Chapter 2

General Methods for Invariant Pattern Recognition

The ability to perceive the permanent features of the visual environment is something which humans take for granted. Indeed, it is hard to imagine a world in which we could not. We do not even notice that we recognize objects and patterns independently of changes in lighting conditions, shifts of the object or observer, or changes in orientation and scale. [Gibson, 1966][1] stated this rather well:

> It can be shown that the easily measured variables of stimulus energy, the intensity of light, sound, odor, and touch, for example, vary from place to place and from time to time as the individual goes about his business in the environment. The stimulation of receptors and the presumed sensations, therefore, are variable and changing in the extreme, unless they are experimentally controlled in a laboratory. The unanswered question of sense perception is how an observer, animal or human, can obtain constant perceptions in everyday life on the basis of these continually changing sensations. For the fact is that animals and men do perceive and respond to the permanent features of the environment as well as to the changes in it.
>
> Besides the changes in stimuli from place to place and from time to time, it can also be shown that certain higher-order variables of stimulus energy – ratios and proportions, for example – do not change. They remain invariant with movements of the observer and with changes in the intensity of stimulation. And it will be shown that these invariants of the energy flux at the receptors of an organism correspond to the permanent properties of the environment.
>
> The active observer gets invariant perceptions despite varying sensations. He perceives a constant object by vision despite changing sensations of light; he perceives a constant object by feel despite changing sensations of pressure; he perceives the same source of sound despite changing sensations of loudness in his ears. The hypothesis is that constant perception depends on the ability of the individual to

[1] quote taken from [Wechsler, 1990, p. 96].

> detect the invariants, and that he ordinarily pays no attention whatever to the flux of changing sensations.

In this thesis we are concerned with only a small subset of the problems outlined above: 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 do unthinkingly when reading a document such as an architectural drawing.

An image in the Cartesian (x, y) domain may be defined as a function $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x, y) = z$, where z is the intensity of the image at coordinates (x, y). The invariant pattern recognition problem is to recognize images as being in some sense "the same" even though they have undergone a variety of allowed transformations. In two-dimensional pattern recognition, one typically desires invariance with respect to a combination of some or all of the shift, rotation and scale transformations.

A transformation may be denoted by $T_{\mathbf{a}}$, where $\mathbf{a}$ is a vector of parameters specifying the transformation. The deformed image is denoted by $f_{T_{\mathbf{a}}} = f\left(T_{\mathbf{a}}x, T_{\mathbf{a}}y\right)$. If the image is not invariant under the action of the transformation, then in general $f_{T_{\mathbf{a}}} \neq f_{T_{\mathbf{a}'}}$ if $\mathbf{a} \neq \mathbf{a}'$. The problem is thus to find a means of identifying f and $f_{T_{\mathbf{a}}}$ as instances of the same image.

Many different approaches to this problem have been used in computer vision, and more specifically in the application of neural networks to invariant pattern recognition. Some involve finding representations of the image in spaces other than the natural (x, y) space such that the representation in the new space is invariant under the desired transformations. Others search for specific features in the image which define "parts", and then seek invariant relationships between these parts. Still others fit analytic curves to parts of the image, and use algebraic or differential invariants of the parameters of these curves. In this chapter an attempt will be made to review these approaches, and to identify their strengths and weaknesses.

2.1 Integral Transforms

The most general form of an integral transform of an image $f(x, y)$ is

$$\mathscr{O}\left[f(x,y)\right] = F(u,v) = \iint\limits_{\mathbb{R}^2} f(x,y)K(u,v;x,y)\,dx\,dy, \tag{2.1}$$

where $K(u, v; x, y)$ is the *kernel* of the transform, and $F(u, v)$ is the representation of the image in the transform domain (u, v). This operation is also frequently described as *filtering* the signal $f(x, y)$. $K(u, v; x, y)$ is the point spread function of the filter, and $F(u, v)$ is the filtered signal.

Integral transforms are useful for invariant pattern recognition if it is possible to choose kernels so that the transformed images are invariant (or have invariant components) under some specified transformations.

It is frequently the case that integral transforms used in invariant pattern recognition have two components:

- one that is invariant under the action of some transformation
- one that encodes the transformation with respect to some origin.

Invariant matching can be achieved using the first component, while the second can be used to recover the transformation. If the transformation can be inverted to recover the original image, then the representation is unique. If the invariant component is unique up to the specified transformations, then the representation can be called "invariant in the strong sense". Invariant, but not unique, representations can be said to be "invariant in the weak sense" [Ferraro and Caelli, 1994].

These two components are frequently the amplitude and phase spectra of the complex transform of the image.[2] The transform may be written as

$$\mathscr{O}\left[f\left(x,y\right)\right] = A\left(u,v\right)e^{-i\phi(u,v)}. \tag{2.2}$$

Considering a one-parameter transformation T_a, the conditions for strong invariance can be met if

$$\mathscr{O}\left[f_{T_a}(x,y)\right] = A\left(u,v\right)e^{-i[\phi(u,v)+ua]}, \tag{2.3}$$

since the amplitude spectrum A is invariant under the transformation T_a (*i.e.* it does not depend on a), and the parameter a of the transformation can be recovered from the phase spectrum. The only further requirement is that $A(u,v)$ is unique for each image.

There are limitations on the construction of strongly-invariant integral transforms, which will be discussed in the following matter. Whether strongly-invariant representations are necessary, or even desirable, is a moot point, especially since in many pattern recognition tasks one wishes to classify a pattern as a member of a class, rather than to recognize uniquely each individual pattern.

[2] Throughout this thesis, the symbol i will be used to denote $\sqrt[+]{-1}$, except when it appears as a subscript, where it indicates an indexing variable.

2.1.1 The Fourier Transform

Consider a one-dimensional function $f(x)$. Its Fourier transform is

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-iux}dx. \tag{2.4}$$

We denote the operation of taking the Fourier transform by the operator $\mathscr{F}$, so that we can write $F(u) = \mathscr{F}f(x)$. Now consider a version of the function f which is transformed by the one-parameter translation T_a specified by a, $f_{T_a}(x) = f(x-a)$. Taking the Fourier transform of $f_{T_a}(x)$, we obtain

$$\mathscr{F}f_{T_a}(x) = \int_{-\infty}^{\infty} f(x-a)e^{-iux}dx. \tag{2.5}$$

Changing variables to $x' = x - a$,

$$\begin{aligned} \mathscr{F}f_{T_a}(x) &= \int_{-\infty}^{\infty} f(x')e^{-iu(x'+a)}dx' \\ &= e^{-iua}\int_{-\infty}^{\infty} f(x')e^{-iux'}dx' \\ &= e^{-iua}\mathscr{F}f(x). \end{aligned} \tag{2.6}$$

Thus we see that a shift in the spatial domain produces only a phase-shift in the complex Fourier frequency domain. Taking the magnitudes of the transforms, we have

$$\left|\mathscr{F}f_{T_a}(x)\right| = \left|\mathscr{F}f(x)\right|. \tag{2.7}$$

The Fourier transform thus has two important properties:

- The magnitude of the Fourier transform is invariant under shifts in the spatial domain.[3]
- The shift a can be recovered, as it is encoded in the phase.

