Chapter 15

II - Feature selection

In previous chapters a few shape representations were discussed. Recognition through the direct comparison of such representations will in general be time consuming and/or unsuccessful. If one would e.g. try to match two curves on the basis of their Freeman code description, slight changes in position or orientation would yield completely different codes and parameters. This would bring the whole matching procedure in disarray. One would rather extract features from those representations, that lend themselves to direct comparison. This chapter discusses how to go about in selecting such features.

In a first section a crude classification of features is presented, along with an overview of properties shared by good features. The bulk of the chapter then elaborates on these "goodness" properties.

15.1 I - Features – a sneak preview

15.1.1 The well-behaved feature

Features are the data about the scene that are used in the final interpretation step. "Interpretation" can mean decision taking as diverse as simple acceptance/rejection in inspection, recognition of object shape, land use classification, etc. At the interpretation level the massive image data will have been squeezed into a few numbers, making explicit these decisions. Feature extraction is the step that preceeds this final level of abstraction. Its input might consist of image type information, e.g. the Fourier transform of the original image, or a more compact scene description as would be the case when using edges. At the feature level typically a first, dramatic reduction in data-stream bandwidth takes place.

The ideal feature set is easy to extract and allows fast and reliable interpretation. Successful applications of vision have been those where such sets are available, usually by carefully controlling the conditions under which the vision system has to operate. Two major problems jeopardize the successful extraction of such feature set.

First, there is the variability in the composition of the scene. Suppose we have to recognize objects that pass by on a conveyor. Depending on the a priori knowledge on

their pose, the problem might be simple or very complicated. If fixtures guarantee a fixed relative position with respect to the camera, simple features such as the presence or absence of a hole at a certain position may suffice. If the objects are flat, they might nevertheless lie at unknown positions and under unknown orientations. The problem is already more difficult, but simple measurements of area could be enough to distinguish between object types. Objects with a complicated 3-D structure hanging on swinging hooks is the vision designer's nightmare. It is very hard to get a handle on such task. The problem is one of *invariance*, i.e. the extraction of features that remain invariant under changing viewpoints. The first requirement for a feature to be useful therefore is to be invariant under expected *geometrical variability* in the images taken from a scene. As will become clear, it is important to identify the group of transformations that captures this variability, e.g. all possible planar rotations or maybe even projective transformations when perspective effects come into play. From the foregoing discussion it will be clear that the complexity of the vision task is largely determined by the degree to which three-dimensional information inference from two-dimensional image data is necessary. The problem arises from the fact that information is lost under projection, which is a non-invertible operation.

A second source of variability are influences that add noise to the measurements, even if they are meant to be invariant. Changing lighting conditions usually are bad news. These tend to result in noisy input data, e.g. edges containing gaps and other imperfections. But also the fact that we have to sample the images contributes to the overal problem, through the addition of *discretization noise*. These and other disturbing factors must not end up in making the features loose their *discriminant power*. Features should have different values for different classes.

This chapter contains separate sections on the invariance and discriminant power issues, but usually both problems go hand in hand. General viewing conditions also entail increasing difficulties in controlling lighting conditions and image quality. Silhouettes created by backlighting are not particularly helpful in recognizing complex objects with an unknown pose. Moreover, if there is a need for general viewpoint vision, controlling the environment usually has to be ruled out altogether, e.g. in outdoor vision. Scanning such scenes with laser spots to directly obtain range information is unacceptable for reasons of health protection and annoyance with the public, whereas stereo is rather fragile when applied to cluttered scenes.

15.1.2 Features à la carte

The requirements of invariance and discriminant power are general. This section discusses some additional considerations to be made when selecting a feature set for object recognition purposes.

First, there is the important question whether the objects will be completely visible or, alternatively, whether occlusion might hide some of their parts to the camera(s). The former situation clearly is the simpler one. In that case so-called *global features* can be used. Global features require the whole shape for their extraction. Area is a simple example. If an object is occluded, the measurement of visible area will be a very doubtful cue for

recognition. Then one has to resort to *local features*, which can be calculated from object portions. An example would be contour curvature.

The global/local issue is related to that of discriminant power, and as was to be expected, they might well turn out to be in conflict for the well-controlled environments vision systems find themselves in. Global features are usually the ones that can robustly be extracted under such circumstances. This follows from the integration-type of operations that are often involved in their calculation, whereas local features tend to entail the calculation of derivatives. Returning to our earlier examples, area is easier to extract than curvature as long as the global outline is available. On the other hand, if the outline has not completely been extracted – and we know of segmentation as a vulnerable processing step – global features are bound to be useless. At the end of the day, global features will often be found in the same systems that work on binary input.

There is one particularly important family of global features that we would like to introduce here: *moments*. Suppose we have an image with intensity $i(x, y)$, then we define the p, qth moment as

$$M_{pq} = \int \int x^p y^q \; i(x, y) \; dxdy$$

where the integration takes place over the whole image. We say that $p + q$ is the order of the moment. For binary image with 1 for objects and 0 for backgrounds these calculations become particularly useful. Suppose e.g. that we put $p = q = 0$ and calculate this 0th order moment for a binary image with one isolated object in the field of view. The result, M_{00}, is the object area. M_{10}/M_{00} yields the x coordinate of the center of gravity and the y coordinate of this point is found as M_{01}/M_{00}. The center of gravity is thus specified by the 1st order moments. Higher order moments can be shown to yield information on object orientation.

In principle, feature sets that allow to distinguish between the particular objects that might appear will do. For small sets of objects the difference in at least one feature can easily be checked by human inspection. However, if the database of possible objects gets large or has to be extended or updated every so often, it might get hard to find out whether the selected features have the necessary discriminant power. A way out is to use features that are guaranteed to unambiguously describe object shape, i.e. *complete descriptions*. "Shape" has to be interpreted with care here, since on the other hand image projections of the same shape can be substantially different from different viewpoints. Basically, the feature set should allow telling shapes apart as soon as they cannot result from the same object seen under different, allowable viewing conditions. Curvature as a function of arclength is an example of a complete (i.e. unambiguous) description when using an overhead camera that looks down on planar shapes from a perpendicular direction. It completely characterizes shape of a flat outline, up to position and orientation in the object plane. Although clearly desirable if obtainable without additional cost, completeness is not sufficient a guarantee for discriminant power, however. First, the discrete treatment of shape rules out genuine completeness. Also might noise render two descriptions that differ in theory indistinguishable in practice. On the other hand, incomplete descriptions may offer all discriminant power needed. A feature set like area and perimeter, both Euclidean

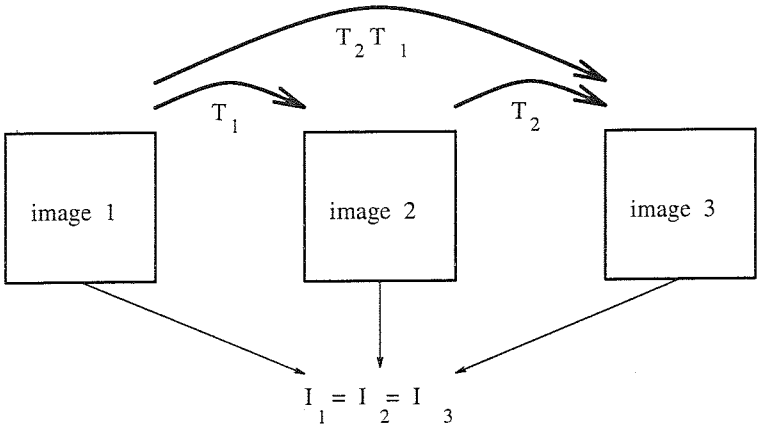

Figure 15.1: *If a feature remains invariant under two transformations, it also remains invariant under their product.*

invariants, is not complete. Nevertheless, it is succesfully applied in several industrial applications. If sufficient a priori information is available to guarantee that no two shapes have the same area and perimeter, then this feature set can prove efficient and robust. It is clear that if any object is added to the list of model objects, one should check whether this pair of features still has the necessary discriminating power.

15.2 I - Invariance,part 1

15.2.1 Invariants and groups

In the previous section the connection between the concepts of invariance and transformation groups was already mentioned. Consider figure 15.1. Invariance under two transformations (i.e. shape deformations) implies invariance under their composition for one thing. This illustrates the importance of the closure property of groups. It is self-evident that features remain invariant under the identity element (i.e. no deformation) and under the inverse transformation if it exists. Features that remain invariant under the transformations of a certain group are said to be invariants for that group. Invariance is a crucial issue in geometry and was even declared its basic ordering principle by Felix Klein [1]. In fact, invariants for other algebraic structures than groups are possible. However, it is for

[1]Felix Klein (1849-1925), who had a degree in mathematics and physics from the University of Bonn, concentrated mainly on physics at the beginning of his career. Soon, however, he also became active in the field of geometry, influenced by people like Plücker (1801-1868), Clebsh (1833-1872) - he has been an assistant of both - and Cayley (1821-1895). To take up a teaching post at the University of Erlangen, Klein had to give an inaugural lecture. In this lecture, which later became famous as the "Erlanger Programm", he proposed to build the theory of geometry around the central concept of invariance. As a professor in Göttingen Klein has played an important role in assuring that girl students could obtain a degree, which was e.g. impossible in Berlin at the time. Kowalewskaja was his student and Noether was one of his assistants.

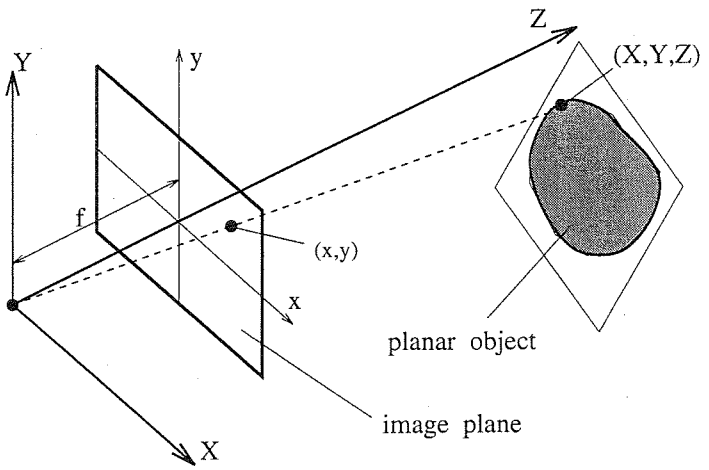

Figure 15.2: *Pinhole camera model, f is the focal length, X, Y, Z are coordinate in the world coordinate frame and x, y coordinates in the image coordinate frame. A planar object contour is observed.*

groups that the mathematical theory for their extraction is reasonably developed.

It is certainly not possible to write a self-contained and complete overview of this theory in a few pages. Nevertheless, we will try to give the persistent reader some intuitive understanding of one technique for the extraction of invariants, based on Lie group theory. We'll come back to this issue in section 15.2.4.

The first problem in finding invariants is the identification of the transformation group that underlies the problem at hand. Here we consider this question for the simplest case, that of planar shapes. It is shown that different viewing conditions lead to different groups.

15.2.2 Deformation of planar curves under projection

The study of shape will be taken to be synonymous to the study of object outlines here. Moreover, we assume that these outlines (also referred to as curves or contours) are planar. This of course is a serious restriction but is not unreasonable when industrial workpieces are of interest, since the large majorities of these have planar faces. The situation is depicted in fig. 15.2 and is a special case of projection with a pinhole camera (cf. chapter 4).

In this section we discuss the different types of deformations to be expected under projection for such planar curves. We study 3 cases:

1. The curves are viewed from a perpendicular direction and can only rotate about an axis parallel to this direction, any 3D translation is allowed.

2. The curves can be viewed from any direction but should be at a sufficient distance from the camera so as to make the differences in depth (Z) values small relative to the distance to the camera.

3. The curves can be viewed from any direction and can be placed at arbitrary distances from the camera.

It is clear that the camera-object configurations become increasingly arbitrary going down this list. As we will see, the corresponding geometries are of increasing complexity.

Case 1

We suppose the curve is described by specifying the three-dimensional coordinates as a function of a parameter t: $(X(t), Y(t), Z(t))$. As we know, in this case we only allow the curve to rotate around an axis parallel to the Z axis. Furthermore any three-dimensional translation can be applied. If we denote the coordinates of the curves after such transformation as $(X'(t), Y'(t), Z'(t))$, we find a relationship of the form

$$\begin{aligned} X' &= \cos\theta X - \sin\theta Y + t_1, \\ Y' &= \sin\theta X + \cos\theta Y + t_2, \\ Z' &= Z + t_3, \end{aligned}$$

Now, suppose that the curve points $(X(t), Y(t), Z(t))$ project to image points $(x(t), y(t))$ and the points $(X'(t), Y'(t), Z'(t))$ to $(x'(t), y'(t))$. We are interested in the transformation bringing the coordinates $(x(t), y(t))$ into correspondence with $(x'(t), y'(t))$, i.e. in the group of transformations describing the deformations between two camera views.

This relationship is particularly easy to derive in this case. Since the planes containing the two curves, i.e. the curve before and the curve after the transformation, are both parallel to the image plane, points of each of the curves will have a constant Z value. This brings us in the situation where the projection equations simplify to a scaling. We have that

$$\begin{aligned} x &= kX, & x' &= k'X', \\ y &= kY, & y' &= k'Y', \end{aligned}$$

with $k = f/Z$ and $k' = f'/Z'$ two constant scale factors. Thus, using the relationships between X' and X we e.g. find that

$$\frac{x'}{k'} = \cos\theta \frac{x}{k} - \sin\theta \frac{y}{k} + t_1.$$

Also using the relationship between Y' and Y we obtain

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{k'}{k} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

We conclude that the image projection of the curve undergoes a similarity transformation. In particular, the curve is transformed via combined scaling, rotation, and translation in the image plane. As a further simplification we could consider the case where the object doesn't come closer to or move away from the camera, i.e. $t_3 = 0$. Then $Z' = Z$, hence $k' = k$ if also the camera focal length remains unchanged and the transformation becomes

a Euclidean motion, i.e. a combination of rotation and translation (in the image plane). This case is highly relevant in industrial applications where cameras often are placed at a fixed distance from e.g. a conveyor, and looking perpendicarly upon it.

