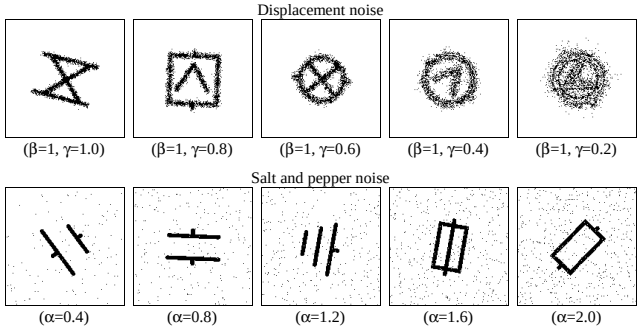

Fig. 8. Examples of library symbols after rotation and addition of noise

Table 1. Summary of run times of the matching process

	Feature extraction	Matching	Total	
			On-line	Off-line
θ-vector	126.0 ms	2.5 ms	3339 ms	189 ms
Full matrix	72.0 ms	19.0 ms	2347 ms	547 ms

and translation-invariant matching can be obtained by calculating their distance as:

$$d_{TS}(R,S) = \sum_{i=1}^{N}\sum_{j=1}^{M} S_{ij} \cdot \min_{R_{kl}\neq 0}\left\{(i-k)^2+(j-l)^2\right\}. \quad (6)$$

In other words, for each nonzero coefficient S_{ij}, we calculate its distance in the parameter space to the nearest nonzero coefficient R_{ij}. This can be performed in $O(K^2)$ time, which is computationally feasible if the feature vector is small.

Using Eq. 6, we define rotation-invariant matching by rotating the feature matrix along the θ-axis, and calculating the distance in all positions. The distance is defined as the minimum distance of all possible fittings:

$$d_{RTS}(R,S) = \min_{m}\left\{d_{TS}\left(R,S^{(m)}\right)\right\} . \quad (7)$$

In the database, the feature matrix would be stored as a list of the preserved values: $\{(\theta_1,\rho_1),(\theta_2,\rho_2),...,(\theta_K,\rho_K)\}$. In this way, the feature vector can be stored compactly, and the space requirement (in bits) is:

$$K\cdot\log_2\cdot q_\rho + K\cdot\log_2\cdot q_\theta \ , \quad (8)$$

where q_ρ and q_θ are the numbers of quantization steps for the ρ and θ.

The feature extraction process is illustrated in Fig. 4 in the case of two sample images. Column **a** shows the original parameter space of the HT, and the column **b**, the parameter space after thresholding and binarization. In column **c**, we have combined the features of the image (black pixels) with the features of the same image after rotation by 45° degrees (gray pixels). It can be seen that the features of the rotated image are shifted to the right approximately by the amount of the rotation.

5 Experiments

We first test the matching accuracy of the proposed methods using the two images of Fig. 2. A small database is generated by rotating these images by 5, 10, 15, 20, and 25 degrees around the y-axis (Fig. 5). The idea is that the rotation is performed in 3-D vector space, which should gradually destroy the 2-D similarity of the images. The original images are denoted as images A and B, and their rotated versions as images $A1$, $A2$, $A3$, $A4$, $A5$, $B1$, $B2$, $B3$, $B4$, and $B5$.

As query images, we use the same images processed as follows: the first query image ($a1s$) is a scaled version of the image A, and the second one ($a1sr$) is a scaled and 2-D rotated version of the same image. The amount of rotation is shown in Fig. 2. The third ($b1s$) and fourth ($b1sr$) images are generated from image B.

The first hypothesis is that the matching should be equally good with and without 2-D rotation ($a1s$ vs. $a1sr$ and $b1s$ vs. $b1sr$). The second hypothesis is that the first two query images should match the database image $A1$ best, since only scaling and 2-D rotation were applied. The matching should be slightly weaker for the images $A2$, $A3$, $A4$, and $A5$. Moreover, the images $a1s$ and $a1sr$ should not match the images $B1$, $B2$, $B3$, $B4$, and $B5$.

The results are show in Fig. 6 as the distances between the query and database images. The first observation is that the matching accuracy is virtually the same with and without the rotation, which supports the hypothesis that the matching is rotation invariant. The second observation shows that the image $A1$ is the closest to both $a1s$ and $a1sr$; and the image $B1$ is the closest to $b1s$ and $b1sr$. The results also show that images $A2$ and $A3$ are rather close according to the measured distance values of both approaches. However, in the case of the θ-based approach, images $A4$ and $A5$ do not match as well, as their distance is already as great as that of the best matched images $B4$ and $B5$ in the wrong test set. The full matrix approach, however, is capable of differentiating the two image sets in the case of all five images.

A second experiment was carried out with the symbol library of the symbol recognition contest in *International Conference on Pattern Recognition* (ICPR′2000, Barcelona, Spain) (Fig. 7). As query images, we use the same set of images distorted by the following processes: the image is first rotated by a random amount, and then noise is included in the image. The noise model was also provided by the organizers of the symbol recognition contest.

