Invariance Signatures: Characterizing contours by their departures from invariance

Squire, David; Caelli, Terry M.

Invariance Signatures: Characterizing contours by their departures from invariance

David McG. Squire

Computer Vision Group, Computer Science Department,
University of Geneva, Switzerland

E-mail: David.Squire@cui.unige.ch

and

Terry M. Caelli

School of Computing,
Curtin University of Technology, Australia

In this paper, a new invariant feature of two-dimensional contours is reported: the Invariance Signature. The Invariance Signature is a measure of the degree to which a contour is invariant under a variety of transformations, derived from the theory of Lie transformation groups. It is shown that the Invariance Signature is itself invariant under shift, rotation and scaling of the contour. Since it is derived from local properties of the contour, it is well-suited to a neural network implementation. It is shown that a Model-Based Neural Network (MBNN) [8, 37] can be constructed which computes the Invariance Signature of a contour, and classifies patterns on this basis. Experiments demonstrate that Invariance Signature networks can be employed successfully for shift-, rotation- and scale-invariant optical character recognition.

Key Words: invariance signature, Lie group, optical character recognition

1. INTRODUCTION

In this paper we are concerned in particular with the invariant perception of two-dimensional patterns under shift, rotation and scaling in the plane. This corresponds to the ability of humans to recognize patterns such as typed or handwritten characters independently of their size, orientation or position, which they are able to do with little or no difficulty when reading a document such as an engineering or architectural drawing.

The aim of any invariant pattern recognition technique is to obtain a representation of the pattern in a form that is invariant under some transformations of the original image. It is often claimed that an ideal technique would produce a representation of the pattern which was not only invariant, but which also uniquely characterized the input pattern. This goal is not always as desirable as it might seem.

In many pattern recognition problems, the aim is to produce a system which classifies input patterns as belonging to a particular *class*, rather than to identify uniquely every input pattern presented. In such cases a unique representation for each possible input pattern can actually be a disadvantage. All that is required is an invariant representation which retains enough information for distinct classes to be distinguished. Indeed, if all members of a class are identical, or nearly so, in the invariant representation this can greatly reduce the size of the training set required by many recognition algorithms. The Invariance Signature provides a means of realizing this goal.

2. PRIOR METHODS FOR INVARIANT PATTERN RECOGNITION

A large number of techniques for invariant pattern recognition has been proposed. They vary greatly. Some treat the image as a pattern of pixels, others operate on higher level representations, such as contours. Many are expensive in computation time and/or space. They have varying degrees of sensitivity to noise. A brief review of popular techniques follows.

2.1. Integral Transforms

An approach that has long been popular is to seek an integral transform of the image which is invariant under some specified transformations. Most such techniques are based upon the Fourier transform: in the transform domain the amplitude spectrum is invariant under shifts of the image and the phase encodes the shift[1]. This idea can be generalized into a scheme for computing the kernel of an integral transform which will map an image into a space in which specified transformations are reduced to shifts. This is possible if the desired transformations are specified by Lie transformation groups, and the coordinates in which the transformations reduce to shifts are the canonical coordinates of the generators of the transformation groups [15, 33, 35, 16]. Such transforms, in various guises, form the bases of many invariant pattern recognition techniques [1, 26, 7, 15]. It is important to note that the transformed image is of (at least) the same dimensionality as the original. The invariant component does not uniquely characterize the input, and the issue of pattern recognition is not addressed.

2.1.1. Moments

Integrals can also be used to compute geometrical moments of the image. Certain functions of the moments are invariant under transformations of the image [21]. Geometrical moments are not fault-tolerant, and generally produce disappointing pattern recognition results [44]. Better results have been obtained using alternative

[1] It should be noted, however, that the phases of the spatial frequency components can be varied *independently*. There are thus many possible images of greatly varying visual appearance which have the same amplitude spectrum.

sets, such as the Zernike moments [22]. Again, computing a set of moments is expensive.

2.2. Matched Filtering and Convolution

Matched Filtering is one of the oldest techniques for pattern recognition [32, 2]. Combined with convolution it becomes cross-correlation: effectively an exhaustive search of the image for all possible transformed versions of a template pattern. Some version of cross-correlation plays a role in many invariant pattern recognition techniques [1, 7, 30, 46, 23, 26], despite the fact that it is computationally-expensive, and this cost scales exponentially with the number of transformations with respect to which invariance is desired.

2.3. Parts and Relationships

In the above techniques, an invariant representation of the whole image is sought. Matching is done on a pixel-by-pixel basis. A different approach is to view an image as a set of parts and relationships. For example, the angle between two lines is invariant under shifts, rotations and scalings. The lines are parts, and the angle is their relationship. In "parts and relationships" techniques, matching is done between abstracted properties of the image, rather than between pixels. These techniques require some form of sub-graph matching [24], which is known to be an NP-complete problem [2]. The challenge is thus to find an algorithm that can obtain an acceptable solution in reasonable time. Explicit search may be satisfactory for sufficiently small graphs. Another approach is relaxation labeling [24].

2.4. Cross-ratios

The techniques discussed in this section are often designed for the projective geometry appropriate in three-dimensional object recognition, rather than two-dimensional shift, rotation and scale invariance. Projective transformations preserve neither lengths, nor ratios of lengths. The ratio of two ratios of lengths, however, *is* invariant [44]. Such a ratio is known as a *cross-ratio.* Since projective transformations are many-to-one, matching cross-ratios are a necessary, but not sufficient, condition for feature-matching. Cross-ratios are used in many recognition problems involving projective geometry [17, 4, 43], and can be defined on the basis of four collinear points or four concurrent lines [28]. In practice, this means that such *labelled* reference points or lines must be extracted from the image before the technique can be applied. Candidate points and lines can, for example, be defined using points of inflection, corners or bitangents. The extraction of such reference points, even from synthetic images, and their ordering is in itself a non-trivial problem [13, 39].

2.5. Contour-Based Methods

In this paper we are specifically interested in the invariant recognition of contours. There are two basic approaches to contour recognition: the contour may be represented by a function fitted to image points belonging to the contour, or treated as a group of pixels. Contour recognition is of particular interest because there are many applications in which patterns naturally consist of line drawings (e.g. character recognition, circuit diagrams, engineering or architectural drawings). In other applications contours may be extracted via some form of edge detection. Moreover, there is evidence that the human visual system applies a contour-based approach to pattern recognition even when no contour exists in the image: an implied contour is interpolated [9].

2.5.1. *The Hough Transform*

Perhaps the simplest contour-based technique is the Hough transform [2]. Variants of the Hough transform occur frequently in the invariant pattern recognition literature [30, 12, 25]. The Hough transform and its generalizations can be interpreted as cross-correlation techniques, where the template is a parametric curve rather than an image [30]. Its disadvantage is that it is restricted in practice to a reasonably small set of template curves, since both the computation time and the data storage requirements increase exponentially with the number of parameters.

2.5.2. *Algebraic Invariants*

Another approach is to compute invariants of a continous function fitted to the contour. This promises to avoid the expenses of exhaustive methods such as the Hough transform. Algebraic invariants are well-suited for use with contours which can be expressed by an implicit polynomial $f(x, y) = 0$. An example is the family of conic sections, which can be defined in matrix form: $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X} = 0$. The coordinates can be transformed so that $\boldsymbol{A}$ is diagonal. Thus properties of $\boldsymbol{A}$ invariant under similarity transforms are invariant descriptors of the conic. One such feature is the determinant. In fact, any symmetric function of the eigenvalues of $\boldsymbol{A}$ is an invariant. $\boldsymbol{A}$ must be obtained by fitting a polynomial to the image – a difficult and potentially computationally-expensive problem in many real applications. The matching process, however, is cheap. A high-resolution image of the curve is required for accurate coefficient estimation, though Forsyth *et al.* claim that "For applications in model-based vision, it is far more important that a representation be projectively invariant than that it be a good approximation" [17].

2.5.3. *Differential Invariants*

Differential Invariants arise most naturally when the points on a curve, $\boldsymbol{x}$, are expressed as a function of a parameter t, $\boldsymbol{x} = \boldsymbol{x}(t)$, rather than by an implicit function. The natural shape descriptors in such a representation are the derivatives $\frac{d^n x_i}{dt^n}$. These descriptors are *local*, since they are evaluated at t, unlike the global descriptors derived using algebraic invariants. A differential invariant is a function of the $\frac{d^n x_i}{dt^n}$ which does not change under transformations of $\boldsymbol{x}$ and t. Various differential invariants have been applied: curvature, torsion and Gaussian curvature, for instance, are all invariant under Euclidean transformations [17]. Differential invariants are complete: a small set of invariants contains all the essential information about the curve. Also, their locality makes them insensitive to occlusion. Whilst elegant, their application requires the computation of high-order derivatives of the contour, which is known often to be infeasible [42, 45]. Moreover, these derivatives are raised to high powers, magnifying the estimation error [5].

2.5.4. *Semi-differential Invariants*

In the above methods, there is often a choice between making a search for reference points or having to compute high-order derivatives of the extracted contours, both of which processes are prone to error. A compromise between these approaches leads to a class of invariants known as *semi-differential Invariants* [42, 45].

3. LIE TRANSFORMATION GROUPS AND INVARIANCE

One approach to invariant pattern recognition is to consider how local features change under global transformations. This leads naturally to the study of Lie transformation groups, which have been a component of many, varied invariant pattern recognition techniques [18, 19, 15, 14, 33, 35, 38].

We derive a new shift-, rotation- and scale-invariant function of a two-dimensional contour, the *Invariance Measure Density Function.* It is shown that several such functions can be combined to yield an *Invariance Signature* for the contour. This Invariance Signature has several properties that make it attractive for implementation in an MBNN: it is based on local properties of the contour, so initial calculations are inherently parallel; it is statistical in nature, and its resolution can be chosen at the designer's discretion, allowing direct control over the dimensionality of the network implementation. The use of the Invariance Signature, however, is not limited to neural implementations.

Whilst patterns are not represented uniquely by the Invariance Signature, it will be shown in §6 that *classes* of patterns are represented sufficiently differently for optical character recognition applications.

3.1. One Parameter Lie Groups in Two Dimensions

A Lie group is a continuous transformation group with a differentiable structure [33]. For our purposes, the most interesting groups are the one-parameter Lie groups defined on the plane. These include rotation, dilation and translation. They are smooth transformations of the form

$$x' = \mu(x, y, \varepsilon) \qquad y' = \nu(x, y, \varepsilon). \tag{1}$$

The parameter ε determines which element of the group the transformation is. For instance, if ε_0 corresponds to the identity element, we have

$$x' = \mu(x, y, \varepsilon_0) = x \qquad y' = \nu(x, y, \varepsilon_0) = y. \tag{2}$$

There is a vector field $\vec{g} = \left[\begin{array}{cc} g_x & g_y \end{array}\right]^{\mathrm{T}}$ associated with each Lie group G, which gives the direction in which a point (x, y) is "dragged" by an infinitesimal transformation under the group. It is given by

$$g_x(x, y) = \left.\frac{\partial \mu}{\partial \varepsilon}\right|_{\varepsilon=\varepsilon_0} \qquad g_y(x, y) = \left.\frac{\partial \nu}{\partial \varepsilon}\right|_{\varepsilon=\varepsilon_0}. \tag{3}$$

This vector field $\vec{g}$ allows an operator $\mathcal{L}_G$ to be defined,

$$\mathcal{L}_G = g_x \frac{\partial}{\partial x} + g_y \frac{\partial}{\partial y}. \tag{4}$$

$\mathcal{L}_G$ is called the *generator* of G, because it can be used to construct the finite transformation corresponding to the infinitesimal dragging described in Eqs. (3).