The Fourier transform is thus strongly-invariant with respect to shifts. These properties extend to multidimensional Fourier transforms [Rosenfeld and Kak, 1982], and form the basis of many invariant pattern recognition techniques [for example, Altmann and Reitböck, 1984; Lin and Brandt, 1993; Caelli and Liu, 1988; Ferraro and Caelli, 1988; Ferraro and Caelli, 1994; Rubinstein, Segman and Zeevi, 1991; Segman, Rubinstein and Zeevi, 1992].

[3] The amplitude of the power spectrum is thus invariant. Since the Fraunhofer (far-field) diffraction pattern of an aperture is just its two-dimensional Fourier transform, and the intensity of light corresponds to its power, a marvelous demonstration of this invariance can be carried out with a laser, lenses and an optical bench.

2.1.2 The Fourier-Mellin Transform

The Mellin[4] transform of a function $f(x)$ defined over the positive reals is the complex function $M(s)$, where

$$M(s) = \int_0^\infty f(x)x^{s-1}dx. \tag{2.8}$$

The Mellin transform of a function is closely related to its Fourier transform, and this is important for its application to invariant pattern recognition. If we change variables in Equation 2.8 to

$$x' = \frac{-\ln(x)}{i}, \tag{2.9}$$

we obtain

$$M(s) = -i\int_{-\infty}^\infty f(e^{-ix'})e^{-isx'}dx'. \tag{2.10}$$

Equation 2.10 shows that the Mellin transform of a function over the positive reals is, up to a multiplicative constant, just the Fourier transform of the function after the coordinate has been logarithmically-distorted. This means that the magnitude of the Mellin transform will be invariant under shifts in the logarithmically-distorted domain.

To see the utility of this property for invariant pattern recognition, consider a representation of an image in polar coordinates: $r \in [0,\infty)$ and $\theta \in [0, 2\pi)$. The domain of r makes it a suitable candidate for the application of the Mellin transform. Thus, we change variables to $\rho = \frac{-\ln(r)}{i}$. We know that the magnitude of the Mellin transform will be invariant under transformations $T_\mathbf{a}$ of the image such that $T_\mathbf{a}\rho = \rho+\alpha$, where α is some constant.

Such a transformation is a scaling of the image by some positive constant λ. We see that

$$T_\mathbf{a}r = \lambda r, \tag{2.11}$$

so that

$$\begin{aligned} T_\mathbf{a}\rho &= \frac{-\ln(\lambda r)}{i} \\ &= \frac{-(\ln(r)+\ln(\lambda))}{i} \\ &= \rho + \alpha \end{aligned} \tag{2.12}$$

[4] Hjalmar Mellin, 1854-1933.

where $\alpha = \frac{-\ln(\lambda)}{i}$. Thus the magnitude of the Mellin transform of the radial coordinate of the image is invariant under scaling of the image.

We observe that a rotation of image by an angle ϕ corresponds to a shift in the angular coordinate θ:

$$\theta' = \theta + \phi. \tag{2.13}$$

Thus the magnitude of the Fourier transform of the angular coordinate will be invariant under rotations of the image. The image representation given jointly by the Mellin transform of the radial coordinate and the Fourier transform of the angular coordinate is thus invariant under rotation and scaling of the image. This is the Fourier-Mellin transform, which has been widely employed for rotation- and scale-invariant pattern recognition.

Biological Instances of Log-Polar Mapping

The notion of changing a scaling to a shift by a logarithmic transformation of the coordinates occurs in many areas. In fact, the log-polar coordinate system described above seems to have a biological analogue. Altmann and Reitböck (1984, p. 46) say that

> The principle of logarithmic mapping seems to play a role in biological sensory systems too. In the simian visual system, as well as in the visual system of the cat, the mapping of the central $20° - 30°$ of retinal space onto area 17 of the visual cortex approximates a polar coordinate transformation together with a logarithmic distortion of the r-axis.

Since biological systems can shift objects to the centre of the field of view by movement of the eyes, it seems logical that a mapping that facilitates scale and rotation invariant recognition would be most useful. This is not to say that biological systems also compute Fourier transforms of these representations. Other methods can take advantage of representations in which transformations are reduced to shifts, as will be discussed in §2.3.

Other Invariances Using the Fourier and Mellin Transforms

We have seen above that a scale- and rotation-invariant representation of an image can be obtained by taking the Mellin transform of its radial coordinate and the Fourier transform of its angular coordinate. By combining these transforms in other ways, it is possible to obtain representations of images which are invariant under other transformations. As an example, we will consider Altmann and Reitböck (1984), who obtain a scale- and shift-invariant representation by taking the absolute value of the Mellin transform of the normalized Fourier amplitude spectrum of an image.

We have seen that the Fourier amplitude spectrum is shift-invariant. It is easy to show that a coordinate scaling in the image domain ($x' = \alpha x$) produces a coordinate scaling[5] and a change in magnitude in the Fourier domain. The magnitude change can be removed by normalizing the amplitude spectrum by its magnitude at $u = 0$. This normalized Fourier amplitude spectrum is thus shift-invariant, and changes only in scale when the image is scaled. The Mellin transform of this will thus be invariant under both shifts and scalings of the image.

It is important to note that, since the Fourier phase spectrum is discarded and the amplitude is normalized, this representation is not unique, and therefore cannot be inverted. It thus does not meet the criteria for strong invariance. Nevertheless, Altmann and Reitböck (1984) show that it can still be a useful representation. Whatever the properties of the exact, continuous transform derived in theory, these will not be preserved in a discrete digital implementation. Exact invariance is not preserved due to sampling and border effects, so *strong invariance* can *never* be achieved in a discrete system. This is a limitation common to all digital invariant pattern matching methods.

Altmann and Reitböck (1984) derive criteria for the preservation of "essential information" in the discrete Fourier transform of an image. For an $N \times N$ image, the total object size should be $n_t \lesssim N/4$, and the smallest permissible detail size is $n_d \gtrsim 4$, measured in pixels. Of course, even once suitable images have been obtained, and the invariant representations computed, the problem of matching these representations to those of the template patterns remains.

2.1.3 Canonical Coordinates

We have seen above that the notion of transforming a pattern into a space in which the action of a transformation $T_{\mathbf{a}}$ reduces to a shift along a coordinate axis is useful for invariant pattern recognition. Indeed, it turns out that this notion can be generalized into a scheme for computing the kernel $K(u, v; x, y)$ of an integral transform which will map an image into a space in which specified transformations are reduced to shifts, and which satisfies the strong invariance condition. This is possible if we consider the transformations of the image to be Lie transformation groups.[6] This is not overly restrictive, as most transformations of interest for (especially two-dimensional) invariant pattern recognition (*e.g.* shift, rotation and scaling) are Lie transformation groups. It will be seen that the coordinates in which the transformations reduce to shifts are the *canonical coordinates* of the generators of the transformation groups.

