Machine Vision and Applications (1996) 9:73–86

Machine Vision and
Applications

Applying algebraic and differential invariants for logo recognition

D. Doermann[1], **E. Rivlin**[1,2], **I. Weiss**[1]

[1]Document Processing Group, Center for Automation Research, University of Maryland, College Park, MD 20742-3275, USA
[2] Department of Computer Science, Technion, Israel Institute of Technology, Haifa, 32000, Israel

Abstract. The problem of logo recognition is of great interest in the document domain, especially for document databases. By recognizing the logo we obtain semantic information about the document which may be useful in deciding whether or not to analyze the textual components. Given a logo block candidate from a document image and a *logo* database, we would like to determine whether the region corresponds to a logo in the database. Similarly, if we are given a logo block candidate and a *document* database, we wish to determine whether there are any documents in the database of similar origin. Both problems require indexing into a possibly large model space.

In this contribution, we present a novel application of algebraic and differential invariants to the problem of logo recognition. By using invariants we have shape descriptors for matching that are unique and independent of the point of view. The algebraic invariants handle cases in which the whole shape of the logo is given and it is easy to describe. The differential invariants cover complex arbitrary logo shape and handle situations in which only part of the logo is recovered.

We outline a hierarchical approach to logo recognition and define methods for page segmentation, feature extraction, and indexing. We demonstrate our approach and present results on a database of approximately 100 logos.

Key words: Document understanding – Logos – Document databases – Algebraic and differential invariants – Application

1 Introduction

Logos typically appear as mixed text and graphic icons which, when recognized, trigger an association of the object to which they are attached, with a given group or organization. In the document domain, logos are a valuable device for identifying the source of a document. By recognizing the logo we can establish a link to a specific organization or publication, and we may be able to make a high-level decision as to whether the information in the document is of potential interest and should be further analyzed.

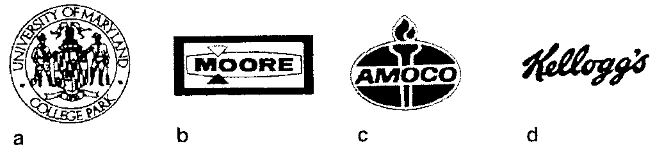


Fig. 1a–d. Examples from the logo database

In the document domain two analogous logo recognition tasks are of interest. First, given a document which contains a logo, classify the logo as one of a finite set of known logos in a logo database or conclude that the logo is not present in the database. Second, given a representative logo (known or unknown), index into a database of documents and extract all documents which contain that logo. Both of these problems can be viewed as indexing into a possibly large database (in one case logos and in the other documents) based on features found in candidate logo regions.

Several examples of logos are shown in Fig. 1. Logos typically consist of an iconic or graphic portion and possibly some associated text. The iconic region may be as simple as a combination of geometric shapes or as complex as a line drawing.

In some cases a logo is composed of artistic text or text arranged in a unique configuration. These text components may correspond to either the name or an acronym of the organization, and are thus a key which can be used to immediately index into a database of known organizations. The extraction of such keys is an important step in reducing the number of possible matches.

Correspondence to: D. Doermann

1.1 Previous work

1.1.1 Logo recognition

The problem of logo recognition has received very little attention in the literature, but some publications which deal with the tasks of Chinese seal identification and extraction of text from trademarks have appeared.

In Oriental cultures *seals* are used to identify the sources of documents, paintings and related objects in much the same way as a signature is used in Western cultures. As in many domains, the pattern may be unstable due to noise in the imprints even from a single seal. Ueda and Nakamura (1984) describe a system which attempts to align and verify round seals with limited success. Fan and Tsai (1984) use a distance-weighted correlation on aligned skeletal representations of the seals to perform recognition. Their system is restricted to square seals for alignment and cannot handle occlusion. Lee and Kim (1989) apply graph matching to the thinned seal under the assumption that the topology of the seal is unique and remains constant. In a domain containing only strokes such an approach may be valid. In the logo domain a much wider diversity of patterns can appear.

In the trademark domain Brossman and Cross (1990) describe the initial stages of the trademark reasoning and retrieval system, which attempts to reason about the similarity of trademarks for possible trademark infringement. The portion of the system which they describe attempts to locate and recognize characters which are embedded in the artwork. They attempt to find characters which share strokes or characters that are composed of stylized non-traditional stroke features, such as bird wings. The difficulty is that many patterns can *appear as characters*, even if the association is never intended.

1.1.2 Algebraic and differential invariants

The subject of viewpoint invariants in vision has developed rapidly in recent years. A simple projective, or viewpoint, invariant, namely the cross-ratio of four points on a line, was introduced in vision by Duda and Hart (1973). However, its domain of applicability was very limited. More general invariants were studied in the nineteenth century, and were introduced in the field of computer vision by Weiss (1988). They are of two main types: (a) Algebraic invariants based on a global description of the shapes by algebraic entities such as lines, conics and polynomials. Details of these methods can be found in Grace and Young (1903) and Springer (1964). (b) Differential invariants are based on describing the shape by arbitrary differentiable functions. These methods were developed by Halphen (1880), Wilczynski (1906), Cartan (1955), and Lane (1942).

These methods have been applied to various vision problems. The algebraic approach was used by Forsyth et al. (1990) and Taubin and Cooper (1992) while differential invariants were used by Weiss (1988, 1992) and Bruckstein and Natravali (1990). Each method proved to have advantages and disadvantages. The algebraic method, while simple and easy to implement, is quite limited in the kinds of shapes that it can handle because most shapes are not representable by simple low order polynomials. The differential method is more general because it can handle arbitrary curves, but it relies on the use of local information such as derivatives (of quite high orders).

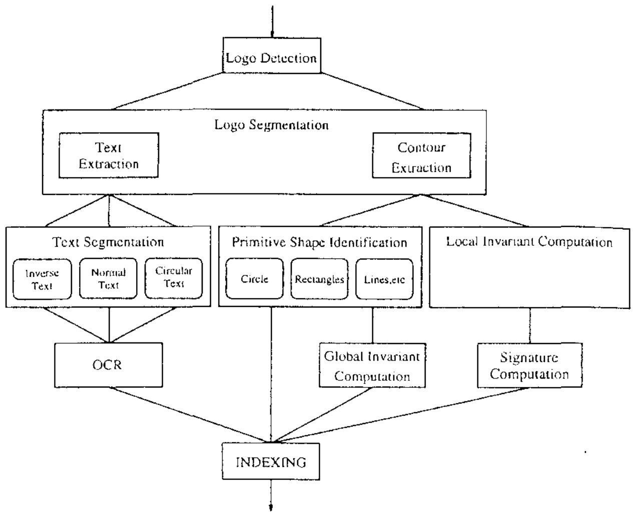


Fig. 2. Overview of the logo recognition system. *Rectangular boxes* processes

This situation has led to the introduction of various kinds of intermediate, or hybrid, methods that try to combine the advantages of the algebraic and differential methods but hopefully not their disadvantages. Van Gool et al. (1991), Barrett et al. (1991), and others introduced invariants that contain both derivatives and reference points. Each reference point reduces the number of derivatives that one needs in order to obtain invariants. Weiss (1992) used a "canonical" coordinate system without curve parameterization to obtain the same goal. This resulted in fewer derivatives and in the capability of using feature lines in addition to points. However, in all these methods the correspondence must be established between the reference points of the two images that are being matched.

1.2 The approach

This paper describes the major components in our approach for logo recognition. An overview of the approach is presented in Fig. 2. To address the logo recognition problem, several tasks must be considered, including: (a) detection and extraction of candidate logo regions from the document image, (b) segmentation and extraction of meaningful features for classification, and (c) indexing into a large database of logos or a database of documents which may contain logos. More importantly, these tasks must be accomplished quickly and in the presence of geometric transformation, noise, and possible occlusion. Our approach is based on a high-level process which attempts to match a candidate logo by applying a set of processes to prune the database using different features at the different stages of the approach. The use of invariants is the major part of the process of handling shape. In parallel we try to extract text from the logo and proceed with OCR.

In the first phase we try to select the logo in the page and separate it from other graphic components. Section 2 describes how logo block candidates are extracted from a document image using a page segmentation algorithm. Section 3 discusses the segmentation of the logo into text and graphic features and describes the algorithms for text labeling and contour extraction. We extract text and the presence of simple geometric shapes, such as circles and lines. The process of extraction of the indexing features is described in Sect. 4. To obtain index features, we apply OCR to the text (Sect. 4.1) and attempt to extract primitive shapes from the contours (Sect. 4.2).