3.2. From Infinitesimal to Finite Transformations

Consider the case in which the vector field specifying the infinitesimal transformation at each point (Eqs. (3)) is known. We wish to construct Eqs. (1), specifying the finite transformation. We will consider the transformation of x in detail. For a small change in the group parameter from the identity element, $\varepsilon = \varepsilon_0 + \Delta\varepsilon$, we can approximate the change in x by

$$x' = x + \Delta x \approx x + \Delta\varepsilon \left.\frac{\partial\mu}{\partial\varepsilon}\right|_{\varepsilon=\varepsilon_0} . \tag{5}$$

We now wish to find a finite transformation corresponding to n applications of the $\Delta\varepsilon$ transformation. This will approximate the finite transformation corresponding to the group element specified by parameter $\varepsilon = n\Delta\varepsilon$. Let x_i be the value of x' after i applications of the $\Delta\varepsilon$ transformation. We obtain

$$\begin{aligned} x_0 &= x \\ x_1 &= x_0 + \frac{\varepsilon}{n}\mathcal{L}_G x_0 = \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right) x \\ x_2 &= x_1 + \frac{\varepsilon}{n}\mathcal{L}_G x_1 \\ &= \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right) x_1 \\ &= \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right)^2 x \end{aligned} \tag{6}$$

and thus

$$x_n = \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right)^n x.$$

In the limit as $n \to \infty$, the approximation becomes exact, giving the finite transformations

$$\mu(x,y,\varepsilon) = \lim_{n\to\infty} \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right)^n x \qquad \nu(x,y,\varepsilon) = \lim_{n\to\infty} \left(1 + \frac{\varepsilon}{n}\mathcal{L}_G\right)^n y. \tag{7}$$

An example of the direct application of the derivation of the familiar rotation transformation using Eqs. (7) may be found in [37].

3.3. Functions Invariant Under Lie Transformations

A function is said to be invariant under a transformation if all points of the function are mapped into other points of the function by the transformation. Consider a function $F(x,y)$. We wish to determine its invariance with respect to a Lie transformation group G. Let

$$\vec{g}(x,y) = \left[\ g_x(x,y)\ \ g_y(x,y)\ \right]^{\mathrm{T}} \tag{8}$$

be the associated vector field, and $\mathcal{L}_G$ be the generator of G. F is constant with respect to the action of the generator if

$$\mathcal{L}_G F = 0. \tag{9}$$

This can be written in terms of the vector field as

$$\nabla F \cdot \vec{g}(x, y) = 0. \tag{10}$$

Now consider a contour C parameterized by t specified by the implicit function

$$\forall t \;\; F(x(t), y(t)) = K. \tag{11}$$

Since F is constant on the contour, we have

$$\frac{dF}{dt} = \frac{\partial F}{\partial x}\frac{dx}{dt} + \frac{\partial F}{\partial y}\frac{dy}{dt} = 0. \tag{12}$$

Combining Eqs. (10) and (12) shows that F is invariant under the Lie transformation generated by $\mathcal{L}_G$ if

$$\frac{dy}{dx} = \frac{g_y}{g_x}. \tag{13}$$

everywhere on the contour.

The condition derived in Eq. (13) has a very natural interpretation: a contour is invariant under a transformation group G if the tangent to the contour at each point is in the same direction as the vector field $\vec{g}$ corresponding to the infinitesimal transformation that generates the group.

4. THE INVARIANCE SIGNATURE: FROM LOCAL INVARIANCE MEASURES TO GLOBAL INVARIANCE

We now propose a new shift-, rotation- and dilation-invariant signature for contours. We call this an *Invariance Signature*, since it is derived from the degree to which a given contour is consistent with invariance under a set of Lie transformation groups. We show that although a given contour may not be invariant under a transformation group G (i.e. Eq. (13) does not hold), *the overall departure from invariance under G of the contour is invariant* under the similarity group of transformations. This is the central idea of this paper. It is important to note that the invariance of the Invariance Signature of a contour C under G does not imply the invariance of C under G: it is the manner in which C departs from invariance under G which is invariant.

4.1. The Local Measure of Consistency

We have seen in Eq. (13) that in order for a contour C to be invariant under a transformation group G the tangent to the contour must be everywhere parallel to the vector field defined by the generator of the group. We now define the *Local Measure of Consistency* with invariance under a transformation group G at a point (x, y) on C, $\iota_G(x, y)$.

$$\iota_G(x, y) = \left| \hat{\theta}(x, y) \cdot \hat{g}_G(x, y) \right| \tag{14}$$

The absolute value is used because only the orientation of the tangent vector is significant, not the direction. At each point both the tangent vector to the contour,

$\vec{\theta}(x,y)$ and the vector field $\vec{g}(x,y)$ are normalized:

$$\hat{g}(x,y) = \frac{g_x(x,y)\hat{\imath} + g_y(x,y)\hat{\jmath}}{\sqrt[+]{g_x^2(x,y) + g_y^2(x,y)}} \tag{15}$$

and

$$\hat{\theta}(x,y) = \frac{\hat{\imath} + \frac{dy}{dx}\hat{\jmath}}{\sqrt[+]{1 + \left(\frac{dy}{dx}\right)^2}}, \tag{16}$$

where $\hat{\imath}$ and $\hat{\jmath}$ are unit vectors in the x and y directions respectively. Substituting Eqs. (15) and (16) in Eq. (14), we obtain

$$\iota_G(x,y) = \frac{1}{\sqrt[+]{1 + \left[\frac{g_y(x,y) - g_x(x,y)\frac{dy}{dx}}{g_x(x,y) + g_y(x,y)\frac{dy}{dx}}\right]^2}}. \tag{17}$$

4.2. The Invariance Measure Density Function

Eq. (14) is a mapping $C \mapsto [0,1]$, which gives a measure of the degree to which the tangent at each point is consistent with invariance under G. We now seek a function which characterizes the consistency of the entire contour C with invariance under G. Such a function is the density function for the value of ι_G in $[0,1]$, $I(\iota_G)$, which we will call the *Invariance Measure Density Function.* The more points from C that are mapped close to 1 by Eq. (14), the more consistent C is with invariance under G. $I(\iota_G)$ is a descriptor of C, and we will show that $I(\iota_G)$ is invariant under rotations and dilations of C. Translation invariance of $I(\iota_G)$ is obtained by choosing the centroid of C as the origin of coordinates.[2]

It is interesting to note that there is evidence from psychophysical experiments that a measure of the *degree of invariance* of a pattern with respect to the similarity group of transformations (rotations, translations and dilations) is important in human pattern recognition [6]. The measure proposed here might be seen as a mathematical formalization of this notion. Moreover, its implementation in a neural network architecture is consistent with Caelli and Dodwell's statement [6, p. 159] of a proposal due to Hoffman [18, 19]: 'Hoffman's fundamental postulate was that the coding of orientation at various positions of the retinotopic map by the visual system, discovered by Hubel and Wiesel [20] and others, actually provides the visual system with "vector field" information. That is, the visual system, on detecting specific orientation and position states ("Φ/P codes"), spontaneously extracts the path curves (interpreted as visual contours) of which the local vectors are tangential elements.'

[2] It should be noted that the translation invariance of $I(\iota_G)$ is a separate notion from that of the translation invariance of the contour C as defined in §4.4.

First, however, we must establish the form of $I(\iota_G)$. Let C be parameterized by $t : t \in [t_0, T]$. The arc length s along C is

$$s(t) = \int_{t_0}^{t} \sqrt{\left(\left. \frac{dx}{dt} \right|_{\tau} \right)^2 + \left(\left. \frac{dy}{dt} \right|_{\tau} \right)^2} d\tau. \tag{18}$$

The total length of C is thus $S = s(T) = \oint_C ds$. For well-behaved functions $F(x, y)$, we can construct $s(t)$ such that we can reparameterize C in terms of s. Thus we can rewrite Eq. (14) to give ι_G in terms of s. For simplicity, we will first consider the case in which ι_G is a monotonic function of s, as shown in Figure 1. The Invariance Measure Density is:

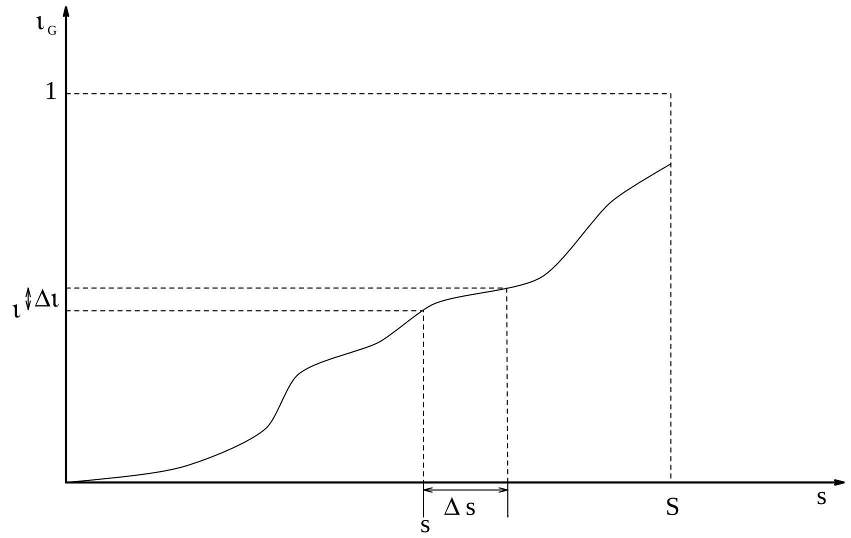


FIG. 1. Local Measure of Consistency as a function of arc length.

$$I(\iota_G) = \lim_{\Delta\iota_G \to 0} \left| \frac{\Delta s}{S \Delta \iota_G} \right| = \frac{1}{S} \left| \frac{ds}{d\iota_G} \right|. \tag{19}$$

$I(\iota_G)$ can be interpreted as the probability density function for ι_G at points $(x(s), y(s))$, where s is a random variable uniformly distributed on $[0, S]$. It is clear that for the general case the function could be broken into piecewise monotonic intervals and their contributions to the density summed. The general form for a specific value $\iota_G{}'$ is thus

$$I(\iota_G{}') = \frac{1}{S} \sum_{s \in [0,S] : \iota_G(s) = \iota_G{}'} \left| \frac{ds}{d\iota_G} \right|. \tag{20}$$

THEOREM 4.1. *The Invariance Measure Density Function, $I(\iota_G)$, is invariant under translations, rotations and dilations of the contour C with respect to which it is calculated.*

Proof. That $I(\iota_G)$, defined in Eq. (20), is invariant under translations of the contour C is trivial, since, as defined in §4.2, ι_G is calculated with the origin at the centroid of the contour C. Rotation and dilation invariance can be proved by construction. Since the transformations for rotation and dilation in the plane commute, we can consider a two-parameter Abelian group corresponding to a rotation by an angle ϕ and a dilation by a positive factor β. The coordinates x and y are transformed according to

$$x' = \beta\left(x\cos\phi - y\sin\phi\right) \qquad\qquad y' = \beta\left(x\sin\phi + y\cos\phi\right). \tag{21}$$

Consider the relationship between the arc length $s(t)$ for the original parameterized contour $(x(t), y(t))$ and $s'(t)$, after the coordinates are transformed according to Eqs. (21). We find that

$$\frac{dx'}{dt} = \beta\cos\phi\frac{dx}{dt} - \beta\sin\phi\frac{dy}{dt} \qquad\qquad \frac{dy'}{dt} = \beta\sin\phi\frac{dx}{dt} + \beta\cos\phi\frac{dy}{dt}. \tag{22}$$

Combining these, we obtain

$$\left(\frac{dx'}{dt}\right)^2 + \left(\frac{dy'}{dt}\right)^2 = \beta^2\left[\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2\right]. \tag{23}$$

This result can be substituted into Eq. (18), giving

$$s'(t) = \beta s(t). \tag{24}$$

This indicates that the total arc length is $S' = \beta S$. The derivative of $s(t)$ is also scaled. Substituting into Eq. (20), we obtain

$$\begin{aligned} I'\left(\iota_G{}'\right) &= \frac{1}{S'}\sum_{\iota_G(s)=\iota_G{}'}\left|\frac{ds'}{d\iota_G}\right| \\ &= \frac{1}{\beta S}\sum_{\iota_G(s)=\iota_G{}'}\beta\left|\frac{ds}{d\iota_G}\right| \\ &= \frac{1}{S}\sum_{\iota_G(s)=\iota_G{}'}\left|\frac{ds}{d\iota_G}\right| \\ &= I\left(\iota_G{}'\right) \end{aligned} \tag{25}$$

Thus $I\left(\iota_G{}'\right)$ is invariant under rotations and dilations of C. ■

4.3. Invariance Measure Densities For Specific Contours

To demonstrate the application of the Invariance Measure Density, we will evaluate $I(\iota_G)$ for a specific contour. Let C be a square of side $2L$ centred at the origin, as shown in Figure 2. We will find the Invariance Measure Density for C with respect to rotation, $I_{C_{rot}}(\iota)$. By symmetry, we need only find $I_{C_{rot}}$ for one side of the square. On the side indicated by the dashed line in Figure 2, x and y