This formalism was introduced by Ferraro and Caelli (1988), and independently by Rubinstein et al. (1991). The results are extended, developed and applied in Segman et al. (1992) and Ferraro and Caelli (1994). The development presented here is an

[5]Dilations in the image domain become contractions in the frequency domain, and *vice versa*.

[6]The theory of Lie transformation groups is recapitulated in Chapter 5.

amalgam of these papers. We have seen that the conditions for strong invariance with respect to a transformation T_a, as given by Equation 2.3, can be satisfied if we find a change of coordinates $(x, y) \to (\xi, \eta)$ such that T_a becomes $\widetilde{T}_a$, a *shift*, in the new coordinate system (ξ, η).

Canonical Coordinates for a One-Parameter Transformation

Consider a one-parameter Lie Transformation Group T_a specified by

$$\begin{aligned} x' &= \alpha\,(x, y, a) \\ y' &= \beta\,(x, y, a)\,. \end{aligned} \tag{2.14}$$

Let $\mathscr{L}_T$ be the generator of T in the original (x, y) coordinates:

$$\mathscr{L}_T = \mu\,(x, y)\,\frac{\partial}{\partial x} + \nu\,(x, y)\,\frac{\partial}{\partial y}. \tag{2.15}$$

The action of T_a in the new coordinates is specified by

$$\begin{aligned} \xi' &= \xi\,(\alpha(x, y, a), \beta(x, y, a)) = \bar{\alpha}\,(\xi, \eta, a) \\ \eta' &= \eta\,(\alpha(x, y, a), \beta(x, y, a)) = \bar{\beta}\,(\xi, \eta, a)\,, \end{aligned} \tag{2.16}$$

and the new generator is

$$\widetilde{\mathscr{L}}_T = \left(\mu\frac{\partial\xi}{\partial x} + \nu\frac{\partial\xi}{\partial y}\right)\frac{\partial}{\partial\xi} + \left(\mu\frac{\partial\eta}{\partial x} + \nu\frac{\partial\eta}{\partial y}\right)\frac{\partial}{\partial\eta}. \tag{2.17}$$

In order to meet the strong invariance condition, we require that $\widetilde{G}$ be a shift in the new coordinates:

$$\begin{aligned} \xi' &= \xi + a \\ \eta' &= \eta. \end{aligned} \tag{2.18}$$

The generator of the transformation corresponding to this shift along the ξ-axis is

$$\widetilde{\mathscr{L}}_T = \frac{\partial}{\partial\xi}. \tag{2.19}$$

Equation 2.17 thus leads to the equations

$$\begin{aligned} \mu(x, y)\frac{\partial\xi}{\partial x} + \nu(x, y)\frac{\partial\xi}{\partial y} &= 1 \\ \mu(x, y)\frac{\partial\eta}{\partial x} + \nu(x, y)\frac{\partial\eta}{\partial y} &= 0. \end{aligned} \tag{2.20}$$

The new coordinates (ξ, η) specified by Equations 2.20 are called the *canonical coordinates* corresponding to the transformation T_a. It is well-known that this system of partial differential equations has an infinite number of solutions [see Rubinstein et al., 1991].

To apply this result to the derivation of a kernel for an integral transform, we recall that the Fourier transform satisfies the strong invariance condition with respect to along-axis shifts in its coordinate system. We now show how to combine the solutions to Equation 2.20 with the normal two-dimensional Fourier transform to find a new invariance kernel. Let $\widetilde{f}(\xi, \eta)$ be a function defined on $\mathbb{R}^2$. Its two-dimensional Fourier transform is

$$\widetilde{F}(u,v) = \iint_{\mathbb{R}^2} e^{i(u\xi + v\eta)} \widetilde{f}(\xi,\eta) d\xi d\eta. \tag{2.21}$$

The transformation T_a changes $\widetilde{f}$ according to $\widetilde{f}_{T_a}(\xi,\eta) = \widetilde{f}(\xi + a, \eta)$. We know from Equation 2.6, therefore, that

$$\widetilde{F}_{T_a}(u,v) = e^{-iua} \widetilde{F}(u,v). \tag{2.22}$$

Changing back to the original coordinates in Equation 2.21, we obtain

$$\begin{aligned} \widetilde{F}(u,v)[f] &= \iint_{\mathbb{R}^2} e^{i(u\xi(x,y) + v\eta(x,y))} J(x,y) f(x,y) dx dy \\ &= e^{-iua} \widetilde{F}(u,v)[f_{T_a}], \end{aligned} \tag{2.23}$$

where

$$J(x,y) = \det \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \tag{2.24}$$

is the Jacobian of the change of coordinates. Thus we have found the kernel

$$K(u,v;x,y) = e^{i(u\xi(x,y) + v\eta(x,y))} J(x,y) \tag{2.25}$$

which satisfies the strong invariance conditions of Equation 2.3. Since Equations 2.20 have an infinite number of solutions, there are infinitely many invariance kernels for the one-parameter group G. This allows a particular kernel to be chosen so that it is convenient from a computational point of view.

Canonical Coordinates for Two Transformations

The method above can be extended to find canonical coordinates for two-parameter Lie transformation groups.[7] Unlike the one-parameter case, in which infinitely many kernels giving integral transforms satisfying the strong invariance conditions could be found, in the two parameter case the method leads to unique coordinate changes, and to the discovery of important restrictions on the existence of strongly-invariant transforms.

In order to satisfy the strong invariance condition simultaneously for two Lie transformation groups T_a and S_b, we require a representation

$$\mathscr{O}\left\{S_b\left[T_a f(x,y)\right]\right\} = A\left(u,v\right) e^{i\phi(u,v)+ua+vb}. \tag{2.26}$$

Again, the amplitude is invariant under the action of the transformations. The parameters the transformations are encoded *orthogonally* in shifts along the two axes in the transformation domain, in the phase.