Depending on the uniqueness of the text and geometric features, a more refined match may be necessary based on the iconic components of the logo. We attempt to refine the match using several *geometric invariants* of contour and shape features. Invariants are used rather than the contours themselves to speed up searching and matching queries in databases. Section 4.3.1 describes how algebraic invariants are used for rapid pruning, and local, differential invariants are used to describe complex shapes and lessen the effects of occlusion. Section 4 outlines the computation of the invariant signatures which are used for matching.

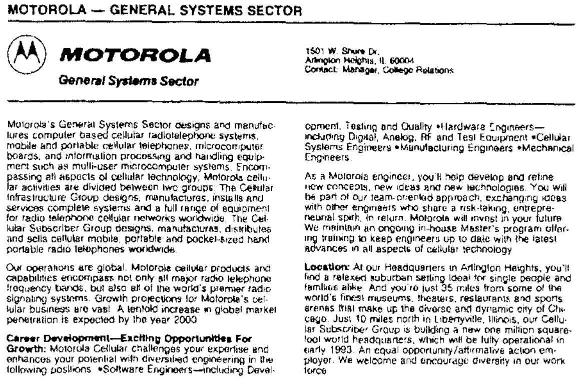
To demonstrate our approach we have produced a database of over 100 logos which are representative of logos found on business correspondence. Section 5 describes the organization of the database and the indexing techniques used. Section 6 presents experimental results, and Sect. 7 provides an overview of the approach and conclusions.

2 Logo detection

Logos appear on documents as "advertisements" for the companies or publications which they represent. A logo must be a unique, perceptually salient trademark that can be quickly recognized by the reader. For this reason logos are typically placed in a prominent position on the document, are larger then mainline text, and are smaller then other graphics or figures. Logos are typically confined to compact regions and are located in isolated portions of the document, and are not embedded in the text structures as are other graphic or iconic figures. Thus, we can separate a logo candidate from other graphic components by its position on the page and by using knowledge of the document type (e.g. memo or letter vs. report) if this is known. *Logo block location hypotheses* are generated as the output of a page decomposition module.

The general goal of document page decomposition is to segment the page into meaningful components based on the physical attributes of the region. In simple documents the text, graphics and half-tones are examples of three component classes which can be extracted with relatively high confidence. Regions are classified by their size, position, and distribution of components.

We have found that with detailed analysis of a region a finer classification can be obtained. Our system attempts to segment an image into the main classes of text/graphics/halftones and then attach additional attributes to each region. In the case of text, we can distinguish between handwritten and machine-printed text blocks by measuring variance in character height. For machine printed text we attempt to identify the font, point size, and other attributes such as style (bold, italic, etc.). For graphic regions we attempt to achieve a classification into tables, charts and graphs, rule lines, and logo/icons.


MOTOROLA — GENERAL SYSTEMS SECTOR

MOTOROLA
General Systems Sector

1501 W. Shure Dr.
Arlington Heights, IL 60004
Contact: Manager, College Relations

Motorola's General Systems Sector designs and manufactures computer based cellular radiotelephone systems, mobile and portable cellular telephones, microcomputer boards, and information processing and handling equipment such as multi-user microcomputer systems. Encompassing all aspects of cellular technology, Motorola cellular activities are divided between two groups: The Cellular Infrastructure Group designs, manufactures, installs and services complete systems and a full range of equipment for radio telephone cellular networks worldwide. The Cellular Subscriber Group designs, manufactures, distributes and sells cellular mobile, portable and pocket-sized hand portable radio telephones worldwide.

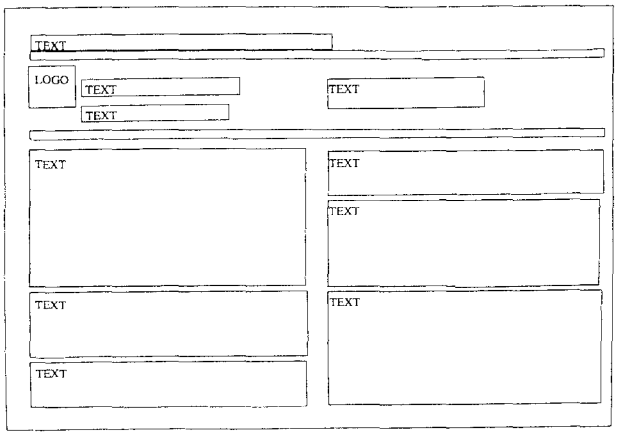
Our operations are global. Motorola cellular products and capabilities encompass not only all major radio telephone frequency bands, but also all of the world's premier radio signaling systems. Growth projections for Motorola's cellular business are vast. A tenfold increase in global market penetration is expected by the year 2000

Career Development—Exciting Opportunities For Growth: Motorola Cellular challenges your expertise and enhances your potential with diversified engineering in the following positions •Software Engineers—including Development, Testing and Quality •Hardware Engineers—including Digital, Analog, RF and Test Equipment •Cellular Systems Engineers •Manufacturing Engineers •Mechanical Engineers.

As a Motorola engineer, you'll help develop and refine new concepts, new ideas and new technologies. You will be part of our team-oriented approach, exchanging ideas with other engineers who share a risk-taking, entrepreneurial spirit. In return, Motorola will invest in your future. We maintain an ongoing in-house Master's program offering training to keep engineers up to date with the latest advances in all aspects of cellular technology

Location: At our Headquarters in Arlington Heights, you'll find a relaxed suburban setting ideal for single people and families alike. And you're just 35 miles from some of the world's finest museums, theaters, restaurants and sports arenas that make up the diverse and dynamic city of Chicago. Just 10 miles north in Libertyville, Illinois, our Cellular Subscriber Group is building a new one million square-foot world headquarters, which will be fully operational in early 1993. An equal opportunity/affirmative action employer. We welcome and encourage diversity in our work force

a



b

Fig. 3a,b. Results of page decomposition. A single logo candidate region is generated

The page segmentation is based on the approach described by Etemad et al. (1995). The algorithm implements a layout-independent document image segmentation scheme in which text, image, and graphics regions in a document image are treated as three different "texture" classes. Feature vectors based on multiscale wavelet packet representation are used for local classification. Segmentation is performed by propagating soft local decisions made on small windows across neighboring blocks and integrating them to reduce their "ambiguities" and increase their "confidence" as more contextual evidence is obtained from the image data. Local votes propagate in a neighborhood, within and across scales, and majorities of weighted votes give the final decisions.

In our experiments, small graphic regions (smaller than 750×750 pixels at 300 dpi) are considered. Such blocks (and all text which falls within it) are is extracted as logo -candidates. Figure 3 shows an example of the results of document segmentation. A single logo candidate region, corresponding to the Motorola logo, was identified.

3 Logo segmentation

In this phase in our approach we attempt to segment the logo into text and graphics features. Basically the segmentation is divided into two parts, text extraction and contour extraction. If textual components can be extracted, the recognized strings are used to index into a keyword index. In the absence of text or when text does not provide sufficient constraint, contour features are used to compute shape signatures and are qualitatively matched against contour signatures from the database.

3.1 Text extraction and text segmentation

The inclusion of text in logo or trademark designs is common, because it provides an immediate association with the acronym or name of the company or publication which the logo represents. Since the logo itself is typically artistic in nature, the text is often found printed in a fancy script, inverted on a dark background, or lying on a circular arc. In the simplest case we find text regions below or horizontally adjacent to the logo. If these regions are not part of the initial logo block hypothesis, as in Fig. 3, they can be analyzed for relevance to the logo region. A more difficult task is extracting text which is embedded within the logo boundaries.

In our system, we attempt to isolate (a) standard text (including banner text having size larger then 14 pts), (b) inverse text, and (c) text which lies on a circular arc. Other configurations in which text occurs, but which are not considered here, include script, tapered text, which is fit within geometric boundaries, such as ellipses, and text composed of features which appear as part of the icon. Even a partial match, which identifies selected characters, can be a valuable indexing tool. The remainder of this section describes our approach to text extraction.

The original documents are scanned at 300 dpi and stored as 8-bit gray-scale images. Since we are interested only in a first estimate of the underlying text components, the logo block image is thresholded by the following procedure: (a) compute the gradient of the image using the Sobel operator. (b) suppress pixels which have non-maximal gradient magnitude in the gradient direction, and (c) use the mean gray level of the remaining edge candidate pixels as a threshold. We then extract connected components of the thresholded image and compute properties of each component, including its bounding box, height, width, centroid, and component area/box area ratio. Using the histogram of component heights, we group components which are approximately the same height. Those components which correspond to text must also satisfy collinearity properties (Fletcher and Kasturi 1988) as described below. We have found that text which is all of the same height (e.g., block caps) is more common then mixed upper/lower case text (Fig. 1). This provides subgroups from which we attempt to identify *phrases*.