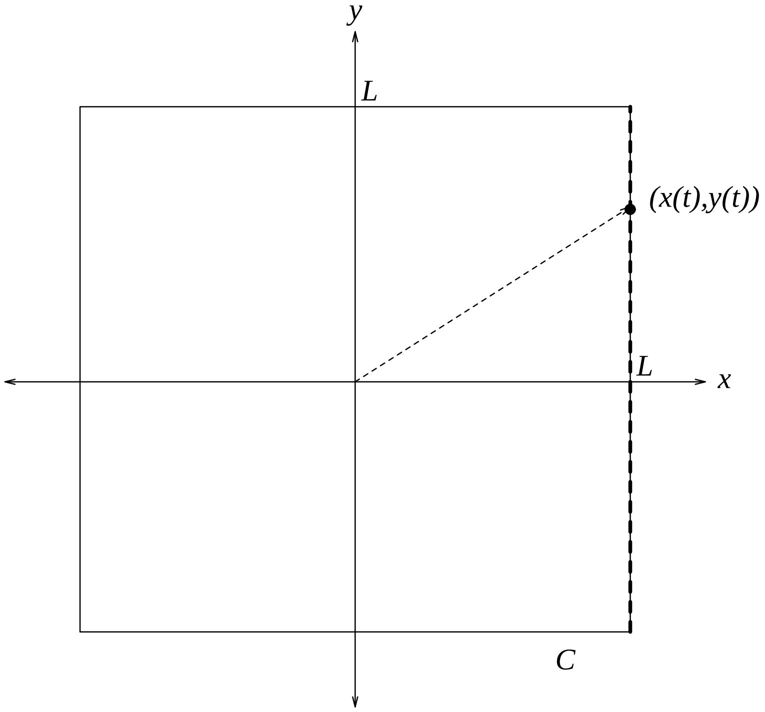


FIG. 2. Square of side 2L.

can be expressed in terms of a parameter t as

$$x = L \qquad\qquad y = t, \qquad -L \leq t \leq L. \tag{26}$$

Thus the arc length is $s(t) = t + L$, and the total is $S = 2L$. If $\dot{x}(t)$ and $\dot{y}(t)$ are the derivatives of x and y with respect to t, Eq. (17) can be rewritten as

$$\iota_{C_{rot}}(s) = \frac{1}{\sqrt{1 + \left[\frac{g_y(s)\dot{x}(s) - g_x(s)\dot{y}(s)}{g_x(s)\dot{x}(s) + g_y(s)\dot{y}(s)}\right]^2}} \tag{27}$$

Here $\dot{x}(t) = 0$ and $\dot{y}(t) = 1$. The generator of the rotation group is

$$\mathcal{L}_R = -y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}. \tag{28}$$

Eqs. (28) and (26) can be substituted into Eq. (27) to give

$$\iota_{C_{rot}}(s) = \frac{1}{\sqrt{1 + \left(\frac{s-L}{L}\right)^2}}, \tag{29}$$

which can be inverted to yield

$$s = L\left(1 + \sqrt{\frac{1}{\iota^2_{C_{rot}}} - 1}\right); \tag{30}$$

differentiating,

$$\frac{ds}{d\iota_{C_{rot}}} = \frac{-L}{\iota_{C_{rot}}^2 \sqrt{1-\iota_{C_{rot}}^2}}. \tag{31}$$

Using Eq. (20), we arrive at our final result:

$$\begin{aligned} I_{C_{rot}}\left(\iota_{C_{rot}}\right) &= \frac{1}{2L} \sum_{\iota'_{C_{rot}}=\iota_{C_{rot}}} \left|\frac{ds}{d\iota_{C_{rot}}}\right|_{\iota'_{C_{rot}}} \\ &= \frac{1}{2L} \times 2 \times \frac{L}{\iota_{C_{rot}}^2 \sqrt{1-\iota_{C_{rot}}^2}} \\ &= \frac{1}{\iota_{C_{rot}}^2 \sqrt{1-\iota_{C_{rot}}^2}}, \qquad \iota_{C_{rot}} \in \left[\frac{1}{\sqrt{2}}, 1\right]. \end{aligned} \tag{32}$$

Note that, as required, the scale of the square L does not appear. The factor of 2 arises because $\iota_{C_{rot}}$ for one side of the square $(0 \leq s \leq 2L)$ is symmetric about $s = L$, so the sum has two terms. This function, shown in Figure 3, is characteristic of the square.

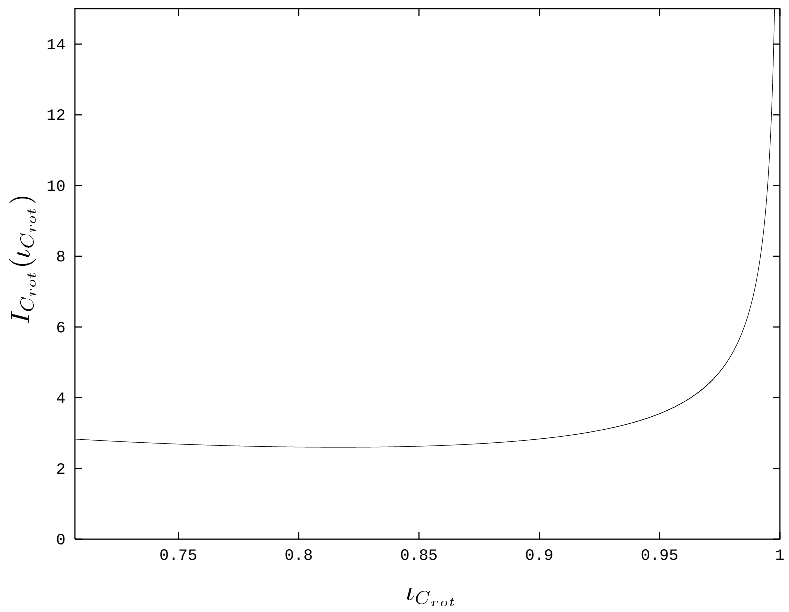


FIG. 3. Invariance Density Measure with respect to rotation for a square.

4.4. Invariance Space: Combining Invariance Measure Densities

We now consider the case in which the Invariance Measure Density Function is calculated with respect to a number of groups, and the results combined to provide a more complete characterization of the transformational properties of a contour C. This operation maps each point from the two dimensional image space to the interior of a unit hypercube in an n-dimensional *invariance space*, where each of the n dimensions corresponds to a particular sort of invariance. Eq. (33) shows this for the case of a three-dimensional invariance space where the dimensions correspond to the Local Measure of Consistency ι with respect to rotation, dilation and translation:

$$(x, y) \mapsto \left[\begin{array}{lll} \iota_{rot} & \iota_{dil} & \iota_{trans} \end{array}\right]^{\mathrm{T}}. \tag{33}$$

The distribution of points in this invariance space is characteristic of the contour C. Since each of the component invariance measure densities is invariant, this n-dimensional Invariance Signature is invariant under rotations, dilations, translations (and reflections) of the input image. It will be shown later that the projections on to the axes of this three-dimensional invariance space (i.e. the Invariance Signatures with respect to each transformation) give adequate performance in an experimental application of Invariance Signatures. The idea of plotting one invariant against another in order to create an invariant signature has also been employed by Weiss [45], who used differential invariants.

The vector fields for the generators of the transformation groups for rotation, dilation and translation are given in normalized form. All can be derived using Eq. (3):
for rotation invariance

$$\vec{g}_{rot}(x, y) = \frac{1}{\sqrt{x^2 + y^2}} \left[\begin{array}{ll} -y & x \end{array}\right]^{\mathrm{T}}, \tag{34}$$

and for dilation invariance

$$\vec{g}_{dil}(x, y) = \frac{1}{\sqrt{x^2 + y^2}} \left[\begin{array}{ll} x & y \end{array}\right]^{\mathrm{T}}. \tag{35}$$

The translation invariance case is somewhat different. The vector field corresponding to the generator of a translation transformation is constant for all (x, y). Here we choose the direction of this field to be that of the unit eigenvector corresponding to the largest eigenvalue, $\vec{e}_1$, of the coordinate covariance matrix of all points in the contour: the direction of maximum variance of the contour. The Invariance Measure Density for translation thus measures the degree to which the contour is invariant under translation along its principal axis. An infinite straight line, for example, is perfectly invariant under this transformation. Since $\vec{e}_1$ is calculated from the image each time it is required, this measure is invariant under rotations, dilations and translations of the image. The vector field for the translation invariance case is thus:

$$\vec{g}_{trans}(x, y) = \left[\begin{array}{ll} e_{1x} & e_{1y} \end{array}\right]^{\mathrm{T}} \tag{36}$$

It should be noted that this representation of the image is not unique. The combination of individual Invariance Measure Densities into an Invariance Space does, however, increase its discriminating properties. As an example, removing two opposite sides of a square will not alter its rotation or dilation Invariance Signatures, but it will change the translation Invariance Signature. Likewise, a single straight line has the same translation Invariance Signature as any number of parallel straight lines, however they are spaced. The rotation and dilation Invariance Signatures, however, are sensitive to these changes.

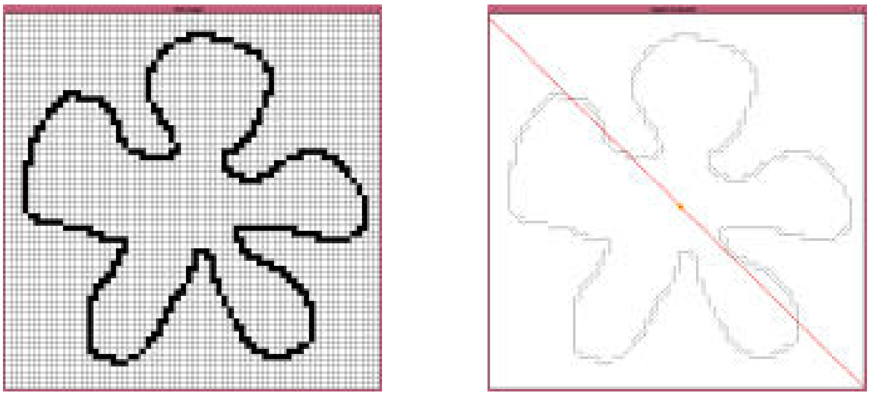
4.5. Discrete Invariance Signatures

For a computer application of Invariance Signatures, a discrete version is required. The natural choice is the frequency histogram of ι_G. For a continuous contour, this is obtained by dividing the interval $[0, 1]$ into n "bins" and integrating $I(\iota_G)$ over each bin. For bins numbered from b_0 to b_{n-1}, the value in bin k is

$$b_k = \int_{\frac{k}{n}}^{\frac{k+1}{n}} I(\iota_G) d\iota_G . \tag{37}$$

Since $I(\iota_G)$ is a probability density function, the sum of the values of the bins must be one.

When using sampled images, a true frequency histogram of the estimated local measures of consistency may be used. The system designer must choose the number of bins, n, into which the data is grouped. It will be seen in §6 that this choice is not arbitrary.



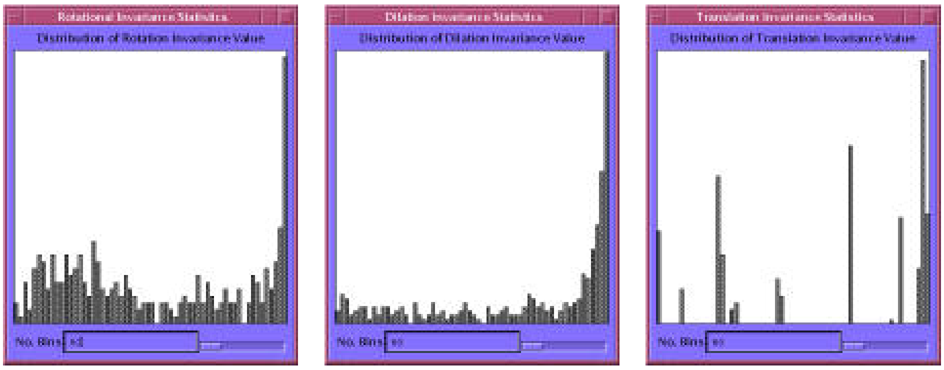
1. Original Contour

2. Tangent Estimates

FIG. 4. Example of a sampled contour and its estimated tangents. In Figure 4(2), the filled circle indicates the centroid of the contour, and the dashed line shows the direction of $\vec{e}_1$.