Ferraro and Caelli (1994) derive the conditions under which such a representation exists, and specify the analytical form of the related integral transform kernel. If $\mathscr{L}_a$ and $\mathscr{L}_b$ are the generators of T_a and S_b respectively, then (ξ, η) are canonical coordinates for the transformations if

$$\begin{aligned} \mathscr{L}_a\xi &= 1, \qquad \mathscr{L}_a\eta = 0, \\ \mathscr{L}_b\xi &= 0, \qquad \mathscr{L}_b\eta = 1. \end{aligned} \tag{2.27}$$

The existence of such canonical coordinates is a sufficient condition for the existence of a strongly-invariant representation, as a simple consequence of the properties of the Fourier transform. It can also be shown that this condition is necessary [Ferraro and Caelli, 1988]. Canonical coordinates only exist if $\mathscr{L}_a$ and $\mathscr{L}_b$ are linearly independent and commute, that is, the Lie bracket vanishes:

$$[\mathscr{L}_a, \mathscr{L}_b] = \mathscr{L}_a\mathscr{L}_b - \mathscr{L}_b\mathscr{L}_a = 0. \tag{2.28}$$

Moreover, the change of coordinates is one-to-one. Equation 2.28 allows one to determine whether or not two transformations admit a strongly-invariant representation via a straightforward calculation. For example, consider rotation and dilation. The generator of the rotation transformation is

$$\mathscr{L}_R = -y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y} \tag{2.29}$$

[7] This analysis is extended to k-parameter groups in Segman et al. (1992).

and that for dilation is

$$\mathscr{L}_D = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}. \tag{2.30}$$

By simple calculation,

$$\begin{aligned} \mathscr{L}_R\mathscr{L}_D &= -y\frac{\partial}{\partial x}\left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right) + x\frac{\partial}{\partial y}\left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right) \\ &= -y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}. \end{aligned} \tag{2.31}$$

Similarly,

$$\begin{aligned} \mathscr{L}_D\mathscr{L}_R &= x\frac{\partial}{\partial x}\left(-y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}\right) + y\frac{\partial}{\partial y}\left(-y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}\right) \\ &= x\frac{\partial}{\partial y} - y\frac{\partial}{\partial x}. \end{aligned} \tag{2.32}$$

Trivially, the commutator is

$$[\mathscr{L}_R, \mathscr{L}_D] = 0. \tag{2.33}$$

Since $\mathscr{L}_R$ and $\mathscr{L}_D$ are orthogonal, and we have just shown that they commute, there exists a representation which is simultaneously strongly-invariant under both the corresponding transformations.

Conversely, consider $\mathscr{L}_X = \frac{\partial}{\partial x}$, corresponding to a translation along the x-axis. It is easy to show that the commutator with the rotation transformation is

$$[\mathscr{L}_X, \mathscr{L}_R] = \frac{\partial}{\partial y} \neq 0. \tag{2.34}$$

Similarly, $[\mathscr{L}_X, \mathscr{L}_D] \neq 0$, and the same is clearly true for translations along the y-axis. Thus Equation 2.28 allows us to show that there exists no representation which is strongly-invariant under both rotation and shift, or under dilation and shift. This refutes some claims in the literature [*e.g.* Casasent and Psaltis, 1976], and explains why the representations of Altmann and Reitböck (1984) discussed above could not be *strongly*-invariant.

The second benefit of the canonical coordinates approach is that it provides an equation for the kernel of the integral transform giving the strongly-invariant representation. This is of exactly the same form as that found above in Equation 2.25, with the condition that the commutation and independence conditions above must now be applied to the two transformation groups with respect to which (ξ, η) are canonical coordinates. An excellent treatment of this, and a generalization to k-parameter groups, is given by Segman et al. (1992).

We have seen that the canonical coordinates approach provides a formalism which allows one to determine for which transformations a strongly-invariant integral transform representation exists, and moreover, when it exists, it shows how to calculate its kernel. These are important and powerful results.

2.2 Moments

Although calculated by integrating a kernel over an image, moments are not integral transforms in the sense used above. An integral transform transforms an image into a space of the same dimensionality as the image space (although the transform coordinates are often complex). A moment of a given order $(p+q)$, however, is a single number. If one thinks of the "moment transform" of an image as the set of all its moments, then one can consider the image to have been mapped into a discrete "moment space", indexed by p and q. In this sense moments can be considered as integral transforms.

2.2.1 Geometrical Moments

The geometrical moment of order $(p+q)$ of an continuous image function $f(x,y)$ is defined by:

$$m_{pq} = \iint\limits_{\mathbb{R}^2} x^p y^q f(x,y)\,dx\,dy \qquad p,q = 0,1,2,\dots\,. \tag{2.35}$$

For invariance purposes, it is more convenient to use the central moments, which are, by definition, translation invariant:

$$\mu_{pq} = \iint\limits_{\mathbb{R}^2} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,dx\,dy \qquad p,q = 0,1,2,\dots\,, \tag{2.36}$$

where the centroid of the image can be found using $(\bar{x},\bar{y}) = (m_{10}/m_{00}, m_{01}/m_{00})$. Altmann and Reitböck (1984) use μ_{20} and μ_{02} to estimate the factor by which an image has been scaled. This is then used as the starting point for a cross-correlation-based search for the image template in a reduced search space.

If the image is scaled according to $x' = \alpha x$, $y' = \alpha y$, a set of normalized central moments can be defined by the equations

$$\begin{aligned} \mu'_{pq} &= \frac{\mu_{pq}}{\alpha^{p+q+2}} \\ \eta_{pq} &= \frac{\mu'_{pq}}{{\mu'_{00}}^{\gamma}}, \qquad \gamma = \frac{p+q}{2} + 1. \end{aligned} \tag{2.37}$$

These normalized central moments are scale invariant, and the $\{\eta_{pq}\}$ can be used to define a set of moment features which are invariant under shifts, rotations and scalings of the image. Such a set [Khotanzad and Lu, 1990; Jain, 1989][8] is:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.38a}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.38b}$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.38c}$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.38d}$$
$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{2.38e}$$
$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) \tag{2.38f}$$
$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & - (\eta_{30} - 3\eta_{21})(\eta_{12} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right].\end{aligned} \tag{2.38g}$$

Functions ϕ_1 through ϕ_6 are also invariant under reflection, while ϕ_7 changes sign.

Wechsler (1990) notes that these moments are not fault-tolerant, and in general produce disappointing results when used in pattern recognition experiments. The major sources of deviation from theoretical invariance are discretization, and quantization noise. They are also defeated by occlusion (as are many of the techniques discussed in this chapter). It is possible to define moments in polar coordinates too:

$$\psi_{kpq} = \int_0^\infty \int_{-\pi}^{\pi} r^k g(r,\theta) \cos^p\theta \sin^q\theta \, d\theta dr. \tag{2.39}$$

This allows a somewhat simpler expression of Equations 2.38 (and a very much simpler proof of rotation invariance), but the drawbacks remain the same.

2.2.2 Zernike Moments

Geometrical moments, as defined by Equation 2.35, have the form of a projection of the image $f(x,y)$ onto the monomial $x^p y^q$. The basis set $\{x^p y^q\}$ lacks some desirable features, such as orthogonality. Consequently, the resultant geometrical moment features are sub-optimal in terms of information redundancy [Khotanzad and Lu, 1990].