Starting with the largest bin, we apply a Hough transform on the centroids of the bounding boxes to detect linear components and a Hough transform to detect circular components. A threshold of four components is used to filter out components which align by chance. If text candidates are found, other components of similar sizes are checked for inclusion in the phrase, and in the case of circular text hypothesis the position of the center of the arc is checked to determine whether it is in a reasonable position near the center of the logo region. After a line or arc is identified, other components, which may have been missed because of size or association with other components, are included by relaxing the size requirements. Figure 4 shows text extracted from three logos in the database. (Only simple block text is currently detected; merged, tapered, or script text is not yet considered.)

Components which have a background/foreground ratio greater than 2:1 are marked as candidates for reversed text. To detect reverse text a background connected component algorithm is applied to the region, and the components are grouped as above.

3.2 Contour extraction

After text components are extracted, the graphic components remain, along with banner or complex text components which could not be classified by the above approach. The contour features are used to extract primitive shapes (Sect. 4.2), and to compute shape signatures (Sects. 4.3.1, 4.3.2) to further refine the match in the database.

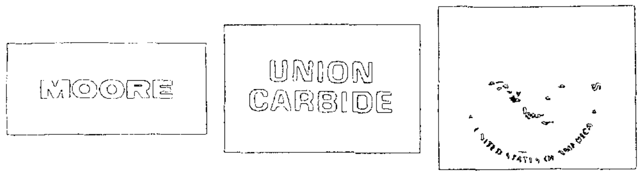
As described in the previous section, edge candidates for these remaining graphic components are identified by areas of high gradient activity. A Sobel operator is applied to the gray level image. From the resulting gradient images *step edges* are computed. These edges are points of locally maximal gradient magnitude; they are recorded with edge angle and location information.

The edges are grouped into chains by starting with edges having high gradient magnitudes and iteratively extending their end points to include neighboring edges. The decision on whether to include a given edge is based on the smoothness between the two edges. A similar definition of smoothness for strokes is given by Doermann and Rosenfeld (1993). *Smoothness* is a local measure of the confidence that a given pair of edges k and l are portions of the same contour. In general, we define smoothness, Ψ, for a pair of adjacent edges k and l with n properties as

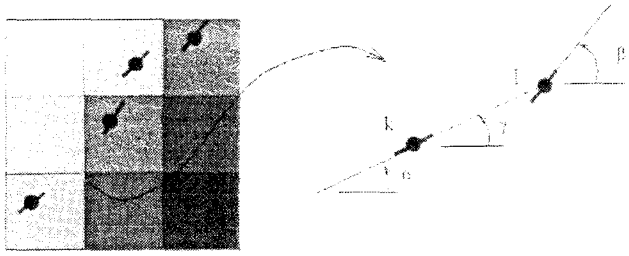
$$\Psi(k,l) = F(\mathbf{w}, \mathbf{P}_k, \mathbf{P}_l) \qquad \mathbf{P} = (p_1, p_2, \ldots, p_n)$$
$$= \bigodot_{m=1}^{n} \omega_m f_m(p_{km}, p_{lm}) \tag{1}$$

where $\mathbf{P}_k$ and $\mathbf{P}_l$ are property vectors computed from the edges k and l, the f_m's are smoothness functions derived for each pair of properties, and ω is a weight vector. The $\odot$ operator is a (possibly nonlinear) combination of the smoothness parameters. The smoothness computation is based on perceptual organization criteria involving the position, curvature and feature consistency of the two edges.

The properties can be computed as follows: Recall that for each edge, its position and orientation are computed (Fig. 5). For any pair of edge pixels, k and l, the following properties can be computed:



4a 4b 4c



5

Fig. 4a–c. The first set of text extracted. **c** The remainder of the text is in the second cluster which has slightly larger height. Recognition is performed on slightly expanded regions in the original image

Fig. 5. Smoothness parameters. Each edge point has a location and orientation. α, Orientation of the first edge; β, orientation of the second edge; γ, angle of the line segment connecting the two centers

$$
\begin{aligned}
\text{bend} &= \text{abs}(\beta - \alpha) \\
\text{total bend} &= \text{abs}(\beta - \gamma) + \text{abs}(\gamma - \alpha) \\
\Delta\text{bend} &= (\beta - \gamma) - (\gamma - \alpha) \\
\Delta X &= k_x - l_x \\
\Delta Y &= k_y - l_y \\
\|kl\| &= \sqrt{\Delta X^2 + \Delta Y^2}
\end{aligned} \tag{2}
$$

A number of different smoothness functions can be defined, as described by Doermann and Rosenfeld (1993). For example, the function used in the experiments reported here is

$$
\begin{aligned}
\Psi &= \odot(\omega, \text{bend}, \Delta\text{bend}) \\
&= 1 - [\omega_1 * \text{bend} + \omega_2 * \text{abs}(\Delta\text{bend})]
\end{aligned} \tag{3}
$$

which is maximum for a smooth transition between edges and favors the straightest path hypothesis. In our experiments the parameters ω_1 and ω_2 simply scale the smoothness measures to values between 0 and 1 (e.g. $\omega_1 = 1/\pi$ and $\omega_2 = 1/2\pi$), but in general, they can be used to weight the terms based on analytical experimentation.

Once chains are constructed, chains shorter than an empirically obtained length are removed and the chains are sorted by length. A simple logo tends to have a small number of long chains corresponding to the boundaries of the graphic components.

4 Extracting index features

The results obtained from the logo segmentation and feature extraction are passed to several processes. First, the text components resulting from text segmentation are passed to OCR engine to provide a fast pruning of the database (Sect. 4.1). Second, the nontext contours extracted from the logo are processed to attempt to identify primitive shapes including circles, conics, and lines which allow us to index directly into the database (Sect. 4.2) Finally, if there are still a large number of candidates, another process computes geometric invariants from the contours to obtain additional indexing features (Sect. 4.3).

4.1 OCR

The text components preprocessed and passed to an OCR system. Text which lies on a horizontal or vertical line is assumed to be upright. Text which is on a circle is cut component by component, and rotated accordingly. The image is passed to the Xerox ScanWorX Package for recognition.

4.2 Primitive shape identification

Lines and circles are common in logos; nevertheless, they provide a meaningful way of pruning the database. Obviously, more complex configurations of features can be used and will eventually uniquely identify the logo. It may not, however, be possible to extract them with the desired simplicity or reliability.

Given the set of contours obtained above, each contour is tested for circularity or linearity. To test for line segments each contour is divided into approximately collinear segments using Pavlidis' (1982) collinearity algorithm. The length of each linear segment is compared with the overall size of the logo. Segments which are long enough to correspond to a border or a significant feature are used to prune the database.

To test for circular arcs, a circle is fit to the contour points and the mean-squared error of the fit is evaluated. Both circular arcs and line segments are used to index into the database. Note, however, that size and position, or in the case of line segments orientation, cannot be used because of possible translation, rotation, and scaling of the original logo.

The next several sections focus on various issues involved in the application of geometric invariants in a logo recognition process. In view of the diversity of possible document styles we must deal with logos under a variety of transformations including translation, scale, rotation, and possibly skew. Rotation is less probable than the other transformations since a logo is likely to be oriented consistently with the text on the page. In the document domain output from transformations which distort the shape of a logo are recognized as different logos. In more general domains, such as a warehouse with a vision system which identifies logos on boxes, a logo may also undergo perspective transformations.

4.3 Invariant computations

Given an image of a logo, we want to extract a unique invariant of the logo so that given another image of the same logo, differing from the first by scale and rotation, for example, we obtain the same invariant. To do this we must eliminate the effects of the transformations that gave rise to the differences between the images.

There are several methods of eliminating transformations between images. The simplest way is to perform every possible transformation of one image and see whether any of the transformed versions matches the other image. For instance, in template matching (Ballard and Brown 1982) it is assumed that a template and an image differ only by translation, and the template is moved pixel by pixel over the image until a match is found. However, when more complicated transformations are involved, such as rotation or projection, the search space becomes overwhelmingly large.