Case 2

In this case the curve can undergo arbitrary three-dimensional rotations and translations but the projection model is simplified to a scaling:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

with (r_{ij}) a rotation matrix and (t_i) a translation vector, and

$$\begin{aligned} x &= kX, & x' &= k'X', \\ y &= kY, & y' &= k'Y'. \end{aligned}$$

Using much of a procedure as in case 1, we easily derive

$$x' = \frac{k'}{k} r_{11} x + \frac{k'}{k} r_{12} y + k'\, r_{13} Z + k'\, t_1.$$

In order to eliminate Z we use the planarity of the curve $(X(t), Y(t), Z(t))$ which implies that real numbers a, b, c and d can be found such that

$$aX + bY + cZ + d = \frac{a}{k} x + \frac{b}{k} y + cZ + d = 0.$$

Hence

$$Z = -\frac{a}{kc} x - \frac{b}{kc} y - \frac{d}{c}.$$

Using this equation to eliminate Z and carrying out similar calculations for y' we arrive at a relationship of the type

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

This is a two-dimensional *affine transformation* or *affinity*.

Case 3

We again assume that the curve undergoes an arbitrary three-dimensional motion (rotation + translation), but we take the perspective projection model this time:

$$\begin{aligned} x &= f\tfrac{X}{Z}, & x' &= f'\tfrac{X'}{Z'}, \\ y &= f\tfrac{Y}{Z}, & y' &= f'\tfrac{Y'}{Z'}. \end{aligned}$$

Expanding the expression for x' yields:

$$\begin{aligned} x' &= f'\frac{X'}{Z'} \\ &= f'\frac{r_{11}X + r_{12}Y + r_{13}Z + t_1}{r_{31}X + r_{32}Y + r_{33}Z + t_3} \\ &= f'\frac{r_{11}\frac{fX}{Z} + r_{12}\frac{fY}{Z} + r_{13}\frac{fZ}{Z} + t_1\frac{f}{Z}}{r_{31}\frac{fX}{Z} + r_{32}\frac{fY}{Z} + r_{33}\frac{fZ}{Z} + t_3\frac{f}{Z}} \end{aligned}$$

Thus,

$$\begin{aligned} x' &= f'\frac{r_{11}x + r_{12}y + r_{13}f + \frac{ft_1}{Z}}{r_{31}x + r_{32}y + r_{33}f + \frac{ft_3}{Z}} \\ y' &= f'\frac{r_{21}x + r_{22}y + r_{23}f + \frac{ft_2}{Z}}{r_{31}x + r_{32}y + r_{33}f + \frac{ft_3}{Z}} \end{aligned}.$$

Because we know the point (X, Y, Z) lies in our contour plane, we can express Z as a function of (x, y) as follows

$$Z = \frac{-df}{ax + by + cf}.$$

Thus

$$\begin{aligned} x' &= f'\frac{(r_{11} - \frac{t_1}{d}a)x + (r_{12} - \frac{t_1}{d}b)y + (r_{13} - \frac{t_1}{d}c)f}{(r_{31} - \frac{t_3}{d}a)x + (r_{32} - \frac{t_3}{d}b)y + (r_{33} - \frac{t_3}{d}c)f}, \\ y' &= f'\frac{(r_{21} - \frac{t_2}{d}a)x + (r_{22} - \frac{t_2}{d}b)y + (r_{23} - \frac{t_2}{d}c)f}{(r_{31} - \frac{t_3}{d}a)x + (r_{32} - \frac{t_3}{d}b)y + (r_{33} - \frac{t_3}{d}c)f}. \end{aligned}$$

We find a two-dimensional *projective transformation* or *projectivity*, i.e. a transformation of the type

$$\begin{aligned} x' &= \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + p_{33}} \\ y' &= \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + p_{33}} \end{aligned}$$

Remarks

Now our recognition strategy can be simple, at least in principle. Depending on the case, invariants under the corresponding transformation group will be extracted. On the one hand, these invariants are stored in memory, e.g. averaged over several views, to form a model data-base built off-line. On the other hand, during the actual recognition task, invariants are extracted from the image and compared to the invariants stored in the database. In order for an object to be recognized, it should yield the same values for the invariants as those found for that object in the database.

As already pointed out in the introduction to this section, the complexity of the relevant transformation groups goes up with the generality of the camera-object configuration. In order to give a more intuitive idea of the difference between these cases figure 15.3 shows typical deformations for a square under the three types of transformations.

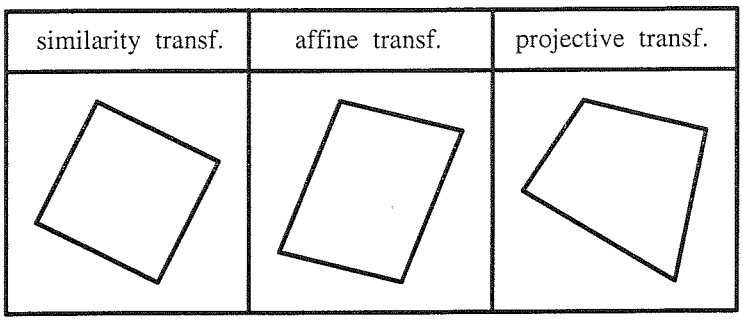

Figure 15.3: *Effects of projection of three types of transformations. The reference object is a square. As can be seen, similarity transformations don't alter the shape (but possibly the scale, the position, and the orientation). Affine transformations lead to deformations, but parallelism is preserved. Projective transformations induce even stronger deformation.*

Notice that for the affine and projective cases (cases 2 and 3) we have actually never used the fact that (r_{ij}) is a rotation matrix. Therefore, we can replace the rotation matrix by an arbitrary 3×3 matrix, thereby replacing the three-dimensional motion of (X, Y, Z) by an arbitrary *three-dimensional* affine transformation. We would still find two-dimensional affine and projective transformations between the different projections of a shape. The interested reader can check that in case 3 one might even apply an additional 3D projective transformation to (X, Y, Z) and still find a projective deformation between image views. What this means is that accepting invariance under affine or projective transformations as proof for shape equivalence for cases 2 and 3, resp., entails a risk of the system being confounded by shapes that are actually different. As an example, suppose a square has to be recognised under the conditions of case 2. Applying an affine transformation to a square yields a parallelogram (cf. fig. 15.3). Affine invariants do not allow to tell both shapes apart. This is not a drawback which is specific to the use of invariants. If no information other than a single image is available (additional information could e.g. be the orientation of the object plane), then there is no way to know for sure that the observed shape is a square. Both will in general project as a parallelogram on the image. If both a square and a parallelogram are in the database, the system cannot decide. All squares, rectangles, and parallelograms are affinely equivalent. From a mathematical point of view, the affine and projective transformations form the smallest groups that include all the deformations between different image projections of the shapes. Therefore, they are the optimal choice nevertheless.

As a final remark, we raise a related issue. The transformation between the original shape and its projection under the conditions of case 3 is called a *perspectivity*. This situation corresponds with setting $Z = f$ and with restricting r_{ij} to represent the components of a 3D rotation matrix. Perspectivities form a proper subset of the projectivities, but not a subgroup. In particular, although two views of a shape will differ from that shape by a perspectivity, the transformation between the two views is not a perspectivity in the general case, but a projectivity as calculated for case 3.

15.2.3 The reparameterization group

In the previous section, the analysis of deformations through projection led to several relevant groups. Especially for the description of contours, besides contour coordinates themselves, contour coordinate derivatives are used. Such derivatives are taken with respect to some contour parameter, say t. When the parameterization is changed, the derivatives will take on different values, however. As the parameterization of the contours in different images will be unrelated, the features should be invariant under such reparameterization as well. Choosing corresponding parameterizations, i.e. giving the same points the same parameter value in different views, clearly is impossible without knowledge about the mapping from one view to the other. This is part of the problem rather than the solution.

Consider a general curve parametrised as $(x(t), y(t))$ and define a reparametrization $t = \phi(t^*)$. Denoting differentiation with respect to t by a dot, and differentiation with respect to t^* by a prime, the following transformation matrix for the derivatives is obtained:

$$\begin{pmatrix} x' \\ x'' \\ x''' \\ \vdots \end{pmatrix} = \begin{pmatrix} \phi' & 0 & 0 & \dots \\ \phi'' & \phi'^2 & 0 & \dots \\ \phi''' & 3\phi'\phi'' & \phi'^3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \\ \vdots \end{pmatrix} \tag{15.1}$$

Similar relations hold for y of course. Taking into account that the function ϕ is an arbitrary monotone function, we see that the derivatives transform as follows under reparameterization:

$$\begin{pmatrix} x \\ x' \\ x'' \\ x''' \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & a & 0 & 0 & \dots \\ 0 & b & a^2 & 0 & \dots \\ 0 & c & 3ab & a^3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \\ \vdots \end{pmatrix} \tag{15.2}$$

where $a > 0$ (or $a \neq 0$) for orientation-preserving (resp. arbitrary) reparametrisations and where $b, c, \dots$ can take on arbitrary values.

It is easy to check that the set of the above type of matrices is closed under matrix multiplication. In fact, they form a group, which we'll refer at as the *reparametrization group*. Note that for both x and y coordinates of a single point the same reparameterization matrix applies. But at different points a different matrix will have to be considered and each such matrix will add to the list of parameters to remain invariant under.

In summary, in order to extract useful features for the recognition of plane curves, both the group induced by projection ánd the reparameterization group will have to be considered and the features should be invariant under both.

15.2.4 * Introduction to Lie group theory

This section gives a concise and admittedly oversimplified introduction to Lie group theory. The aim is to arrive at a method for the systematic extraction of invariants. Although

different strategies for the systematic extraction of invariants exist, it is to the content of the authors that the Lie approach is one of the more systematic, elegant, clarifying, and generally applicable ones.

Let us explain the problem via the simple example of the group of rotations in the plane (i.e. SO(2)). The action of this group on the points of the plane is well-known:

$$p \equiv \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto R_\theta\, p \equiv \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where R_θ stands for rotation over an angle θ. A function f is invariant for this action if $f(R_\theta p) = f(p)$ for all θ and $(x, y)^T$. It is clear that in this case invariants are functions which depend on the radial distance only ($f = f(x^2 + y^2)$). If we enlarge the group to the group of Euclidean motions (i.e. $E(2)$, the group of rotations + translations), then the only invariants that can be defined using no other information than x and y are constant functions. The essential difference between the two cases is that in the latter case, given two arbitrary points we can find a transformation (i.e. a Euclidean motion) which maps the first onto the second. We can "fill" the complete space of (x, y) points by considering a single point and letting all transformations act on this point. This is of course not the case if we restrict ourselves to rotations. In that case we only obtain circles through the initial point and centered around the origin. The set of points that can be reached from a given point by applying all possible transformations is usually referred to as an *orbit*.

We begin to see an interpretation of invariants. Consider again $SO(2)$. Parameterizing the space by taking polar coordinates, the radial coordinate is seen to identify the circular orbits in the sense that all points along the circle have the same radial coordinate. This radial coordinate is an invariant for the rotations. In order to be able to construct a parametrization of the space such that at least one of the corresponding coordinates remains invariant along a single orbit whereas transversal coordinates do not, the orbits should have a lower dimension than the space itself. This observation might seem almost trivial but will play an important part in the methods to be expounded. As is so often the case, the fundamental observations underlying a technique are pretty simple. In the case of the Euclidean motions such parameterisation of the space cannot be found, since the transformations act so as to fill the space completely. In order to find non-trivial invariants in a two-dimensional space as our image plane, the orbits should be curves, as is the case with one-dimensional translations, scaling, or rotation around the optical axis. Combinations are not allowed.

Does this mean that we can only deal with the very simplest of groups? The situation is not that gloomy in fact. In general, invariants can be found if the group acts on the data in such a way as to generate an orbit of lower dimension than the space of the data. The "data" should not necessarily be the coordinates of a single point though. Let us illustrate this suggestion on the basis of the same examples. Indeed, we can obtain a non-trivial invariant for the Euclidean motions if we are prepared to look at functions which depend on the coordinates of two points (x_1, y_1) and (x_2, y_2) (*2-point invariants*). Take for instance f to be the Euclidean distance between the two points, then it is well-known that f remains

invariant under the action of $E(2)$. The coordinates of the 2 points yield a 4-dimensional space. A point in this 4-dimensional space, say having coordinates (x_1, y_1, x_2, y_2) cannot be brought to any other point in this space, as we know, since the distance between the points if preserved under rotation and translation. The orbit created under the action of the group when starting with a single point is 3-dimensional in this case, 3 being the number of independent transformation parameters: one for rotation and two for translation.

Both $SO(2)$ and $E(2)$ are in fact *Lie groups*, as are most of the groups vision people could get interested in. Loosely speaking this means that such groups can be (at least locally) described by means of differentiable functions of k (say) continuously varying parameters. This number k is called the *dimension* of the Lie group. The difference between the dimension of the group and the space on which it acts yields the number of independent [2] invariants that can be expected [3]. In the examples above, the 1-dimensional rotation group $SO(2)$ acting on the 2-dimensional image plane ($\mathbb{R}^2$ for that matter) yielded one independent invariant, being the radial distance. As for the second example, since dim $E(2) = 3$ exceeds the dimension of $\mathbb{R}^2$, there cannot be a non-trivial invariant based on the coordinates of a single point. But looking at 2-point invariants the space on which the group acts is actually 4-dimensional, thus indicating there should be 1 independent invariant.

Instead of simply realizing that invariants should exist, we would rather have a method to actually derive them. To this end we consider the following situation: suppose we start with some data vector $\bar{d}(0)$ and then change the data by the application of a continuously changing transformation, written $a(t)$ and such that $a(0) = e$, the identity element of the group. The result is a continuous change in the data vector, expressed as

$$\bar{d}(t) = a(t).\bar{d}(0) \ ,$$

where the dot indicates the action of the group on the data vector. It is especially instructive to think of t as time. We could e.g. consider rotations for $a(t)$ and the coordinates of a point for $\bar{d}(t)$:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{pmatrix} \begin{pmatrix} x(0) \\ y(0) \end{pmatrix}$$

describes the group action on the point, with e.g.

$$\theta(t) = \omega t \ .$$

In order for a function f of these data to be an invariant, we should have

$$\frac{d}{dt} f(d_1(t), d_2(t), \ldots, d_m(t)) = 0 \ ,$$

[2] By "independent" we mean that none of these invariants can be expressed as a function of the others.