The Zernike polynomials $\{V_{nm}(x,y)\}$ are a set of complex polynomials which form

[8] Both sources cite Hu (1962) as the originator of these invariant moment combinations, but I have not obtained that paper.

a complete orthogonal set over the interior of the unit circle:

$$V_{nm}(x,y) = V_{nm}(r,\theta) = R_{nm}(r)e^{im\theta} \qquad \begin{cases} n \in \mathbb{J},\ n \geq 0 \\ m \in \mathbb{J},\ n - |m| \text{ even},\ |m| \leq n \end{cases} \tag{2.40}$$

where

$$R_{nm}(r) = \sum_{s=0}^{n-\frac{|m|}{2}} \frac{(-1)^s[(n-s)!]r^{n-2s}}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!}. \tag{2.41}$$

These polynomials, unlike the monomials $x^p y^q$, are orthogonal, satisfying

$$\iint\limits_{x^2+y^2\leq 1} V^*_{nm}(x,y)V_{pq}(x,y)dxdy = \frac{\pi}{n+1}\delta_{np}\delta_{mq}, \tag{2.42}$$

where δ_{ab} is the usual Kronecker delta: $\delta_{ab} = 1$ if $a = b$, and 0 otherwise.

The Zernike moments $\{A_{nm}\}$ of an image $f(x,y)$ are the projections of the image onto these basis functions. Expressed in polar coordinates,

$$A_{nm} = \frac{n+1}{\pi} \iint\limits_{r\leq 1} f(r,\theta)V^*_{nm}(r,\theta)drd\theta. \tag{2.43}$$

The orthogonality property means that the image can be reconstructed from these moments (albeit an infinite number of them). To compute the Zernike moments for a digital image, the usual summation approximation of the integral is used. The centre of the image is taken as the origin, and the pixel coordinates are scaled to the unit circle. Pixels falling outside the unit circle are discarded [Khotanzad and Lu, 1990].

The Zernike moments provide rotation-invariant features. Consider the Zernike moment A_{nm} of an image, and A'_{nm}, the Zernike moment of the image after rotation through an angle ϕ. Since the radial and angular components of the Zernike polynomials are separable, it is trivial to note that

$$A'_{nm} = A_{nm}e^{-im\phi}. \tag{2.44}$$

This is reminiscent of the property of the Fourier transform discussed in §2.1.2. It is clear then that the magnitudes of the Zernike moments are a set of rotationally-invariant image features. This invariance is of course not exact in a digital setting due to the usual problems of digitization and quantization noise, but Khotanzad and Lu (1990) show σ/μ values[9] of less than 7% for a range of Zernike moments for variously rotated images of real characters.

[9] σ is sample standard deviation, μ is sample mean.

Zernike moments are only rotation-invariant. To obtain shift and scale invariance, the images must first be normalized with respect to these actions. This is done by using the geometric moments of the raw image. In short, the normalized image is

$$f'(x,y) = f\left(\bar{x} + \frac{x}{a}, \bar{y} + \frac{y}{a}\right) \tag{2.45}$$

where $(\bar{x}, \bar{y}) = (m_{10}/m_{00}, m_{01}/m_{00})$ is the image centroid, and $a = \sqrt{\beta/m_{00}}$, where β is a pre-determined image "size" (in pixels, for a binary image). This normalization renders $|A_{00}|$ identical for all images, and $|A_{11}| \equiv 0$.

Khotanzad and Lu (1990) compared a number of classifiers (multilayer perceptron (MLP), Bayes, nearest-neighbour and minimum mean distance) using features based on geometrical moments and Zernike moments for classifying arbitrarily transformed uppercase Roman letters. They found that the neural network classifier outperformed all the others, especially in noisy cases. In agreement with Wechsler (1990), they found that the geometric moment features became useless in the presence of noise, whereas the Zernike moment features proved robust, although the higher order moments were more sensitive. Perantonis and Lisboa (1992) compared a MLP trained with Zernike moment features with a third-order network (see §3.4.3) for classifying hand-written numerals, and found that the third-order network performed better. Higher-order neural networks are, however, extremely expensive in space and computation time.

2.3 Matched Filtering and Convolution Techniques

The integral transform techniques discussed in §2.1 address the problem of finding a representation of an image which is invariant under certain transformations of the image. The amplitude spectra obtained by these techniques, the invariant part of the representation, are themselves images. The "Invariant" part of the phrase "Invariant Pattern Recognition" has been addressed, but not yet the actual "Recognition". Matched filtering provides one answer to the recognition problem, and convolution, or cross-correlation, techniques employ it to provide invariance as well.

2.3.1 Matched Filtering

One of the oldest techniques for pattern recognition is matched filtering, descriptions of which can be found in many standard reference books on computer vision [Rosenfeld and Kak, 1982; Ballard and Brown, 1982]. Matched filtering allows the computation of R_{fg}, a measure of the similarity, between an image $f(x,y)$ and a template $g(x,y)$:

$$R_{fg} = \iint f(x,y)g(x,y)dxdy, \tag{2.46}$$

where the integration is over the entire domains of x and y.

If the image and template are normalized so that they have the same total intensity *i.e.* $\iint f^2(x,y)dxdy = \iint g^2(x,y)dxdy$, then R_{fg} is proportional to the interesting part of the mean-squared distance between the images d^2_{fg}, since

$$\begin{aligned} d^2_{fg} &= \iint (f(x,y) - g(x,y))^2 dxdy \\ &= \iint \left(f^2(x,y) - 2f(x,y)g(x,y) + g^2(x,y)\right) dxdy \\ &= 2\iint f^2(x,y)dxdy - 2R_{fg}. \end{aligned} \tag{2.47}$$

The first term is constant due to the normalization, so it is clear that R_{fg} measures the least-squared "similarity" between the image and the template: the larger R_{fg} is, the closer the images are to identical. In a pattern recognition problem, for an image and each of a set of possible templates, R_{fg} is computed and the image is assigned to the class of the template for which R_{fg} is greatest. No classification may be made if R_{fg} is less than some threshold. It is known that this is the optimal technique for detecting templates in the presence of additive white noise [Rosenfeld and Kak, 1982].

The matched filter output is extremely sensitive to shifts or distortions of the image and it is rarely used in this simple form. Nevertheless, it is a candidate for the in-place matching of the amplitude spectra obtained using the integral transforms described in §2.1.

2.3.2 Cross-correlation

A more common approach is to compute the cross-correlation $C_{fg}(u,v)$ between the image and a template, frequently with a template smaller than the image:

$$C_{fg}(u,v) = \iint f(x,y)g(x+u, y+v)dxdy. \tag{2.48}$$

$C_{fg}(u,v)$ is again an "image", in which the intensity at coordinates (u,v) is just R_{fg} calculated for the template shifted to those coordinates before the comparison with $f(x,y)$.

The presence of the template $g(x,y)$ centred at (u,v) in the image $f(x,y)$ can be detected if $C_{fg}(u,v)$ is greater than some threshold at that point. It is possible to detect the presence of multiple instances of a template in the image using this technique. Some version of cross-correlation plays a role in many invariant pattern recognition techniques [Altmann and Reitböck, 1984; Caelli and Liu, 1988; Pintsov, 1989; Zetzsche and Caelli, 1989; Lenz, 1990; Ohlsson, 1992; Lin and Brandt, 1993].