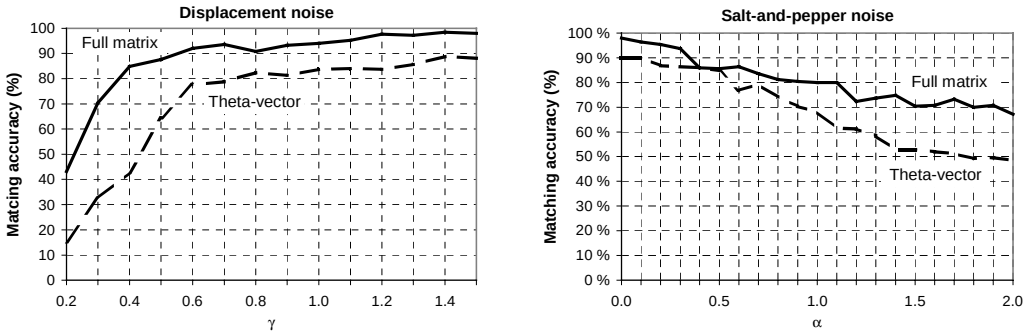

Fig. 9. Matching performance as a function of noise parameters

The noise consists of two separate components: *displacement* and *salt-and-pepper* noise. The noise is modeled as the probability function for a pixel to change its color to the opposite value. In the salt-and-pepper noise, the probability is constant. In the displacement noise, the probability is a negative exponential function of

A. The distance to the closest pixel of the opposite color.
B. The distance to an open area of a given size of the opposite color.

In other words, the closer a pixel is to an edge, the higher is the probability of noise. The probability function for background pixels (bg) is

$$P(fg|bg) = \alpha_{bg} + \beta_{bg} e^{-\gamma_{bg} \times (BtoF + BtoFBlock/4)} , \qquad (9)$$

where α_{bg} is a parameter of the salt-and-pepper noise; β_{bg} and γ_{bg} are the parameters of the displacement noise; $BtoF$ and $BtoFBlock$ are the distances to the closest pixel of the opposite color, and the distances to the open area. The corresponding equation for the foreground pixels (fg) is

$$P(bg|fg) = \alpha_{fg} + \beta_{fg} e^{-\gamma_{fg} \times (FtoB + FtoBBlock/5)} . \qquad (10)$$

The effect of the noise parameters is illustrated in Fig. 8, and the matching performance is summarized in Fig. 9. Note that the values of α are presented as percentages. The matching process has been repeated ten times for each image, and the reported results are the averages of the ten runs. These results indicate that the full matrix version is the better choice of the two. The matching accuracy is surprisingly good with the presence of the displacement noise. Problems arise only with the highest noise values ($\gamma = 0.2$), but this would be hard even for a human observer. In the case of the salt-and-pepper noise, the classification accuracy decreases faster than expected. This is not necessarily a problem in practice because this kind of noise can be removed quite efficiently with a median filtering technique.

More detailed examination reveals that there are two groups of symbols that tend to fail more often than the others. The first group includes the symbols *ground*, *capacitor*, and *battery*, which all contain lines parallel to each other. The second problematic group includes the symbols *voltage meter* and *current meter*. The shapes of these two are quite similar, and the main difference is their orientation. At the same time, the matching is designed to be rotation invariant, which turns out not necessarily to be a good thing in this particular case.

The dimensions of the parameter space are selected to maximize the overall matching accuracy. In the case of the full matrix variant, the dimensions are $q_\rho = 32$ and $q_\theta = 32$, and 40 maxima are preserved. Moderate changes in the parameter set-up did not have a significant effect on the matching accuracy. The space requirement of the feature vectors would be 50 bytes/vector using 5 bits/maxima. In the θ-vector approach, the length of a feature vector is 64, which requires 128 bytes of storage space.

The run times of the matching process with an Intel Celeron 400 MHz and a Linux 2.2 environment are summarized in Table 1. In *on-line matching*, feature extraction must also be performed for the database images. Thus, the overall run time consists of $1 + 25$ feature extractions and 25 matching procedures, summing up to 3.3 and 2.3 s in the case of the θ-vector and full-matrix approaches. In *off-line matching*, the database images have been indexed beforehand, and only one feature extraction is required in the matching process. The corresponding run times are approximately 0.2 and 0.5 s for the θ-vector and full-matrix approaches.

6 Conclusions

We have introduced two methods for content-based matching of line-drawing images using the HT. The methods were shown to work well in the case of small images. They are capable of reliably detecting the given query images and tolerating noise and errors in orientation of lines. The methods are invariant in scaling, translation, and rotation, and the matching is relatively fast.

Additional experiments, however, revealed that the methods cannot be used reliably for very large and complex images because these images have many more lines with a more uniform distribution in the feature space. Further improvements are therefore needed if the method is to be used in image retrieval. However, the method can be applied as such for on-line matching of smaller images, and for finding similar objects from a larger image. In this case, the method should be integrated with a suitable image-segmentation algorithm.

