Chapter 3

2D Moment Invariants to Translation, Rotation, and Scaling

3.1 Introduction

In this chapter, we introduce 2D moment invariants with respect to the simplest spatial in-plane transformations – translation, rotation, and scaling (TRS). Invariance with respect to TRS is widely required in almost all practical applications, because the object should be correctly recognized regardless of its particular position and orientation in the scene and of the object to camera distance (see Figure 3.1). On the other hand, the TRS model is a sufficient approximation of the actual image deformation if the scene is flat and (almost) perpendicular to the optical axis. Due to these reasons, much attention has been paid to TRS invariants. While translation and scaling invariants can be mostly derived in an intuitive way, derivation of invariants to rotation is far more complicated.

Before we proceed to the design of the invariants, we start this chapter with a few basic definitions and with an introduction to moments. The notation introduced in the next section will be used throughout the book if not specified otherwise.

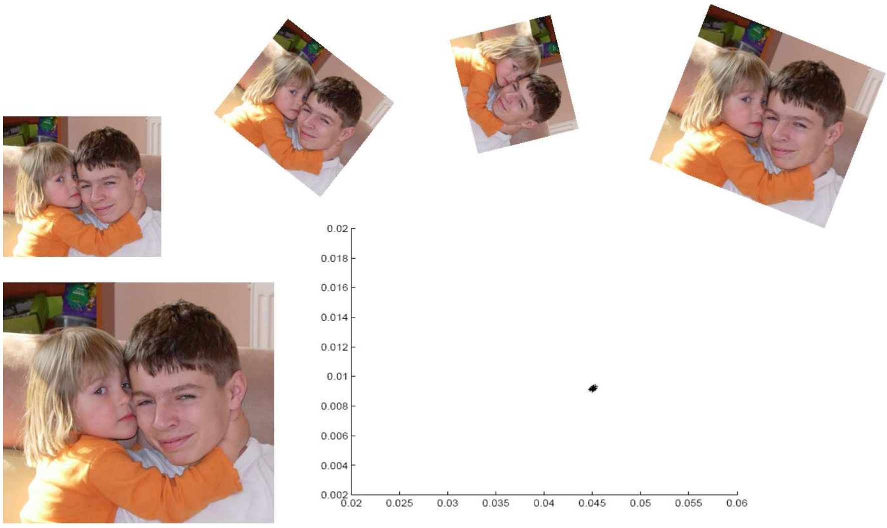


Figure 3.1: The desired behavior of TRS moment invariants – all instances of a rotated and scaled image have almost the same values of the invariants (depicted for two invariants.

3.1.1 Mathematical preliminaries

Spatial coordinates in the image domain are denoted as $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$, where d is the dimension of the space. In 2D and 3D domains, if there is no danger of misunderstanding, we sometimes use a simpler (and more common in the literature) notation $\mathbf{x} = (x, y)^T$ and $\mathbf{x} = (x, y, z)^T$, respectively. The supperscript $(\cdots)^T$ means a transposition, so our coordinates are arranged into a column vector.

Definition 3.1: By an *image function* (or *image*) we understand any piece-wise continuous real function $f(\mathbf{x})$ defined on a compact support $\Omega \subset \mathbb{R}^d$, which has a finite nonzero integral.

According to this definition, our "images" need not be non-negative. The piece-wise continuity and the compact support are assumed to make the operations, we are going to apply on the images, well defined, which enables a comfortable mathematical treatment without a tedious verifying of the existence of the operations in each individual case. From a purely mathematical point of view, these requirements may seem to be too restrictive because certain operations are well defined on broader classes of functions such as integrable or square integrable functions or infinitely supported functions of fast decay. However, these nuances make absolutely no difference from a practical point of view when working with digital images. The non-zero integral is required because its value will be frequently used as a normalization factor[1].

If the image is mathematically described by the image function, we sometimes speak about the *continuous representation.* Although the continuous representation may not reflect certain properties of digital images, such as sampling and quantization errors, we will adopt the continuous formalism because of its mathematical transparency and simplicity. If the discrete character of the image is substantial (such as, for instance, in chapter 8 where we explain numerical algorithms for moment computations), we will work with a *discrete representation* of the image in a form of a finite-extent 2D matrix $\mathbf{f} = (f_{ij})$ or 3D matrix $\mathbf{f} = (f_{ijk})$, which is supposed to be obtained from the continuous representation $f(\mathbf{x})$ by sampling and quantization.

The image function from Definition 3.1 represents *monochromatic* images (sometimes also called graylevel or scalar images). If f has only two possible values (which are usually encoded as 0 and 1), we speak about a *binary* image. Color images and vector-valued images are represented as a vector image function, each component of which satisfies Definition 3.1.

Convolution is an operation between two image functions[2], the result of which is another image function defined as

$$(f * g)(\mathbf{x}) = \int\limits_{\mathbb{R}^d} f(\mathbf{t}) g(\mathbf{x} - \mathbf{t}) \mathrm{d}\mathbf{t} \ . \tag{3.1}$$

Fourier transformation (FT) of image function f is defined as

$$\mathcal{F}(f)(\mathbf{u}) \equiv F(\mathbf{u}) = \int\limits_{\mathbb{R}^d} e^{-2\pi i \mathbf{u}\mathbf{x}} f(\mathbf{x}) \mathrm{d}\mathbf{x} \ , \tag{3.2}$$

where i is the imaginary unit, the components of $\mathbf{u}$ are spatial frequencies, and $\mathbf{u}\mathbf{x} = u_1 x_1 + \ldots + u_d x_d$ means scalar product[3]. Note that $\mathcal{F}(f)$

[1] If we released this assumption, the invariants still could be constructed provided that at least one moment is non-zero.

[2] Note that the convolution is not a point-wise operation between the function values, so the notation $f(x) * g(x)$, commonly used in many engineering textbooks, is misleading and mathematically incorrect.

[3] A more correct notation of the scalar product is $\mathbf{u}^T\mathbf{x}$; we drop the superscript ^{T} for simplicity.

always exists thanks to the integrability of f, and its support cannot be bounded[4]. Fourier transformation is invertible through the *inverse Fourier transformation* (IFT)

$$\mathcal{F}^{-1}(F)(\mathbf{x}) = \int\limits_{\mathbb{R}^d} e^{2\pi i \mathbf{u}\mathbf{x}} F(\mathbf{u}) \mathrm{d}\mathbf{u} = f(\mathbf{x}) \ . \tag{3.3}$$

We recall two important properties of FT, which we employ in this book. The first one, also known as the *Fourier shift theorem*, tells that the FT of a translated image function equals the FT of the original up to a phase shift

$$\mathcal{F}(f(\mathbf{x} - \mathbf{t}))(\mathbf{u}) = e^{-2\pi i \mathbf{u}\mathbf{t}} \mathcal{F}(f(\mathbf{x}))(\mathbf{u}) \ . \tag{3.4}$$

The second property, known as the *convolution theorem*, shows why FT is so useful in signal processing – it transfers the convolution into a pointwise multiplication in the frequency domain

$$\mathcal{F}(f * g)(\mathbf{u}) = F(\mathbf{u}) G(\mathbf{u}) \ . \tag{3.5}$$

3.1.2 Moments

Moments are scalar real or complex-valued features which have been used to characterize a given function. From the mathematical point of view, moments are "projections[5]" of function f onto a polynomial basis (similarly, Fourier transformation is a projection onto a basis of the harmonic functions).

Definition 3.2: Let $\{\pi_{\mathbf{p}}(\mathbf{x})\}$ be a d-variable polynomial basis of the space of image functions defined on Ω and let $\mathbf{p} = (p_1, \ldots, p_d)$ be a multi-index of non-negative integers which show the highest power of the respective variables in $\pi_{\mathbf{p}}(\mathbf{x})$. Then the *general moment* $M_{\mathbf{p}}^{(f)}$ of image f is defined as

$$M_{\mathbf{p}}^{(f)} = \int\limits_{\Omega} \pi_{\mathbf{p}}(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} \ . \tag{3.6}$$

[4] More precisely, $\mathcal{F}(f)$ cannot vanish on any open subset of $\mathbb{R}^d$.