The use of cross-correlation with raw images and templates can be seen as a form of shift-invariant pattern recognition, since the template can be detected no matter where it is in the image. It is perhaps not a true invariant pattern recognition technique, since the image is effectively exhaustively-searched for the template. Computing $C_{fg}(u, v)$ is a very computationally-expensive process, especially if the template is large.

Pintsov (1989) uses the idea of "the cross-correlation function with respect to a group of transformations". He views Equation 2.48 as cross-correlation with respect to the group of translations of the plane:

$$\begin{aligned} x' &= x + u, \\ y' &= y + v. \end{aligned} \tag{2.49}$$

The cross-correlation process calculates R_{fg} between the image and all possible versions of the template transformed by the group of translations. This notion can be extended to other groups. In general, if a group G has n parameters, then locating instances of a transformed template in an image by means of cross-correlation is the problem of finding the local maxima of the cross-correlation function in an n-dimensional space. Pintsov (1989) shows how this approach leads directly to the Hough transform which is discussed in detail in §2.5.1, and the closely-related Radon transform. It is noted that the computational cost may be reduced if the transformation has known symmetries.

Extending the cross-correlation function to higher dimensions in this way as a means of achieving invariance was discussed earlier by Caelli and Liu (1988, p. 207), but, as they noted:

> ... if in the cross-correlation process, we explicitly include α and θ as variables, (by generating all possible templates) then the correlation process is invariant with respect to rotation (θ) and size change (α) as well as translation (x, y) in a four dimensional space. However, the large amount of computation involved in calculating the quadruple integration ... prohibits the process from being practical, even with the help of fast Fourier transform techniques.

Again, this is really just exhaustive search: all possible transformations of the template are computed and compared with the image. Pintsov (1989) shows how known symmetries of transformations can be used to reduce the space which must be searched. Altmann and Reitböck (1984) give a method that reduces this cost by only searching for the template in regions of the image (in their case actually a Fourier transform of an image) which are likely to contain it. This is possible because they are using a Fourier-Mellin technique in which scale changes are mapped to shifts in the transform domain, and the normalized central second moments (see §2.2) are used to estimate the scale factor.

2.3.3 Filters Based On Multiple Templates

We have seen that computing a cross-correlation is equivalent to doing a matched filtering of the image with all possible transformed versions of a template. The computational cost increases dramatically as the number of parameters of the allowed transformation group increases, rendering this approach largely impractical. This cost can be greatly reduced if it is necessary to compute the matched filter output of the image with only a small set of transformed patterns. This is possible, for instance, if the number of possible transformations of the image is known *a priori*.

One approach is to use a filter H_j which is a linear combination of the n allowed transform states f_{ij} of the jth pattern:

$$H_j = a_{1j} f_{1j} + a_{2j} f_{2j} + \cdots + a_{nj} f_{nj}, \tag{2.50}$$

where the $\{a_{ij}\}$ are determined by some suitable constraints. In order to achieve simultaneous invariance to translation, rotation and scaling, a very large number of templates is still required, and finding the $\{a_{ij}\}$ is again computationally-expensive. This approach can be effective, however, if the template design procedure is based on some optimization criteria.

Caelli and Liu (1988) address this problem by defining the "invariance surface"[10] of an image. The coordinates of an pattern $f_j(x, y)$ may be rotated and scaled, giving new coordinates (ξ, η) according to

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \alpha \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2.51}$$

The invariance surface $I_j(\alpha, \theta)$ for pattern j is the maximum of the shift cross-correlation function between the untransformed pattern and the rotated and scaled versions,

$$I_j(\alpha, \theta) = \max_{u,v} \left\{ \iint_{\mathbb{R}^2} f_j(\xi, \eta) f_j(x + u, y + v) dx dy \right\}. \tag{2.52}$$

The degree of invariance of a particular pattern with respect to the transformations is given by $I_j(\alpha, \theta)$: the degree of invariance is measured by the difference between the cross-correlation function and its maximum value, when computed between the untransformed template and itself. The degree of invariance of a pattern depends on the pattern structure: the letter "O" is much more rotationally-invariant than the letter "T", and its $I_j(1, \theta)$ curve is correspondingly smoother, and deviates less from $\max_\theta\{I_j(1, \theta)\}$. This indicates that the number of templates required to construct a

[10] This is a different use of this term than that used later in this thesis.

satisfactory filter H_j depends of the invariance surface of pattern j. This number can be determined by choosing a recognition threshold criterion C, and choosing to generate a new template at each transformation state (α, θ) where $I_j(\alpha, \theta)$ falls below C. Clearly this will automatically take account of any symmetries of the patterns, and of the transformations.

This technique allows a computationally feasible application of the matched filtering method to invariant pattern recognition. The number of templates used to construct the filter for each pattern depends on the invariance properties of the pattern with respect to the transformations for which invariance is desired. Moreover, this can be controlled by manipulating the threshold C, trading matching accuracy for computational expense.

Zetzsche and Caelli (1989) employ a related method. They use a 4-dimensional pattern representation based on banks of filters. The representation is encapsulated in a set of filtered images, each containing the original Cartesian coordinates of the image, and indexed in terms of the orientation- and scale-specific filters used to produce them. This is based on transformations to a space in which rotations and scalings become shifts, as discussed extensively above, and again claims biological inspiration. Rather than filters based on transformed target images, the filters are in this case generic, based on two-dimensional Gabor functions, which are discussed in Chapter 4. This is thus not a direct matched filtering technique, but cross-correlation is used for recognition of the filter outputs.

2.4 Parts and Relationships

In all the techniques discussed so far in this chapter, an attempt has been made to find an invariant representation of the image as a whole, and to match images to images (or transformed images) on a pixel-by-pixel basis. A different approach to the invariant pattern recognition problem is to view an image as a group of component *parts*, and *relationships* between those parts. As a trivial example, if an image consists of two straight lines, then the angle between those lines is invariant under shifts, rotations and scalings of the image. The lines are the parts, and the angle between them is their relationship. Matching is done between these abstracted properties of the image, rather than between pixels. In more realistic examples, there is the sub-problem of (possibly invariantly) recognizing the parts in the image, and extracting their relationships.

There is a large body of literature concerning the application of the parts and relationships approach to three-dimensional object recognition. Here we will consider only a small sample of two-dimensional invariant pattern recognition applications. It should be noted that the distinction between a "holistic" pattern recognition approach and the use of parts and relationships is not as clear-cut as it might seem. Holistic approaches depend implicitly on the constituent parts of the image, and their relationships.

2.4.1 Sub-Graph Matching

Li (1992) addresses the problem of matching between an image and a model under arbitrary translations, rotations and scale changes in noisy conditions. Only very simple objects (or patterns) are considered, the parts being points or line segments. An image can consist of only sub-parts of a model object, and there can be extra lines and points in the image. Li (1992) calls his object representation an *attributed relational structure* (ARS). An ARS consists of a set of nodes, their properties (unary relations), and binary and ternary relations between them. The ARS is an invariant image representation, because the node properties and relations are chosen to be invariant under arbitrary 2D geometric transformation. Possible choices for these invariant attributes include curvature properties, relative angles and distance and area ratios.