To reduce the search space "invariant features" can be extracted. These are features in the image that stay invariant under some transformation and can be matched directly between the two images. For example, an edge remains an edge, and edges can therefore be used for matching. The problem here is that features such as edges are not distinctive. Any edge in one image can match any edge in the other. This leads to a *correspondence problem*, which can easily lead to a combinatorial explosion. Invariant constraints on edges (Grimson and Lozano-Pérez 1987) can be used, but they still leave a very large search space.

The correspondence problem can be solved by using more distinctive invariant descriptors, that is, descriptors that are invariant only to the transformation that we are interested in and not to others. For instance, a descriptor of logo A should be distinct from a descriptor of logo B, that is, it should not be invariant to a transformation that maps the shape of logo A into that of logo B. Edges are invariant to such a transformation since they can appear in both shapes. In other words, they are "too" invariant, that is, they are invariant to too wide a set of transformations. Thus, we must try to find features that are invariant only to the transformations that we want to eliminate and not to others, such that they are distinctive enough to match without ambiguity.

This contribution deals only with purely geometric invariants, that is, ones that can be calculated from the shape alone. Other logo properties, such as color, can also be considered as invariants in more general domains, subject to the same considerations as above, but they are beyond the scope of this paper.

The most desirable invariants for logo applications are those which are invariant to similarity transformations (translation, rotation, and scale). A simple example of an invariant to Euclidean transformations (translation, and rotation) can be described using the length of a line segment. In a simple document consisting only of very few line segments we can identify a particular segment by measuring its length on the image and comparing it to a database of lengths. The line's orientation is irrelevant and can be ignored. As another example, when a 2-D curve is rotated or translated, its curvature at each point does not change. Thus curvature is an invariant of the Euclidean transformations. It is common to plot the curvature of such a curve as a function of its arc length (which is invariant up to a choice of starting point; Guggenheimer 1963) to obtain a 2-D Euclidean invariant representation of the curve.

There are two main kinds of invariants: algebraic, or global, and differential, or local. The calculation of global invariants requires knowledge of the entire shape. Because the method is based on a global description of the shapes by algebraic entities such as lines, conics, and polynomials, it is simple and easy to implement, but, on the other hand, quite limited in the kinds of shapes that it can handle (because most shapes are not representable by simple low-order polynomials).

Differential invariants are based on describing the shape by arbitrary differentiable functions. As such the method is more general because it can handle arbitrary curves, but, on the other, it relies on the use of local information such as derivatives (of quite high orders). The differential approach avoids the occlusion problem by performing the calculation pointwise, in small neighborhoods around each point of a visible contour of the shape. Below we describe some global and local invariants under similarity transformations.

4.3.1 Global algebraic geometric invariants

We are interested in global shape descriptors that are invariant to similarity transformations. As mentioned above, the calculation of global invariants requires knowledge of the entire shape. For example, we need a whole contour to find the area, which is a Euclidean invariant. Moments and Fourier coefficients are also examples of global shape descriptors with some invariant properties. As an example, consider the moment of inertia. The (i, j)th moment of an image I is defined by

$$m_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

The moment of inertia of I around the origin m_0 is

$$\sum_x \sum_y (x^2 + y^2) I(x, y) = m_{20} + m_{02}$$

It is clear that m_0 is invariant under rotation of I about the origin, that if I is scaled by a factor of s, for example, m_0 is scaled by s^2. We can thus normalize I with respect to scale by requiring m_0 to be a constant. Similarly, a ratio of two moments that have the same value of $i + j$ (e.g., m_{10}/m_{01}) is invariant to scale.

Other possible shape descriptors may include roundedness and elongation, for example (for a survey of such descriptors see Rosenfeld and Kak 1982). Global invariants are relatively easy to calculate but they are sensitive to occlusion. That is, if part of the shape is missing from the image or is occluded by another object, we obtain a totally incorrect value for the descriptor.

Below we describe a set of possible global geometric invariants (under similarity) and then describe a method of obtaining local geometric invariants which avoids the occlusion problem.

Algebraic invariants of similarities. In this subsection we describe the similarity invariants of algebraic forms, such as points, lines and conics, and their combinations. The similarity transformation in the plane has four parameters: translation (in the x and y directions), rotation, and scaling. This means that in most cases we need to obtain more than four quantities from the image. Four quantities are needed in the

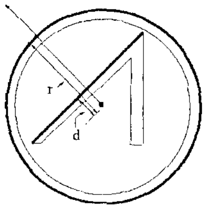


Fig. 6. A circle and a line: the distance-radius ratio $\frac{r}{d}$ is a global invariant

process of eliminating the similarity transformation, and the remaining quantities are invariant. This argument is modified for the case of a configuration in which the form is symmetric with respect to one of the transformation parameters, as we shall see below.

We can differentiate among the following cases

- *Two lines:* These have only four parameters and, by the general guideline above, we might expect no invariants. However, there exists one similarity invariant: the angle between the lines. The reason is that the two lines are symmetric (or invariant) with respect to scaling. If we magnify the two lines (keeping the origin at the intersection point), the magnified configuration is identical to the original one. Therefore we do not need to eliminate scaling in this case; we only need to eliminate the three Euclidean parameters. The angle between the two lines:

$$a_1x + a_2y = 1$$

$$b_1x + b_2y = 1$$

is calculated using the scalar product:

$$\frac{a_1b_1 + a_2b_2}{\sqrt{a_1^2 + a_2^2}\sqrt{b_1^2 + b_2^2}}$$

- *Three points:* With six parameters two of the angles of the triangle formed by these points are independent invariants. Equivalently, two of the ratios of the sides of the triangle are invariants.
- *A conic:* With five parameters a general conic has one similarity invariant, its eccentricity. However, a degenerate case such as the circle has only three parameters and no invariants.
- *A circle and a point:* This system has five parameters and one invariant, the ratio of the radius to the distance between the point and the circle's center. The distance of a point x_1, y_1 from a line $a_1x + a_2y = 1$ is given by:

$$\frac{a_1x_1 + a_2y_1}{\sqrt{a_1^2 + a_2^2}}$$

- *A circle and a line:* With five parameters there is one independent invariant: the ratio between the radius and the distance of the circle's center from the line. Figure 6 shows the major contour extracted from logo 25 and the parameters used to compute the global invariant.
- *Two circles:* Six parameters yield two independent invariants. One is the ratio between the radii; the other is the ratio between a radius and the distance between the two centers.

Other simple examples can be generated in a similar way. In addition, combinations of simpler configurations provide additional joint invariants. For example, if we add a line to a conic, we can easily find two invariants in addition to the eccentricity. These are formed by extending the (say) major axis of the conic to meet the line and looking at the appropriate angle and ratio of distances. Two conics can be treated similarly.

Global geometric invariants give us the ability to index more accurately into the database. In addition to the existence of the basic features which are used, they provide more information since these invariants characterize the relative positions of the basic features. As such, they are more meaningful as indexing features.

4.3.2 Local differential geometric invariants

Local invariants avoid the occlusion problem by performing the calculation pointwise, in small neighborhoods around each point of a visible contour of the shape. For example, it is quite common to plot the curvature of a contour with respect to the arclength, that is, the length along the contour from some starting point to some given point. Local invariants allows us to deal with "incomplete" queries, in which part of the information expected in the query is missing due to occlusion, but there is still enough information to retrieve the desired record. Denoting the curvature and arclength by κ and s respectively, we obtain a $\kappa(s)$ curve representing the visible contour. Both arclength and curvature are invariant to Euclidean transformation, and thus we obtain an "invariant signature" that can identify the curve. This curve can be stored in a database; it matches the signature of a similar logo (contour) presented as a query, even if the query logo is translated or rotated with respect to the logo originally stored in the database.

To obtain a local signature we use a shape descriptor which remains unchanged under similarity transformation and can be extracted from a partial contour representation of a logo. The result is a similarity invariant signature of the image which can be matched against the signatures extracted from a database of known logos.

Our method of obtaining local projective and affine invariants is described and illustrated by Rivlin and Weiss (1995). Being local, the invariants are much less sensitive to occlusion than are global invariants. Computation of the invariants is based on a normalization method. This consists of defining a canonical coordinate system in terms of intrinsic properties of the shape, independently of the given coordinate system. Since this canonical system is independent of the original one, it is invariant, and all quantities measured in it are invariant.