[5] The moments in general are not the coordinates of f in the given basis in the algebraic sense. That is true for orthogonal bases only.

The number $|\mathbf{p}| = \sum_{k=1}^{d} p_k$ is called the *order* of the moment. We omit the superscript $^{(f)}$ whenever possible without confusion.

Depending on the polynomial basis $\{\pi_{\mathbf{p}}(\mathbf{x})\}$, we recognize various systems of moments. The most common choice is a standard power basis $\pi_{\mathbf{p}}(\mathbf{x}) = \mathbf{x}^{\mathbf{p}}$ which leads to *geometric moments*

$$m_{\mathbf{p}}^{(f)} = \int\limits_{\Omega} \mathbf{x}^{\mathbf{p}} f(\mathbf{x}) \mathrm{d}\mathbf{x} \ . \tag{3.7}$$

In the literature, one can find various extensions of Definition 3.2. Some authors allow non-polynomial bases (more precisely, they allow basis functions which are products of a polynomial and some other – usually harmonic – functions) and/or include various scalar factors and weighting functions in the integrand. Some other authors even totally replaced the polynomial basis by some other basis but still call such features moments – we can find *wavelet moments* [1] and *step-like moments* [2], where wavelets and step-wise functions are used in a combination with harmonic functions instead of the polynomials. These modifications broadened the notion of moments but have not brought any principle differences in moment usage.

3.1.3 Geometric moments in 2D

In case of 2D images, the choice of the standard power basis $\pi_{pq}(x, y) = x^p y^q$ yields *2D geometric moments*

$$m_{pq} = \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} x^p y^q f(x, y) \mathrm{d}x \mathrm{d}y. \tag{3.8}$$

Geometric moments have been widely used in statistics for description of the shape of a probability density function and in classic rigid-body mechanics to measure the mass distribution of a body. Geometric moments of low orders have an intuitive meaning – m_{00} is a "mass" of the image (on binary images, m_{00} is an area of the object), m_{10}/m_{00} and m_{01}/m_{00} define the *center of gravity* or *centroid* of the image. Second-order moments m_{20} and m_{02} describe the "distribution of mass" of the image with respect to the coordinate axes. In mechanics, they are called the *moments of inertia*. Another popular mechanical quantity, the *radius of gyration* with respect to an axis, can be also expressed in terms of moments as $\sqrt{m_{20}/m_{00}}$ and $\sqrt{m_{02}/m_{00}}$, respectively.

If the image is considered to be a joint probability density function (PDF) of two random variables (i.e., its values are non-negative and normalized such that $m_{00} = 1$), then the horizontal and vertical projections of the image are the 1D marginal densities, and m_{10} and m_{01} are the mean values of the variables. In case of zero means, m_{20} and m_{02} are their *variances*, and m_{11} is a *covariance* between them. In this way, the second-order moments define the principal axes of the image. As will be seen later, the second-order geometric moments can be used to find the normalized position of the image. In statistics, two higher-order moment characteristics of a probability density function have been commonly used – the *skewness* and the *kurtosis*. These terms are mostly used for 1D marginal densities only. Skewness of the horizontal marginal density is defined as $m_{30}/\sqrt{m_{20}^3}$ and that of the vertical marginal distribution as $m_{03}/\sqrt{m_{02}^3}$. The skewness measures the deviation of the PDF from symmetry. If the PDF is symmetric with respect to the mean (i.e., to the origin in this case), then the corresponding skewness equals zero. The kurtosis measures the "peakedness" of the probability density function and is again defined separately for each marginal distribution – the horizontal kurtosis as m_{40}/m_{20}^2 and the vertical kurtosis as m_{04}/m_{02}^2.

Characterization of the image by means of the geometric moments is complete and unambiguous in the following sense. For any image function, its geometric moments of all orders do exist and are finite. The image function can be exactly reconstructed from the set of all its moments (this assertion is known as the *uniqueness theorem* and holds thanks to the Weirstrass theorem on infinitely accurate polynomial approximation of continuous functions)[6].

Geometric moments of function f are closely related to its Fourier transformation. If we expand the kernel of the Fourier transformation $e^{-2\pi i(ux+vy)}$ into a power series, we realize that the geometric moments form Taylor coefficients of F

$$F(u,v) = \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} \frac{(-2\pi i)^{p+q}}{p!q!} m_{pq} u^p v^q \ .$$

[6] A more general moment problem is well known in statistics: can a given sequence be a set of moments of some compactly supported function? The answer is yes if the sequence is completely monotonic.

This link between the geometric moments and Fourier transformation is employed in many tasks, for instance in image reconstruction from moments as we will see in detail in Chapter 7.

3.1.4 Other moments

Geometric moments are very attractive thanks to the formal simplicity of the basis functions. This is why many theoretical considerations about moment invariants have been based on them. On the other hand, they have also certain disadvantages. One of them is their complicated transformation under rotation. This has led to introducing a class of *circular moments*, which change under rotation in a simple way and allow to design rotation invariants systematically, as we will show later in this chapter for the 2D case and in chapter 4 for the 3D case.

Another drawback of the geometric moments are their poor numerical properties when working in a discrete domain. Standard powers are nearly dependent both for small and large values of the exponent and increase rapidly in range as the order increases. This leads to correlated geometric moments and to the need for high computational precision. Using lower precision results in unreliable computation of geometric moments. This has led several authors to employing *orthogonal* (OG) moments, that is, moments the basis polynomials of which are orthogonal on Ω.

In theory, all polynomial bases of the same degree are equivalent because they generate the same space of functions. Any moment with respect to a certain basis can be expressed in terms of moments with respect to any other basis. From this point of view, OG moments of any type are equivalent to geometric moments. However, a significant difference appears when considering stability and other computational issues. OG moments have the advantage of requiring lower computing precision because we can evaluate them using recurrent relations, without expressing them in terms of the standard powers. OG moments are reviewed in chapter 7.

3.2 TRS invariants from geometric moments

Translation, rotation, and scaling in 2D is a four-parameter transformation, which can be described in a matrix form as

$$\mathbf{x}' = s\mathbf{R}_\alpha \mathbf{x} + \mathbf{t},$$

where $\mathbf{t}$ is a translation vector, s is a positive scaling factor (note that here we consider *uniform* scaling only, that is, s is the same both in horizontal and vertical directions), and $\mathbf{R}_\alpha$ is a rotation matrix by the angle α

$$\mathbf{R}_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} .$$

Note that the TRS transformations actually form a group because

$$\mathbf{R}_0 = \mathbf{I},$$

$$\mathbf{R}_\alpha \mathbf{R}_\beta = \mathbf{R}_{\alpha+\beta}$$

and

$$\mathbf{R}_\alpha^{-1} = \mathbf{R}_{-\alpha} = \mathbf{R}_\alpha^T .$$

3.2.1 Invariants to translation

Geometric moments are not invariant to translation. If f is shifted by vector $\mathbf{t} = (a,b)^T$, then its geometric moments change as

$$\begin{aligned} m'_{pq} &= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} x^p y^q f(x-a, y-b)\mathrm{d}x\mathrm{d}y = \\ &= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} (x+a)^p (y+b)^q f(x,y)\mathrm{d}x\mathrm{d}y = \\ &= \sum_{k=0}^{p}\sum_{j=0}^{q} \binom{p}{k}\binom{q}{j} a^k b^j m_{p-k,q-j}. \end{aligned} \tag{3.9}$$

Invariance to translation can be achieved simply by seemingly shifting the object into a certain well-defined position before the moments are calculated. The best way is to use its centroid – if we shift the object such that its centroid coincides with the origin of the coordinate system, then all the moments are translation invariants. This is equivalent to keeping the object fixed and shifting the polynomial basis into the object centroid[7]. We obtain so-called *central* geometric moments

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y)\mathrm{d}x\mathrm{d}y, \tag{3.10}$$

where

$$x_c = m_{10}/m_{00}, \quad y_c = m_{01}/m_{00}$$

are the coordinates of the object centroid. Note that it always holds $\mu_{10} = \mu_{01} = 0$ and $\mu_{00} = m_{00}$. Translation invariance of the central moments is straightforward.