It is apparent that, once parts and relationships have been extracted from an image, the parts and relationships constitute a graph, with labeled nodes and edges. This is true of all parts and relationships approaches to object or pattern recognition, whether or not the attributes chosen are invariant. The greatest limitation of these techniques is thus that recognition requires a solution of the sub-graph isomorphism problem, which is known to be NP-complete [Ballard and Brown, 1982]. 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, which is employed by Li (1992).

2.4.2 Deformable Templates

Less general, but related, are the deformable template techniques. Typically they are used when the aim is to match a distorted version of a known model, for which the possible distortions are limited and non-arbitrary. These conditions allow costs to be assigned to various deformations, and constraints to be placed on the matching problem. One advantage of these techniques is that they not only allow matching of distorted templates, but allow the distortion to be recovered, and its cost to be estimated.

Chow and Li (1993) apply deformable template matching to the automatic detection of features in front-view images of human faces. Domain knowledge, such that the eyes are a horizontal pair of dark regions on a light surface, is used to provide initial plausible estimates of feature locations, for features such as the eyebrows and mouth. A generalized Hough transform is then used to locate the irises, modeled as circles. These estimated iris positions are in turn used as a starting point for the application of a deformable template, consisting of two parabolas, to fitting the eye boundaries. The optimization method used is based on the simplex method, which allows the reduced search space to be taken into account [Press, Teukolsky, Vetterling and Flannery, 1992]. A similar method is used to locate the mouth. Experiments with this system showed that it was able to locate facial features satisfactorily in over 80% of test images.

Deformable template techniques share several features with the model-based methods developed in this thesis. The system designer's *a priori* knowledge of the domain of possible patterns is used to constrain the search for features. This domain knowledge is directly encoded in the system, rather than being abstracted from training images.

2.5 Contour-Based Methods

An important class of two-dimensional invariant pattern recognition techniques, especially in the context of this thesis, are those designed for the recognition of contours. These contours may be extracted through some form of edge detection, or may be a natural pattern representation, such as in character recognition. Broadly speaking, approaches to contour recognition may be divided into two classes: those which represent the contour by a parameterized algebraic expression fitted to the image, and those which treat a contour as a group of pixels. Contour recognition is of particular interest because contours corresponding to object boundaries are frequently used in three-dimensional object recognition, and also because there are many applications for the application of machine pattern recognition to domains in which the patterns naturally consist of line drawings (*e.g.* character recognition, circuit diagrams, engineering drawings, *etc.*). Moreover, there is evidence that the human visual system applies a contour-based approach to pattern recognition even when no contour actually exists in the image: it interpolates an implied contour [Caelli, Preston and Howell, 1978].

2.5.1 The Hough Transform

Perhaps the simplest "contour-based" technique is the Hough transform, which is described in standard computer vision references [*e.g.* Ballard and Brown, 1982]. Variants of the Hough transform occur frequently in the invariant pattern recognition literature [Pintsov, 1989; Chow and Li, 1993; Li and Roeder, 1994]. We have seen in §2.3.2 that the Hough transform and its generalizations can be interpreted as cross-correlation techniques, where the template is a parametric curve rather than an actual image. The most basic Hough transform is that for detecting straight lines.

Consider an image in which the intensity $I(x, y)$ at a point (x, y) is a measure of that point's importance (typically the "strength" assigned to it by an edge detection algorithm). All straight lines passing through (x, y) obey the equation $y = mx+c$. Each individual line is represented by a point (m, c) in a two-dimensional parameter space. The $\{(m, c)\}$ corresponding to all possible lines passing through (x, y) form a line in the parameter space. This parameter space is unbounded, so in practical applications, the parameterization

$$x \cos \theta + y \sin \theta = r \tag{2.53}$$

is usually used. This produces a sinusoid in (r, θ) space for each point (x, y). The parameter space can be divided into a number of bins, or accumulators, each corresponding to a range of values of r and θ. For each point (x, y), $I(x, y)$ is added to the bins corresponding to all the (r, θ) satisfying Equation 2.53. As with cross-correlation, a threshold can then be used to chose which bins have accumulated sufficient evidence for the presence of a line.

The Hough transform in this form is usually thought of as a line detection algorithm, rather than an invariant pattern recognition technique. It can, however, be considered to be a method for recognizing a straight line segment invariant under shifts and rotations.[11] The use of the Hough transform for detecting more complex parameterized contours is more readily recognizable as invariant pattern recognition. The circles $(x - a)^2 + (y - b)^2 = r^2$ can be detected in exactly the same way, this time with a three-dimensional parameter space (a, b, r). This form of the Hough transform is used by Chow and Li (1993), and, slightly modified, by Li and Roeder (1994).

The Hough transform is thus a generalized matched filtering technique for parameterized curves. It can be used to discover the set of parameters that best explain the image data, given that the parametric form of the curve producing the data is *a priori* known. Its disadvantage is that it is practically restricted to a reasonably small set of template curves, since both the computation time and the size of the array of bins increase exponentially with the number of parameters. Moreover, it is heavily dependent on the quality of the edge detection which produces $I(x, y)$.

2.5.2 Algebraic and Differential Invariants

Another approach to invariant contour matching involves calculating invariants of the contour, and matching these. This offers the chance to avoid the computation time and space expenses of methods such as cross-correlation or the Hough transform, in which the contour is effectively matched against all possible transformed versions of the template.

Such techniques are often employed when seeking three-dimensional projective invariants of contours, a more difficult problem than invariance under the subset of affine transformations in the plane discussed so far. The researchers involved in this work are frequently interested in issues such as camera calibration, stereo matching or photometrics, rather than with invariant object (or pattern) recognition *per se* [Forsyth,

[11] The choice of threshold influences the scale invariance of this technique.

Mundy and Zisserman, 1992; Weiss, 1993b; Faugeras, 1993; Barrett, Gheen and Payton, 1993; Gros, 1993].