Our method is as follows:

1. We repeat the following steps for each pixel that belongs to the logo:
   - Define a window around the pixel and fit an implicit polynomial curve to it. We use a quartic:

$$\begin{aligned} f(x, y) &= a_0 + a_1x + a_2y + a_3x^2 + a_4xy \\ &\quad + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 \end{aligned}$$

$$+a_9y^3 + a_{10}x^4 + a_{11}x^3y + a_{12}x^2y^2$$
$$+a_{13}xy^3 + a_{14}y^4 = 0 \quad (4)$$

Once the curve order and window size have been chosen, the fitting itself can be done by standard methods. Simple least square fitting is quite ill conditioned because of the relatively large number of unknowns. The singular-value decomposition method is very successful in overcoming this problem, and we obtain a quite reliable fit.

The following stages are performed analytically.

- Derive a canonical, intrinsic coordinate system based invariantly on the properties of the shape itself, independently of the given coordinate system. By doing so we eliminate all the unknown quantities of the original system (e.g., the similarity parameters). To accomplish this: define an "auxiliary curve" which osculates the original fitted curve with a known order of contact. The canonical system is defined so that in it the osculating curve has a particularly simple, predetermined form.
- Transform the original fitted curve to this new system. Since the system is canonical, all shape descriptors defined in it are independent of the original coordinate system and are therefore invariants. Pick two invariants that are independent of the window size or the order of the fitted curve, and depend only on the shape itself.

2. We plot one invariant against the other to obtain an invariant signature curve.
3. If an invariant fit is needed, we repeat the previous steps, i.e., redo the curve fitting in the new canonical system, and iterate until convergence.

Below we describe the steps performed to obtain the canonical system and then use it to obtain the affine invariants.

Euclidean canonization. First we detail the Euclidean canonization stage. As a convention, we denote the new coordinates after each canonization step by $\bar{x}, \bar{y}$ and drop the bars before going to the next step, and similarly for other quantities.

The first step is translation, moving the origin to our curve point. Our initially chosen pixel x_0, y_0 does not necessarily lie on the fitted curve, but is close to it. Thus, we find a point x_0, y_0^* which does lie on the curve, i.e. we solve Eq. 4 for y_0^*, given x_0. This is easily accomplished using Newton's method because y_0 is a close initial guess. We then translate the origin to x_0, y_0^*. (We could simplify the solution by first translating so that $x_0 = 0$ and then solving for y_0^*.) We drop the star from y^*. We then transform the curve coefficients to the new system and obtain new $\bar{a}_i$. This is done by expressing the old coordinates in terms of the new, $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{x}_0$, substituting in Eq. 4 and rearranging. In this new system we have $\bar{a}_0 = 0$ which can be seen by simply substituting the point (0,0) in Eq. 1.

The next step is to rotate the coordinates so that the x axis is tangent to the curve. It is easy to see that in the rotated system we must have $\bar{a}_1 = 0$ because $df(x,y)/dx = 0$. To satisfy this condition we again express the old coordinates in terms of the new, with the rotation factor u_r:

$$x = \frac{\bar{x} + u_r\bar{y}}{(1 + u_r^2)^{1/2}} \qquad y = \frac{\bar{y} - u_r\bar{x}}{(1 + u_r^2)^{1/2}} \quad (5)$$

Now a_1 is transformed to

$\bar{a}_1 = a_1 - u_r a_2$

To make this term vanish we thus have to rotate by the amount

$u_r = a_1/a_2$

Since translation and rotation generate the Euclidean transformations, we have reached a Euclidean canonical system. All quantities defined in it are Euclidean invariants. The curvature at $\mathbf{x}_0$ is now simply the second derivative, d^2y/dx^2. The arclength is $|dx|$ since $dy = 0$.

Obtaining similarity invariants. For the Euclidean case we used the tangent to obtain a canonization process that met our requirements of invariance and locality. We can generalize this method by using an oscilating curve, which is a generalization of the tangent. A tangent is a line having at least two points in common with the curve in an infinitesimal neighborhood, i.e., two "points of contact." This can be expressed as a condition on the first derivative. Similarly, a higher order oscilating curve has more independent contact points, and the condition on the derivatives can be written as

$$\frac{d^k}{dt^k}(f^*(x,y) - f(x,y)) = 0, \qquad k = 0, \ldots, n \quad (6)$$

with f^* being the oscilating curve, f the given curve, and n the order of contact. Since the derivatives vanish, this condition is invariant to the parameter t. We will derive the oscilating curve without this parameter. Since it has a geometric interpretation in terms of points of contact, the condition is also projectively invariant, and since it is expressed as derivatives, it is also local. The derivatives are calculated analytically from f.

In the following we use an oscilating implicit curve f^* satisfying the above condition. This curve is chosen as the simplest one that meets our needs; its shape is thus known. Thus it is easier to handle than the original f, which can be any function that fits. According to our needs we find either a cubic or a conic which oscilates our fitted curve. We then transform the coordinates so that this cubic or conic takes on a particularly simple, predetermined form, i.e., we eliminate all its coefficients. In this new canonical system all quantities are invariants, and we pick the ones that best suit our needs.

Of the four parameters of the general similarity transformation we have already eliminated three by translation and rotation, so that our oscilating curve should have at least one coefficient, while passing through the origin and being tangent to the x axis. We also need two independent invariants. Therefore we choose a conic with three parameters:

$$f^* = c_0x^2 + c_1y^2 + c_2xy + y = 0 \quad (7)$$

To go further, we need to calculate the derivatives d^ny/dx^n of the fitted curve. This is done analytically from

$f(x, y)$. To do it we use the fact that all the total derivatives of f vanish, since f vanishes identically (Eq. 1). The first derivative, for example, is

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = 0$$

This is a linear equation for dy/dx. It is superfluous because we have already used its vanishing (tangency). However, each successive differentiation gives one linear equation for one higher $y^{(n)}$, in terms of lower derivatives.
Setting $a_2 = 1$ and denoting $d_n = \frac{1}{n!}\frac{d^n y}{dx^n}(0)$ we have:

$$d_2 = -a_3 \tag{8}$$

$$d_3 = -a_6 - d_2 a_4 \tag{9}$$

Given these derivatives we find the coefficients c_n of the conic:

$$c_0 = -d_2 \tag{10}$$

$$c_1 = \frac{d_2 d_4 - d_3^2}{d_2^3} \tag{11}$$

$$c_3 = -\frac{d_3}{d_2} \tag{12}$$

To eliminate the scaling we choose a scaled canonical system in which the sum of the coefficients in the new canonical system is 1, i.e. $c_0^2 + c_1^2 + c_3^2 = 1$. This is done by scaling the x, y coordinates by the factor:

$$s_{xy} = (c_0^2 + c_1^2 + c_2^2)^{1/2}$$

This scaling transforms the c_i to new ones:

$$\bar{c}_0 = \frac{c_0}{s_{xy}}, \qquad \bar{c}_1 = \frac{c_1}{s_{xy}} \qquad \bar{c}_2 = \frac{c_2}{s_{xy}}$$

These quantities are *local similarity invariants* because we have reached a similarity-invariant canonical system. We have used all possible similarity transformations (translation, rotation and scaling) to eliminate all the possible similarity transformation factors and arrive at the above form of the conic; thus the remaining independent coefficients are uniquely defined regardless of which system we started with. In Fig. 7 the logo on the left was processed, and the signature was extracted from the bounding curve of the internal shape (middle). The signature is on the right.

4.3.3 Scale space of invariants under similarity

Another robust method that we use for logo indexing is based on extraction of scale space of invariants. This method was developed by Bruckstein et al. (1995). It consists of defining an invariant arclength (using the lowest possible order of derivatives in given schemes) and then defining invariant finite differences using this arclength. These differences replace the higher order derivative in the traditional invariants. The differences are not necessarily small and do not tend to zero. Rather, their variable size creates the "scale space."

We briefly describe here an illustrative example of the method. Given a curve, with the Euclidean invariance in mind, we start from a point $P(\tau)$ on the curve, and we want to find invariants there. We choose an interval size Δ and find two points on the curve, $P^+(\tau+\Delta)$, $P^-(\tau-\Delta)$, located at distances $+\Delta$ and $-\Delta$ (measured on the curve) from the point $P(\tau)$ at which we want to calculate the invariants. Given these three points, we can calculate any Euclidean invariant involving them, such as the area $A(\tau)$ of the triangle formed by them. $A(\tau)$ is then a new type of invariant signature. This is much more robust than a derivative, if Δ is not too small. In this way we reduce the number of derivatives needed without needing any fixed reference points or their correspondence. The scale parameter Δ can now be varied to obtain a whole range of scale dependent invariants. In this way we obtain whole ranges of invariants at each point rather than single values. The signature functions for the curves then become signature vectors or even continua of values, i.e., surfaces or hypersurfaces. Matching them is robust because it is less sensitive to peculiarities that may exist at some fixed pre-set value of the locality (scale) parameters.