[3] Some care should be taken in applying this "counting argument" though, since invariants might sometimes exist even in cases where one would not expect to find them. Examples are given in later sections.

writing the components of the data vector $\bar{d}(t)$ as $d_j(t)$ and supposing there are m of them. Similarly, writing the k group parameters as a_i, we reformulate the previous equation as

$$\sum_{j=1}^{m}\sum_{i=1}^{k}\frac{\partial f}{\partial d_j}\frac{\partial d_j}{\partial a_i}\frac{da_i}{dt}=0\ .$$

Two important remarks are in place here. First, notice that we want to find functions that remain invariant irrespective of the precise sequence of transformations $a(t)$, i.e. irrespective of the precise values of the $\frac{da_i}{dt}$. Invariance should be realized under any transformation of the group. This implies that we should solve a system of k equations of the type

$$\sum_{j=1}^{m}\frac{\partial f}{\partial d_j}\frac{\partial d_j}{\partial a_i}=0\ .$$

Lie groups have the property that any element can be obtained as the composition of elements arbitrarily close to the identity. As an example, any rotation around the origin can be obtained as a sequence of arbitrarily small rotations. Each rotation brings us at a new point, which serves as starting point for the next rotation. As Lie group theory shows, it is then sufficient to demand invariance under transformations in a small neighbourhood around the identity for arbitrary elements of the measurements. Since a rigorous mathematical underpinning is out of the scope of this introduction, it is hoped the reader will find the following argument helpful. Moving along an orbit, the function f will in general change its values, and we obtain a curve $f(t)$. If f is continuous in the measurements (we will need differentiability later on), it will be continuous in t (we only consider Lie groups in which case the group parameters are continuous in t). A continuous function with $\frac{df}{dt}=0$ for all t has to be a constant. Hence, it is sufficient to demand invariance locally for each "small" transformation separately, i.e. evaluate the expressions $\frac{\partial d_j}{\partial a_i}$ around the identity element. We obtain

$$\sum_{j=1}^{m}\frac{\partial f}{\partial d_j}\frac{\partial d_j}{\partial a_i}|_{a_k=e_k}=0$$

where e_k stands for the parameter values corresponding to the identity element. Such system of linear, partial differential equations is called a *Lie prolongation.* In order to have a non-trivial solution, the number of independent variables should surpass the number of *independent* equations. This *counting argument* tallies with our observations, but at the same time is more sophisticated. Usually, the number of independent equations is the number of parameters, but it might be lower. In that case, invariance might exist where we didn't expect them before.
In summary, for a Lie group each transformation can be written as a product of transformations that come arbitrarily close to the identity transformation. Realizing invariance under each such individual transformation then guarantees invariance under the whole transformation and we find a system of linear, first-order partial differential equations, the solutions of which are the invariants for the selected measurement set.

Let us try to clarify these ideas on the basis of a simple example: good old $SO(2)$. Let us first derive the system of equations for the case where the data are two-dimensional

vectors $(d_1, d_2) = (x, y)$, i.e. as data we simply take the coordinates of an image point. Using

$$\begin{aligned} \frac{\partial x}{\partial \theta}|_{\theta=0} &= -y \ , \\ \frac{\partial y}{\partial \theta}|_{\theta=0} &= x \ . \end{aligned}$$

we find as Lie prolongation system for this one-parameter group a single equation:

$$-y\frac{\partial f}{\partial x} + x\frac{\partial f}{\partial y} = 0 \ .$$

Solutions for this equation (invariants!) will be described later. As another simple example, let us consider the group of two-dimensional translations:

$$\begin{aligned} x(t) &= x(0) + a_1(t) \ , \\ y(t) &= y(0) + a_2(t) \ . \end{aligned}$$

We immediately see that

$$\frac{\partial x}{\partial a_1} = 1 \ , \qquad \frac{\partial y}{\partial a_1} = 0 \ , \qquad \frac{\partial x}{\partial a_2} = 0 \ , \qquad \frac{\partial y}{\partial a_2} = 1 \ .$$

We then find the system of equations

$$\begin{aligned} \frac{\partial f}{\partial x} &= 0 \ , \\ \frac{\partial f}{\partial y} &= 0 \ . \end{aligned}$$

An invariant should satisfy both equations. A discussion follows.

15.2.5 The counting argument

Readers who have read the previous section can skip this one.

It has already been suggested that a group that introduces a wider diversity of deformations, allows fewer invariants. The reader can easily verify that fact for the different groups discussed for the case of planar shapes. Ratios of lengths are invariant under similarities, but not under affinities or projectivities. On the other hand, the ratio of lengths of two segments on the same straight line is affine invariant, but is not projective invariant either.

Fortunately, there is a rather simple rule on how complex an invariant for a given group may get, the so-called *counting argument*, and it goes like this: the number of independent invariants that one can calculate for a transformation group with p independent parameters from a set of m independent measurements is at least $m - p$. As mentioned before, sometimes invariance under more than one group has to be acquired. If one were e.g. to use coordinate derivatives as measurements, then also invariance under reparameterisation

has to be reckoned with. The above counting rule can be generalised by adding the number of parameters of the different groups involved.

As an example, assume an affine invariant has to be found from a contour. This is to be achieved by selecting a number of points on the contour and by combining their coordinates. Since the affine group has 6 independent parameters, at least 7 independent measurements will have to be combined into the expression of the invariant. Since point coordinates come as pairs (the affine group action is only defined on *pairs* of x and y coordinates), one has to use at least 4 points, which yields 8 measurements. The counting argument predicts the existence of at least $8 - 6 = 2$ independent invariants, which are to be derived later.

Now assume that only the coordinates of 3 points are to be used, but one can also use the first derivatives of the coordinates of one point with respect to the parameterisation of the contour, which however is arbitrary. The reparameterisation group for the first derivatives introduces a single parameter for the single point where it operates. Hence, there are now 6 affine transformation parameters and 1 reparameterisation parameter, totaling to 7 parameters. On the other hand, one still has 8 measurements. The counting argument predicts the existence of at least one invariant, which will be derived later as well.

It should be mentioned that the counting argument is sometimes formulated without the "at least". One should beware, however, since invariants can well exist without the counting argument predicting them. It is but a rule-of-thumb and readers interested in a mathematically rigorous version should consult section 15.2.4 on Lie theory.

15.2.6 Relative invariants

Invariants discussed so far are *absolute invariants*. In fact, they are a special case of so-called *relative invariants*. Relative invariants do change under the transformations of the corresponding group, but only in a very restricted way. Upon transformation, a relative invariant must only change by a factor that depends solely on the transformation parameters. As an example, consider simple scaling operations in the plane. Lengths are relative invariants. All lengths change with the same factor, the scaling parameter. Relative invariants are generally easier to obtain than absolute invariants, where this factor should be 1 irrespective of the precise transformation. [4] Let it suffice to mention – as a rule of thumb – that a relative invariant can be found as soon as the number of measurements equals the number of independent equations in the prolongation, but there is no guarantee as with the counting argument for absolute invariants.

A special role played by relative invariants for the reparameterization group is worth noting. As already mentioned, for expressions containing coordinate derivatives, invariance under both the deformation group and the reparameterization group should be considered. But, as we now illustrate, even if an expression is only relative invariant under reparameterization, it is highly useful. Actually, an invariant parameter can be generated via simple

[4] Lie group theory can be extended to include the extraction of relative invariants, but we will not discuss this possibility.

integration. If an expression F remains invariant up to a factor of the type a^w with w a real number and a having the meaning of section 15.2.3, then

$$\int \text{abs}\left(F^{1/w}\right) dt = \int \text{abs}\left(F^{*1/w}\right) dt^* \tag{15.3}$$

with expression F calculated by differentiation with respect to t and F^* the same expression be evaluated with respect to a parameter t^*: starting from the same contour point, subsequent contour points that are traversed when following the contour correspond to montonously increasing, identical values. These can be compared to arclength in the Euclidean case. We have an invariant parameter at our hands: the contour "length" between points calculated with such parameter is always the same. In practice, the starting point will probably differ (otherwise we have to find corresponding points), replacing the original factor a by an offset. Invariant parameters are so useful because they impose a natural, invariant ordering on the contours.

15.2.7 Invariant features for planar curves

This section gives invariants for the viewpoint invariant description and recognition of planar shapes. In particular, invariants that are extracted from the shape contours and for the groups derived in section 15.2.2 will be derived.

Readers who have taken the time to read section 15.2.4 will have noticed that Lie theory offers a systematic but equally tedious strategy for the actual calculation of the invariants. Hence, this section takes a shortcut. It should be noted, however, that the Lie prolongation approach nevertheless is both highly useful and practical. It allows to precisely predict the minimal amount of information that is required to arrive at an invariant.

First, section 15.2.7 fixes some issues of notation. Section 15.2.7 introduces some special expressions in the contour coordinates and their derivatives that will serve as building blocks for the generation of invariants, as discussed in section 15.2.7. Finally, some examples of invariants are given in section 15.2.8.

Notation

Henceforth, the image coordinates of a point on a contour or edge are denoted by $\mathbf{x} = (x, y)^T$, where ^{T} stands for the transpose, as usual. Subscripts as in $\mathbf{x}_i = (x_i, y_i)^T$ identify specific points. Superscripts between curved brackets indicate that a derivative of the coordinates with respect to an arbitrary curve parameter, denoted t, is taken:

$$\mathbf{x}^{(j)} = \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} = \begin{pmatrix} \frac{d^j x}{dt^j} \\ \frac{d^j y}{dt^j} \end{pmatrix} .$$

The notation $\mathbf{x}_i^{(j)}$ then indicates the jth derivative of the coordinates at $= \mathbf{x}_i$. The notation $\mathbf{x}^{(j:s)} = (x^{(j:s)}, y^{(j:s)})^T$ is used to denote derivatives with respect to a specific parameter s. As a matter of fact, the notation s will be reserved for Euclidean arclength.

Single bars (|) indicate determinants, double bars (||) norms. As an example

$$|\mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_3| = (x_1 - x_2)(y_1 - y_3) - (y_1 - y_2)(x_1 - x_3)$$

and

$$\|\mathbf{x}^{(1)}\| = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \ ,$$

with t an arbitrary parameterisation of the curve.

Building blocks for invariants

As noted earlier, if an expression is invariant under a group, it is invariant under all its subgroups as well. Invariance under a subgroup does not imply invariance under the group, however. Hence, in studying invariants it is natural to start with the largest group under scrutiny. For the current discussion this means that the projectivities are studied first, followed by the affinities, and subsequently the similarities and the Euclidean motions. Of course, deriving the invariants, care will be taken as to use as few points and derivatives as possible, i.e. to generate the simplest invariants.

As already mentioned in the introduction of section 15.2.7, the invariants will be derived on the basis of building blocks. These building blocks behave in a particularly simple manner under the different transformations and are now derived first.

Projectivities First, recall the expression for planar projective transformations:

$$\begin{aligned} x' &= \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + p_{33}} \ , \\ y' &= \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + p_{33}} \ . \end{aligned}$$

In the sequel, N will serve as a shorthand notation for the denominator of these expressions:

$$N = p_{31}x + p_{32}y + p_{33}$$

and similarly $N_i = p_{31}x_i + p_{32}y_i + p_{33}$. Note that the projective transformation can then be rewritten as

$$N \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \ .$$

Hence, for three transformed points the action of the projectivity can be summarized as

$$\begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1/N_1 & 0 & 0 \\ 0 & 1/N_2 & 0 \\ 0 & 0 & 1/N_3 \end{pmatrix} \ .$$

Taking the deteriminant of the left and right hand sides immediately yields

$$|\mathbf{x}'_1 - \mathbf{x}'_2 \;\; \mathbf{x}'_1 - \mathbf{x}'_3| = \frac{|P|}{N_1 N_2 N_3} |\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3|$$

where P denotes the matrix of the transformation parameters p_{ij}. The determinant expressions on both hand sides, i.e. before and after the transformation, are related by a factor. This factor in turn is composed of a number of factors. In order to find invariants one can resort to the simple strategy of eliminating all these factors. One should be careful not to generate trivial invariants (i.e. constants) though.

This type of determinant is not the only kind of building block one may use. The following list gives an overview of those used in this volume, together with the factors by which they change under a projective transformation:

building block		factor
$\|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3\|$	$\rightarrow$	$\frac{\|P\|}{N_1 N_2 N_3}$
$\|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)}\|$	$\rightarrow$	$\frac{\|P\|}{N_1^2 N_2}$
$\|\mathbf{x}_1^{(1)} \;\; \mathbf{x}_1^{(2)}\|$	$\rightarrow$	$\frac{\|P\|}{N_1^3}$

Each of these building blocks has an especially simple interpretation if the derivatives are calculated with respect to Euclidean arclength s,

$$s(t_x) = \int_0^{t_x} \|\mathbf{x}^{(1:t)}\| \; dt$$

with t an arbitrary parameter and t_x its particular value at the point $\mathbf{x}$ which moves along the curve when t changes. It corresponds to length measured along the contour. The first building block gives the area of the parallelogram defined by the three points. Using s for the parameter, the second block measures the distance of point $\mathbf{x}_2$ to the tangent line to the contour at point $\mathbf{x}_1$. This is the case because the tangent vector $\mathbf{x}_1^{(1)}$ is a unit vector (choice of Euclidean arclength as parameter!). The third block expresses (Euclidean) curvature at the point $\mathbf{x}_1$. Again, this interpretation hinges on the choice of Euclidean arclength for the parameter, since in general curvature is given by

$$\frac{|\mathbf{x}^{(1)} \;\; \mathbf{x}^{(2)}|}{\|\mathbf{x}^{(1)}\|^3} \; .$$

These interpretations are illustrated in fig. 15.4. The reader should keep in mind that the second and third building blocks no longer assume these simple meanings when arbitrary parameterisations are used.