Functions of Contour Coefficients

Algebraic invariants are well suited for use with algebraic contours: contours which can be expressed by an implicit polynomial $f(x, y) = 0$. As an example we will consider the family of contours known as conic sections. The general equation for a conic section is

$$f(x, y) = a_{11}x^2 + a_{22}y^2 + 2a_{12}xy + 2a_{13}x + 2a_{23}y + a_{33} = 0. \tag{2.54}$$

If we define the matrices

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}. \tag{2.55}$$

Equation 2.54 can then be written succinctly in matrix form as

$$\mathbf{X}^{\mathrm{T}}\mathbf{A}\mathbf{X} = 0. \tag{2.56}$$

The shape of a particular conic is determined by the coefficients $\{a_{ij}\}$. Since $\mathbf{A}$ is a real symmetric matrix, the coordinates can be transformed using a similarity transform so that $\mathbf{A}$ is diagonal. This implies that properties of the matrix $\mathbf{A}$ which are invariant under similarity transforms will be invariant descriptors of the shape of the conic under translation and rotation. One such feature is the determinant $\mathbf{D}$, another is the trace $\mathbf{T}$. Two invariant descriptions of the conic $f(x, y)$ are thus

$$\mathbf{D} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix} \qquad \mathbf{T} = a_{11} + a_{22} + a_{33}. \tag{2.57}$$

Further invariant properties of $\mathbf{A}$ can be found.[12] These functions of the polynomial coefficients can therefore be used for matching a transformed conic to its untransformed template. This approach can be extended to higher dimensions (*e.g.* quartic surfaces), and to other families of curves. The coefficients must be obtained by fitting a polynomial to the image data – in itself a far from trivial problem, and potentially computationally-expensive. Clearly a high resolution image of the curve will be required if the coefficients are to be estimated sufficiently well to be useful for recognition, although Forsyth et al. (1992, p. 43) say “For applications in model-based vision, it is far more important that a representation be projectively invariant than that it be a good approximation.” If

[12] For instance, any symmetric function of the eigenvalues.

the coefficients can be determined sufficiently well, however, the matching process is very cheap.

Cross-ratios

The techniques discussed in this section are often more applicable to the projective geometry appropriate in three-dimensional object recognition than in 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 [Wechsler, 1990]. 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 [*e.g.* Forsyth et al., 1992; Barrett et al., 1993; Vanderkooy, 1996].

An example of a possible cross-ratio that might be used comes from Vanderkooy (1996). Consider the detection of four coplanar lines, A, B, C and D. Under a perspective projection, these lines will intersect at a single point: the vanishing point.[13] The cross-ratio of the sines of the angles between these lines is:

$$[A, B; C, D] = \frac{\sin(\angle AC)}{\sin(\angle AD)} \times \frac{\sin(\angle BD)}{\sin(\angle BC)}. \tag{2.58}$$

If the lines are in fact coplanar, this corresponds to an invariant structure. If the lines are not coplanar, the cross-ratio will vary from view to view.

Note that this requires that the lines be labeled, so that the appropriate angles can be identified. This requires some prior point-matching. If this is not done, then the technique faces a combinatorial explosion. The number of cross-ratios, C, that can be calculated for N lines is

$$C = \frac{N!}{(N-4)!4!}. \tag{2.59}$$

Differential Invariants

Differential Invariants arise most naturally when the coordinates of points on a curve, $\mathbf{x}$, are expressed explicitly as a function of some local parameter t, $\mathbf{x} = \mathbf{x}(t)$, rather than by an implicit function such as that in Equation 2.54. The natural shape descriptors in such a representation are the derivatives $\frac{d^n x_i}{dt^n}$. These descriptors are *local*, since they depend on the derivatives at a particular value of t, unlike the global descriptors derived from the coefficients of the implicit function in the case of algebraic invariants. A differential invariant is a function of the $\frac{d^n x_i}{dt^n}$ which does not change under a transformation of the coordinates $\mathbf{x}$ and the parameter t (or which changes in a limited way, which will be

[13] Lines can be extended to the vanishing point; they need not intersect in the actual image.

explained in the following matter) [Weiss, 1993a]. Various differential invariants have been widely applied in computer vision: curvature, torsion and Gaussian curvature, for instance, are all invariant under Euclidean transformations [Forsyth et al., 1992].

It is possible to characterize a curve using the differential equation to which the curve is one of the possible solutions. The advantage of this approach is that some constants are eliminated. Curves are represented parametrically, $\mathbf{x} = \mathbf{x}(t)$, where $\mathbf{x}$ is a point on the curve, expressed in homogeneous coordinates.

As a simple example, consider the second order linear differential equation

$$\mathbf{x}'' = 0. \tag{2.60}$$

The solutions to this equation constitute the family of all straight lines. Equation 2.60 is thus an invariant representation of straight lines – it is invariant with respect to the gradient and y-intercept of the line (*i.e.* shift, rotation and scale).

Generalizing, any curve expressed in n-dimensional homogeneous coordinates satisfies the linear differential equation [Weiss, 1993b]:

$$\mathbf{x}^{(n)} + \binom{n}{1} p_1 \mathbf{x}^{(n-1)} + \binom{n}{2} p_2 \mathbf{x}^{(n-2)} + \cdots + p_n \mathbf{x} = 0. \tag{2.61}$$

It is clear that multiplying $\mathbf{x}$ by a constant matrix $\mathbf{T}$ will have no effect on Equation 2.61, since $\mathbf{T}$ will factor out. The p_i are thus invariant under the linear transformation $\mathbf{T}$. The solutions are curves in $(n-1)$-dimensional space, determined up to projection. The p_i in Equation 2.61 are scalar functions of the parameter t. Projective invariants can be defined in terms of these p_i.

For plane curves in 3-space, the differential equation corresponding to Equation 2.61 is

$$\mathbf{x}''' + 3p_1\mathbf{x}'' + 3p_2\mathbf{x}' + p_3\mathbf{x} = 0. \tag{2.62}$$

In order deal with coordinate scaling, it is necessary to define *semi-invariants*, P_2 and P_3, which are not invariant under changes in parameterization:

$$\begin{aligned} P_2 &= p_2 - p_1^2 - p_1' \\ P_3 &= p_3 - 3p_1p_2 + 2p_1^3 - p_1''. \end{aligned} \tag{2.63}$$

P_2 and P_3 are invariant under projection and coordinate scaling. We now need to find quantities invariant under transformations of the curve parameter t. We may write this transformation as $t \to \tilde{t}(t)$. *Relative invariants of weight w* are defined to be quantities which transform as:

$$\tilde{\theta}_w = \frac{1}{(\tilde{t}')^w}\theta_w. \tag{2.64}$$

Two such relative invariants can be found using the quantities defined in Equation 2.63:

$$\begin{aligned}\theta_3 &= P_3 - \frac{3}{2}P_2' \\ \theta_8 &= 6\theta_3\theta_3'' - 7(\theta_3')^2 - 27P_2\theta_3^2.\end{aligned} \tag{2.65}$$

Absolute invariants ($w = 0$) can be found by combining relative invariants. For plane curves, an absolute invariant is θ_3^8/θ_8^4. This approach can be extended to higher dimensions.

One advantage of differential invariants is that they are complete – a small set of invariants contains all the essential information about the curve. They are also local, in that differential properties at one point determine the entire curve. This means that differential invariants are invulnerable to occlusion.

Whilst mathematically elegant, this approach has one great disadvantage. Its application to digital images requires the computation of extremely high-order derivatives of the contour in the image (as high as the eighth). This process is well-known to be error-prone. Moreover, these derivatives are raised to high powers, magnifying the estimation error.

Weiss (1993b), noting that the method above is unreliable and hard to use in practice, defines some modified semi-invariants which reduce the number of derivatives required. He also introduces a filter-based method for improving the accuracy of the estimation of high derivatives. Nonetheless, the results presented are for noiseless synthetic curves of very much higher resolution than those typically available for pattern recognition.