Below we give an example of an extraction of scale-space invariants under the similarity group of transformations from a logo.

As was pointed out by Bruckstein et al. (1990, 1993), the similarity invariant arclength parameter is given in this case by

$$d\tau = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t))^2 + (y'(t))^2}dt$$

After the curve is reparametrized by the invariant arclength we can call upon several types of scale-dependent similarity invariants. Here we plot the angle $[P(t-\Delta)\ P(t)P(t+\Delta)] = \varphi_\Delta(\tau)$ as a function of τ.

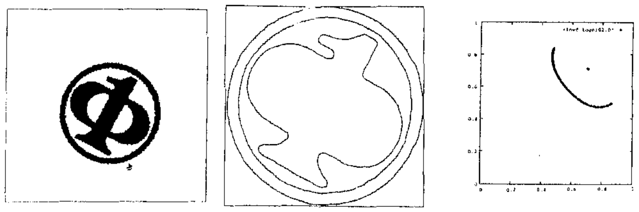
Figure 8 shows a logo from our database before and after transformation. The multi-valued signature for the curve which represents the letters *Kell* is presented in Fig. 9. The invariant arc length is represented by the horizontal axis which represents position along the reparametrized curve. The vertical-axis represents the values of the scale parameters Δ. In the example each image contains 20 different signatures for 20 different parameter values. For each signature different Δs were used. For a constant vertical value one obtains single-valued signatures for the curve. The gray level encodes the similarity invariant for a particular arclength and parameter value. The full display represents an "invariant signature surface." For each curve the starting position is marked by a white square. Due to the different starting position one multivalued signature is shifted relative to the other. To check for a match between two signatures one should match one multivalued signature to the other while shifting it in a cyclic manner. One can see that a match is achieved when one of the signatures is shifted.

Signatures for indexing. The invariant signature for each logo is simply the set of points in the space defined in Sect. 4.3. As described in the next section, indexing is accomplished by a pointwise matching procedure between signatures.

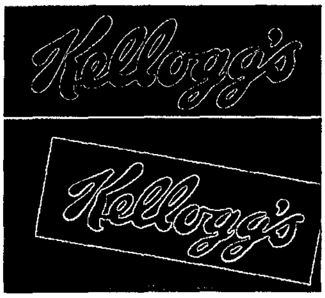
5 Recognition

5.1 Database organization

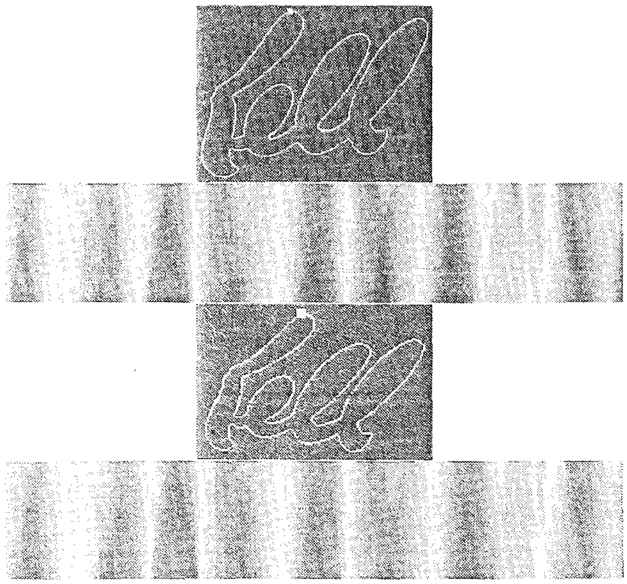
The database currently contains approximately 100 logos. Each entry in the database consists of the contour and text



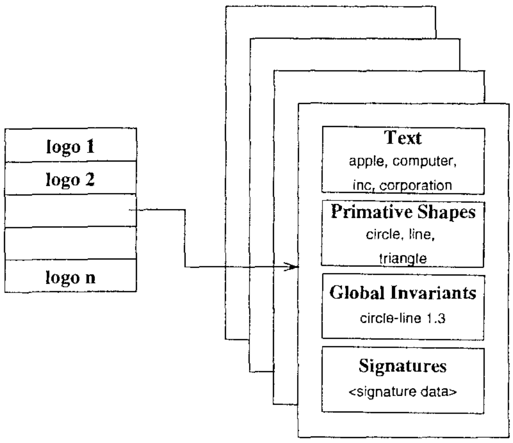
7a 7b 7c



8



9



10

Fig. 7a–c. A logo and its similarity invariant signature. **a** The logo was processed and the signature extracted from the bounding curve of the internal shape (**b**). **c** The signature

Fig. 9. The multivalued signature for each of the curves extracted from the logo (taken from the four letters *Kell*). The horizontal axis represents position along the reparametrized curve. Signatures for 20 different parameter values are displayed along the vertical axis. The starting position for each curve is marked (*white square*). A match is achieved when one of the signatures is shifted

Fig. 8. A logo before and after transformation (scaling and rotation) The logo was processed as five different curves

Fig. 10. Database organization

information from the original logo, as well as a set of key fields for associated attributes (see Fig. 10). The logo candidate is represented as a set of extracted contours, their signatures, and a set of text strings extracted from the logo. We have found that logos rarely differ in only minute details, so we can ignore many of the very small or fractalized contours, and consider only long, stable contours. Once features are extracted from the candidate logo, we prune the database by eliminating logos which do not contain the extracted features.

5.2 Indexing

Feature-based indexing into a database is difficult because of the possibility of missing the correct entry because of an error in feature extraction. Searching for basic features such as circles, long lines, and recognized text components minimizes the possibility of misidentifying features and attempting to index on features which do not exist in the original logo. Depending on the composition of the database, using basic features can reduce its size significantly before attempting to match signatures.

Since the database is assumed to be complete and accurate, if a logo candidate has a matching logo in the database, the features extracted from the candidate logo must be a subset of the features of the logo in the database. For example, if a candidate contains a circle feature, its match must contain at least one circular arc, and all logos which are known to contain no circle features are eliminated. The opposite, however, is not true since we cannot guarantee that we extract all features correctly from each candidate logo. Our goal is to avoid false positives at the expense of failing to identify meaningful features.

Since the task of logo indexing can follow any page decomposition process, we cannot assume a perfect segmentation. We must therefore assume that although a logo block hypothesis has been generated, portions may be missing, regions may be corrupted, or additional nonlogo components may be present. "Partial match" and "not present in the database" may be valid interpretations of the hypothesized logo region.

Our approach to matching consists of two stages. In the first stage we use simple shape properties and text indexing to prune the database. In the second stage we use geometric invariants (Weiss 1988) to generate a logo hypotheses. Both text and shape indexing are completed before the signatures are matched.

5.2.1 Text indexing

Recall that text regions are extracted and recognized in the original logo region, and only text which is recognized with a high confidence is passed on to avoid false positives. A dictionary entry is created a priori for each unique string which is associated with a logo in the database. Each text fragment or substring extracted from a candidate logo is matched against the dictionary. Although standard techniques exist to speed up substring matching, they often require nontrivial preprocessing of the dictionary (Knuth 1973).

For our system a brute force algorithm for string matching is sufficient given the number of items in the database. A substring match produces a list of all logos which contain it. When a logo candidate contains multiple substrings, say n of them, a hit percentage is also produced for each partial match. The hit percentage is computed for each logo as the percentage of the n substrings which match it. Clearly, the higher the hit percentage is, the more likely that we have identified the logo correctly.

There is currently no restriction on the read ordering of the strings or on the orientation of a matched substring in the original image.

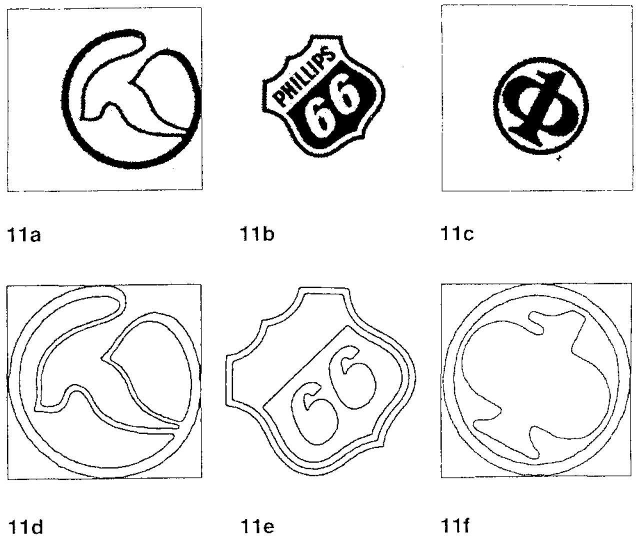
5.2.2 Shape indexing

For each shape feature computed (Sect. 4.2) the database contains either a logical or a numeric attribute field indicating the existence of the feature or its number of occurrences, respectively. If a logo candidate contains n occurrences of a given feature, then all logos which contain fewer than n occurrences are pruned.