An example of a sampled contour and the estimated tangent vectors at each point is shown in Figure 4. The estimated discrete Invariance Signatures are shown in Figure 5, for 60 bins. It would be expected that this "flower"-shaped contour would have Invariance Signatures which reflect a quite strong dilation-invariant component corresponding to the approximately radial edges of the "petals". These edges should also cause several distinct modes in the Invariance Signature with respect to translation, corresponding to edges at given angles to the principal direction $\vec{e}_1$. The Invariance Signature with respect to rotation should be more uniform, though with a rotation-invariant component due to the ends of the petals which are approximately orthogonal to the radial edges. This is indeed what is observed in Figure 5.

4.5.1. Occlusion



No. Bins 60

No. Bins 60

No. Bins 60

1. Rotation 2. Dilation 3. Translation

FIG. 5. 60 bin discrete Invariance Signatures for the contour in Figure 4. Histogram bin heights are normalized by the height of the largest bin (rightmost bin of the Dilation Invariance Signature), which has a value of 23%.

The Invariance Signature is vulnerable to occlusions of the contour, since these can alter the position of the contour centroid and the direction of $\vec{e}_1$. In the applications for which the Invariance Signature was designed, i.e. recognizing text and/or symbols in engineering or architectural drawings, occlusion should never occur. Indeed an occlusion can transform an alphabetic character into a completely different valid character. Nevertheless, if invariant contour points (e.g. bitangents, corners, etc.) were used to define this point and direction, occlusion invariance (in the histogram intersection sense, see §7.2) could be achieved.

4.5.2. Hough Transform Interpretation

It is interesting to note that the discrete Invariance Signature of a contour can be interpreted as a sort of Hough Transform [2]. Tangents at points on the contour are mapped to accumulators in Invariance Space. These can be interpreted as votes for the presence of a contour with a particular degree of invariance with respect to a given transformation. A circle, for example, is perfectly rotation-invariant, and all its tangents will be mapped to the same "circle" bin in the Invariance Signature with respect to rotation. Likewise, there is a trajectory in invariance space to which all circles are mapped.

5. THE INVARIANCE SIGNATURE NEURAL NETWORK CLASSIFIER

We have proposed a Model-Based Neural Network to compute the discrete Invariance Signature of an input pattern and to classify it on that basis [38, 37]. This system will be referred to as the Invariance Signature Neural Network Classifier (ISNNC). MBNNs, introduced in [8] and further elaborated in [37], allow a network to be constructed in which the supervisor's knowledge of the task to be performed is used to specify, partially or completely, the roles of some hidden units, or of whole hidden layers or modules, in advance. Thus the supervisor's knowledge of which features of the training data are significant for the task is incorporated into the network geometry and connection weighting functions, serving as a constraint on the state space searched during training.

The ISNNC consists of a system of neural network modules, some hand-coded and some trained on sub-tasks. A schematic diagram is shown in Figure 6. Whilst the ISNNC appears complex, it retains the basic characteristics of a traditional feed-forward neural network (The term MLP will be used to denote a fully-connected, feed-forward multilayer perceptron). It consists entirely of simple nodes joined by weighted connections.[3]. Each node i in the network computes the sum of its j weighted inputs, $\mathrm{net}_i = \sum_j w_{ij} x_j$. This is used as the input to a transfer function f, which is either linear, $f(\mathrm{net}_i) = \mathrm{net}_i$, or the standard sigmoid, $f(\mathrm{net}_i) = \frac{1}{1+e^{-\mathrm{net}_i}}$.

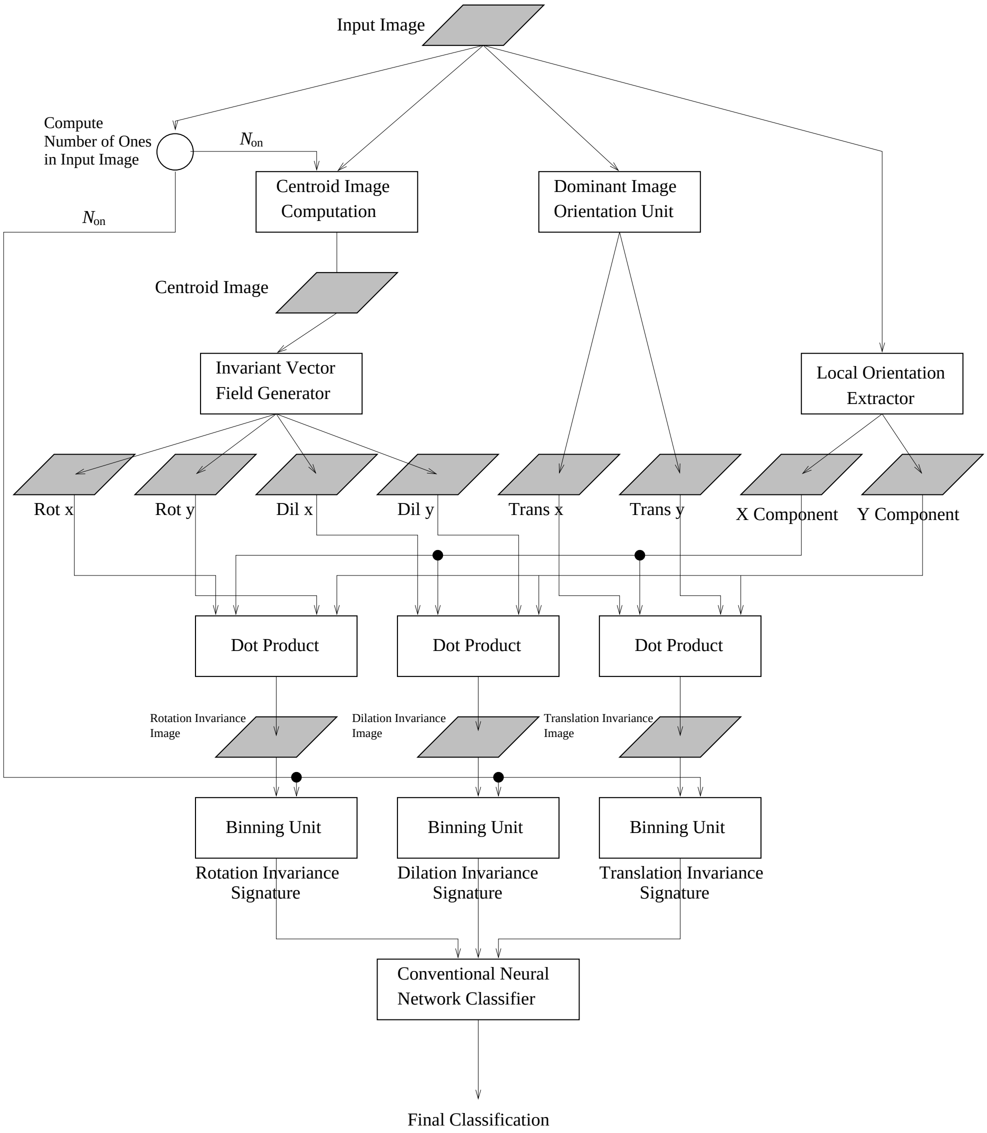

Input Image
Compute Number of Ones in Input Image
N_{on}
N_{on}
Centroid Image Computation
Dominant Image Orientation Unit
Centroid Image
Invariant Vector Field Generator
Local Orientation Extractor
Rot x
Rot y
Dil x
Dil y
Trans x
Trans y
X Component
Y Component
Dot Product
Dot Product
Dot Product
Rotation Invariance Image
Dilation Invariance Image
Translation Invariance Image
Binning Unit
Binning Unit
Binning Unit
Rotation Invariance Signature
Dilation Invariance Signature
Translation Invariance Signature
Conventional Neural Network Classifier
Final Classification

FIG. 6. Invariance Signature-based contour recognition system.

[3] With the exception of the *Dominant Image Orientation Unit*, for which a neural network solution is still to be developed

There is a number of distinct levels. Computation at one level must be completed before computation at the next level begins.

The only departure from a traditional neural network is that some weights are calculated at runtime by nodes at prior levels. We call these *dynamic weights.* They allow the ISNNC to compute dot products, and for nodes to act as gates controlling the transmission of the output of another node. Since connection weights in any implementation are only references to a stored value, this should not present any difficulty. Alternatively, the same functionality can be achieved by allowing nodes which multiply their inputs, as used in Higher Order Neural Networks [29, 36].

It is acknowledged that several calculations performed by modules of the ISNNC need not be carried out by neural networks. Indeed tasks such as centroid location and local orientation extraction can be more simply performed by direct calculation or table lookup, particularly if the ISNNC is implemented as a software simulation, as is in the present work. The classifications of the Invariance Signatures could be done by computing multidimensional histogram intersections with those of a database of known patterns [40, 41]. This work, however, was carried out as part of an investigation of the capabilities of MBNNs, with a view to hardware implementation. Consequently, neural solutions for all modules were sought.

6. CHARACTER RECOGNITION WITH INVARIANCE SIGNATURE NETWORKS

It remains now to demonstrate that Invariance Signatures retain enough information to be usable for pattern recognition, and that they are not unduly sensitive to the noisy data encountered in real applications. To this end, the system is applied to the classification of Roman alphabetic characters, both for "perfect" machine-generated training and test data, and for scanned data. The term "perfect" will be used throughout to describe data which is both noise-free and unambiguous.

6.1. Perfect Data

6.1.1. Departures from Exact Invariance

Despite the proven invariance properties of Invariance Signatures calculated for continuous contours in the plane, departures from invariance occur in real applications in several ways. Scanned data contains noise from the sensor, although the present quality of scanners makes this negligible for this application. More important sources of error are discussed below.

Quantization Noise. Noise is introduced into the tangent estimation procedure by the sampling of the contour. Since the estimated orientation is quantized, the Local Measure of Consistency can change when a contour is quantized at a new orientation. It is possible to compensate partially for this effect by using sufficiently wide bins when calculating the Invariance Signature, but errors still arise when the new estimated orientation moves the resultant ι_G across bin boundaries.

Ambiguous Characters. In many fonts some letters are rotated or reflected versions of others, such as {b, d, p, q} and {n, u}. Consequently, it is impossible to classify isolated characters into 26 classes if shift, rotation, scale and reflection invariance is desired. Admittedly, reflection invariance is not usually desired, but it is an characteristic of the ISNNC. In commercial OCR systems, context information (i.e. surrounding letter classifications and a dictionary) is used to resolve ambiguities, which occur even in systems without inherent invariances. This approach would be equally applicable as a post-processing stage for the ISNNC.

6.1.2. The Data Set

A computer can be used to produce a perfect data set, which is free from quantization noise and contains no ambiguous characters. This set can be used to show that Invariance Signatures retain sufficient information for classification in the absence of noise, and these results can be used to assess the performance of the system on real data.

A training data set was created using a screen version of the Helvetica font. Only the letters {a, b, c, e, f, g, h, i, j, k, l, m, n, o, r, s, t, v, w, x, y, z} were used, so that ambiguity was avoided. An 18×18 binary image of each letter was produced. This training data set is shown in Figure 7.

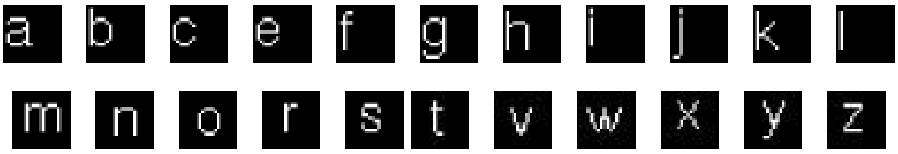


FIG. 7. Training set of canonical examples of unambiguous characters.

A perfect test data set was created by computing reflected and rotated versions of the training data, where rotations were by multiples of $\frac{\pi}{2}$ radians, so that there was no quantization error. This test data set is shown in Figure 8.