The central moments can be expressed in terms of geometric moments as

$$\mu_{pq} = \sum_{k=0}^{p}\sum_{j=0}^{q} \binom{p}{k}\binom{q}{j} (-1)^{k+j} x_c^k y_c^j m_{p-k,q-j}.$$

Although this relation has little importance for theoretic consideration, it is sometimes used when we want to calculate the central moments by means of some fast algorithm for geometric moments computation.

3.2.2 Invariants to uniform scaling

Scaling invariance is obtained by a proper normalization of each moment. To find a normalization factor, let us first look at how the moment is changed under image scaling[8] by a factor s

$$\mu'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x/s, y/s)\mathrm{d}x\mathrm{d}y = \tag{3.11}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s^p (x - x_c)^p s^q (y - y_c)^q f(x, y) s^2 \mathrm{d}x\mathrm{d}y = s^{p+q+2}\mu_{pq} \tag{3.12}$$

[7] We recall that the use of the centralized coordinates is a common way of reaching translation invariance also in case of many other than moment-based features.

[8] There is a difference between scaling the image and scaling the coordinates – upscaling the image is the same as downscaling the coordinates and vice versa, the scaling factors are inverse. However, for the purpose of deriving invariants, it does not matter which particular scaling we consider.

In particular,

$$\mu'_{00} = s^2 \mu_{00}.$$

In principle, any moment can be used as a normalizing factor provided that it is non-zero for all images in the experiment. Since low-order moments are more stable to noise and easier to calculate, we normalize most often by a proper power of μ_{00}

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{00}^w} \ . \tag{3.13}$$

To eliminate the scaling parameter s, the power must be set as

$$w = \frac{p+q}{2} + 1. \tag{3.14}$$

The moment ν_{pq} is called the *normalized* central geometric moment[9] and is invariant to uniform scaling:

$$\nu'_{pq} = \frac{\mu'_{pq}}{(\mu'_{00})^w} = \frac{s^{p+q+2}\mu_{pq}}{(s^2\mu_{00})^w} = \nu_{pq} \ .$$

The moment that has been used for scaling normalization can no longer be used for recognition because the value of the corresponding normalized moment is always one (in the above normalization, $\nu_{00} = 1$). If we want to keep the zero-order moment valid, we have to normalize by another moment. Such kind of scaling normalization is used very rarely; probably the only meaningful normalization is by $(\mu_{20} + \mu_{02})^{w/2}$.

3.2.3 Invariants to non-uniform scaling

Non-uniform scaling is a transformation beyond the TRS framework that maps a unit square onto a rectangle. It is defined as

$$x' = ax,$$

$$y' = by,$$

where $a \neq b$ are positive scaling factors. Invariants to non-uniform scaling are sometimes called *aspect-ratio invariants.*

[9] This normalization was proposed already in Hu's paper [3], but the exponent was stated incorrectly as $w = \frac{p+q+1}{2}$. Many authors adopted this error, and some other authors introduced a new one claiming $w = [\frac{p+q}{2}] + 1$, where [.] denotes an integer part.

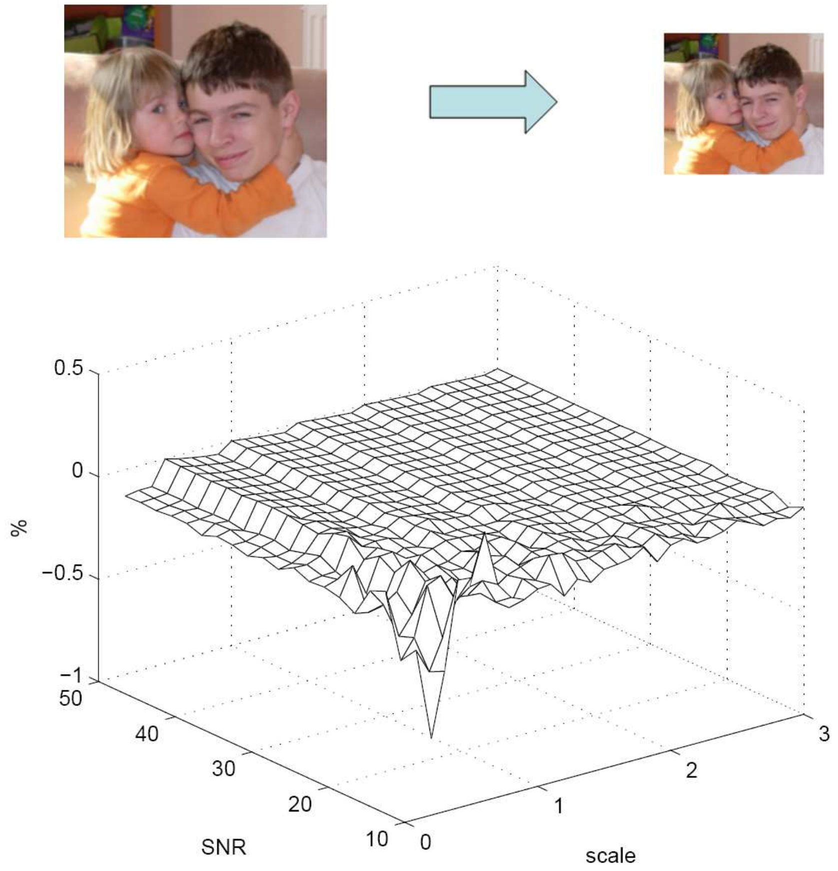


Figure 3.2: Numerical test of the normalized moment ν_{20}. Computer-generated scaling of the test image ranged form $s = 0.2$ to $s = 3$. To show robustness, each image was corrupted by additive Gaussian white noise. Signal-to-noise ratio (SNR) ranged from 50 (low noise) to 10 (heavy noise). Horizontal axes: scaling factor s and SNR, respectively. Vertical axis – relative deviation (in %) between ν_{20} of the original and that of the scaled and noisy image. The test proves the invariance of ν_{20} and illustrates its high robustness to noise.

Geometric central moments change under non-uniform scaling simply as

$$\mu'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a^p(x-x_c)^p b^q(y-y_c)^q f(x,y)ab \,\mathrm{d}x\,\mathrm{d}y = a^{p+1}b^{q+1}\mu_{pq} \ .$$

To eliminate the scaling factors a and b, we need at least two normalizing moments. When using for instance μ_{00} and μ_{20}, we get the invariants

$$A_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\alpha}\mu_{20}^{\beta}},$$

where

$$\alpha = \frac{3q-p}{2} + 1$$

and

$$\beta = \frac{p-q}{2}.$$

One can derive many different invariants to non-uniform scaling. For instance, Pan and Keane [4] proposed more "symmetric" normalization by three moments μ_{00}, μ_{20} and μ_{02}, which leads to

$$S_{pq} = \frac{\mu_{00}^{(p+q+2)/2}}{\mu_{20}^{(p+1)/2} \cdot \mu_{02}^{(q+1)/2}} \cdot \mu_{pq}.$$

Invariance to non-uniform scaling cannot be combined in a simple way with rotation invariance. The reason is that rotation and non-uniform scaling are not closed operations. When applied repeatedly, they generate an affine group of transformations, which implies that if we want to have invariance to non-uniform scaling and rotation, we must use affine invariants (they will be introduced in the chapter 5). There is no transformation group "between" the TRS and affine groups and, consequently, no special set of invariants "between" the TRS, and affine moment invariants may exist. This is why the invariants to non-uniform scaling described above are of only little importance in practical applications, and we presented them here merely for illustration.