Our current implementation deals with circular arcs, ellipses, rectangular blocks, triangles, and long line segments.

5.2.3 Global invariants

From the shape features we compute the appropriate global invariants as described in Sect. 4.3.1. For example, in Fig. 12c three global invariants can be extracted (two from the line/circle pairs and one from the circle/circle pair). These global invariants maintain the relative positioning of the shapes involved and are thus more powerful then local methods with respect to preserving spatial relationships.



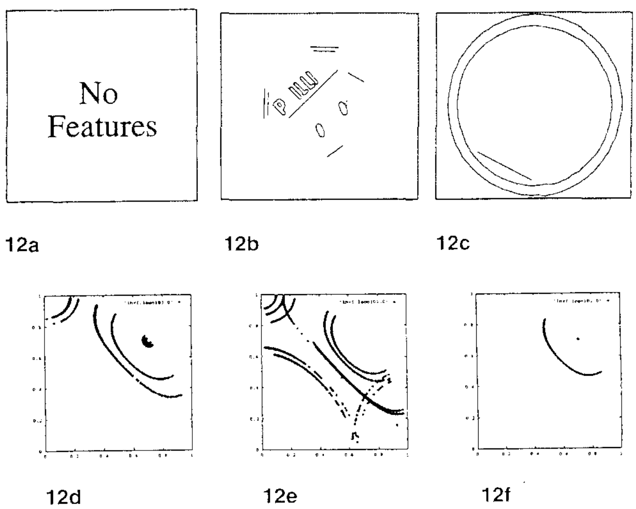


Fig. 11a–f. Three test logos and the "long" contours

Fig. 12a–f. The candidate text, line and circle features from the logos in Fig. 11 (a–c) and the resulting contour signatures (d–f)

5.2.4 Local and scale space invariants: signature matching

Having found some candidate logos we want to compare their signatures and verify a match. For each logo we obtain its local similarity invariant signature according to the procedure described in Sect. 4.3.2. Usually the logo contour is given as a set of contours, and the procedure is repeated for each contour segment. The signature for the logo is the collection of all the contour segments.

A method for automatic matching of the signatures was successfully used by Wolfson (1990) in the Euclidean case (curvature vs. arclength): draw a circle of radius ϵ around a point in one signature, and measure how much of the other signature enters inside that circle. This gives a measure of the local overlap between the two signatures, taking into account the noise level ϵ. Then move the circle along the signature and repeat the process for each point. Add up the local similarity measurements to obtain a global measure of similarity. We used a similar approach to compare two similarity signatures. The program is given the radius ϵ around the curve and computes the percentage of the pixels within the circle in the matched curve.

For the scale space signatures we check a match in the following method. The two signature surfaces are shifted horizontally with respect to each other by an unknown amount, due to the different starting point of the parameterization of the curves. We can deal with the problem in two ways. In the simple, but more expensive way, we shift one surface relative to the other by an increasing amount, trying to match the surfaces for each shift until the best match is found. In the second way, we eliminate the shift constant by differentiation (or differencing). If the surface is denoted as $z(x, y)$, the shift is along a constant y. For each constant y we can invert the function as $x(z, y)$, and take the derivative dx/dz. This eliminates any additive constant in the x direction. Plotting this derivative as a (possibly multivalued) function of z, we obtain a signature which is invariant to the horizontal shift.

6 Experiments and discussion

Experiments were conducted using rotated, scaled and translated logos from the database. (The database is available via ftp from documents.cfar.umd.edu.) For each logo the indexing for text, circles, lines, local invariant signatures, and global invariants were performed independently. In most cases the text, shape, and global invariant features reduced the search space significantly. (We attempted to match the signatures on all 100+ logos. Even for very large databases, the number of candidates should be only one or two dozens as a result of the pruning.) In any case, the results of the indexing can be combined to produce an ordered list of logos for verification.

Figure 11a–c shows logo candidate regions, and contours which are used to compute properties for indexing into the database (Fig. 11d–f). The results of text extraction, circle identification and line extraction are shown in Fig. 12a–c. The individual characters in the string can be matched, and we attempt to identify a logo which contains a superset of the symbols.

The extracted contours were used to produce local geometric invariants. The signatures of the three candidate logos are shown in Figures 12d–f. These signatures were matched to the database. In the cases where text or primitive shape information was present the size of our database resulted in only two or three candidates. From these candidates the signature matching picked the best match.

6.1 Local signature matching

To evaluate the power of signature matching more realistically (i.e., in a scenario where the pruned database contains on the order of 60–80 logos), we tested the system using *only* the local invariant signatures for matching. The three best matches for the signatures are presented in Fig. 13. The top candidates are fed to the scale space signature matching whose results are described in the next section.

Table 1. A set of ten randomly chosen logos, and their top four matches in the database

Trial	Actual logo	Match 1	Match 2	Match 3	Match 4
1	1	1	36	50	35
2	3	3	17	48	57
3	11	11	93	69	25
4	25	25	54	11	2
5	32	32	105	21	87
6	33	33	39	87	24
7	59	59	51	57	60
8	89	89	53	68	74
9	94	94	22	37	41
10	95	95	57	48	28

In all queries the correct logo was among the top three. Since the logo is given by the segmentation process as a set of contours, each contour is treated as an individual feature, and the spatial relationship between any pair of features in not preserved. The breaking therefore increases the number of false positives. Even when false positives are present, the signature computation prunes a large portion of the database and leaves us with a relatively small number of possible candidates.

6.2 Scale-space signature matching

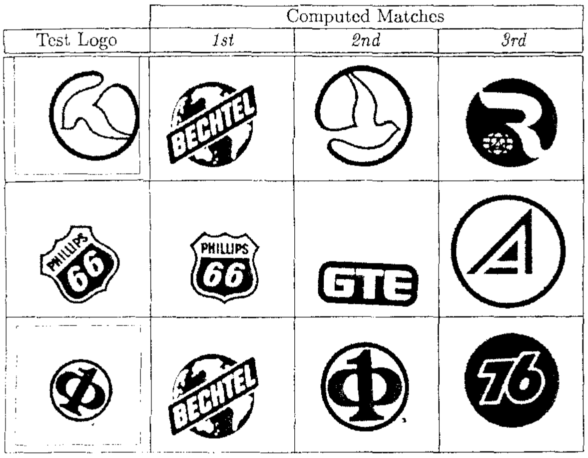
To illustrate the use of the scale-space signatures we took a logo from our database (Fig. 14) and obtained its multivalued signature before and after it went through scaling and rotation. As mentioned above, each image contains 20 different signatures for 20 different parameter values. For each signature different Δs were used. For a constant Y value one obtains single-valued signatures for the curve. The gray level encodes the similarity invariant for a particular arclength and parameter value. The results are presented in Figs. 15 and 16. The shape is symmetric, and only one of the two curves comprising it (the lower) was processed. The processed curve was itself symmetric. One can see the symmetry in the shape from the structure of the signature. A good match is achieved when one of the signatures is shifted (Fig. 17). In Fig. 18 we compared two scale-space signatures of the two logos presented. The signatures were checked for a match. No match was found.

Matching signatures in the scale space method is much more expensive than a checking for a match in the other method that we used. This is because it involves construction and matching of surfaces rather than curves. Table 1 shows the scale space matching results of ten additional logos.

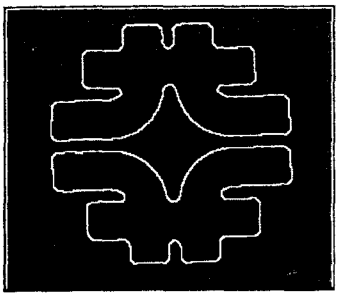
7 Conclusions

The problem of logo recognition is of great interest in the document domain, especially for document databases. The recognition process and subsequent association can be used to determine the need for further processing of a document. A similar situation can be imagined in a warehouse where decisions about storage can be made based on identification marks such as logos.