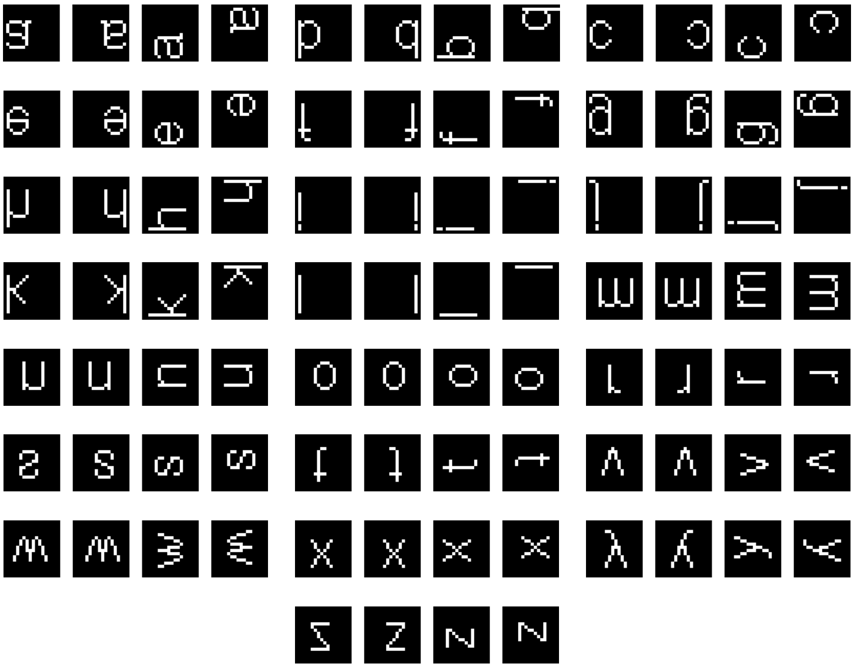


FIG. 8. Test set of ideally shifted, rotated and reflected letters.

6.1.3. Selected Networks Applied to this Problem

Simulations showed that this training set could be learnt by a network with no hidden layer: it is linearly separable [27]. This was was also be true of the Invariance Signatures calculated from these data. Two different network architectures were constructed for comparison. The first was a MLP with a 18 × 18 input layer, no hidden layers, an 1 × 22 output layer and standard sigmoid transfer functions. The other was an ISNNC, with an 18×18 input layer. Within this ISNNC, the Invariance Signature was calculated using 5 bins for each transformation. The resultant 15 node Invariance Signature layer was connected to a 1 × 22 output layer, forming a linear classifier sub-network.

6.1.4. Reduction of Data Dimensionality

Although the Invariance Signature calculation stage of the ISNNC has to be run every time a new pattern is classified, Invariance Signatures of the training and test data need only be calculated once. The classification stage of the ISNNC can then be trained and tested as a separate module. This can lead to a great reduction in development time. The number of weights n_p in a MLP is:

$$n_p = \sum_{i=1}^{N-1} (\text{nodes in layer})_{i-1} \times (\text{nodes in layer})_i \tag{38}$$

where N is the total number of layers, and i is the layer number, (the input layer is layer 0). The iteration time during training and testing is proportional to n_p, so, for this example, each iteration for the Invariance Signature classification module will be $\frac{18\times18}{3\times5} = 21.6$ times faster than for the MLP.

The calculation of the Invariance Signatures is time-consuming, but this time is recouped during training and testing when the input image is large, which is typically the case in real applications. ISNNCs can reduce the dimensionality of the training and testing data significantly, and thus the development time for the classification network. Moreover, an ISNNC simulation on a sequential computer cannot take advantage of the parallel, local computations that characterize many of the ISNNC modules. A parallel implementation would be much faster.

6.1.5. Perfect and Network-Estimated Local Orientation

The neural orientation extraction module (NOEM) had some residual error [37]. Consequently, two versions of the Invariance Signature training and test data were created, one with the tangents calculated directly from the covariance matrix eigenvectors, and the other using the NOEM. Results from these are compared to evaluate the importance of accurate tangent extraction.

Ten instances of each network were made, each with a different parameter initialization. All were trained for 1000 iterations, using backpropagation. Patterns were assigned to the class corresponding to the highest output node value, so no patterns were rejected.

Results for MLPs trained using backpropagation. The results obtained with MLPs are summarized in Table 1. In this and subsequent tables, the “Best Performance” figures show the classification accuracies on both test and training data at the iteration at which the best *test data* performance was obtained during training.

The "Final Performance" figures are those obtained after the full (fixed) number of training iterations. The final test data performance is often inferior to the best performance, due to the over-fitting of the training data. The "best" performing network could easily be obtained by storing the state of the network at this iteration, or by applying a comprehensive cross-validation scheme.

As expected, the MLPs do not exhibit invariance, since the training data contained no transformed versions of the patterns.

TABLE 1
Classification performance (% correct) of traditional neural network classifiers trained for 1000 iterations with the data shown in Figure 7 and tested with the perfect data set in Figure 8.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
$\mu \pm \sigma$	100 ± 0.0	15 ± 0.5	100 ± 0.0	14 ± 0.0

The final performance of the MLPs is better than chance ($\frac{1}{22} = 4.5454\%$). It might be thought that this is because some transformations of highly symmetrical training patterns resulted in patterns very similar to the untransformed version (e.g. o, s, x and z). Analysis, however, shows that this is not the case. The 12 correctly classified test patterns are shown in Figure 9.

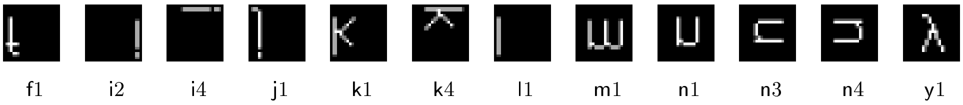


FIG. 9. Test patterns classified correctly by the MLPs.

No reason for these patterns being classified correctly is apparent. It seems clear that chance must play a part. For instance, i4 shares no "on" pixels with the training example of i. Marginally better performance on the test data could have been achieved by employing an early-stopping scheme. This would, however, have been at the cost of less than 100% correct performance on the training data. It should be noted that the sum squared-error on the test set decreased throughout training. A scheme based on the test set error would not improve performance in this case.

It must be acknowledged that MLPs could not be expected to perform better than chance on this task. Their architecture provides no invariances, and generalization cannot be expected unless transformed versions of the patterns are included in the training set. This argument can be used against all the comparisons between MLPs and MBNNs in this paper: they are not fair. Nevertheless, these comparisons between naive applications of MLPs and specifically-designed MBNNs demonstrate that MBNNs can perform successfully using training sets completely inadequate for MLPs. Moreover, these MBNNs are of lower dimensionality than the MLPs. Providing the MLPs with sufficiently large training sets would only make their training still more computationally-expensive, with no *guarantee* of invariant performance.

Results with Perfect Local Orientation. The results for the ten ISNNCs which used perfect Local Orientation Extraction are summarized in Table 2. The average number of iterations for 100% correct classification to be achieved was 220. Since the problem is linearly-separable, it could in fact be solved directly, using a technique such as singular-valued decomposition [31]. Since weights may set by any method at all in the MBNN paradigm, this makes the comparison of convergence times somewhat irrelevant. Nevertheless the MBNN modules, although taking on average 4.4 times as many iterations to converge as the MLPs, were 4.9 times faster to train, due to their lower dimensionality.

TABLE 2
Classification performance (% correct) of 5 Bin Invariance Signature Neural Network Classifiers (with perfect Local Orientation Extraction) trained for 1000 iterations with the data shown in Figure 7 and tested with the perfect data set in Figure 8.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
$\mu \pm \sigma$	100 ± 0.0	100 ± 0.0	100 ± 0.0	100 ± 0.0

The ISNNCs generalize perfectly to the the test set. The network architecture constrains the system to be shift-, rotation-, scale- and reflection-invariant in the absence of quantization noise, so this is no surprise. Importantly, the result indicates that sufficient information is retained in the 5 bin Invariance Signatures for all 22 unambiguous letters of the alphabet to be distinguished. It is clear that if this is the case, then it will also be so for any Invariance Signatures with higher resolution. Inspection of the sum squared error values after each iteration indicated that the error on the test set was indeed identical to that on the training set: for perfect data, the ISNNC produces perfect results.

Results using the NOEM. The results above were obtained using a hybrid system, which used a non-neural module to calculate the tangent at each point. The Invariance Signatures for the test and training data were recalculated using the NOEM, and classification modules were trained using these Invariance Signatures. Systems were produced with both 5 and 10 bin Invariance Signatures. The results obtained are summarized in Tables 3 and 4.

TABLE 3
Classification performance (% correct) of 5 Bin Invariance Signature Neural Network Classifiers (with neural Local Orientation Extraction) trained for 1000 iterations with the data shown in Figure 7 and tested with the perfect data set in Figure 8.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
$\mu \pm \sigma$	100 ± 0.0	96.6 ± 0.00	100 ± 0.00	96.6 ± 0.0

TABLE 4
Classification performance (% correct) of 10 Bin Invariance Signature Neural Network Classifiers (with neural Local Orientation Extraction) trained for 1000 iterations with the data shown in Figure 7 and tested with the perfect data set in Figure 8.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
$\mu \pm \sigma$	100 ± 0.0	95.5 ± 0.00	100 ± 0.00	93.2 ± 0.0

The misclassified patterns for the 5 bin ISNNCs are shown in Figure 10. t1 and t2 were classified as f, which is understandable, since there is very little difference between the patterns. t4 was misclassified as a. All ten networks had these same misclassifications.

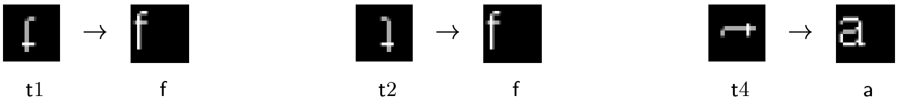


FIG. 10. Test Patterns Misclassified by the 5 Bin Invariance Signature Neural Network Classifiers, and the training examples as which they were incorrectly classified.

These results show that residual error in the NOEM causes a degradation of the classification performance of the ISNNCs. They are, however, still far superior to those for the MLPs. Moreover, a more accurate NOEM could be constructed: it is simply a question of network size and training time.

The results for the 10 bin system in Table 4 show that the effects of inaccuracies in the NOEM are greater when the number of bins is increased. This is due to the fact that the errors can cause the consistency measure at a point to change bins more easily this way, thus altering the Invariance Signature histogram. This is of course not the case when perfect Local Orientation Extraction is used.

6.2. Optical Character Recognition

Having demonstrated that Invariance Signatures retain sufficient information for the classification of "perfect" data, it remains to show that the system can be used for transformation invariant pattern recognition in the presence of sensor and quantization noise. To this end, it was decided to apply ISNNCs to the classification of scanned images of shifted and rotated printed alphabetic characters.

6.2.1. The Data Set

The training and test sets were created using all 26 letters of the English alphabet, with each character appearing in 18 different orientations, at increments of 20 degrees. Shifts arose also, because characters were extracted from the scanned image of all these characters using a technique which took no account of centroid position.

An A4 page with these characters printed on it by a laser printer at 300 dots per inch was scanned at 75 dots per inch using a UMAX Vista-S6 Scanner. The connected regions in this image were detected, and the minimum bounding-box for the largest character was calculated (56 × 57 pixels). The image was then segmented into separate characters, and each character was thinned using an algorithm due to Chen and Hsu [11]. The training set consisted of the 234 characters rotated by angles in the range $[0°, 160°]$ relative to the upright characters, and the test set of the 234 characters in the range $[180°, 340°]$. Examples of the resultant scanned, extracted and thinned characters are shown in Figures 11 and 12.

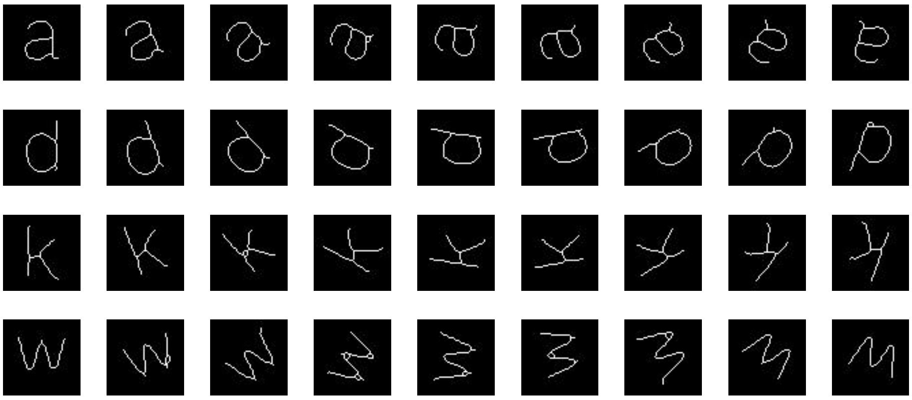


FIG. 11. Examples from the training set of thinned, scanned characters.