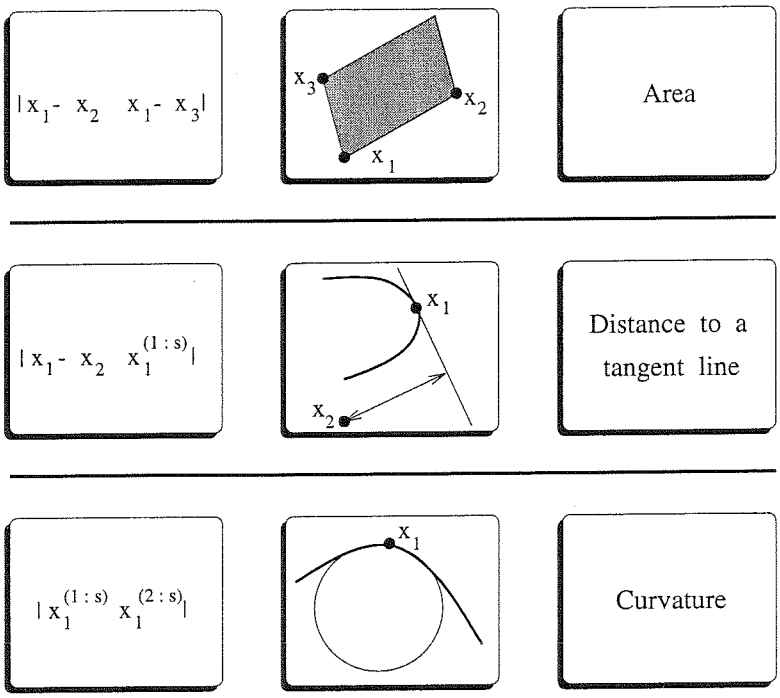

Figure 15.4: *Euclidean interpretations for the different projective building blocks. Left: formula for the building block; middle: graphical interpretation; right: interpretation in wording.*

Finally, an example is given to illustrate how the combination of building blocks can yield invariants. Consider

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_5|\,|\mathbf{x}_3 - \mathbf{x}_4 \quad \mathbf{x}_3 - \mathbf{x}_5|}{|\mathbf{x}_1 - \mathbf{x}_3 \quad \mathbf{x}_1 - \mathbf{x}_5|\,|\mathbf{x}_2 - \mathbf{x}_4 \quad \mathbf{x}_2 - \mathbf{x}_5|} .$$

It is an easy exercise to check that all the factors produced under a projective transformation cancel. Notice that this invariant uses 5 points. According to the counting argument (section 15.2.5), this is the minimum number of points one would expect, if no derivatives are used. The 5 points introduce 10 measurements (10 coordinates), which is sufficient to surpass the 8 degrees of freedom (independent parameters) of the projective transformation by 2. As a consequence, a second, independent invariant on the basis of the same 5 points can be constructed:

$$\frac{|\mathbf{x}_3 - \mathbf{x}_2 \quad \mathbf{x}_3 - \mathbf{x}_5|\,|\mathbf{x}_1 - \mathbf{x}_4 \quad \mathbf{x}_1 - \mathbf{x}_5|}{|\mathbf{x}_1 - \mathbf{x}_3 \quad \mathbf{x}_1 - \mathbf{x}_5|\,|\mathbf{x}_2 - \mathbf{x}_4 \quad \mathbf{x}_2 - \mathbf{x}_5|} .$$

Additional invariants based on the coordinates of the 5 points will necessarily be functionally dependent on the 2 foregoing invariants.

Affinities Of course, all projective invariants are also affine invariants. Using projective invariants where affine invariants suffice is certainly not to be recommended. Since the group of planar affine transformations has only 6 independent parameters, affine invariants can be expected to be simpler than projective invariants. Indeed, recall that affinities are transformations of the type

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} .$$

In order to identify building blocks for the affine invariants, the following observations are useful. First we note that we can eliminate the influence of the translation by considering coordinate differences or derivatives, i.e. using measurements of the type $(x_i - x_j, y_i - y_j)^T$ or $(\frac{d^n x}{dt^n}, \frac{d^n y}{dt^n})^T$ with $n \geq 1$. Writing these vectors as $\mathbf{v} = (v_x, v_y)^T$ we find

$$\begin{pmatrix} v'_x \\ v'_y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

Having two such vectors $\mathbf{v}$ and $\mathbf{w}$ it follows that

$$\begin{pmatrix} v'_x & w'_x \\ v'_y & w'_y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} v_x & w_x \\ v_y & w_y \end{pmatrix}$$

and hence

$$|\mathbf{v}' \quad \mathbf{w}'| = |A|\,|\mathbf{v} \quad \mathbf{w}| ,$$

where A is the matrix with elements a_{ij}. It immediately transpires that such expressions $|\bar{v} \ \bar{w}|$ are absolute invariants under area preserving affine transformations (i.e. equiaffine or unimodular transformations for which the determinant $|A|$ is 1) and are in general relative invariants. The following is a list of affine building blocks with their factors:

building block		factor
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_3 \rvert$	$\rightarrow$	$\lvert A \rvert$
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1^{(1)} \rvert$	$\rightarrow$	$\lvert A \rvert$
$\lvert \mathbf{x}_1^{(1)} \quad \mathbf{x}_1^{(2)} \rvert$	$\rightarrow$	$\lvert A \rvert$
$\lvert \mathbf{x}_1^{(1)} \quad \mathbf{x}_2^{(1)} \rvert$	$\rightarrow$	$\lvert A \rvert$

Note that, compared to the projective case, a building block has been added. Upon choosing Euclidean arclength as the contour parameter, the second and third block obtain the same interpretation as for the projective case (fig. 15.4) and the fourth block becomes the sine of the angle subtended by the tangent vectors $\mathbf{x}_1^{(1)}$ and $\mathbf{x}_2^{(1)}$.

The extraction of invariants is simplified by the simpler block factors and the addition of the fourth block. Since all factors are equal, invariants under affine transformations are found by taking the ratio of 2 blocks. An example invariant is

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_3 \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_2 \quad \mathbf{x}_1 - \mathbf{x}_4 \rvert} .$$

This invariant requires only 4 points, as opposed to the 5 points used by the projective invariants given earlier. This is not surprising since the 8 coordinates offered by 4 points are higher in number than the 6 degrees of freedom (independent parameters) to be lifted. Since the difference is 2, a second independent invariant can be found, e.g.

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_3 \quad \mathbf{x}_1 - \mathbf{x}_2 \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_3 \quad \mathbf{x}_1 - \mathbf{x}_4 \rvert} .$$

Similarities A further specialisation of the transformations leads to similarities. These combine rotation, translation, and scaling. Coordinates are thus related by a transformation of the type

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s\cos\theta & -s\sin\theta \\ s\sin\theta & s\cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} .$$

It is easy to check that the following list of blocks and factors apply:

building block		factor
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert$	$\rightarrow$	s^2
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert$	$\rightarrow$	s^2
$\lvert \mathbf{x}_1^{(1)} \;\; \mathbf{x}_1^{(2)} \rvert$	$\rightarrow$	s^2
$\lvert \mathbf{x}_1^{(1)} \;\; \mathbf{x}_2^{(1)} \rvert$	$\rightarrow$	s^2
$\lVert \mathbf{x}_1 - \mathbf{x}_2 \rVert$	$\rightarrow$	s
$\lVert \mathbf{x}_1^{(1)} \rVert$	$\rightarrow$	s

These building blocks are relative invariants.

The group of similarities has 4 independent parameters. The following pair of independent invariants can be formed with 2 points

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert}{\lVert \mathbf{x}_1 - \mathbf{x}_2 \rVert^2}$$

and

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert}{\lVert \mathbf{x}_1 - \mathbf{x}_3 \rVert^2} \; .$$

Euclidean motions Euclidean motions combine rotations and translations in the plane. What distinguishes them from the similarities is that they do not include scaling. Returning to the similarities and putting $s = 1$ shows that the building blocks for the similarities are absolute invariants for the Euclidean motions. The group of Euclidean motions has 3 degrees of freedom. If one picks two points $\mathbf{x}_1$ and $\mathbf{x}_2$ the distance $\lVert \mathbf{x}_1 - \mathbf{x}_2 \rVert$ between them is an invariant. The use of 4 coordinates suffices to lift the group's 3 degrees of freedom.

The reparameterisation group If coordinate derivatives are used for the calculation of invariants, then also the reparamerisation group has to be reckoned with. Indeed, it will only rarely happen that one can use the same parameterisation for two different views of a planar object. This would require complete knowledge of which points correspond in the two views. Hence, the invariants must not change if different parameterisations are used. Fortunately, switching between parameterisations can be regarded as the action of the reparameterisation group, as explained in section 15.2.3. Having identified the relevant group, invariance theory applies again. However, as in the case of the previous groups, also here the shortcut of using building blocks will be taken.

Keeping the discussion of section 15.2.3 in mind, the following list of blocks and factors are seen to apply:

building block		factor
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert$	$\rightarrow$	1
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert$	$\rightarrow$	a_1
$\lvert \mathbf{x}_1^{(1)} \;\; \mathbf{x}_1^{(2)} \rvert$	$\rightarrow$	a_1^3
$\lvert \mathbf{x}_1^{(1)} \;\; \mathbf{x}_2^{(1)} \rvert$	$\rightarrow$	$a_1 a_2$
$\lVert \mathbf{x}_1 - \mathbf{x}_2 \rVert$	$\rightarrow$	1
$\lVert \mathbf{x}_1^{(1)} \rVert$	$\rightarrow$	a_1

The meaning of a is that of section 15.2.3. The subscripts indicate values for the different points used in the expression. It has to be emphasised again that in contrast to the other groups, the parameters of the reparameterisation will differ for each point.

An example of an invariant under reparameterisation is

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert}{\lVert \mathbf{x}_1^{(1)} \rVert} \; .$$

The calculation of invariants

Assuming that the group under which invariance should be obtained is known, the generation of invariants can then proceed as follows.

First one should decide on the kind of measurements on the image that one would like to use. So far, the discussion has been restricted to point coordinates and their derivatives. But even then, one has quite some leeway in as far as the number of points and the orders of derivatives in these points.

Second, the counting argument yields a rule of thumb to estimate the amount of information that one should combine. For instance, if the goal is to find an affine invariant that must not depend on derivatives higher than the first, then the following inequality yields the minimal number of points required:

$$\text{number of points} \times 4 \;\; > \;\; 6 + \text{number of points} \; .$$

On the left hand side of this equation the total number of available measurements are given: for every point 2 point coordinates and their 2 first-order derivatives are available. On the right hand side the total number of group parameters appear. First, there is the affine transformation group with its 6 parameters. Reparameterisation then adds another parameter for every point used. In order for the number of measurements to surpass the number of parameters, at least 3 points have to be used. If a single invariant is required, this is still suboptimal, since the number of measurements is now outnumbering the number of parameters by 3, whereas 1 should suffice. Moreover, coordinates will have higher precision

than their first derivatives, the use of which one would therefore like to restrict. Hence, we have two spare degrees of freedom. If we now only consider the first derivative in a single point, then we loose 4 measurements but also is the number of parameters reduced by 2. This yields 6 coordinates for three points and 2 first-order derivatives for one point – i.e. 8 in total – against 6 affine parameter and 1 reparameterisation parameter, i.e. a total of 7.

Third, the appropriate building blocks are selected to find the invariants. It should be mentioned that the building blocks given in the previous section have already been selected with this step in mind. Additional building blocks for these groups can be found, but only those are interesting that are common to both the reparameterisation group and at least one of the coordinate transformation groups (projective, affine, Euclidean, etc.). Combining three points the following independent building blocks apply for both the affine and the reparameterisation groups:

building block		aff. factor	repar. factor
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert$	$\rightarrow$	$\lvert A \rvert$	1
$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert$	$\rightarrow$	$\lvert A \rvert$	a_1
$\lvert \mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1^{(1)} \rvert$	$\rightarrow$	$\lvert A \rvert$	a_1

Finally, invariant combinations of these blocks are searched. In particular, the search can be focused on expressions that are multiplied powers of the blocks. For the example, expressions of the type

$$\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3 \rvert^a \; \lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert^b \; \lvert \mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1^{(1)} \rvert^c$$

are considered. Such expression will change by a factor

$$\lvert A \rvert^{a+b+c} \; a_1^{b+c} \; .$$

The elimination of the affine and reparameterisation factors then amounts to looking for solutions a, b, and c of the equations

$$\begin{cases} a + b + c = 0 \\ \quad\;\; b + c = 0 \end{cases}$$

Solving these equation yields $a = 0$ and $b = -c$. Hence, the simplest invariant, found for $b = 1$, is

$$\frac{\lvert \mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)} \rvert}{\lvert \mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1^{(1)} \rvert} \; .$$

This invariant extraction scheme is remarkably simple when compared to the solution of e.g. Lie prolongation equations. All example invariants in the following section have been extracted this way.

15.2.8 Examples

This section gives a few more examples of invariants, found by combining building blocks so as to eliminate the factors. It is a useful exercise to check their invariance. The examples are restricted to the two most complicated cases, i.e. affine and projective transformations.

Projectivities

Examples of projective invariants are:

$$\frac{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1-\mathbf{x}_5|\,|\mathbf{x}_3-\mathbf{x}_4\ \ \mathbf{x}_3-\mathbf{x}_5|}{|\mathbf{x}_1-\mathbf{x}_3\ \ \mathbf{x}_1-\mathbf{x}_5|\,|\mathbf{x}_2-\mathbf{x}_4\ \ \mathbf{x}_2-\mathbf{x}_5|}$$

$$\frac{|\mathbf{x}_1-\mathbf{x}_4\ \ \mathbf{x}_1-\mathbf{x}_5|\,|\mathbf{x}_3-\mathbf{x}_2\ \ \mathbf{x}_3-\mathbf{x}_5|}{|\mathbf{x}_1-\mathbf{x}_3\ \ \mathbf{x}_1-\mathbf{x}_5|\,|\mathbf{x}_2-\mathbf{x}_4\ \ \mathbf{x}_2-\mathbf{x}_5|}$$

$$\int \mathrm{abs}\left(\frac{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1^{(1)}|\,|\mathbf{x}_2-\mathbf{x}_3\ \ \mathbf{x}_2-\mathbf{x}_4|}{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1-\mathbf{x}_3|\,|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1-\mathbf{x}_4|}\right) dt$$

$$\frac{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_2^{(1)}|^3\ \ |\mathbf{x}_1^{(1)}\ \ \mathbf{x}_1^{(2)}|}{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1^{(1)}|^3\ \ |\mathbf{x}_2^{(1)}\ \ \mathbf{x}_2^{(2)}|}$$

$$\frac{|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_1^{(1)}|\,|\mathbf{x}_2-\mathbf{x}_3\ \ \mathbf{x}_2^{(1)}|\,|\mathbf{x}_1-\mathbf{x}_3\ \ \mathbf{x}_3^{(1)}|}{|\mathbf{x}_1-\mathbf{x}_3\ \ \mathbf{x}_1^{(1)}|\,|\mathbf{x}_1-\mathbf{x}_2\ \ \mathbf{x}_2^{(1)}|\,|\mathbf{x}_2-\mathbf{x}_3\ \ \mathbf{x}_3^{(1)}|}$$

Note how points can be exchanged for derivatives: if one wants to use fewer points for the invariants, then higher derivatives can be introduced. It is useful to browse through the above list and to consider the number of points and the highest derivatives used for each of them.