3.2.4 Traditional invariants to rotation

Rotation moment invariants were firstly introduced in 1962 by Hu [3], who employed the results of the theory of algebraic invariants and derived

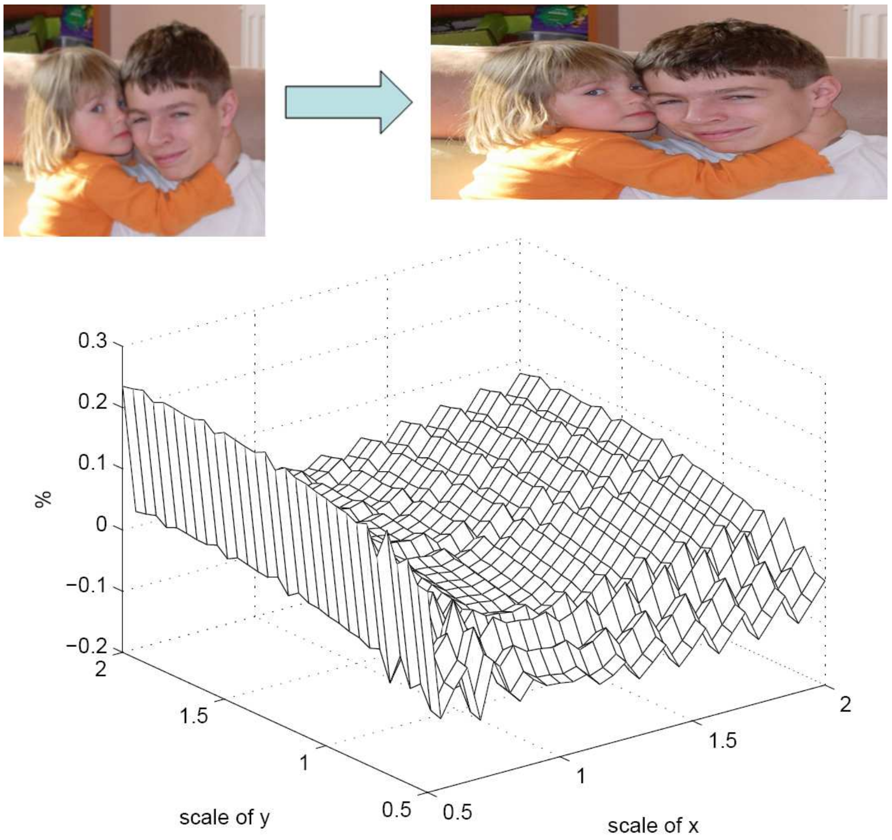


Figure 3.3: Numerical test of the aspect-ratio invariant A_{22}. Computer-generated scaling of the test image ranged form 0.5 to 2 in both directions independently. Horizontal axes: scaling factors a and b, respectively. Vertical axis – relative deviation (in %) between A_{22} of the original and that of the scaled image. The test illustrates the invariance of A_{22}. Higher relative errors for low scaling factors and typical jagged surface of the graph are the consequences of the image resampling.

his seven famous invariants to an in-plane rotation around the origin

$$\begin{aligned}
\phi_1 &= m_{20} + m_{02}, \\
\phi_2 &= (m_{20} - m_{02})^2 + 4m_{11}^2, \\
\phi_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2, \\
\phi_4 &= (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2, \qquad (3.15)\\
\phi_5 &= (m_{30} - 3m_{12})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2) + \\
&\quad (3m_{21} - m_{03})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2), \\
\phi_6 &= (m_{20} - m_{02})((m_{30} + m_{12})^2 - (m_{21} + m_{03})^2) + \\
&\quad 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}), \\
\phi_7 &= (3m_{21} - m_{03})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2) - \\
&\quad (m_{30} - 3m_{12})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2).
\end{aligned}$$

If we replace geometric moments by central or normalized moments in these relations, we obtain invariants not only to rotation but also to translation and/or scaling, which at the same time ensures invariance to rotation around an arbitrary point. Hu's derivation was rather complicated, and that is why only these seven invariants were derived explicitly and no hint how to derive invariants from higher-order moments was given in [3]. However, once we have the formulae, the proof of rotation invariance is easy. Let us demonstrate it for ϕ_1 and ϕ_2.

The second-order moments after rotation by angle α can be expressed as

$$\begin{aligned}
\mu'_{20} &= \cos^2\alpha\cdot\mu_{20}+\sin^2\alpha\cdot\mu_{02}-\sin 2\alpha\cdot\mu_{11}\\
\mu'_{02} &= \sin^2\alpha\cdot\mu_{20}+\cos^2\alpha\cdot\mu_{02}+\sin 2\alpha\cdot\mu_{11}\\
\mu'_{11} &= \frac{1}{2}\sin 2\alpha\cdot(\mu_{20}-\mu_{02})+\cos 2\alpha\cdot\mu_{11}.
\end{aligned}$$

Thus,

$$\phi'_1=\mu'_{20}+\mu'_{02}=(\sin^2\alpha+\cos^2\alpha)(\mu_{20}+\mu_{02})=\phi_1$$

and similarly for ϕ'_2, applying the formula $\cos 2\alpha=\cos^2\alpha-\sin^2\alpha$. □

Although Hu's invariants suffer from the limited recognition power, mutual dependence, and the restriction to the second- and third-order moments only, they have become classics, and, despite of their drawbacks, they have found numerous successful applications in various areas. The major weakness of Hu's theory is that it does not provide for a possibility of any generalization. By means of it, we could not derive invariants from higher-order moments. To illustrate that, let us consider how a general moment is changed under rotation:

$$\begin{aligned}
m'_{pq} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x\cos\alpha-y\sin\alpha)^p(x\sin\alpha+y\cos\alpha)^q f(x,y)\mathrm{d}x\mathrm{d}y=\\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{k=0}^{p}\sum_{j=0}^{q}\binom{p}{k}\binom{q}{j}x^{k+j}y^{p+q-k-j}(-1)^{p-k}(\cos\alpha)^{q+k-j}(\sin\alpha)^{p-k+j}\\
&\qquad f(x,y)\mathrm{d}x\mathrm{d}y=\\
&= \sum_{k=0}^{p}\sum_{j=0}^{q}(-1)^{p-k}\binom{p}{k}\binom{q}{j}(\cos\alpha)^{q+k-j}(\sin\alpha)^{p-k+j}m_{k+j,p+q-k-j}\ .
\end{aligned} \tag{3.16}$$

This is a rather complicated expression from which the rotation parameter α cannot be easily eliminated. An attempt was proposed by Jin and

Tianxu [5], but compared to the approach described in the sequel it is difficult and not very transparent.

In the next section, we present a general approach to deriving rotation invariants, which uses circular moments.

3.3 Rotation invariants using circular moments

After Hu, various approaches to the theoretical derivation of moment-based rotation invariants, which would not suffer from a limitation to low orders, have been published. Although nowadays the key idea seems to be bright and straightforward, its first versions appeared in the literature only two decades after Hu's paper.

The key idea is to replace the cartesian coordinates (x, y) by (centralized) *polar coordinates* (r, θ)

$$\begin{aligned} x &= r\cos\theta \qquad & r &= \sqrt{x^2+y^2} \\ y &= r\sin\theta \qquad & \theta &= \arctan(y/x) \end{aligned} \tag{3.17}$$

In the polar coordinates, the rotation is a translation in the angular argument. To eliminate the impact of this translation, we were inspired by the Fourier shift theorem – if the basis functions in angular direction were harmonics, the translation would result just in a phase shift of the moment values, which would be easy to eliminate. This brilliant general idea led to the birth of a family of so-called *circular moments*[10]. The circular moments have the form

$$C_{pq} = \int\limits_0^\infty \int\limits_0^{2\pi} R_{pq}(r) e^{i\xi(p,q)\theta} f(r,\theta) r \mathrm{d}\theta \mathrm{d}r \tag{3.18}$$

where $R_{pq}(r)$ is a radial univariate polynomial and $\xi(p, q)$ is a function (usually a very simple one) of the indices. Complex, Zernike, pseudo-Zernike, Fourier-Mellin, radial Chebyshev, and many other moments are particular cases of the circular moments.