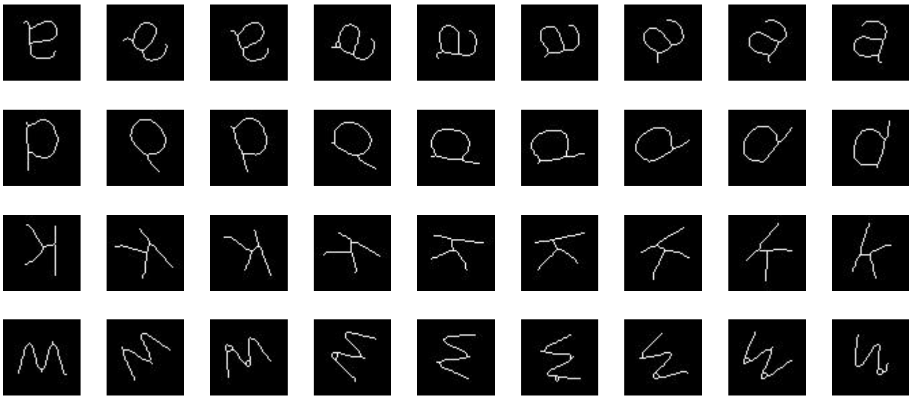
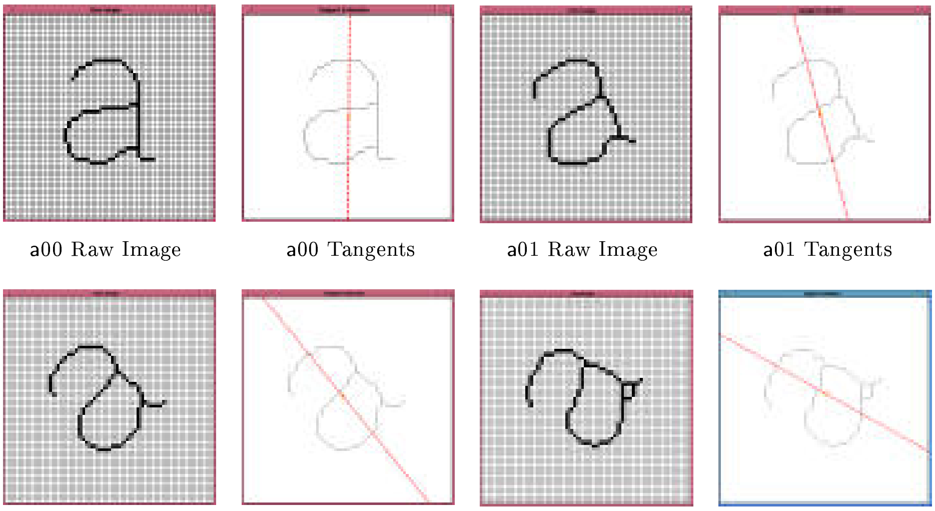


FIG. 12. Examples from the test set of thinned, scanned characters.

It is interesting to compare the subjective visual similarity between the Invariance Signatures both within and between classes for these data. Figure 13 shows the first four training examples of the letter a, accompanied by images showing the tangent estimates. The tangent estimate is represented by a line segment with the orientation of the estimated tangent.

It is not easy to interpret the similarity between these tangent representations of the contours. For this, it is necessary to see the Invariance Signatures. Figure 14 shows the Invariance Signature histograms for the patterns in Figure 13. It is apparent that there is a significant (subjective) similarity between these representations. They are not identical, as a result of the noise discussed in §6.1. For classification purposes, however, it is necessary only that the signatures be more


a00 Raw Image a00 Tangents a01 Raw Image a01 Tangents

a02 Raw Image a02 Tangents a03 Raw Image a03 Tangents

FIG. 13. Tangents estimated for training examples of the letter a.

similar within each letter class than between classes. This must be determined by experiment.

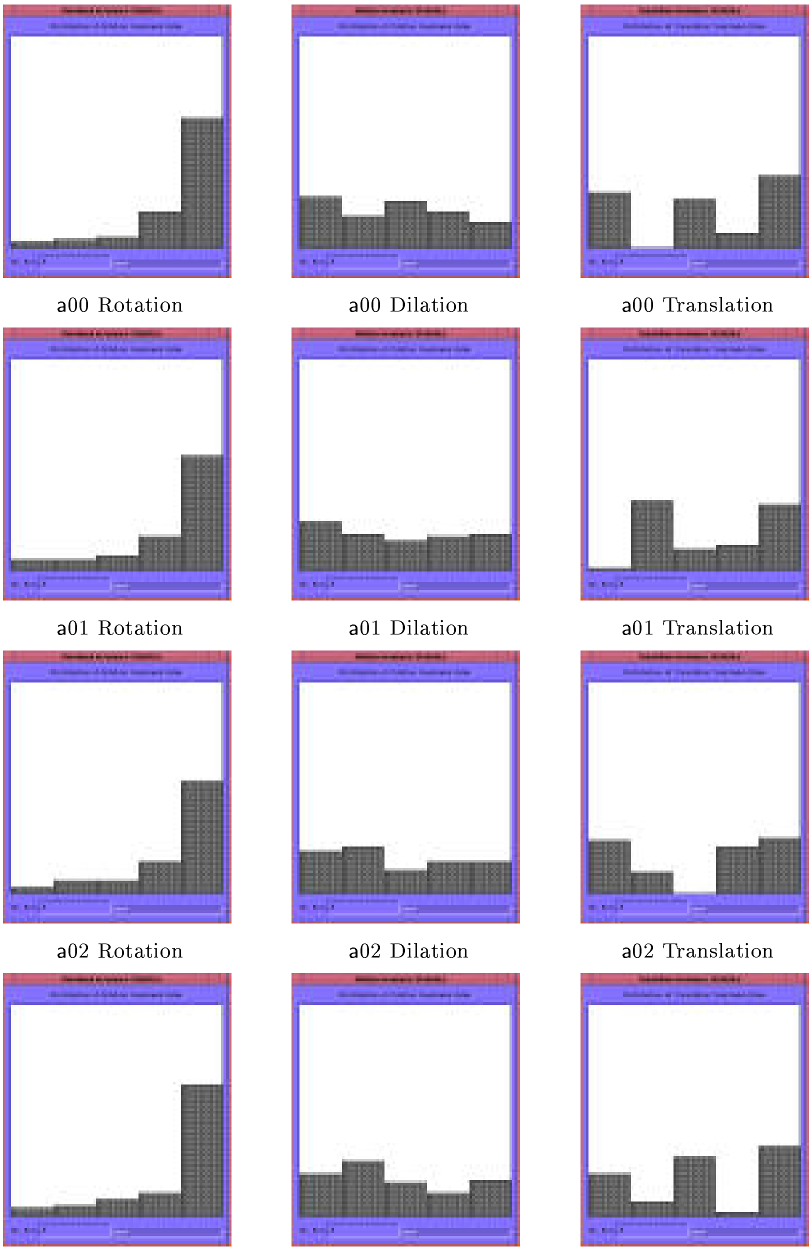
Figures 15 and 16 show the equivalent data for the letter x. These again show the marked within-class similarity, and are distinctly different to the signatures for the letter a: these letters were deliberately chosen since a is “quite rotationally-invariant”, whereas x is “quite dilationally-invariant”.

6.2.2. Selected Networks Employed for this Problem

The methodology employed was identical to that used for the synthetic data, described in §6.1.3. The MLP used had a 56×57 node input layer, and a 1×26 output layer, giving a massive 83018 independent weights to be estimated. With only 234 training patterns and 83018 parameters, this problem is almost certain to be linearly separable. This was verified by simulation.

A variety of classification modules was tried, for ISNNCs with both 5 and 10 bin Invariance Signatures. These included a linear classifier, and a variety of MLPs with differing hidden layer sizes. These experiments indicated that 5 bin Invariance Signatures were insufficient for this problem: there is a trade-off between retaining sufficient information about the Invariance Signature in the histogram and the greater sensitivity to noise caused by using a larger number of bins. It was also found that with these data the problem was not linearly separable. It also became clear that the errors introduced by the slightly inaccurate NOEM caused a significant departure from invariance.

For these reasons, the results presented are for 10 bin ISNNCs with directly-calculated (“perfect”) Local Orientation and a MLP classification module. The MLP classifier had a 3×10 node input layer, a 15 node hidden layer, and a 26 node output layer. This classifier has only 881 weights to be estimated, a reduction of 99% compared to the MLP linear classifier. This translates to a dramatic reduction in both the storage space and the training time required. Only five MLPs were



a00 Rotation a00 Dilation a00 Translation

a01 Rotation a01 Dilation a01 Translation

a02 Rotation a02 Dilation a02 Translation

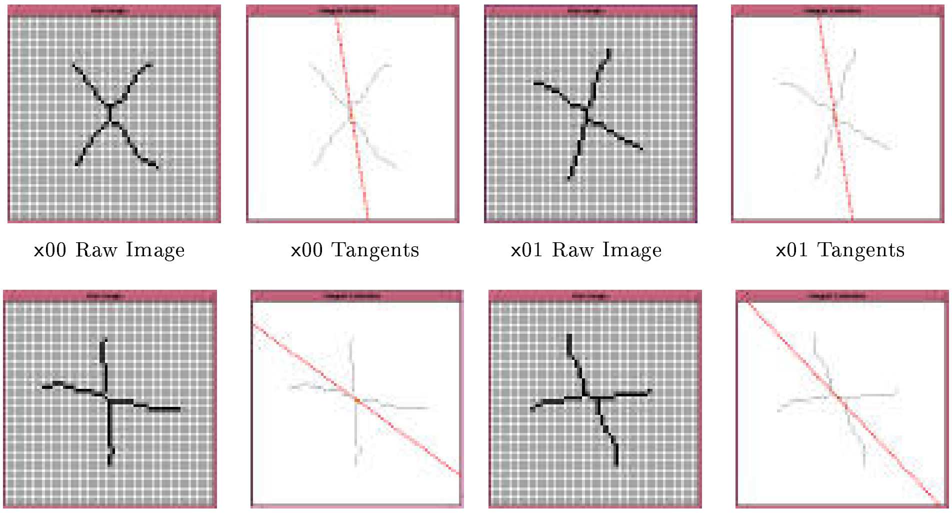
a03 Rotation a03 Dilation a03 Translation

FIG. 14. 5 bin Invariance Signatures for training examples of the letter a.

trained, partly because of the training time needed, and partly because the results were so consistent.

6.2.3. Results for MLPs

It might have been expected that the MLPs would perform better on this task than on that described in §6.1.3, since this training set contains differently transformed versions of the canonical untransformed characters. As can be seen from



x00 Raw Image x00 Tangents x01 Raw Image x01 Tangents

x02 Raw Image x02 Tangents x03 Raw Image x03 Tangents

FIG. 15. Tangents estimated for training examples of the letter x.

Table 5, the results are in fact slightly worse (average best percent correct of 13.932 ± 0.300 compared with 15.000 ± 0.454). Although better than chance (3.85% correct), these results are completely inadequate for an optical character recognition system.