The third invariant in the list warrants some additional comment. As can be seen, this expression is not a simple algebraic expression of building blocks. It rather integrates such expression for changing positions of the point $\mathbf{x}_1$, i.e. this point slides along the contour and the parameter t gives the parameter value at $\mathbf{x}_1$. Some contour point is picked as the starting position of the integration. The other points are to be regarded as fixed points, as usual. The reason for this integration is simple and has been expounded already in section 15.2.6. The integrand is only relative invariant under reparameterisation. The integration eliminates the reparameterisation factor introduced by the integrand, but exchanges this variability by a milder form. Indeed, upon changing the starting point of this "invariant" parameterisation, the value of the integral will change. The situation is comparable to that of the well-known Euclidean arclength though. Picking a different contour point to start the integration will result in a simple shift of all the parameter values.

Although these expressions might look pretty complicated, they often have a simple Euclidean interpretation. As an example, consider the fourth of these expressions. The expression being invariant under reparameterisation (i.e. parameter independent), we may take the natural (Euclidean) arclength as the parameter. Following the interpretation of

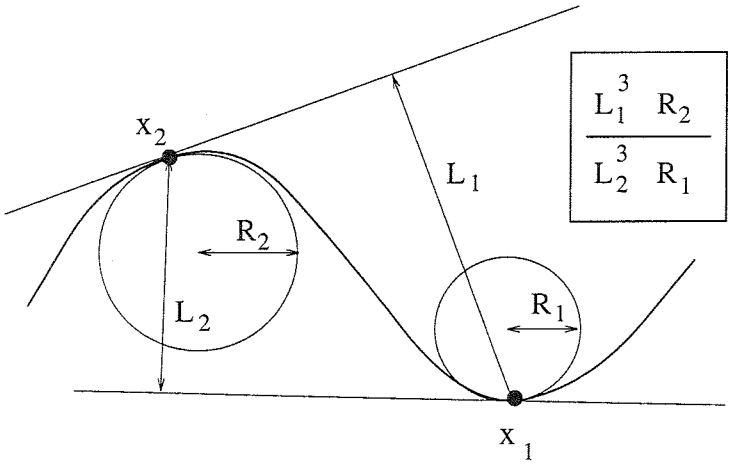

Figure 15.5: *Example of Euclidean interpretation: one of the projective invariants found can be interpreted as a ratio of curvatures and lengths raised to the third power.*

the individual determinants shown in fig. 15.4, this invariant is equivalent to the simple ratio of lengths and curvatures explained in fig. 15.5. This example was highlighted also because of another reason. This invariant combines 12 measurements, and has to lift 12 parameters. Following the counting argument the invariant is not expected to exist! This shows that one should be cautious when applying the counting argument. It is merely a conservative rule of thumb. The Lie theory expounded earlier does account for such cases and is full-proof.

Affinities

Examples of affine invariants are:

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3|}{|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_4|}$$

$$\frac{|\mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1 - \mathbf{x}_2|}{|\mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1 - \mathbf{x}_4|}$$

$$\int \text{abs}\left(\frac{|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)}|}{|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3|}\right) dt$$

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)}|}{|\mathbf{x}_1 - \mathbf{x}_3 \;\; \mathbf{x}_1^{(1)}|}$$

Notice that these invariants are systematically simpler than their projective counterparts. Again, for the invariant parameter (third invariant), the integral is calculated with $\mathbf{x}_1$ sliding along the contour (i.e. the parameter t specifies the position of $\mathbf{x}_1$).

15.3 II - Invariance, Part 2

15.3.1 Invariant configurations from zero-crossings

In section 15.2.7 building blocks were given for the different plane transformation groups. Under the action of the corresponding groups, they change by a factor. This implies, however, that for configurations that make the building blocks vanish, the building block will also vanish after the transformation and therefore has remained invariant. This is a very useful feature from a practical point of view. It implies that the following qualitative aspects are preserved:

1) $|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1 - \mathbf{x}_3| \Rightarrow$ collinearity is preserved

2) $|\mathbf{x}_1 - \mathbf{x}_2 \;\; \mathbf{x}_1^{(1)}| \Rightarrow$ tangency is preserved

3) $|\mathbf{x}_1^{(1)} \;\; \mathbf{x}_1^{(2)}| \Rightarrow$ inflections are preserved

4) $|\mathbf{x}_1^{(1)} \;\; \mathbf{x}_2^{(1)}| \Rightarrow$ parallelism is preserved

15.3.2 Moment invariants

So far, we have focused on the invariant description of curves. Often the intensity or colour information inside the region circumscribed by a curve can also be used for tasks such as recognition of the corresponding object. An example where this information is actually more important than the outline is the recognition of traffic signs. Several of these signs have identical outlines, like circular or triangular. It is the pattern within that sets each apart from the others.

The goal of this section is to introduce a series of invariants that capture information about patterns enclosed by curves. We will assume that these patterns are (more or less) planar. Again, we have to deal with deformations under changing viewpoints. Whereas extracted curves will typically not change much when the illumination changes, this is no longer the case for intensity or colour patterns. Hence, we here also have to deal with *photometric invariance*, i.e. invariance under changing light direction, intensity, and/or colour. In chapter 13 a first step in this direction was taken, but the colour based invariants introduced there don't characterise patterns, only relative colours of isolated points.

For the description of patterns several types of features can be used. There is e.g. quite an extensive literature on the use of histograms or histogram based features. We only consider the use of so-called moments.

Generalised colour moments

A color pattern can mathematically be described as a vector-valued function I defined on a region Ω in the plane; i.e. $I : \Omega \subset I\!R^2 \longrightarrow I\!R^3 : (x, y) \longmapsto I(x, y)$ where x and y stand for the row and column coordinates of an image point and $I(x, y) =$

$(\, i_R(x,y)\,,\, i_G(x,y)\,,\, i_B(x,y)\,)$ is the vector containing the RGB-values of the image point (x,y). We define the *generalized color moments* M_{pq}^{abc} by

$$M_{pq}^{abc} = \iint_\Omega x^p y^q \, [i_R(x,y)]^a \, [i_G(x,y)]^b \, [i_B(x,y)]^c \, dxdy \ .$$

M_{pq}^{abc} is said to be a generalized color moment of *order* $p+q$ and *degree* $a+b+c$. Observe that generalized color moments $M_{pq}000$ of degree 0 in fact are the (p,q)-shape moments of the image region Ω. Furthermore, the generalized color moments of degree 1, viz. $M_{pq}100$, $M_{pq}010$, $M_{pq}001$, are just the (p,q)-intensity moments of respectively the R-, G- and B-color band. On the other hand, the generalized color moments M_{00}^{abc} of order 0 actually are the non-central (a,b,c)-moments of the (multivariate) color distribution of the RGB-values of the pattern. Hence, these generalized color moments actually generalize shape moments of planar shapes, intensity moments of greylevel images, and non-central moments of the color distribution in the image. It is clear that a large number of generalized color moments can be generated with only small values for the order and the degree. This is important as higher values yield values that are more sensitive to outliers, i.e. that are less robust as with high derivative for curves. We therefore only consider generalized color moments up to the first order and the second degree. These generalized color moments will in the sequel be called "moments" for short.

Geometric and photometric transformations

The geometric deformations considered are *affine transformations*:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{15.4}$$

with $|A| = a_{11}a_{22} - a_{12}a_{21} \neq 0$. This implies that the camera is assumed to be relatively far from the object. This assumption together with the fact that the object is planar can greatly simplify the analysis of the photometric changes. Typically, light sources are far from the objects as well. The geometry of light reflection is the same for all points in that case, i.e. they share the same angles of light incidence and camera viewing direction. Also for the more sophisticated models of diffuse reflection the change in camera or light position will in that case result in an overall scaling of intensity. Adding an offset allows to better model the combined effect of diffuse and specular reflection and has been found to give better performance. Thus the photometric changes to be considered ideally comprise *a scaling combined with an offset* for each color band:

$$i'_K(x,y) = s_K \, i_K(x,y) + u_K \qquad \text{with } K \in \{\, R, G, B \,\}.$$

and where $0 \neq s_K, u_K \in I\!R$. The combined effect of the affine and photometric transformations of the moments M_{pq}^{abc} is given by

$$\begin{aligned} \left(M_{pq}^{abc}\right)' = \iint_\Omega & \, [\, a_{11}x + a_{12}y + b_1 \,]^p \, [\, a_{21}x + a_{22}y + b_2 \,]^q \, [\, s_R \, i_R(x,y) + u_R \,]^a \\ & [\, s_G \, i_G(x,y) + u_G \,]^b \, [\, s_B \, i_B(x,y) + u_B \,]^c \, \mathrm{abs}(()|A|) \, dxdy \end{aligned}$$

which, after expansion, yields a linear combination of moments of order $\leq p+q$ and degree $\leq a+b+c$.

Here we drop the photometric offsets, i.e. $u_R = u_G = u_B = 0$. In fact, in all what follows we can deal with such an offset nevertheless, if first the images are always normalised to have zero average values in all three colour bands. So, we silently assume this is the case. A caveat is that by doing so, we loose several of our simplest moments, as the 0th order 1st degree moments M_{00}^{100}, M_{00}^{010}, M_{00}^{001} vanish. This can be avoided by using the absolute values of the zero-mean intensities instead.

It should be noted that the moment invariant based analysis is not restricted to affine rather than projective deformations only for the sake of simplicity. It can be proven that there simply are no projective moment invariants.

Classification of affine/photometric moment invariants

In order to calculate the combined affine/photometric color invariants, the effects of the affine and photometric changes on the images are to be determined first. As already mentioned before, only moments up to the first order and the second degree are considered, and the photometric changes only involve intensity scaling of the different color bands. Hence, the resulting invariants are functions of the color moments M_{00}^{abc}, M_{10}^{abc} and M_{01}^{abc} with $(a,b,c) \in \{(0,0,0), (1,0,0), (0,1,0), (0,0,1), (2,0,0), (0,2,0), (0,0,2), (1,1,0), (1,0,1), (0,1,1$ Under affine image transformations and intensity scaling, these moments transform as

$$\begin{pmatrix} \left(M_{10}^{abc}\right)' \\ \left(M_{01}^{abc}\right)' \\ \left(M_{00}^{abc}\right)' \end{pmatrix} = s_R^a\, s_G^b\, s_B^c\, |A| \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} M_{10}^{abc} \\ M_{01}^{abc} \\ M_{00}^{abc} \end{pmatrix} \qquad \text{for all } a,b,c.$$

As can be seen from the way they transform, the 0th-order moments M_{00}^{abc} can be considered in isolation; and a similar remark holds for the different color bands and for pairs of color bands. Moreover, different powers of the intensity of a single color band can also be taken in isolation. Hence, the invariants can be classified according to 3 degrees of freedom: the order, the degree and the color bands of the moments involved. In order to allow maximal flexibility for the user in the choice of the color bands but also to ensure the highest possible robustness of the invariants in the classification, we have used the following criteria to construct a basis for the moment invariants: (1) Include as much as possible invariants involving fewer color bands; and (2) include as much as possible low-order invariants. The result is the following classification, obtained through Lie group theory (the actual derivation of these invariants is out of the scope of this introductory text).

1-band invariants All affine/photometric invariants involving generalized color moments up to the first order and the second degree in only one color band are functions of the following 2 basis invariants:

$$B_{02} = \frac{M_{00}^2 M_{00}^0}{(M_{00}^1)^2}\,, \tag{15.5}$$

$$B_{12} = \frac{(M^1_{10}M^2_{01}M^0_{00} + M^2_{10}M^0_{01}M^1_{00} + M^0_{10}M^1_{01}M^2_{00} - M^0_{10}M^2_{01}M^1_{00} - M^1_{10}M^0_{01}M^2_{00} - M^2_{10}M^1_{01}M^0_{00})}{M^2_{00}M^1_{00}M^0_{00}} \quad (15.6)$$

where M^i_{pq} stands for either M^{i00}_{pq}, M^{0i0}_{pq} or M^{00i}_{pq}.

2-band invariants All affine/photometric invariants involving generalized color moments up to the first order and the second degree in 2 of the 3 color band are functions of the following 10 basis invariants:

0th order, 2nd degree:

$$B^1_{02} \ , \quad B^2_{02} \ , \quad \text{and} \quad C_{02} = \frac{M^{11}_{00}M^{00}_{00}}{M^{10}_{00}M^{01}_{00}} \ ; \quad (15.7)$$

where B^i_{02} stands for the invariant B_{02} in (15.5) evaluated in the ith (of the 2) color band(s). Note that one can always build other invariants by combining those from the basis. As an example – that will be used in chapter 16 – one can combine the above invariants as

$$\tilde{C}_{02} = \frac{B^1_{02}\,B^2_{02}}{(C_{02})^2} = \frac{M^{20}_{00}\,M^{02}_{00}}{(M^{11}_{00})^2}$$

which yields a nicely symmetrical expression. Of course these invariants depend on those already available and add no new, independent information.

1th order, 1nd degree:

$$C_{11} = \frac{M^{10}_{10}M^{01}_{01}M^{00}_{00} + M^{01}_{10}M^{00}_{01}M^{10}_{00} + M^{00}_{10}M^{10}_{01}M^{01}_{00} - M^{00}_{10}M^{01}_{01}M^{10}_{00} - M^{10}_{10}M^{00}_{01}M^{01}_{00} - M^{01}_{10}M^{10}_{01}M^{00}_{00}}{M^{10}_{00}M^{01}_{00}M^{00}_{00}} \quad (15.8)$$

where M^{ij}_{pq} stands for either M^{ij0}_{pq}, M^{i0j}_{pq} or M^{0ij}_{pq}, depending on which of the 2 color bands are used.