Under coordinate rotation by angle α (which is equivalent to image

[10] They are sometimes referred to as radial or radial-circular moments.

rotation by $-\alpha$), the circular moment is changed as

$$\begin{aligned} C'_{pq} &= \int\limits_0^\infty \int\limits_0^{2\pi} R_{pq}(r) e^{i\xi(p,q)\theta} f(r,\theta+\alpha) r \mathrm{d}\theta \mathrm{d}r = \\ &= \int\limits_0^\infty \int\limits_0^{2\pi} R_{pq}(r) e^{i\xi(p,q)(\theta-\alpha)} f(r,\theta) r \mathrm{d}\theta \mathrm{d}r = \\ &= e^{-i\xi(p,q)\alpha} C_{pq} \ . \end{aligned} \tag{3.19}$$

To eliminate the parameter α, we can take the moment magnitude $|C_{pq}|$ (note that the circular moments are generally complex-valued even if $f(x,y)$ is real) or apply some more sophisticated method to reach the phase cancelation (such a method of course depends on the particular function $\xi(p,q)$).

The above idea can be traced in several papers, which basically differ from each other by the choice of the radial polynomials $R_{pq}(r)$, the function $\xi(p,q)$, and by the method by which α is eliminated.

Teague [6] and Wallin [7] proposed to use Zernike moments, which are a special case of circular moments, and to take their magnitudes. Li [8] used Fourier-Mellin moments to derive invariants up to the order 9, Wong [9] used complex monomials up to the fifth order that originate from the theory of algebraic invariants. Mostafa and Psaltis [10] introduced the idea to use complex moments for deriving invariants, but they focused on the evaluation of the invariants rather than on constructing higher-order systems. This approach was later followed by Flusser [11, 12], who proposed a general theory of constructing rotation moment invariants. His theory, presented in the next section, is based on the complex moments. It is formally simple and transparent, allows deriving invariants of any orders, and enables studying mutual dependence/independence of the invariants in a readable way. The usage of the complex moments is not essential; this theory can be easily modified for other circular moments as was for instance demonstrated by Derrode and Ghorbel [13] for Fourier-Mellin moments and Yang et al. [14] for Gaussian-Hermite moments.

3.4 Rotation invariants from complex moments

In this section, we present a general method of deriving complete and independent sets of rotation invariants of any orders. This method employs *complex moments* of the image.

3.4.1 Complex moments

The complex moment c_{pq} is obtained when we choose the polynomial basis of complex monomials $\pi_{pq}(x,y) = (x+iy)^p(x-iy)^q$

$$c_{pq} = \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} (x+iy)^p(x-iy)^q f(x,y)\mathrm{d}x\mathrm{d}y. \tag{3.20}$$

It follows from the definition that only the indices $p \geq q$ are independent and worth considering because $c_{pq} = c^*_{qp}$ (the asterisk denotes complex conjugate).

Complex moments carry the same amount of information as the geometric moments of the same order. Each complex moment can be expressed in terms of geometric moments as

$$c_{pq} = \sum_{k=0}^{p}\sum_{j=0}^{q}\binom{p}{k}\binom{q}{j}(-1)^{q-j}\cdot i^{p+q-k-j}\cdot m_{k+j,p+q-k-j} \tag{3.21}$$

and vice versa[11]

$$m_{pq} = \frac{1}{2^{p+q}i^q}\sum_{k=0}^{p}\sum_{j=0}^{q}\binom{p}{k}\binom{q}{j}(-1)^{q-j}\cdot c_{k+j,p+q-k-j}. \tag{3.22}$$

We can also see the link between the complex moments and Fourier transformation. The complex moments are almost (up to a multiplicative constant) the Taylor coefficients of the Fourier transformation of $\mathcal{F}(f)(u+v, i(u-v))$. If we denote $U = u+v$ and $V = i(u-v)$, then we

[11] While the proof of (3.21) is straightforward, the proof of (3.22) requires first to express x and y as $x = ((x+iy)+(x-iy))/2$ and $y = ((x+iy)-(x-iy))/2i$.

have

$$\mathcal{F}(f)(U,V) \equiv \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} e^{-2\pi i(Ux+Vy)} f(x,y)\mathrm{d}x\mathrm{d}y = \\ = \sum_{j=0}^{\infty}\sum_{k=0}^{\infty}\frac{(-2\pi i)^{j+k}}{j!k!}c_{jk}u^j v^k. \tag{3.23}$$

When expressed in polar coordinates, the complex moments take the form

$$c_{pq} = \int\limits_0^{\infty}\int\limits_0^{2\pi} r^{p+q}e^{i(p-q)\theta}f(r,\theta)r\mathrm{d}\theta\mathrm{d}r. \tag{3.24}$$

Hence, the complex moments are special cases of the circular moments with a choice of $R_{pq} = r^{p+q}$ and $\xi(p,q) = p - q$. The following lemma is a consequence of the rotation property of circular moments.

Lemma 3.3: Let f' be a rotated version (around the origin) of f, i. e. $f'(r,\theta) = f(r,\theta+\alpha)$. Then

$$c'_{pq} = e^{-i(p-q)\alpha} \cdot c_{pq}. \tag{3.25}$$

3.4.2 Construction of rotation invariants

The simplest method proposed by many authors (see [15] for instance) is to use as invariants the moment magnitudes themselves. However, they do not generate a complete set of invariants – by taking the magnitudes only we miss many useful invariants. In the following theorem, phase cancelation is achieved by multiplication of appropriate moment powers.

Theorem 3.4: Let $n \geq 1$, and let $k_i, p_i \geq 0$, and $q_i \geq 0$ $(i = 1, \ldots, n)$ be arbitrary integers such that

$$\sum_{i=1}^{n} k_i(p_i - q_i) = 0.$$

Then

$$I = \prod_{i=1}^{n} c_{p_i q_i}^{k_i} \tag{3.26}$$

is invariant to rotation.

Proof: Let us consider rotation by angle α. Then

$$I' = \prod_{i=1}^{n} c'^{k_i}_{p_i q_i} = \prod_{i=1}^{n} e^{-ik_i(p_i-q_i)\alpha} \cdot c^{k_i}_{p_i q_i} = e^{-i\alpha \sum_{i=1}^{n} k_i(p_i-q_i)} \cdot I = I. \qquad \square$$

According to Theorem 3.4, some simple examples of rotation invariants are c_{11}, $c_{20}c_{02}$, $c_{20}c_{12}^2$, and so on. Most invariants (3.26) are complex. If real-valued features are required, we consider real and imaginary parts of each of them separately. To achieve also translation invariance, we use central coordinates in the definition of the complex moments (3.20) and we discard all invariants containing c_{10} and c_{01}. Scaling invariance can be achieved by the same normalization as we used for the geometric moments.

3.4.3 Construction of the basis

In this section, we will pay attention to the construction of a basis of invariants up to a given order. Theorem 3.4 allows us to construct an infinite number of invariants for any order of moments, but only few of them being mutually independent. By the term *basis* we intuitively understand the smallest set required to express all other invariants. More precisely, the basis must be *independent*, which means that none of its elements can be expressed as a function of the other elements, and also *complete*, meaning that any rotation invariant can be expressed by means of the basis elements only.

The knowledge of the basis is a crucial point in all pattern recognition tasks because the basis provides the same discriminative power as the set of all invariants and thus minimizes the computational cost. For instance, the set

$$\{c_{20}c_{02}, c_{21}^2 c_{02}, c_{12}^2 c_{20}, c_{21}c_{12}, c_{21}^3 c_{02} c_{12}\}$$

is a dependent set whose basis is $\{c_{12}^2 c_{20}, c_{21}c_{12}\}$.

To formalize these terms, we first introduce the following definitions.

Definition 3.5: Let $k \geq 1$, let $\mathcal{I} = \{I_1, \ldots, I_k\}$ be a set of rotation invariants. Let J be also a rotation invariant. J is said to be *dependent on* $\mathcal{I}$ if and only if there exists a function F of k variables such that

$$J = F(I_1, \ldots, I_k).$$

J is said to be *independent on* $\mathcal{I}$ otherwise.