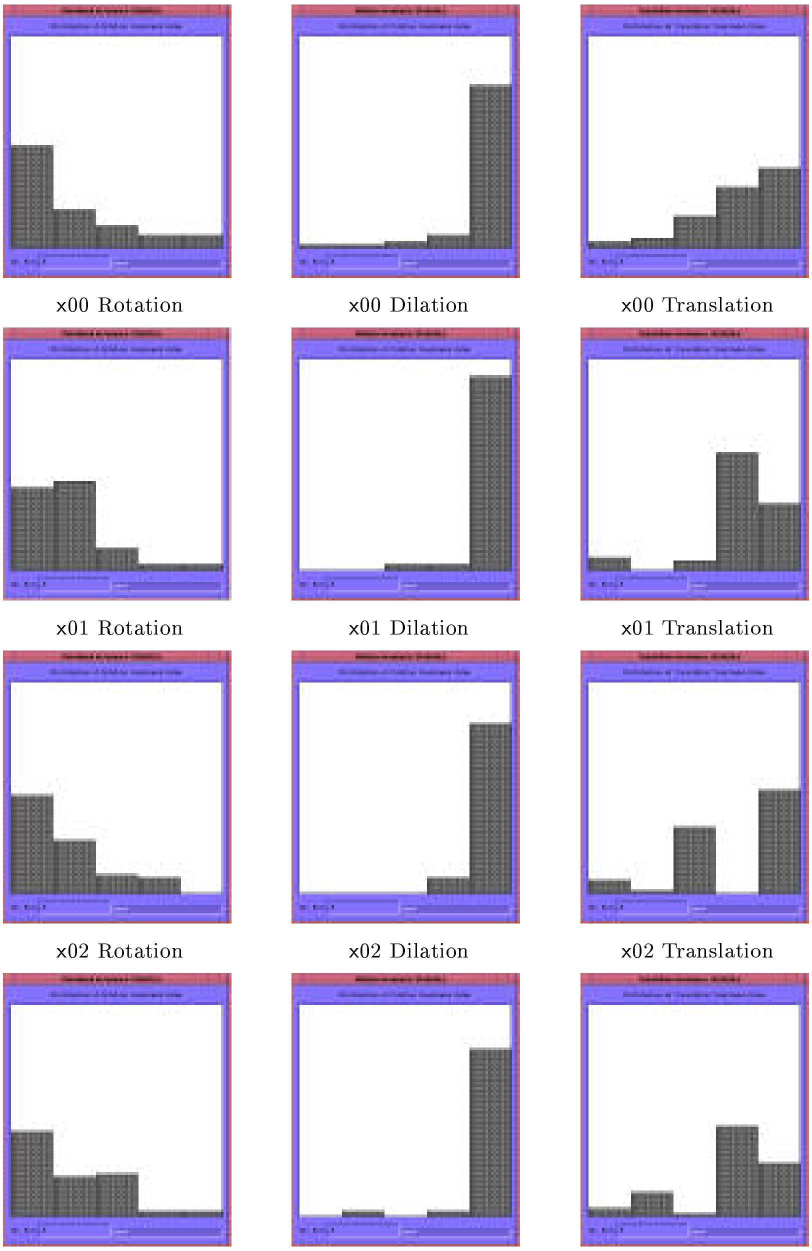
TABLE 5
Classification performance (% correct) of Traditional Neural Network Classifiers trained for 200 iterations with the data described in §6.2.1.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
$\mu \pm \sigma$	100 ± 0.0	13.9 ± 0.3	100 ± 0.0	10.9 ± 0.2

6.2.4. Results for Invariance Signature Neural Network Classifiers

The results obtained with ISNNCs appear in Table 6. The ISNNCs achieved a much higher correct classification rate on the test set than the MLPs. The failure of the ISNNCs to achieve 100% correct classification of the training set is not surprising. The test and training sets used for this problem have each character of the alphabet mapped to a separate class. Yet, as was discussed in §6.1, the sets of characters {b, d, p, q} and {n, u} are identical under rotations and reflections: transformations under which the ISNNC output is invariant. The expected training set performance for noise-free data is thus $100 \times (20 + 0.25 \times 4 + 0.5 \times 2)/26 = 85\%$ correct. Any performance on the training data better than this must be the result of the fitting of noise in the training data, indicating that results might be improved by applying techniques such as cross-validation.

In order to assess this effect, training and test sets were created in which {b, d, p, q} were assigned the same label, as were {n, u}. The results are shown in Table 7.



x00 Rotation x00 Dilation x00 Translation

x01 Rotation x01 Dilation x01 Translation

x02 Rotation x02 Dilation x02 Translation

x03 Rotation x03 Dilation x03 Translation

FIG. 16. 5 bin Invariance Signatures for training examples of the letter x.

The average final test set performance was improved by 14.8% by this re-labeling, which is very close to the maximum possible 15.4% achievable if this were the only source of error.

The residual difference between training and test set error is generalization error, rather than invariance error. These networks were trained with only 9 examples of each character, and these examples are quite noisy. There is the unavoidable quantization noise, but there are also some quite marked artifacts, such as loops

TABLE 6
Classification performance (% correct) of 10 bin Invariance Signature Neural Network Classifiers (with perfect Local Orientation Extraction) trained for 40000 iterations with the data described in §6.2.1.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
Network 1	95.7	72.7	95.7	71.8
Network 2	96.2	72.2	96.2	70.9
Network 3	96.2	73.1	96.2	69.7
Network 4	95.3	73.9	95.3	70.5
Network 5	94.9	71.8	94.9	71.4
Network 6	95.7	72.7	95.7	68.8
Network 7	96.2	73.5	96.2	69.7
Network 8	97.4	72.2	97.4	70.5
Network 9	94.4	74.4	94.4	69.7
Network 10	96.2	70.9	96.2	70.9
$\mu \pm \sigma$	95.8 ± 0.8	72.7 ± 1.0	95.8 ± 0.8	70.4 ± 0.9

TABLE 7
Classification performance (% correct) of 10 bin Invariance Signature Neural Network Classifiers (with perfect Local Orientation Extraction) trained for 10000 iterations with the data described in §6.2.1, modified to label characters which can be transformed into each other as the same character.

	Best Performance		Final Performance	
	Training Data	Test Data	Training Data	Test Data
Network 1	98.7	87.2	98.7	87.2
Network 2	99.1	83.8	99.1	82.9
Network 3	99.6	86.8	99.1	85.9
Network 4	99.6	86.8	99.1	85.0
Network 5	98.7	88.0	98.7	88.0
Network 6	99.6	85.9	99.1	85.5
Network 7	100	85.9	100	83.8
Network 8	98.3	86.3	98.3	85.0
Network 9	97.9	86.8	97.9	85.0
Network 10	99.1	84.6	99.1	83.8
$\mu \pm \sigma$	99.1 ± 0.6	86.2 ± 1.2	99.1 ± 0.6	85.2 ± 1.5

introduced by the thinning algorithm, one of which can be seen in pattern a03 in Figure 13. There are also two erroneous test patterns, the result of clipping in the segmentation process. These were retained, as such errors can and do occur in practical applications.

6.2.5. Failure Analysis

The errors made by the ISNNCs are not random. To illustrate this, a failure analysis is presented in Table 8 for Network 5 from Table 7, showing how the test patterns were misclassified. Patterns which are perceptually similar are responsible for many of the misclassifications. This means that prior information about likely errors could be used in conjunction with these classifications to aid error correction. This analysis indicates that ISNNCs often make errors that appear “human”, which is a promising indication that the Invariance Signature measures contour similarity in a way similar to humans.

TABLE 8
Failure analysis for Network 5 from Table 7.

Misclassifications						
d → n	f → t	f → j	i → l	i → l	i → l	i → l
i → l	i → l	j → r	k → f	k → r	k → f	l → i
m → h	n → b	n → b	q → n	r → f	r → y	r → k
r → i	t → f	t → f	t → f	t → f	u → h	w → m

Inspection of the thinned patterns used indicated that the patterns for the letters {i, j, l} were little more than straight lines. If these were to be relabeled as the same character, final test set performance for Network 5 would improve to 91.026% correct. If the same were done for {f, t}, which are also extremely similar, test set performance would be 93.162% correct.

The misclassifications of patterns to the classes b and n may be due to the fact that the re-labeled dataset implies a non-uniform prior probability distribution for these classes: b was used as the target class for input patterns corresponding to {b, d, p, q}, and thus occurs four times more frequently in the training data than standard classes. Similarly, n was used as the target for inputs {n, u}. Networks will tend to “guess” these classes more frequently than the others [3].

Given the extremely small training set, this is a remarkable result, comparable with character recognition results achieved by others with thousands of training patterns. Such re-relabeling is an essential feature of a *truly* invariant optical character recognition system, since some characters are inherently ambiguous. Others are extremely similar, and noise can render them indistinguishable. In practical optical character recognition systems, a dictionary is used to verify recognized words, and context is used to correct erroneously labeled patterns. Alternatively, the orientation of correctly classified unambiguous characters could be used to infer the correct labeling of ambiguous characters.

Perhaps most importantly, these results show that ISNNCs can correctly classify patterns which have been transformed by arbitrarily large amounts. The shifts and rotations of the input images are not restricted to small perturbations of the training patterns. This indicates that the ISNNCs are performing truly invariant pattern recognition, rather than interpolation-based generalization.

This study should be considered to be a “proof of concept”, both for the Invariance Signature as a contour descriptor, and for MBNNs which classify on that basis. It is not intended to be a large-scale experiment which corresponds to a real application. Such a study would require much greater quantities of data than are used here. We believe, however, that the experiments presented demonstrate that the Invariance Signatures are genuinely useful for robust invariant pattern recognition. The quality of the results obtained is very pleasing, when the size of the training sets is compared to those used in other neural network approaches to optical character recognition.

7. COMPARISON WITH OTHER METHODS

In order to compare the performance of the ISNNC with that of alternative systems designed for shift-, rotation- and scale-invariant pattern recognition, a correlation classifier based on the Fourier-Mellin (FM) transform [10, 7] was applied to the data described in §6.2.1.

For further comparisons, two-dimensional invariance signature histograms for the data were created, in the space defined by $\left[\begin{array}{cc} \iota_{rot} & \iota_{trans} \end{array}\right]^{\mathrm{T}}$. These histograms were classified both by histogram intersection with the training set, and using a neural network.

7.1. The Fourier-Mellin classifier

As discussed in §2.1, the amplitude spectrum of the Fourier transform (denoted $\mathcal{F}$) of an image is shift-invariant. Rotation in the spatial domain maps to rotation in the frequency domain, and scaling to inverse scaling. Applying a log-polar transform in the frequency domain thus leads to a coordinate system (ρ, θ) in which rotations and scalings in the image domain are mapped to shifts. The amplitude spectrum of the Fourier transform of the resultant function is thus shift-, rotation- and scale-invariant. Since the Mellin transform of a function defined on $\mathbb{R}^+$ is just the Fourier transform of the function after the coordinate has been logarithmically distorted, this combination of transforms is sometimes known as the Fourier-Mellin transform. Summarizing:

$\lvert\mathcal{F}(\mathcal{I}(x,y))\rvert = \lvert\mathcal{I}_\omega(\omega_1,\omega_2)\rvert$	magnitude of Fourier transform: shift-invariant
$\mathcal{I}_\omega(\omega_1,\omega_2) \rightarrow \mathcal{I}_{\mathrm{lp}}(\rho,\theta)$	log-polar transform
$\lvert\mathcal{F}(\lvert\mathcal{I}_{\mathrm{lp}}(\rho,\theta)\rvert)\rvert$	magnitude of Fourier-Mellin tranform: shift-, rotation- and scale-invariant

It must be noted that the phases of both the initial and second Fourier transforms have been discarded, meaning that this representation is very "lossy". Nonetheless, as is shown below, sufficient information is retained for classification purposes in some applications.

Classification can be performed by calculating the correlation between the magnitude the FM tranform of the image to be classified with those of the example images for each class. The image is assigned to the class of the image with which it has the greatest correlation.

Experiments were performed where the set of example images contained only the first image of training data described in §6.2.1 for each character, and for the case where correlations were performed with the entire training set.

7.2. The histogram intersection classifier

Histogram intersection is metric which is frequently applied in content-based image retrieval applications, where it is most often used with colour histograms [40]. It is natural to apply it to Invariance Signatures, since they are in fact histograms. The intersection $d(H_1, H_2)$ between two N-bin histograms H_1 and H_2 is

$$d(H_1, H_2) = \frac{\sum_{n=1}^{N} \min(H_1[n], H_2[n])}{\min\left(\sum_{n_1=1}^{N} H_1[n_1], \sum_{n_2=1}^{N} H_2[n_2]\right)}. \tag{39}$$

A simple classification is performed by calculating the intersections between the Invariance Signature of the image to be classified with those of the example images for each class. The image is assigned to the class of the image with which it has the greatest intersection.

As for the FM classifier, experiments were performed where the set of example images contained only the first image of training data described in §6.2.1 for each character, and for the case where intersections were performed with the entire training set.

7.3. The histogram-based neural network classifier

A problem with the simple histogram intersection metric defined above is that it does not take into account the contributions from neighbouring bins. Quadratic histogram intersection measures exist, which use weighted intersections of *all* bins. How these weights should be defined, however, is a difficult problem. An alternative is to train a MLP to classify the histograms. In §5, we described a neural network classifier which took as its input the three one-dimensional histograms corresponding to the Invariance Signatures with respect to shift, rotation and scaling. Here we report results for a system using the 10 × 10 two-dimensional histogram corresponding to the distribution of points in the invariance space with axes ι_{rot} and ι_{trans}.