1th order, 2nd degree:

B^1_{12} , B^2_{12} ,

$$C^1_{12} = \frac{M^{10}_{10}M^{11}_{01}M^{00}_{00} + M^{11}_{10}M^{00}_{01}M^{10}_{00} + M^{00}_{10}M^{10}_{01}M^{11}_{00} - M^{00}_{10}M^{11}_{01}M^{10}_{00} - M^{10}_{10}M^{00}_{01}M^{11}_{00} - M^{11}_{10}M^{10}_{01}M^{00}_{00}}{M^{11}_{00}M^{10}_{00}M^{00}_{00}} \quad (15.9)$$

$$C^2_{12} = \frac{M^{01}_{10}M^{11}_{01}M^{00}_{00} + M^{11}_{10}M^{00}_{01}M^{01}_{00} + M^{00}_{10}M^{01}_{01}M^{11}_{00} - M^{00}_{10}M^{11}_{01}M^{01}_{00} - M^{01}_{10}M^{00}_{01}M^{11}_{00} - M^{11}_{10}M^{01}_{01}M^{00}_{00}}{M^{11}_{00}M^{01}_{00}M^{00}_{00}} \quad (15.10)$$

$$C^3_{12} = \frac{M^{10}_{10}M^{02}_{01}M^{00}_{00} + M^{02}_{10}M^{00}_{01}M^{10}_{00} + M^{00}_{10}M^{10}_{01}M^{02}_{00} - M^{00}_{10}M^{02}_{01}M^{10}_{00} - M^{10}_{10}M^{00}_{01}M^{02}_{00} - M^{02}_{10}M^{10}_{01}M^{00}_{00}}{M^{10}_{00}M^{02}_{00}M^{00}_{00}} \quad (15.11)$$

$$C^4_{12} = \frac{M^{20}_{10}M^{01}_{01}M^{00}_{00} + M^{01}_{10}M^{00}_{01}M^{20}_{00} + M^{00}_{10}M^{20}_{01}M^{01}_{00} - M^{00}_{10}M^{01}_{01}M^{20}_{00} - M^{20}_{10}M^{00}_{01}M^{01}_{00} - M^{01}_{10}M^{20}_{01}M^{00}_{00}}{M^{20}_{00}M^{01}_{00}M^{00}_{00}} \quad (15.12)$$

where B^i_{pq} stands for the 1-band invariant B_{pq} defined above but evaluated in the ith (of the 2) color band(s) and where M^{ij}_{pq} stands for either $M^{ij0}_{pq}, M^{i0j}_{pq}$, or M^{0ij}_{pq}, depending on which of the 2 color bands are used.

3-band invariants All geometric/photometric invariants involving generalized color moments up to the first order and the second degree of all 3 color bands are functions of the following 21 basis invariants:

0th order, 2nd degree: $B_{02}^R, B_{02}^G, B_{02}^B, C_{02}^{RG}, C_{02}^{RB}, C_{02}^{GB}$
1st order, 1st degree: $C_{11}^{RG}, C_{11}^{RB}, C_{11}^{GB}$
1st order, 2nd degree: $B_{12}^R, B_{12}^G, B_{12}^B, C_{12}^{1(RG)}, C_{12}^{1(RB)}, C_{12}^{1(GB)}$
$C_{12}^{2(RG)}, C_{12}^{2(RB)}, C_{12}^{2(GB)}, C_{12}^{3(RG)}, C_{12}^{3(GB)}, C_{12}^{4(RB)}$

where B_{ij}^K and C_{pq}^{KL} are the 2-band invariants defined above, but applied to the colour bands K and L. AS a consequence, *every invariant involving all 3 colour bands can always be written as a function of invariants which involve only 2 of the 3 bands.*

15.3.3 Invariants for three-dimensional shapes

There is no guarantee that invariants exist. Projectively invariant moments e.g. cannot be found. The same is true for most non-planar structures when viewed with a single camera. However, invariants for such structures (e.g. space curves or point clouds) can be calculated from image pairs. The problem gets even worse for curved surfaces, where it is difficult to identify certain points as references. It is not clear as yet what can be done from an invariance point of view for such cases. In summary, features that can be used irrespective of viewpoint can sometimes no longer be found. Usually then, one resorts to extracting features that can be used from subsets of viewpoints, as will be explained in chapter ??.

15.4 II - Pattern recognition, Part 1

As a second, necessary requirement for a feature to be useful, *discriminant power* was highlighted besides invariance. Features, although in principle maybe even forming a complete description, will always be corrupted by noise. Using e.g. a complete signature like affine curvature vs. affine arclength and correlating it with several model descriptions should yield only a perfect correlation with the correct model. Of course, in reality this won't happen. We can then build a vector containing the correlation coefficients with all model descriptions. Similarly, using area and perimeter as input to the final decision stage (e.g. in a simple application with perpendicular overhead views), the actual measurements will vary, under the influence of discretisation noise and other advert influences. Indeed, in taking the decision whether a certain pixel does belong to the object or the background, pixels near the boundaries can be judged to fall on either side of the threshold depending on small fluctuations. Anyway, using such features we finally end up with a list of numbers, the so-called *feature vector.*

The feature vectors that will finally be fed into the recognition procedure (e.g. the correlation coefficient vector or the 2-D vector (area,perimeter)) will therefore occupy a certain subspace of the feature space. The feature space consists of all allowable feature

vectors. Without variability, the set of feature vectors for a certain *class*, i.e. decision [5], would be a single point or a set of points if the class itself comprises distinct subclasses. The subspaces filled with feature vectors of a certain class will normally be connected and compact. They are referred to as *clusters*. The fact of having clusters rather than single points doesn't have to be a problem though. As long as the clusters don't overlap, perfect classification (recognition) is possible. One will not always be that fortunate, however. Returning to the problem of crop classification in chapter ??, the different crop classes will form clusters that do overlap. The actual shapes of the clusters usually have to be assessed from practical data rather than theoretical predictions. Such assessment can best be done visually, by looking at graphical representations of the data. An important problem might be the dimension of the feature space, which usually will be higher than two or three. In that case, visualization of the data is not straightforward. Being able to fit simple parametric models to the data can also be of great help, since efficient computer programming and optimization through mathematical analysis is possible in such cases.

In this section, we will deal with two aspects. The first are measures to quantify the separability of classes, the second ways to lower the dimension of the feature space without (appreciable) loss of separability. But before that, the multivariate, normal distribution is explained, since it is of such paramount practical importance.

15.4.1 The multivariate, normal distribution

A dividing line that goes right through the theory of pattern recognition is the distinction between *parametric* vs *non-parametric* methods. The former assume a parametric model for the probability distributions (clusters), whereas the latter don't impose restrictions on their nature. The parametric probability distribution model that is used most is the *normal distribution.*

For the univariate case (i.e. there is only one feature), the normal distribution takes the well-known form

$$p(x|\omega_i) = \frac{1}{(2\pi)^{1/2}\sigma_i}\exp[-\frac{1}{2}\frac{(x-m_i)^2}{\sigma_i^2}]$$

where ω_i is the class having the normal distribution and m_i is its mean and σ_i^2 its variance. Similarly, for the multi-variate case with n features x_i, one has the conditional probability

$$p(X|\omega_i) = \frac{1}{(2\pi)^{n/2}|C_i|^{1/2}}\exp[-\frac{1}{2}(X-M_i)^T C_i^{-1}(X-M_i)]$$

with the feature vector X, the mean vector M_i, and the covariance matrix C_i given by

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad M_i = \begin{pmatrix} m_{i1} \\ m_{i2} \\ \vdots \\ m_{in} \end{pmatrix} \qquad C_i = \begin{pmatrix} \sigma_{i11} & \sigma_{i12} & \dots & \sigma_{i1n} \\ \sigma_{i21} & \sigma_{i22} & \dots & \sigma_{i2n} \\ \vdots & \vdots & & \vdots \\ \sigma_{in1} & \sigma_{in2} & \dots & \sigma_{inn} \end{pmatrix}.$$

[5] Such as object type or accept/reject

In the case of two features, loci of equiprobability would be ellipses. In the case of three features they would be ellipsoids. The covariance matrix captures the information about the orientation and eccentricity of these structures.

15.5 II - Pattern recognition, Part 2

15.5.1 PCA for classification and inspection

In chapter 6 we have introduced Principal Component Analysis (PCA) for image compression and for the compact parameterisation of the relevant information in constrained image sets. Here we discuss the PCa as a tool for classification and inspection.

Let us first consider the use of PCA for classification. Think of the following artificial problem. We measure the height and the weight of people and want to clasxify them into youngsters and adults. Fig. ?? shows some virtual samples for both groups. **Hier figuur met uitleg classificatie en PCA, deel (a) en (b), met (a) voor de oorspronkelijke assen en (b) na PCA** As can be seen from part (a) of the figure the two classes cannot be separated by considering the height or weight in isolation. Part (b) shows the first and second principal component found with PCA. As can be seen, the first component – which corresponds with the direction with highest variance in the data – is the single feature that best separates the two classes. Of course, in order to calculate it we have to take the corresponding linear combination of height and weight.

The previous example is quite simple, as it is only two-dimensional. But if the dimensionality of the problem increases, it starts becoming quite difficult to find directions of high variance purely through visualisation of the data.

Now, assume we want to perform an inspection task, e.g. to check whether a person has a normal (height, weight) combinatiuon, or whether s/he is an outlier. Fig. ??(a) shows two such outliers. **hier figuur voor inspectie obv PCA; weer met deel (a) en (b)** the one in the upper right could be easily distinguished by the fact that this person is too heavy compared to the normal population. The second outlier poses a more difficult case, however, as now both the height and the weight fall within the range of the normal population when considered separately. Part (b) shows the principal component directions. Both outliers can be detected by projecting the data onto the axis of the second component, where they have both values outside the normal range. Again, the selection of this direction of minimal variance could be done interactively, but the situation would be much more difficult to assess if the data were high-dimensional.

As can be seen from these two over-simplified examples, classification and inspection tend to use PCA in a complementary way. For classification mainly the components with the highest variance are relevant. They help in separating the data. Inspection needs to find samples that fall outside the normal distribution. Here, components with small variances tend to be the most useful, as outliers will be more easily detected for components where the normal range is small than for components where the normal samples already cover a large part of occurring values.



Figure 15.6: *SPOT spectral images, with from left to right: near-infrared, red, and green.*

Next, we give two examples. The first shows the use of PCA for a classification problem in remote sensing. The second illustrates the use of PCA for textile inspection.

PCA application 1: classification of satellite data

As an example of the use of PCA for classification, the following application from remote sensing has been selected. A SPOT satellite produces three spectral images, corresponding to three different spectral bands: near-infrared (N), red (R), and green (G). Based on these multi-spectral data, pixels in the images have to be classified with respect to the land use class they belong to. One pixel corresponds to $20m \times 20m$. In this application five crop types had to be considered.

Fig. 15.6 shows the three satellite images N,R, and G from left to right, resp. The five classes (ground truth, i.e. the real situation) are shown in fig. 15.7. The black pixels belong to other classes and are not considered in the discussion.

As can be seen from fig. 15.6, the difference between R and G is hardly noticeable. It seems R and G are strongly correlated. On the other hand, the outlook of the N image is substantially different. The impression is that N contains uncorrelated data. These conclusions are corroborated by PCA. The covariance matrix C for this particular problem was found to be

$$C = \begin{pmatrix} 127.2447 & 13.3062 & -5.9095 \\ 13.3062 & 34.2264 & 39.2092 \\ -5.9095 & 39.2092 & 54.8805 \end{pmatrix}.$$

The rows and columns correspond to N,R, and G in that order. This symmetric matrix tallies with our initial observations. The first row (or column) indicates from left to right: the variance in the N image (covariance of N with itself), the covariance of N and R, and the covariance of N and G. The *correlation coefficient* for two stochastic variables i and j measures the amount of correlation between them. It can take on values between -1 and 1, with the extremes indicating perfect correlation and 0 no correlation whatsoever [6]. The

[6]It is important to note that the correlation coefficient captures linear relations between the variables.

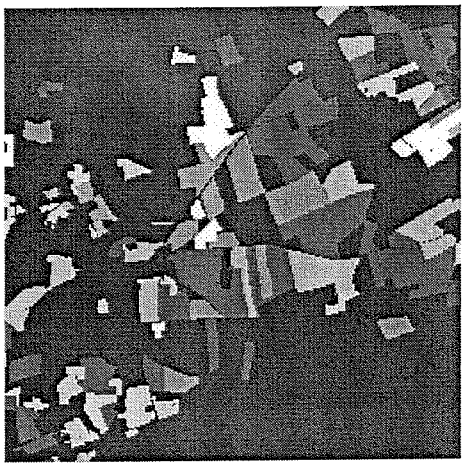

Figure 15.7: *Ground truth data: 5 crop classes are to be distinguished, and to be compared to the reference data of the figure. Black pixels belong to other classes and should not be considered.*

correlation coefficient ρ for two variables i and j is defined as

$$\rho_{ij} = \frac{c_{ij}}{\sigma_i \sigma_j}$$

with $c_{ij} = C(i,j)$, i.e. the covariance, and $\sigma_i^2 = C(i,i)$, i.e. the variance. We e.g. find that $\rho_{12} = 0.2016$ and $\rho_{23} = 0.9047$, indicating a rather low correlation between the N and R band and a high correlation between the R and G bands. The principal components (eigenvectors) are

$$\begin{pmatrix} 0.9907 \\ 0.1360 \\ -0.0070 \end{pmatrix} \quad \begin{pmatrix} -0.0765 \\ 0.5980 \\ 0.7978 \end{pmatrix} \quad \begin{pmatrix} -0.1127 \\ 0.7899 \\ -0.6028 \end{pmatrix}$$

with eigenvalues 129.1135, 84.8359, and 2.4022, resp. Notice that the first PC almost coincides with the N input. The eigenvector coordinates yield the weights to be used in combining the initial data. The PC images are given in fig. 15.8. The decreasing variance of these images is evident. The third PC hardly contains information on the agricultural parcels that are of interest. Dropping this information can therefore be expected to have only a minor influence on the crop classification results. In this particular example, using all three input images yielded 76.3% correct classification, whereas restricting the analysis to the two first principal components leads to a comparable result of 73.5%. Naturally, in this case using three images is not unwieldy. Yet, if more spectral bands or input data acquired at different times have to be combined, then the dimensional reduction offered by PCA will become crucial.

If one finds 0 correlation one must not conclude that the variables are independent, merely that they are not linearly dependent.