Considering the speed, feature extraction is currently the bottleneck of the algorithms. This can be a serious limitation if the features are extracted in real time, the database is very large, or the method is integrated with a more complex image segmentation algorithm. However,

the feature extraction could be speeded up quite easily by a factor of about 10 to 100 times with a variant of the *randomized Hough transform (RHT)* [13]. Effectively, the matching procedure would become the most time-consuming part of the algorithm. The use of the RHT is expected to cause only a minor deterioration in the quality. These matters are points for future studies.

References

1. E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, J.S.B. Mitchell: An efficiently computable metric for comparing polygonal shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 13:3, 209–226 (1991)
2. Y. Chen, A. Langrana, A.K. Das: Perfecting vectorized mechanical drawings. Computer Vision and Image Understanding 63:2, 273–286 (1996)
3. J.Y. Chiang, S.C. Tue, Y.C. Leu: A new algorithm for line image vectorizing. Pattern Recognition 31:10, 1541–1549 (1998)
4. M.S. Costa, L.G. Shapiro: Scene analysis using appearance-based models and relational indexing. Proceedings of the IEEE International Symposium on Computer Vision. Coral Gables, FL: November 1995, pp 103–108
5. P.M. Devaux, D.B. Lysak, R.A. Kasturi: A complete system for the intelligent interpretation of engineering drawings. International Journal on Document Analysis and Recognition 2:2/3, 120–131 (1999)
6. P. Fränti, A. Mednonogov, H. Kälviäinen: Content-based retrieval of line-drawing images using angular information of Hough transform. GraphiCon'2000, Moscow, Russia, August 2000, pp. 51–55
7. P. Fränti, A. Mednonogov, H. Kälviäinen: Hough transform for rotation invariant matching of line-drawing images. International Conference on Pattern Recognition (ICPR′00), Barcelona, Spain, September 2000, pp 379–382, vol. 4
8. B. Furth, S.W. Smoliar, H.J. Zhang: Image and Video Processing in Multimedia Systems. Boston: Kluwer Academic 1995
9. P.C.V. Hough: Methods and means for recognizing complex patterns. U.S. Patent 3,069,654, 1962
10. M.K. Hu: Visual pattern recognition by moment invariants. In: J.K. Aggarval, R.O. Duda, A. Rosenfeld (eds) Computer Methods in Image Analysis. Los Angeles: IEEE Computer Society 1977
11. B. Huet, E.R. Hancock: Relational histograms for shape indexing. Proceedings of the IEEE International Conference on Computer Vision. Bombay, India: January 1998, pp 563–569 (1998)
12. B. Huet, E.R. Hancock: Shape recognition from large image libraries by inexact graph matching. Pattern Recognition Letters 20:11–13, 1259–1269 (1999)
13. H. Kälviäinen , P. Hirvonen, L. Xu, E. Oja: Probabilistic, non-probabilistic Hough transforms: overview and comparisons. Image and Vision Computing 13:4, 239–251 (1995)
14. M.S. Kankanhalli, B.M. Mehtre, H.Y. Huang: Color and spatial feature for content-based image retrieval. Pattern Recognition Letters 20:1, 109–118 (1999)
15. A.A. Kassim, T. Tan, K.H. Tan: A comparative study of efficient generalised Hough transform techniques. Image and Vision Computing 17:10, 737–748 (1999)
16. H. Kauppinen, T. Seppänen, M. Pietikäinen: An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 17:2, 201–207 (1995)
17. V.F. Leavers: Survey: which Hough transform. CVGIP Image Understanding 58:2, 250–264 (1993)
18. W.-Y. Ma, H.J. Zhang: Content-based image indexing and retrieval. In: B. Fuhrt: (ed) Handbook of Multimedia Computing, Boca Raton, FL: CRC Press 1999, pp 227-254
19. B.S. Manjunath, W.Y. MA: Browsing large satellite and aerial photographs. IEEE International Conference on Image Processing, Vol. 2. Lausanne, Switzerland, September 1996, pp 765–768
20. R. Mehrotra, J.E. Gary: Similar-shape retrieval in shape data management. Computer 28:9, 57–62 (1995)
21. J.R. Parker: Algorithms for image processing and computer vision. New York: John Wiley 1996
22. E. Persoon, K. Fu: Shape discrimination using Fourier descriptors. IEEE Transactions on Systems, Man and Cybernetics 7:3, 170–179 (1977)
23. Y. Rui, T.S. Huang, S.-F. Chang: Image retrieval: current techniques, promising directions and open issues. Journal of Visual Communications and Image Representation 10:1, 39–62 (1999)
24. P. Vaxiviere, K. Tombre: Celesstin: CAD Conversion of mechanical drawings. Computer 25:7, 46–54 (1992)
25. L. Wenyin, D. Dori: From raster to vectors: extracting visual information from line drawings. Pattern Analysis & Applications 2:1, 10–21 (1999)
26. I.H. Witten, A. Moffat, T.C. Bell: Managing gigabytes: compressing and indexing documents and images (2nd edn). New York: Van Nostrand Reinhold 1999