Definition 3.6: Let $k > 1$ and let $\mathcal{I} = \{I_1, \ldots, I_k\}$ be a set of rotation invariants. The set $\mathcal{I}$ is said to be *dependent* if and only if there exists $k_0 \leq k$ such that I_{k_0} depends on $\mathcal{I} \dot{-} \{I_{k_0}\}$. The set $\mathcal{I}$ is said to be *independent* otherwise.

According to this definition, $\{c_{20}c_{02},\, c_{20}^2 c_{02}^2\}$, $\{c_{21}^2 c_{02},\, c_{21}c_{12},\, c_{21}^3 c_{02}c_{12}\}$, and $\{c_{20}c_{12}^2,\, c_{02}c_{21}^2\}$ are examples of dependent invariant sets.

Definition 3.7: Let $\mathcal{I}$ be a set of rotation invariants, and let $\mathcal{B}$ be its subset. $\mathcal{B}$ is called a *complete* subset if and only if any element of $\mathcal{I} \dot{-} \mathcal{B}$ depends on $\mathcal{B}$. The set $\mathcal{B}$ is a *basis* of $\mathcal{I}$ if and only if it is independent and complete.

Now we can formulate a fundamental theorem that tells us how to construct an invariant basis of a given order.

Theorem 3.8: Let us consider complex moments up to the order $r \geq 2$. Let a set of rotation invariants $\mathcal{B}$ be constructed as follows:

$$\mathcal{B} = \{\Phi(p,q) \equiv c_{pq}c_{q_0 p_0}^{p-q} | p \geq q \wedge p + q \leq r\},$$

where p_0 and q_0 are arbitrary indices such that $p_0 + q_0 \leq r$, $p_0 - q_0 = 1$ and $c_{p_0 q_0} \neq 0$ for all admissible images. Then $\mathcal{B}$ is a basis of all rotation invariants[12] created from the moments of any kind up to the order r.

Theorem 3.8 is very strong because it claims $\mathcal{B}$ is a basis of *all possible* rotation moment invariants, not only of those constructed according to (3.26) and not only of those which are based on complex moments. In other words, $\mathcal{B}$ provides at least the same discrimination power as any other set of moment invariants up to the given order $r \geq 2$ because any possible invariant can be expressed in terms of $\mathcal{B}$. (Note that this is a theoretical property; it may be violated in the discrete domain where different polynomials have different numerical properties.)

Proof: Let us prove the independence of $\mathcal{B}$ first. Let us assume $\mathcal{B}$ is dependent, that is, there exists $\Phi(p,q) \in \mathcal{B}$, such that it depends on

[12] The correct notation should be $\mathcal{B}(p_0, q_0)$ because the basis depends on the choice of p_0 and q_0; however, we drop these indexes for simplicity.

$\mathcal{B} \doteq \{\Phi(p,q)\}$. As follows from the linear independence of the polynomials $(x+iy)^p(x-iy)^q$ and, consequently, from independence of the complex moments themselves, it must hold that $p = p_0$ and $q = q_0$. That means, according to the above assumption, there exist invariants $\Phi(p_1,q_1),\ldots,\Phi(p_n,q_n)$ and $\Phi(s_1,t_1),\ldots,\Phi(s_m,t_m)$ from $\mathcal{B} \doteq \{\Phi(p_0,q_0)\}$ and positive integers $k_1,\ldots,k_n$ and $\ell_1,\ldots,\ell_m$ such that

$$\Phi(p_0,q_0) = \frac{\prod_{i=1}^{n_1}\Phi(p_i,q_i)^{k_i}\cdot\prod_{i=n_1+1}^{n}\Phi^*(p_i,q_i)^{k_i}}{\prod_{i=1}^{m_1}\Phi(s_i,t_i)^{\ell_i}\cdot\prod_{i=m_1+1}^{m}\Phi^*(s_i,t_i)^{\ell_i}}. \tag{3.27}$$

Substituting into (3.27) and grouping the factors $c_{p_0q_0}$ and $c_{q_0p_0}$ together, we get

$$\Phi(p_0,q_0) = \frac{c_{q_0p_0}^{\sum_{i=1}^{n_1}k_i(p_i-q_i)}\cdot c_{p_0q_0}^{\sum_{i=n_1+1}^{n}k_i(q_i-p_i)}\cdot\prod_{i=1}^{n_1}c_{p_iq_i}^{k_i}\cdot\prod_{i=n_1+1}^{n}c_{q_ip_i}^{k_i}}{c_{q_0p_0}^{\sum_{i=1}^{m_1}\ell_i(s_i-t_i)}\cdot c_{p_0q_0}^{\sum_{i=m_1+1}^{m}\ell_i(t_i-s_i)}\cdot\prod_{i=1}^{m_1}c_{s_it_i}^{\ell_i}\cdot\prod_{i=m_1+1}^{m}c_{t_is_i}^{\ell_i}}. \tag{3.28}$$

Comparing the exponents of $c_{p_0q_0}$ and $c_{q_0p_0}$ on both sides, we get the constraints

$$K_1 = \sum_{i=1}^{n_1}k_i(p_i-q_i) - \sum_{i=1}^{m_1}\ell_i(s_i-t_i) = 1 \tag{3.29}$$

and

$$K_2 = \sum_{i=n_1+1}^{n}k_i(q_i-p_i) - \sum_{i=m_1+1}^{m}\ell_i(t_i-s_i) = 1. \tag{3.30}$$

Since the rest of the right-hand side of eq. (3.28) must be equal to 1, and since the moments themselves are mutually independent, the following constraints must be fulfilled for any index i:

$$n_1 = m_1, \qquad n = m, \qquad p_i = s_i, \qquad q_i = t_i, \qquad k_i = \ell_i.$$

Introducing these constraints into (3.29) and (3.30), we get $K_1 = K_2 = 0$ which is a contradiction.

To prove the completeness of $\mathcal{B}$, it is sufficient to resolve the so-called *inverse problem*, which means to recover all complex moments (and, consequently, all geometric moments) up to the order r when knowing the elements of $\mathcal{B}$. Thus, the following nonlinear system of equations must be resolved for the c_{pq}'s:

$$\begin{array}{rcl}
\Phi(p_0, q_0) & = & c_{p_0q_0}c_{q_0p_0}, \\
\Phi(0,0) & = & c_{00}, \\
\Phi(1,0) & = & c_{10}c_{q_0p_0}, \\
\Phi(2,0) & = & c_{20}c^2_{q_0p_0}, \\
\Phi(1,1) & = & c_{11}, \\
\Phi(3,0) & = & c_{30}c^3_{q_0p_0}, \\
 & \cdots & \\
\Phi(r,0) & = & c_{r0}c^r_{q_0p_0}, \\
\Phi(r-1,1) & = & c_{r-1,1}c^{r-2}_{q_0p_0}, \\
 & \cdots &
\end{array} \tag{3.31}$$

Since $\mathcal{B}$ is a set of rotation invariants, it does not reflect the actual orientation of the object. Thus, there is one degree of freedom when recovering the object moments that correspond to the choice of the object orientation. Without loss of generality, we can choose such orientation in which $c_{p_0q_0}$ is real and positive. As can be seen from eq. (3.25), if $c_{p_0q_0}$ is nonzero then such orientation always exists. The first equation of (3.31) can be then immediately resolved for $c_{p_0q_0}$:

$$c_{p_0q_0} = \sqrt{\Phi(p_0, q_0)}.$$

Consequently, using the relationship $c_{q_0p_0} = c_{p_0q_0}$, we obtain the solutions

$$c_{pq} = \frac{\Phi(p,q)}{c^{p-q}_{q_0p_0}}$$

and

$$c_{pp} = \Phi(p,p)$$

for any p and q. Recovering the geometric moments is straightforward from eq. (3.8). Since any polynomial is a linear combination of standard powers x^py^q, any moment (and any moment invariant) can be expressed as a function of geometric moments, which completes the proof. □

Theorem 3.8 allows us not only to create the basis but also to calculate the number of its elements in advance. Let us denote it as $|\mathcal{B}|$. If r is odd then

$$|\mathcal{B}| = \frac{1}{4}(r+1)(r+3),$$

if r is even then

$$|\mathcal{B}| = \frac{1}{4}(r+2)^2.$$

(These numbers refer to complex-valued invariants.) We can see that the authors who used the moment magnitudes only actually lost about half of the information containing in the basis $\mathcal{B}$.