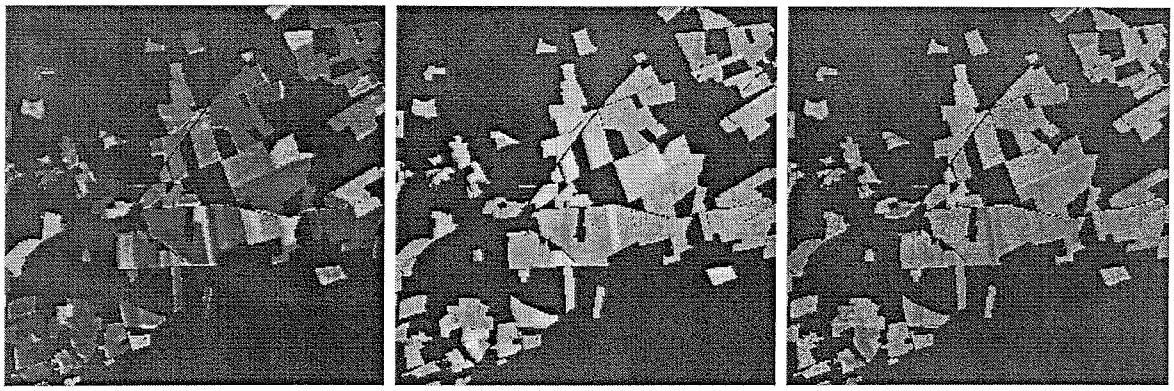

Figure 15.8: *Three principal components for the input of fig. 15.6, in order of decreasing variance.*

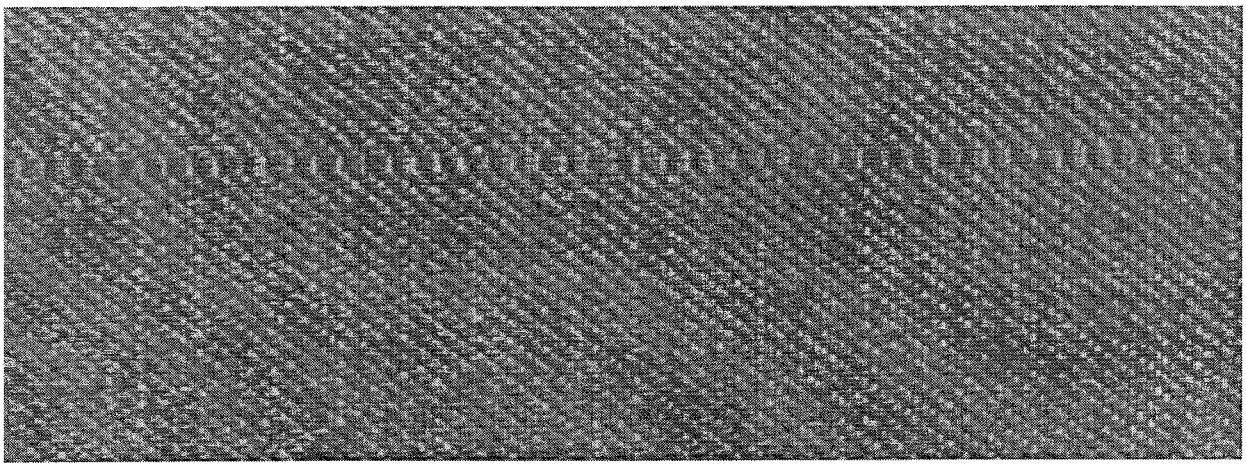

Figure 15.9: *Example of a flaw in a textile sample.*

PCA application 2: visual inspection of fabrics

We discuss textile inspection as an application of PCA for inspection. We start from the eigenfilters introduced in chapter 13. As the reader will remember, these are filters that have been tuned towards the specific texture at hand, through PCA. Although the filters corresponding to the highest variance (in their output) may carry most information about the structure of the correct textile texture, this does not mean that they are best suited for all tasks, including inspection. As already discussed earlier, one can expect that the filters corresponding to lower variance may actually be the better filters for inspection. We now have a more detailed look.

Consider fig. 15.9. It shows a sample of the same textile as that of fig. 13.18. This textile sample contains a flaw (it runs horizontally over the whole width). We investigate the use of the eigenfliters that have been extracted from the flawless texture for the detection of such flaws (see chapter 13). In particular, the goal is to inspect this type of textile on-loom, by convolving newly woven parts by the eigenfilters and comparing observed against normal

outputs. Fig. 15.10 shows the results of applying the eigenfilters on the example sample of fig. 15.9. The flaw shows up in the output of several of the filters. It can also already be seen that both filters with large and small eigenvalues are among the set that best pick up the flaw. This may seem to run counter to the previous claim that components with the smallest eigenvalues are generally the more interesting ones for inspection. There are many different flaw types, and they can take on different appearances. When extracting the filters, only flawless texture is used. The texture corresponding to the flaws has not been considered at all. The response of the filters to these textures can in principle be anything. There is no guarantee that the filters with small eigenvalues will be the ones which bring out the differences best. They only can be expected to have – on the average – the best chance to do so. We now carry on with our inspection example.

The next step is to calculate the features' local 'energy'. This step consists of calculating the sum of absolute or squared values over neighbourhoods around each pixel. **HIER JUISTE PARAMETERS VOOR HET TEXTUURVOORBEELD** The results are shown in fig. 15.11. The goal of this step is to generate images with more homogeneous intensities in the hope of getting at least one or two where the resulting intensity for the flawed region is substantially different from the resulting intensity for good quality fabric. In the example this is e.g. the case for the energy of the first, seventh and eighth eigenfilter's output.

In a last step, the energy values are combined into a single distance to the typical energies. The Mahalanobis distance was used. This distance sums the differences with the average values for good fabric for the different eigenfilter energies, each time weighted by the inverse of the energy variance for flawless texture. The Mahalanobis distance for our example is shown in fig. 15.12. Based on this distance, the region in fig. 15.13 is classified as a flaw.

15.5.2 Statistical separability of classes

It is clear that overlapping probability distributions are bad news: patterns represented by feature vectors in the overlap zone might be attributed to more than one class. In this section measures for the amount of overlap are discussed, usually referred to as measures of the *statistical separability.*

Intraset vs. interset measures

Suppose a cluster representing a class ω_k consists of N_k samples X_k^i, $i = 1...N_k$. The X_k^i are assumed to be column (feature) vectors. Then the *intraset distance* measure for that class is

$$D_k = \frac{1}{N_k(N_k-1)} \sum_{i=1}^{N_k} \sum_{j=1, j\neq i}^{N_k} d^2(X_k^i, X_k^j)$$

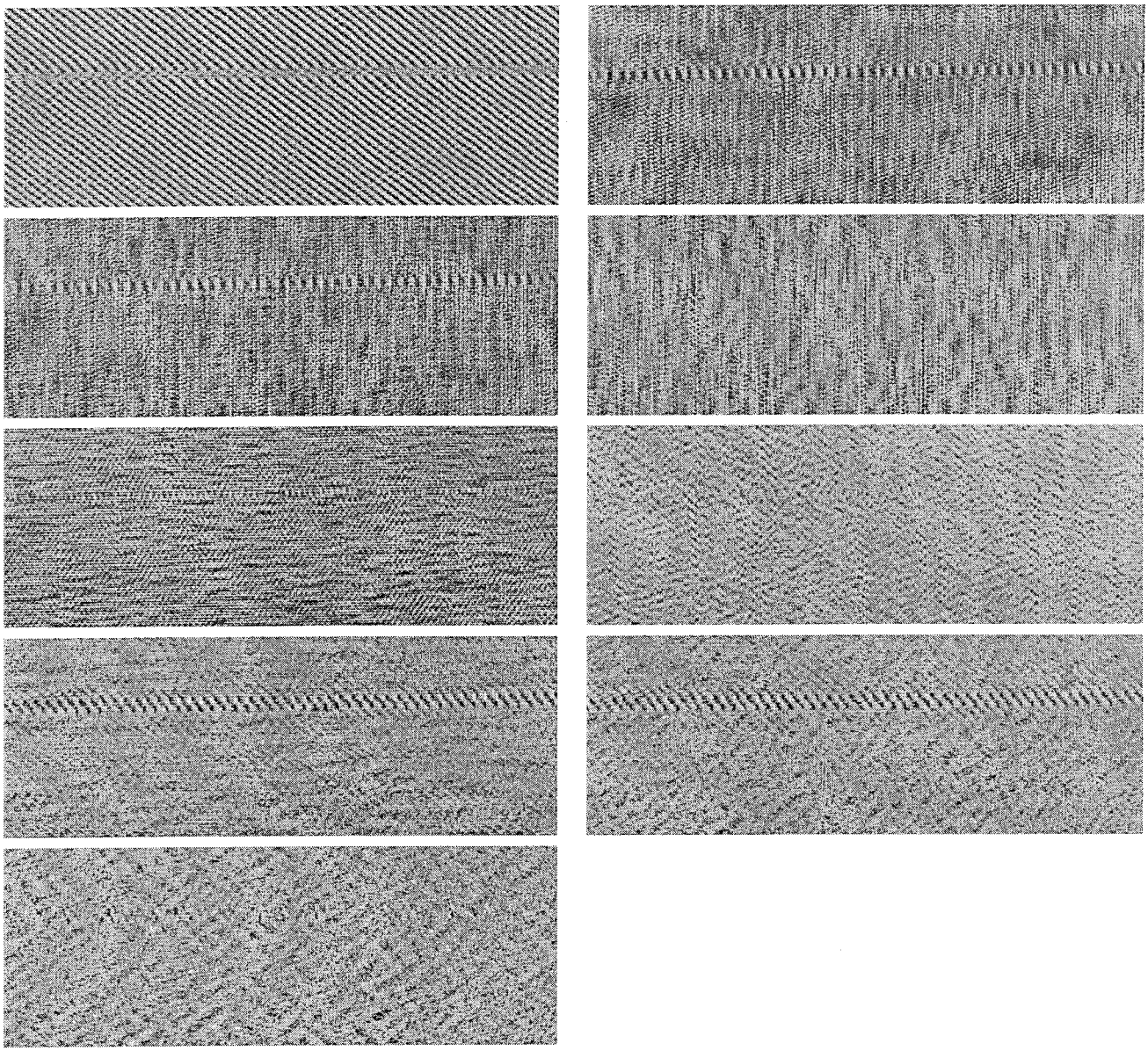

Figure 15.10: *Results of the 9 eigenfilters on the image of fig. 15.9 in lexicographic order.*

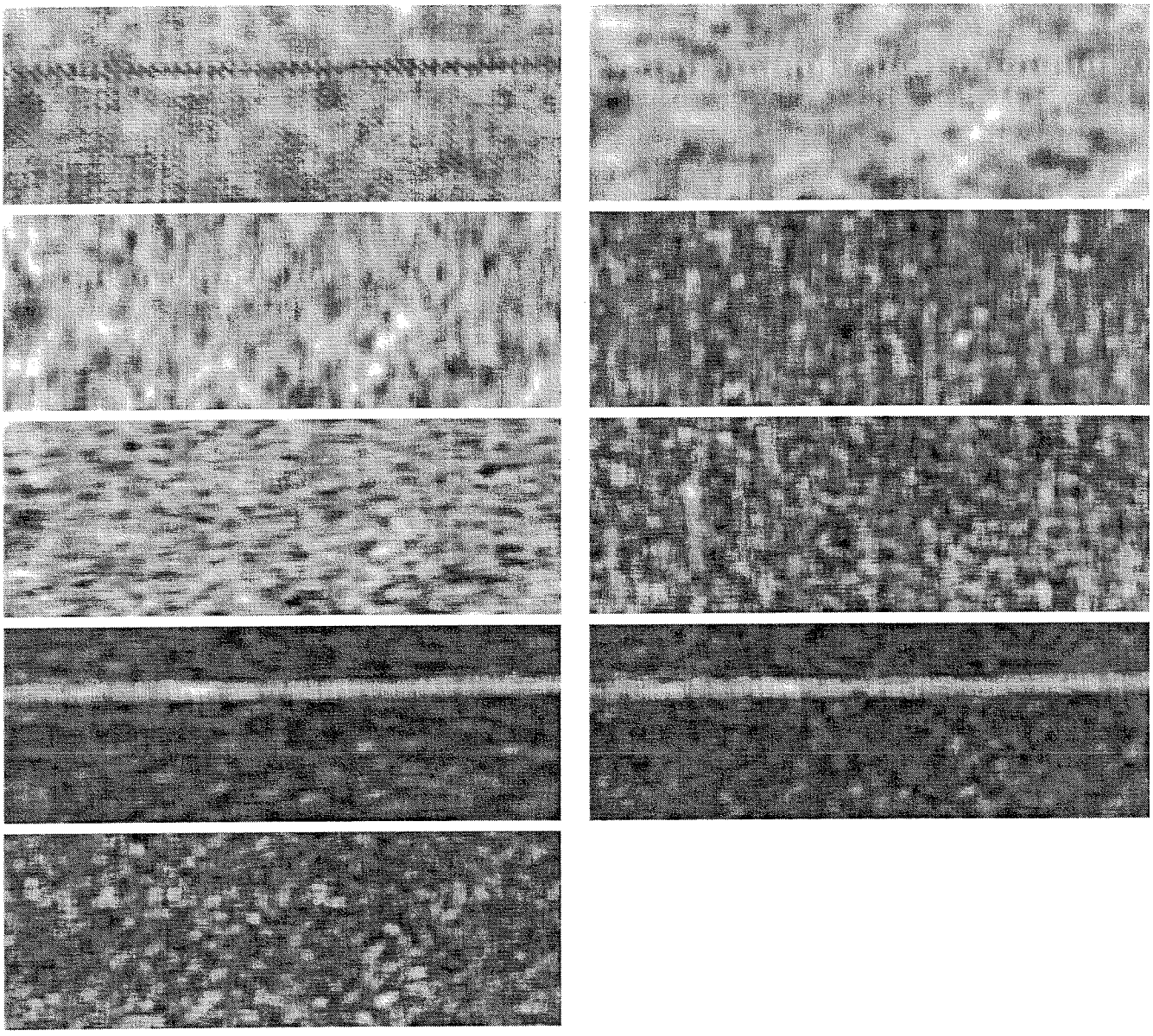

Figure 15.11: *Energies for the outputs of the 9 eigenfilters in fig. 15.10 in lexicographic order.*