A variety of MLPs were tried, beginning with a linear classifier. The data described in §6.2.1 were used for training. In §6.2, it was noted that for the dataset used sets of characters {b, d, p, q} and {n, u} are identical under rotations and reflections. The MLPs were thus trained with 22 output classes, corresponding the the letters of the alphabet, taking into account these equivalences. Results are presented for a network with a 10 node hidden layer. Larger hidden layers were not found to improve classification performance.

7.4. Classification results

Table 9 shows the classification performances of these methods. The shortcomings of simple histogram intersection are apparent. The improvement gained by using a MLP to classify the 2D histograms indicates that the use of relationships between bins can improve performance significantly. These results also show that no significant representational power was lost by considering the three separate Invariance Signatures with the ISNNC, as opposed to the full invariance space. This permits an extremely significant improvement in efficiency.

It is clear that the Fourier-Mellin classifier using all training examples significantly outperforms all the others. It must be remembered, however, that this comes at great computational cost. The classification of a single pattern requires

TABLE 9

Classification performance (% correct) of various invariant classifiers for the the test data described in §6.2.1.

	Classification Performance	
	26 classes	22 classes
FM classifier, single examples	79.1	89.7
FM classifier, all examples	87.6	97.4
2D histogram intersection, single examples	33.3	39.7
2D histogram intersection, all examples	58.5	77.8
MLP with 2D histogram input		86.0 ± 1.1
Invariance Signature Neural Network Classifier	72.7 ± 1.0	86.2 ± 1.2

the computation of two Fourier transforms, a log-polar transform, and a correlation with each of the 234 target pattern exemplars. In this study, each of these operations had to be carried out on a 64 × 64 image. This must be compared with a single pass through the ISNNC.

It is also to be expected that the performance of the ISNNC would improve with higher resolution input images which would allow more accurate tangent estimation, and correspondingly more precise Invariance Signatures. Nevertheless, the performance of the Invariance Signature technique on these low-resolution images is quite impressive for a contour-based technique.

8. CONCLUSION

In this paper an invariant feature of two-dimensional contours was developed: the Invariance Signature. We believe that the Invariance Signature is a powerful descriptor of contour shape, which is closely-related to measures employed in human perception. Its application is by no means limited to neural networks.

The development of the Invariance Signature was inspired by the desire to find an invariant contour descriptor which was suitable for calculation in a neural network, and which corresponded well to theories of human contour perception. Since Lie group theory provides the link between the local changes in the positions of points under the action of a transformation and the global specification of the transformation, it provides the natural starting point. The Invariance Signature is a global measure of the degree of invariance of a given contour with respect to a set of Lie transformations, which, however, is constructed from *local* calculations. It is this that makes the Invariance Signature attractive for use in an MBNN.

The core of the Invariance Signature approach is this: rather than seeking individual invariant features of a contour, the Invariance Signature measures the *degree to which the contour is invariant* under a transformation. The statistics of these departures from invariance are themselves an invariant descriptor of the contour.

In order to be useful, the Invariance Signature must not only be invariant, but must retain sufficient information for contour classes to be distinguished. That this is so is demonstrated in §6. When applied to noise-free, unambiguous data, the ISNNC produced perfect results. Pleasing results were also obtained using scanned data. Although outperformed by 11% by a Fourier-Mellin classifier, the ISNNC uses a more efficient representation of shape. The fact that the entire feature extraction and classification procedure could be compiled into a simple neural network means that the result was obtained at a very much lower computational cost. Moreover, the low-resolution images used in this work are not ideally suited to a contour-based invariant. These experiments indicate that the Invariance Signature can be successfully employed for the recognition of scanned characters independent of rotations and shifts, and that this technique can be implemented in a model-based neural network.

REFERENCES

1. Jürgen Altmann and Herbert J.P. Reitböck. A fast correlation method for scale- and translation-invariant pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):46–57, January 1984.

2. Dana Harry Ballard and Christopher M. Brown. *Computer Vision*. Prentice–Hall, Englewood Cliffs, New Jersey 07632, 1982.

3. Etienne Barnard and Elizabeth C. Botha. Back-propagation uses prior information efficiently. *IEEE Transactions on Neural Networks*, 4(5):794–802, September 1993.

4. Eamon Barrett, Gregory Gheen, and Paul Payton. Representation of three-dimensional object structure as cross-ratios of determinants of stereo image points. In Joseph L. Mundy, Andrew Zisserman, and David Forsyth, editors, *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, pages 47–68, Berlin, 1994. Springer-Verlag.

5. Christopher Brown. Numerical evaluation of differential and semi-differential invariants. In Joseph Mundy and Andrew Zisserman, editors, *Geometric Invariance in Computer Vision*, Series: Artificial intelligence, pages 215–227. The MIT Press, Cambridge, MA, USA, 1992.

6. Terry Caelli and Peter Dodwell. Orientation-position coding and invariance characteristics of pattern discrimination. *Perception and Psychophysics*, 36(2):159–168, 1984.

7. Terry M. Caelli and Zhi-Qiang Liu. On the minimum number of templates required for shift, rotation and size invariant pattern recognition. *Pattern Recognition*, 21(3):205–216, 1988.

8. Terry M. Caelli, David McG. Squire, and Tom P.J. Wild. Model-based neural networks. *Neural Networks*, 6:613–625, 1993.

9. T.M. Caelli, G.A.N. Preston, and E.R. Howell. Implications of spatial summation models for processes of contour perception: A geometric perspective. *Vision Research*, 18:723–734, 1978.

10. D. Casasent and D. Psaltis. Position, rotation, and scale invariant optical correlation. *Applied Optics*, 15(7):1795–1799, July 1976.

11. Y-S. Chen and W-H. Hsu. A modified fast parallel algorithm for thinning digital patterns. *Pattern Recognition*, 7:99–106, 1988.

12. Gloria Chow and Xiaobo Li. Towards a system for automatic facial feature detection. *Pattern Recognition*, 26(12):1739–1755, December 1993.

13. Christopher Coelho, Aaron Heller, Joseph L. Mundy, David A. Forsyth, and Andrew Zisserman. An experimental evaluation of projective invariants. In Joseph Mundy and Andrew Zisserman, editors, *Geometric Invariance in Computer Vision*, Series: Artificial intelligence, pages 87–104. The MIT Press, Cambridge, MA, USA, 1992.

14. James B. Cole, Hiroshi Murase, and Seiichiro Naito. A Lie group theoretic approach to the invariance problem in feature extraction and object recognition. *Pattern Recognition Letters*, 12:519–523, September 1991.

15. M. Ferraro and T. Caelli. The relationship between integral transform invariances and Lie group theory. *Journal of the Optical Society of America (A)*, 5:738–742, 1988.

16. Mario Ferraro and Terry M. Caelli. Lie transform groups, integral transforms, and invariant pattern recognition. *Spatial Vision*, 8(1):33–44, 1994.

17. David Forsyth, Joseph L. Mundy, and Andrew Zisserman. Transformational invariance - a primer. *Image and Vision Computing*, 10(1):39–45, 1992.

18. W. C. Hoffman. The Lie algebra of visual perception. *Journal of Mathematical Psychology*, 3:65–98, 1966.

19. W. C. Hoffman. The Lie transformation group approach to visual neuropsychology. In E. Leewenberg and H. Buffart, editors, *Formal theories of visual perception*, pages 27–66. Wiley, New York, 1978.

20. D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

21. Anil K. Jain. *Fundamentals of digital image processing.* Prentice-Hall information and system sciences series. Prentice-Hall International, London, 1989.

22. Alireza Khotanzad and Jiin-Her Lu. Classification of invariant image representations using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):1028–1038, June 1990.

23. Reiner Lenz. Group invariant pattern recognition. *Pattern Recognition*, 23(1/2):199–217, 1990.

24. S.Z. Li. Matching: Invariant to translations, rotations and scale changes. *Pattern Recognition*, 25(6):583–594, 1992.

25. Xiaobo Li and Nicholas Roeder. Experiments in detecting face contours. In *Proceedings of Vision Interface '94*, pages 96–103, Banff, Canada, May 1994.

26. Feng Lin and Robert D. Brandt. Towards absolute invariants of images under translation, rotation and dilation. *Pattern Recognition Letters*, 14:369–379, May 1993.

27. Marvin Minsky and Seymour Papert. *Perceptrons. An introduction to computational geometry.* The MIT Press, Cambridge, London, 1969.

28. Joseph L. Mundy and Andrew Zisserman. Towards a new framework for vision. In Joseph Mundy and Andrew Zisserman, editors, *Geometric Invariance in Computer Vision*, Series: Artificial intelligence, pages 1–39. The MIT Press, Cambridge, MA, USA, 1992.

29. Stavros J. Perantonis and Paulo J.G. Lisboa. Translation, rotation, and scale invariant pattern recognition by higher-order neural networks and moment classifiers. *IEEE Transactions on Neural Networks*, 3(2):241–251, March 1992.

30. David A. Pintsov. Invariant pattern recognition, symmetry, and radon transforms. *Journal of the Optical Society of America (A)*, 6(10):1544–1554, October 1989.

31. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing.* Press Syndicate of the University of Cambridge, Cambridge, U.K., 2 edition, 1992.

32. A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, Orlando, FL., 1982.

33. Jacob Rubinstein, Joseph Segman, and Yehoshua Zeevi. Recognition of distorted patterns by invariance kernels. *Pattern Recognition*, 24(10):959–967, 1991.

34. S.L. Salas, Einar Hille, and John T. Anderson. *Calculus: One and several variables, with analytic geometry.* John Wiley & Sons, New York, 5 edition, 1986.

35. Joseph Segman, Jacob Rubinstein, and Yehoshua Y. Zeevi. The canonical coordinates method for pattern deformation: Theoretical and computational considerations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1171–1183, December 1992.

36. Lilly Spirkovska and Max B. Reid. Higher-order neural networks applied to 2D and 3D object recognition. *Machine Learning*, 15(2):169–199, 1994.

37. David McG. Squire. *Model-based Neural Networks for Invariant Pattern Recognition.* PhD thesis, School of Computing, Curtin University of Technology, Perth, Western Australia, October 1996. (available from: `http://cuiwww.unige.ch/~squire/publications/thesis.html`).

38. David McG. Squire and Terry M. Caelli. Shift, rotation and scale invariant signatures for two-dimensional contours, in a neural network architecture. In Stephen W. Ellacott, John C. Mason, and Iain J. Anderson, editors, *Mathematics of Neural Networks: Models Algorithms and Applications*, Statistics and OR, pages 344–348, Boston, July 1995. Kluwer Academic Publishers.

39. Sergei Startchik. *Geometric and Illumination Invariant Object Representation: Application to Content-based Image Retrieval.* Ph.D. Dissertation No. 3009, University of Geneva, Switzerland, July 1998.

40. Michael J. Swain and Dana H. Ballard. Indexing via color histograms. In *Proceedings of the DARPA Image Understanding Workshop*, pages 623–630, Pittsburgh, PA, USA, 1990.

41. Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

42. Luc J. Van Gool, Theo Moons, Eric Pauwels, and André Oosterlinck. Semi-differential invariants. In Joseph Mundy and Andrew Zisserman, editors, *Geometric Invariance in Computer Vision*, Series: Artificial intelligence, pages 157–192. The MIT Press, Cambridge, MA, USA, 1992.

43. Geoffrey E. Vanderkooy. Line matching for uncalibrated cameras using projective invariants. Technical report, Vision and Electronic Measurement Laboratory, Department of Mechanical Engineering, University of Victoria, Canada, 1996.

44. Harry Wechsler. *Computational Vision.* Academic Press Inc., 1250 Sixth Avenue, San Diego, CA 92101, 1990.

45. Issac Weiss. Noise-resistant invariants of curves. In Joseph Mundy and Andrew Zisserman, editors, *Geometric Invariance in Computer Vision*, Series: Artificial intelligence, pages 135–156. The MIT Press, Cambridge, MA, USA, 1992.

46. Christopher Zetzsche and Terry Caelli. Invariant pattern recognition using multiple filter image representations. *Computer Vision, Graphics and Image Processing*, 45:251–262, 1989.