The basis defined in Theorem 3.8 is generally not unique. It depends on the particular choice of p_0 and q_0, which is very important. How shall we select these indices in practice? On the one hand, we want to keep p_0 and q_0 as small as possible because lower-order moments are less sensitive to noise than the higher-order ones. On the other hand, the close-to-zero value of $c_{p_0q_0}$ may cause numerical instability of the invariants. Thus, we propose the following algorithm. We start with $p_0 = 2$ and $q_0 = 1$ and check if $|c_{p_0q_0}|$ exceeds a pre-defined threshold for all objects (in practice this means for all given training samples or database elements). If this condition is met, we accept the choice; in the opposite case we increase both p_0 and q_0 by one and repeat the above procedure.

3.4.4 Basis of the invariants of the second and third orders

In this section, we present a basis of the rotation invariants composed of the moments of the second and third orders, that is constructed according to Theorem 3.8 by choosing $p_0 = 2$ and $q_0 = 1$. The basis is

$$\begin{aligned} \Phi(1,1) &= c_{11}, \\ \Phi(2,1) &= c_{21}c_{12}, \\ \Phi(2,0) &= c_{20}c_{12}^2, \\ \Phi(3,0) &= c_{30}c_{12}^3. \end{aligned} \tag{3.32}$$

In this case, the basis is determined unambiguously and contains six real-valued invariants. It is worth noting that formally, according to Theorem 3.8, the basis should contain also invariants $\Phi(0,0) = c_{00}$ and $\Phi(1,0) = c_{10}c_{12}$. We did not include these two invariants in the basis because $c_{00} = \mu_{00}$ is in most recognition applications used for normalization to scaling, and $c_{10} = m_{10} + im_{01}$ is used to achieve translation invariance. Then $\Phi(0,0) = 1$ and $\Phi(1,0) = 0$ for any object and it is useless to consider them. Numerical stability of $\Phi(2,0)$ is illustrated in Figure 3.4.

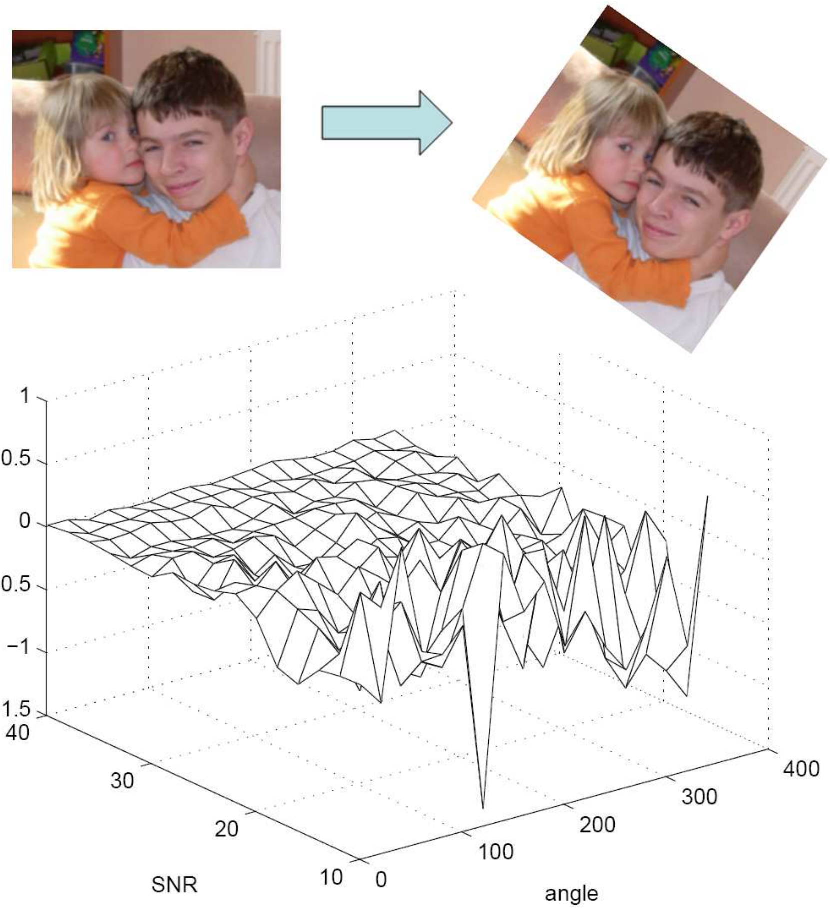


Figure 3.4: Numerical test of the basic invariant $\Phi(2,0)$. Computer-generated rotation of the test image ranged form 0 to 360 degrees. To show robustness, each image was corrupted by additive Gaussian white noise. Signal-to-noise ratio (SNR) ranged from 40 (low noise) to 10 (heavy noise). Horizontal axes: rotation angle and SNR, respectively. Vertical axis – relative deviation (in %) between $\mathcal{R}e(\Phi(2,0))$ of the original and that of the rotated and noisy image. The test proves the invariance of $\mathcal{R}e(\Phi(2,0))$ and illustrates its high robustness to noise.

3.4.5 Relationship to the Hu invariants

In this section, we highlight the relationship between the Hu invariants (3.15) and the proposed invariants (3.32). We show the Hu invariants are incomplete and mutually dependent, which can explain some practical problems connected with their usage.

It can be seen that the Hu invariants are nothing but particular representatives of the general form (3.26)

$$\begin{aligned}
\phi_1 &= c_{11}, \\
\phi_2 &= c_{20}c_{02}, \\
\phi_3 &= c_{30}c_{03}, \\
\phi_4 &= c_{21}c_{12}, \\
\phi_5 &= \mathcal{R}e(c_{30}c_{12}^3), \\
\phi_6 &= \mathcal{R}e(c_{20}c_{12}^2), \\
\phi_7 &= \mathcal{I}m(c_{30}c_{12}^3)
\end{aligned} \tag{3.33}$$

and that they can be expressed also in terms of the basis (3.32)

$$\begin{aligned}
\phi_1 &= \Phi(1,1) \\
\phi_2 &= \frac{|\Phi(2,0)|^2}{\Phi(2,1)^2} \\
\phi_3 &= \frac{|\Phi(3,0)|^2}{\Phi(2,1)^3} \\
\phi_4 &= \Phi(2,1) \\
\phi_5 &= \mathcal{R}e(\Phi(3,0)) \\
\phi_6 &= \mathcal{R}e(\Phi(2,0)) \\
\phi_7 &= \mathcal{I}m(\Phi(3,0)).
\end{aligned}$$

Using (3.33), we can demonstrate the dependency of the Hu invariants. It holds

$$\phi_3 = \frac{\phi_5^2 + \phi_7^2}{\phi_4^3},$$

which means that either ϕ_3 or ϕ_4 is useless and can be excluded from the Hu's system without any loss of discrimination power.

Moreover, we show the Hu's system is incomplete. Let us try to recover complex and geometric moments when knowing $\phi_1, \ldots, \phi_7$ under the same normalization constraint as in the previous case, i.e. c_{21} is required to be real and positive. Complex moments c_{11}, c_{21}, c_{12}, c_{30}, and c_{03} can be recovered in a straightforward way:

$$\begin{aligned} c_{11} &= \phi_1, \\ c_{21} &= c_{12} = \sqrt{\phi_4}, \\ \mathcal{R}e(c_{30}) &= \mathcal{R}e(c_{03}) = \frac{\phi_5}{c_{12}^3}, \\ \mathcal{I}m(c_{30}) &= -\mathcal{I}m(c_{03}) = \frac{\phi_7}{c_{12}^3}. \end{aligned}$$

Unfortunately, c_{20} cannot be fully recovered. We have, for its real part,

$$\mathcal{R}e(c_{20}) = \frac{\phi_6}{c_{12}^2}$$

but for its imaginary part we get

$$(\mathcal{I}m(c_{20}))^2 = |c_{20}|^2 - (\mathcal{R}e(c_{20}))^2 = \phi_2 - \left(\frac{\phi_6}{c_{12}^2}\right)^2.$$

There is no way of determining the sign of $\mathcal{I}m(c_{20})$. In terms of geometric moments, it means the sign of m_{11} cannot be recovered.