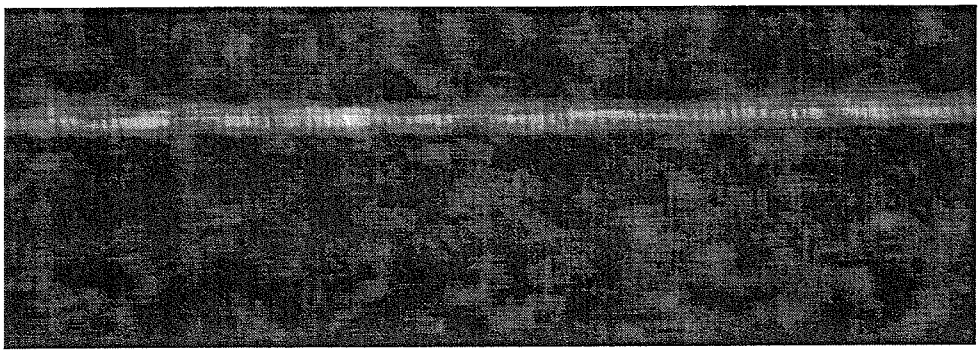

Figure 15.12: *Mahalanobis distance for the image in fig. 15.9.*

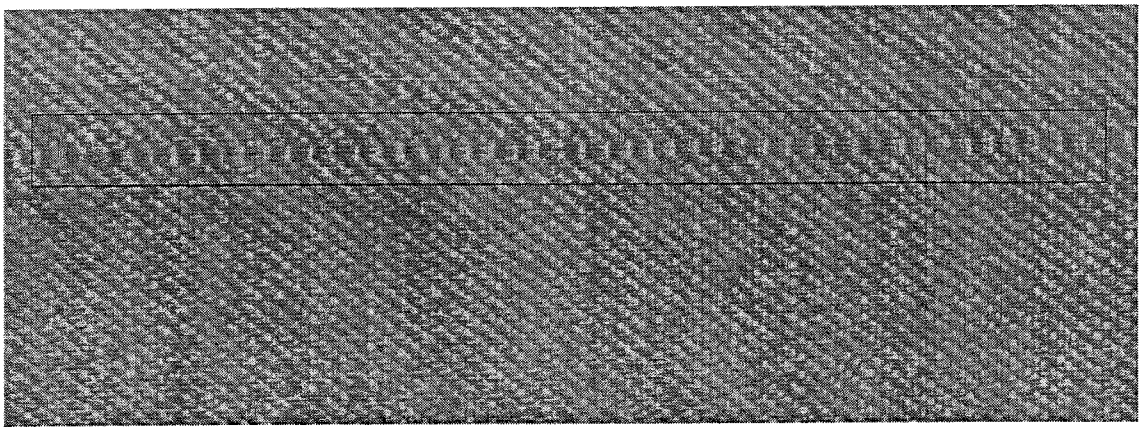

Figure 15.13: *Flaw detected on the basis of the Mahalanobis distance.*

with d some distance measure, e.g. Euclidean distance.
The *interset distance* measure between two classes ω_k and ω_l is defined similarly:

$$D_{kl} = \frac{1}{N_k N_l} \sum_{i=1}^{N_k} \sum_{j=1}^{N_l} d^2(X_k^i, X_l^j).$$

If there are K classes, one can then calculate

$$Q = \frac{\sum_{k=1}^{K} D_k}{\sum_{k=1}^{K} \sum_{l=1, l \neq k}^{K} D_{kl}}$$

as an overall measure of separability.

Since Q requires substantial computational effort, the pairwise distance calculations may be better replaced by comparisons with respect to the mean feature vector M_k for each of the classes or the overall mean M. Thus,

$$\begin{aligned} M_k &= \tfrac{1}{N_k} \textstyle\sum_{i=1}^{N_k} X_k^i \\ M &= \tfrac{1}{N} \textstyle\sum_{k=1}^{K} \sum_{i=1}^{N_k} X_k^i \end{aligned}$$

with

$$N = \sum_{k=1}^{K} N_k.$$

In particular, covariance is a useful measure. The *intraset covariance matrix* is given by

$$W = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N_k} (X_k^i - M_k)(X_k^i - M_k)^T$$

with ^{T} the matrix transpose as usual and the name W standing for w ithin class covariance. Similarly, the *interset covariance matrix* for classes k and l is defined as

$$B = \sum_{k=1}^{K} \frac{N_k}{N} (M_k - M)(M_k - M)^T$$

(the name B stands for b etween classes). Also considering the *total covariance matrix*

$$T = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N_k} (X_k^i - M)(X_k^i - M)^T$$

one can prove that

$$T = B + W.$$

The total variance of the (orthogonal) projection of the data on a one-dimensional axis generated by a (column) vector u is given by

$$u^T T u = u^T B u + u^T W u.$$

A good directional vector u (i.e. an appropriate linear combination of the original features) should at the same time yield a high interset covariance and a small intraset covariance, i.e. we look for that u which realizes

$$\max \frac{u^T B u}{u^T W u}.$$

This criterion refines the PCA technique suggested in chapter ??.

Divergence

Let the probability of occurrence of a pattern X be $p(X|\omega_k)$ if it belongs to class ω_k and $p(X|\omega_l)$ if it belongs to class ω_l. Moreover, suppose that the overall (a priori) chance of running into a pattern of class ω_k is $p(\omega_k)$ and similarly $p(\omega_l)$ for ω_l. How much does the occurrence of X tell us about its class membership? If the chances for X being found and belonging to each of the classes is the same, i.e. if $p(X|\omega_k)p(\omega_k) = p(X|\omega_l)p(\omega_l)$ then clearly there is no information in knowing that X was found as far as the class membership is concerned. This is the worst situation. On the other hand, if $p(X|\omega_k)p(\omega_k) = 1$ and $p(X|\omega_l)p(\omega_l) = 0$, then there is not a scintilla of doubt about class membership left. Information is complete.

From information theory there is the celebrated result by Shannon that for the quantification of information logarithmic functions ought to be used. Suppose there are a number of possible outcomes (classes), each with their own probability. Further suppose the observer can ask binary questions, i.e. membership to a certain set of outcomes or not. Instead of singling out a single outcome at a time, a better strategy would be to group outcomes in a hierarchical manner, first taking them together in two equiprobable groups (0.5 probability each), which are then subdivided into groups with probability 0.25, etc. In that case gradual restriction of options is possible and the number of questions to ask is $\log_2 p_{min}$ with p_{min} the smallest probability among the original outcomes. For instance, if four outcomes are possible, with probabilities $p_1 = \frac{1}{2}, p_2 = \frac{1}{4}, p_3 = \frac{1}{8}, p_4 = \frac{1}{8}$, then the series of questions 'class 1 or otherwise?', 'class 2 or otherwise?', and 'class 3 or otherwise' is sufficient. The maximum number of questions to remove all uncertainty about class membership is three. That many questions will have to be asked to be sure of outcome identity for the third and fourth class. On the other hand, a single question suffices to find out about class 1. Two questions have to be asked for class 2. This fits with the values of the base-2 logarithm: $\text{abs}(\log_2 p_1) = 1$, $\text{abs}(\log_2 p_2) = 2$, and $\text{abs}(\log_2 p_3) = \text{abs}(\log_2 p_4) = 3$. Of course, the example was artificial in that all probabilities were powers of 2, enabling finer subdivisions into equiprobable subclasses. Nevertheless, it will be clear that logarithmic functions have the desired qualitative behaviour to capture the essence of uncertainty. The base chosen

is quite immaterial since changing it amounts to multiplying by a factor, keeping relative values intact.

What is relevant in classification is the relative uncertainty. Even if the probability of having found a certain sample feature vector for one class is very small, there is not ground for confusion if chances for other classes having supplied the sample are even more minute. Consider the difference in uncertainty given by

$$\log p(X|\omega_k)p(\omega_k) - \log p(X|\omega_l)p(\omega_l) = \log \frac{p(X|\omega_k)p(\omega_k)}{p(X|\omega_l)p(\omega_l)}.$$

This function is zero for equal probabilities, and its magnitude increases when one probability gets larger with respect to the other. This function has a large magnitude if the relative uncertainty is small and v.v. It is clear that uncertainty – or missing information – has to be kept to a minimum if classification results are to be acceptable. The average discriminating information for class ω_k with respect to class ω_l is then found by integrating over all possible values of the feature vector X:

$$I_{kl} = \int p(X|\omega_k)p(\omega_k) \ \log \frac{p(X|\omega_k)p(\omega_k)}{p(X|\omega_l)p(\omega_l)} dX.$$

As it stands the "missing information" can be positive and negative, what we don't want of course. Also is the definition of relative uncertainty asymetrical: interchanging k and l yields different values. Both problems are solved simultaneously by defining the *divergence* as

$$\begin{aligned} D_{kl} &= I_{kl} + I_{lk} \\ &= \int (p(X|\omega_k)p(\omega_k) - p(X|\omega_l)p(\omega_l)) \ \log \frac{p(X|\omega_k)p(\omega_k)}{p(X|\omega_l)p(\omega_l)} dX. \end{aligned}$$

The integrand is now guaranteed to be positive and swapping k and l doesn't affect its value. Good features, if they exist, lead to high D_{kl} values.

For normal, N-variate distributions

$$\begin{aligned} p(X|\omega_k) &= \frac{1}{(2\pi)^{n/2}|C_k|^{1/2}} \exp[-\tfrac{1}{2}(X - M_k)^T C_k^{-1}(X - M_k)] \\ p(X|\omega_l) &= \frac{1}{(2\pi)^{n/2}|C_l|^{1/2}} \exp[-\tfrac{1}{2}(X - M_l)^T C_l^{-1}(X - M_l)] \end{aligned}$$

where we also assume $p(\omega_k) = p(\omega_l)$ for the ease of notation, we have

$$\log \frac{p(X|\omega_k)}{p(X|\omega_l)} = \frac{1}{2}\log \frac{|C_k|}{|C_l|} - \frac{1}{2}\mathrm{tr}[C_k^{-1}(X - M_k)(X - M_k)^T] + \frac{1}{2}\mathrm{tr}[C_l^{-1}(X - M_l)(X - M_l)^T]$$

using

$$\mathrm{tr}[C^{-1}XX^T] = X^T C^{-1} X$$

and tr standing for the matrix trace. Rewriting

$$\begin{aligned} C_l^{-1}(X - M_l)(X - M_l)^T = C_l^{-1}(X - M_k)(X - M_k)^T + \\ C_l^{-1}(M_k X^T + X M_k^T - M_l X^T - X M_l^T - M_k M_k^T + M_l M_l^T) \end{aligned}$$

and using equations like

$$\int p(X|\omega_k)\ \mathrm{tr}[C^{-1}(X-M_k)(X-M_k)^T]dX = \mathrm{tr}[C^{-1}C_k]$$
$$\int p(X|\omega_k)\ \mathrm{tr}[C^{-1}MX^T]dX = \mathrm{tr}[C^{-1}MM_k^T]$$

I_{kl} is found to be

$$\log\frac{p(X|\omega_k)}{p(X|\omega_l)} = \frac{1}{2}\log\frac{|C_k|}{|C_l|} - \frac{1}{2}\mathrm{tr}[(C_l^{-1}-C_k^{-1})C_k] + \frac{1}{2}\mathrm{tr}[C_l^{-1}(M_k-M_l)(M_k-M_l)^T]$$

and hence

$$D_{kl} = \frac{1}{2}\mathrm{tr}[(C_k-C_l)(C_l^{-1}-C_k^{-1})] + \frac{1}{2}\mathrm{tr}[(C_k^{-1}+C_l^{-1})(M_k-M_l)(M_k-M_l)^T].$$

As a further simplification, $C_k = C_l = C$ yields

$$(M_k-M_l)^TC^{-1}(M_k-M_l).$$

This distance measure is usually referred at as the *Mahalanobis distance.*

15.5.3 Lowering feature space dimension

Using many features, i.e. feature spaces of high dimension, makes the choice of a good classifier difficult and its use time consuming. Therefore, one often has to strike a balance between classification performance and cost. The above criteria can be used to select the most promising features as linear combinations of the original ones (and forget about the rest).

Interset vs. Intraset covariance

From the previous section the following feature selection criterion was obtained:

$$\max\frac{u^TBu}{u^TWu}$$

where u is the optimal direction of a line on which the samples are to be projected. It is clear that the above requirement is equivalent to

$$\min\frac{u^TWu}{u^TBu},$$

which in turn has the same solution as

$$\min\frac{u^TWu}{u^TBu} + 1 = \min\frac{u^TTu}{u^TBu} = \max\frac{u^TBu}{u^TTu}$$

Introducing the Lagrange multiplier λ this amounts to extremizing

$$u^TBu - \lambda(u^TTu).$$

This problem is comparable to the one encountered with PCA (chapter ??). It leads to the eigenvalue problem

$$T^{-1}Bu = \lambda u.$$

The eigenvector corresponding to the largest eigenvalue yields to the optimal direction. In fact the method boils down to applying PCA on the mean vectors using the T^{-1}-metric:

$$d(X,Y) = (X-Y)^T T^{-1}(X-Y).$$

We recognize the *Mahalanobis distance* [7].

The two-class case leads to the *Fisher discriminant method.* In that case we have

$$B = (M_1 - M_2)(M_1 - M_2)^T.$$

First note that Bu is always a vector in the direction of $(M_1 - M_2)$. Moreover, since the optimal direction u should satisfy

$$T^{-1}Bu = \lambda u,$$

we can conclude that the optimal u is found as

$$u = T^{-1}(M_1 - M_2).$$

Note that the scale factor to be used is immaterial, hence no care was taken of it. Only the direction is important. This is *Fisher's linear discriminant.* One can consider reducing the problem to a one-dimensional analysis by projecting all sample vectors on that direction. It will show the maximal interset vs. intraset variance.
In the case of normal distributions with equal covariance matrices C, T in the above expression can be replaced by C.

Divergence

A general analysis of divergence maximization is rather involved (e.g. drawing heavily on properties of generalized eigenvalue analysis). Let it suffice to mention that in the two-class case of equiprobable, normal distributions and equal covariance matrices the result is exactly the same: the Fisher linear discriminant is found again.

15.5.4 Support Vector Machines

Not available yet

[7] In the case of normal distributions with identical covariance matrices C, the intraclass covariance matrix W can be replaced by C, and T can be replaced by W as shown earlier, thereby completing the chain.