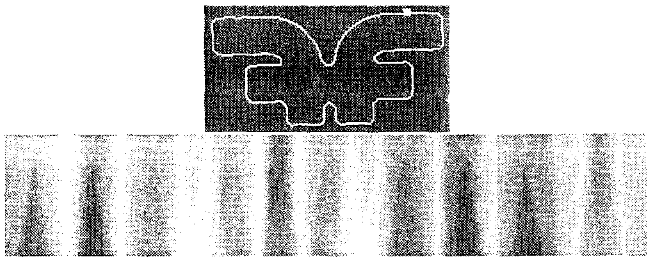
We have presented a hierarchal approach to logo recognition which uses text and contour features to prune the



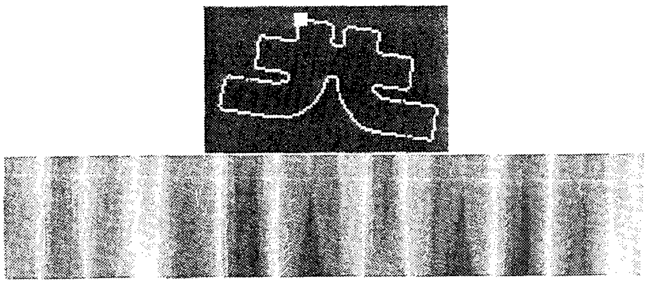
13



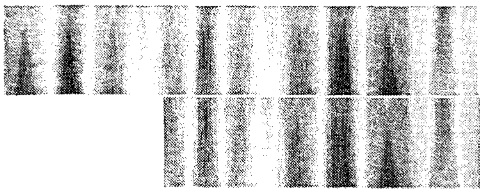
14



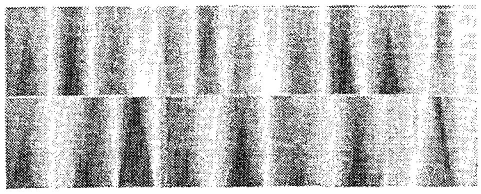
15



16



17



18

Fig. 13. Signature matching restuls for the three candidate logos in Fig. 11

Fig. 15. The lower curve and its signature. One can see the symmetry of the shape from the structure of the signature

Fig. 17. Shifting the lower curve's multivalued signature to the right gives the highest correlation value, and achieves a match

Fig. 14. An input logo from the database. Only the lower curve is processed

Fig. 16. The lower curve after scaling and rotation

Fig. 18. The multivalued signature from logo 27 was compared to the multivalued signature from the Kellogg's logo (logo 30) No match was achieved

database, and similarity invariants to obtain a more refined match. We have outlined the recognition process from page segmentation and feature extraction through index computation and matching. The recognition process was tested with a database of more than a hundred logos and very good results were obtained.

The strength of our approach lies in the fusion of approaches to matching. The combination of text, shape, and local and global invariants adds robustness to the recognition process. We have provided new techniques for signature computation and shown their value to dealing with matching in large databases.

Future work will include testing the system on a larger database and improving the local signature computation to include affine and projective invariants.

Acknowledgements. The support of this research by the Advanced Research Projects Agency (ARPA Order No. A550), under contract MDA 9049-3C-7217, is gratefully acknowledged.

References

1. Ballard D, Brown CM (1982) Computer vision. Prentice Hall, Englewood Cliffs
2. Barrett E, Payton P, Haag N, Brill M (1991) General methods for determining projective invariants in imagery. CVGIP:IU 53:45–65
3. Brossman C, Cross, G (1990) Model-based recognition of characters in trademark artwork. Pattern Recognition Letters 11:363–370
4. Bruckstein AM, Katzir A, Lindenbaum M, Porat M (1990) Similarity invariant recognition of partially occluded planar curves and shapes. Technion CIS report
5. Bruckstein A, Holt J, Netravali AN, Richardson TJ (1993) Invariant signatures for planar shape recognition under partial occlusion. CVGIP:IU 58:49–65

6. Bruckstein AM, Rivlin E, Weiss I (1995) Scale space local invariants. Technion CIS report
7. Bruckstein AM, Netravali AN (1990) On differential invariants of planar curves and the recognition of partially occluded planar shapes. AT&T technical memo
8. Cartan E (1955) La théorie des groupes continus et la géometrie. Oeuvres complètes, III/2, 1727–1861, Gauthier-Villars, Paris
9. Doermann D, Rosenfeld A (1993) The interpretation and recognition of interfering strokes. In: Proceedings of the international workshop on frontiers in handwriting recognition, pp 41–50
10. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
11. Etemad K, Doermann D, Chellappa R (1995) Multiscale document page segmentation using soft decision integration. IEEE Transactions on Pattern Analysis and Machine Intelligence (submitted)
12. Fan T, Tsai W (1984) Automatic Chinese seal identification. Computer Vision, Graphics and Image Processing 25:311–330
13. Fletcher L, Kasturi R (1988) A robust algorithm for text string separation from mixed text/graphics images. IEEE Transactions on Pattern Analysis and Machine Intelligence 10:910–918
14. Forsyth D, Mundy JL, Zisserman A, Brown CM (1990) Projectively invariant representations using implicit algebraic curves. Image and Vision Computing 8:130–136
15. Grace JH, Young A (1903) The algebra of invariants. Chelsea, New York
16. Grimson W, Lozano-Pérez T (1987) Localizing overlapping parts by searching the interpretation tree. IEEE Transactions on Pattern Analysis and Machine Intelligence 9:469–482
17. Guggenheimer H (1963) Differential geometry. Dover, New York
18. Halphen M (1880) Sur les invariants différentiels des courbes gauches. J. Ecole Polytechnique 28:1
19. Knuth D (1973) The art of computer programming: sorting and searching, vol 3. Addison-Wesley, Reading
20. Lane EP (1942) A treatise on projective differential geometry. University of Chicago Press
21. Lee S, Kim J (1989) Attributed stroke graph matching for seal imprint verification. Pattern Recognition Letters 9:137–145
22. Pavlidis T (1982) Algorithms for graphics and image processing. Computer Science Press, Rockville
23. Rivlin E, Weiss I (1995) Local invariants for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 17:226–238
24. Rosenfeld A, Kak AC (1982) Digital picture processing, 2nd edn. Academic, New York
25. Springer CE (1964) Geometry and analysis of projective spaces. Freeman, San Francisco
26. Taubin G, Cooper DB (1992) Object recognition based on moment (or algebraic) invariants. In: Mundy JL, Zisserman A (eds) Geometric invariance in machine vision. MIT Press, Cambridge, pp 375–397
27. Ueda K, Nakamura Y (1984) Automatic verification of seal-impression pattern. In: Proceedings of the international conference on pattern recognition, pp 1019–1021
28. Van Gool L, Kempenaers P, Oosterlinck A (1991) Recognition and semi-differential invariants. Proc CVPR 454–460
29. Weiss I (1996) Geometric invariants and object recognition. Int J Computer Vision (accepted)
30. Weiss I (1992) Local projective and affine invariants, TR 612. Center for Automation Research, University of Maryland
31. Weiss I (1988) Projective invariants of shapes. In: Proceedings of the DARPA image understanding workshop. Cambridge, pp 1125–1134
32. Wilczynski EJ (1906) Projective differential geometry of curves and ruled surfaces. Teubner, Leipzig
33. Wolfson HJ (1990) On curve matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 483–489

David S. Doermann received a B.Sc. degree in Computer Science and Mathematics from Bloomsburg Univewrsity in 1987. He studied in the Department of Computer Science at the University of Maryland, College Park, where he earned an M.Sc. degree in 1989. From 1987 to 1989 he worked as a Teaching Assistant at Maryland and from 1989 to 1993 was a researcher in the Computer Vision Laboratory. In 1993 he earned a Ph.D. with a thesis entitled "Document Image Understanding: Integrating Recovery and Interpretation." He is currently Assistant Research Scientist in the Center for Automation Research at the University of Maryland and head of the Document Processing Group. His research interests include intelligent document analysis, image processing, and video analysis.

Ehud Rivlin received B.Sc. and M.Sc. degrees in Computer Science, M.B.A degree from the Hebrew University in Jerusalem, and Ph.D from the University of Maryland. Currently he is Assistant Professor in the Computer Science Department at Technion, Israel Institute of Technology. His current research interests are in machine vision and robot navigation.

Isaac Weiss received his Ph.D. in physics from the Tel-Aviv University. He was subsequently a research scientist at New York University's Courant Institute of Mathematics and then at the Massachusetts Institute of Technology. He joined the Center for Automation Research of the University of Maryland in 1985. His current research interests are computer vision, image processing, pattern recognition, and robotics