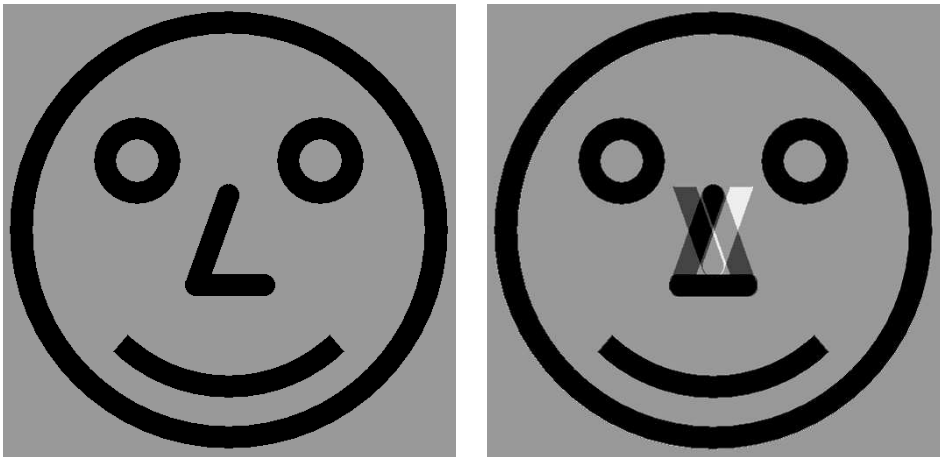
The incompleteness of the Hu invariants implicates their lower discrimination power compared to the proposed invariants of the same order. Let us consider two objects $f(x, y)$ and $g(x, y)$, both in the normalized central positions, having the same geometric moments up to the third order except m_{11}, for which $m_{11}^{(f)} = -m_{11}^{(g)}$. In the case of artificial data, such object $g(x, y)$ exists for any given $f(x, y)$ and can be designed as

$$g(x, y) = (f(x, y) + f(-x, y) + f(x, -y) - f(-x, -y))/2, \qquad (3.34)$$

see Figure 3.5 for an example. It is easy to prove that under (3.34) the moment constraints are always fulfilled. While the basic invariants (3.32) distinguish these two objects by the imaginary part of $\Phi(2, 0)$, the Hu invariants are not able to do so (see Table 3.1), even if the objects are easy to discriminate visually.

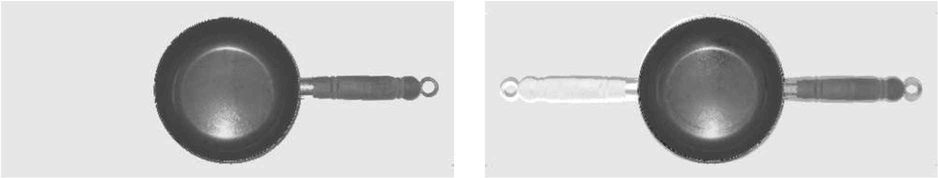
This property can be demonstrated also on real data. In Figure 3.6 (left), one can see the photograph of a pan. A picture of a virtual "two-handle" pan (Figure 3.6 right) was created from the original image according to (3.34). Although these two objects are apparently different, all their Hu invariants are exactly the same. On the other hand, the new invariant $\Phi(2,0)$ distinguishes these two objects clearly thanks to the opposite signs of its imaginary part.

This ambiguity cannot be avoided by a different choice of the normalization constraint when recovering the moments. Let us, for instance, consider another normalization constraint, which requires c_{20} to be real and positive. In terms of geometric moments, it corresponds to the requirement $m_{11} = 0$ and $m_{20} > m_{02}$. Then, similarly to the previous case, the signs of $(m_{30} + m_{12})$ and $(m_{03} + m_{21})$ cannot be unambiguously determined.



(a) (b)

Figure 3.5: The smiles: (a) original and (b) another figure created from the original according to Eq. (3.34). For the values of the respective invariants see Table 3.1.



(a) (b)

Figure 3.6: (a) Original image of a pan and (b) a virtual "two-handle" pan. These objects are distinguishable by the basic invariants but not by the Hu invariants.

It is worth noting that also other earlier sets of rotation invariants are

Table 3.1: The values of the Hu invariants and the basic invariants (3.32) of "The smiles" in Figure 3.5. The only invariant discriminating them is $\mathcal{I}m(\Phi(2,0))$. (The values shown here were calculated after bringing Figure 3.5 into normalized central position $c_{21} > 0$ and nonlinearly scaled to the range from -10 to 10 for display.)

Hu	Fig. 3.5(a)	Fig. 3.5(b)	Basic	Fig. 3.5(a)	Fig. 3.5(b)
ϕ_1	4.2032	4.2032	$\Phi(1,1)$	4.2032	4.2032
ϕ_2	1.7563	1.7563	$\Phi(2,1)$	6.5331	6.5331
ϕ_3	3.6395	3.6395	$\mathcal{R}e(\Phi(2,0))$	8.6576	8.6576
ϕ_4	6.5331	6.5331	$\mathbf{Im(\Phi(2,0)})$	**7.6437**	**-7.6437**
ϕ_5	1.0074	1.0074	$\mathcal{R}e(\Phi(3,0))$	1.0074	1.0074
ϕ_6	8.6576	8.6576	$\mathcal{I}m(\Phi(3,0))$	-6.9850	-6.9850
ϕ_7	-6.9850	-6.9850			

dependent and/or incomplete. This was, however, not mentioned in the original papers, partly because the authors did not pay attention to these issues and partly because their methods did not allow them to find out the dependencies among the invariants in a systematic way. Li [8] published a set of invariants from moments up to the ninth order. Unfortunately, his system includes the Hu invariants and therefore it also cannot be a basis. Jin and Tianxu [5] derived twelve invariants in explicit form but only eight of them are independent. Wong [9] presented a set of sixteen invariants from moments up to the third order and a set of "more than forty-nine" invariants from moments up to the fourth order. It follows immediately from Theorem 3.8 that the basis of the third-order invariants has only six elements and the basis of the fourth-order invariants has nine elements. Thus, most of the Wong's invariants are dependent and of no practical significance. Even Mukundan's monograph [16] presents a dependent and incomplete set of twenty-two rotation invariants up to the sixth order (see [16], p. 125). The construction of the invariants from the complex moments has resolved the independence and completeness issues in an elegant way.

3.5 Pseudoinvariants

In this section, we investigate the behavior of rotation invariants under mirror reflection of the image (see Figure 3.7). Since the mirror reflection is not a special case of the TRS transformation, the rotation invariants may change when the image is mirrored. The mirror reflection can occur in some applications only, but it is useful to know how the rotation invariants behave in that case. We show they cannot change arbitrarily. There are only two possibilities – a (real-valued) invariant can either stay constant or change its sign. The invariants, which preserve their value under reflection are traditionally called *true invariants* while those that change the sign are called *pseudoinvariants* [6] or, misleadingly, *skew invariants* [3]. Pseudoinvariants discriminate between mirrored images of the same object, which is useful in some applications but may be undesirable in other cases. We show which invariants from the basis introduced in Theorem 3.8 are pseudoinvariants and which are the true ones.

Let us consider a basic invariant, and let us investigate its behavior under a reflection across an arbitrary line. Due to the rotation and shift invariance, we can limit ourselves to the reflection across the x-axis.

Let $\overline{f}(x, y)$ be a mirrored version of $f(x, y)$, i.e. $\overline{f}(x, y) = f(x, -y)$. As follows from the definition,

$$\overline{c_{pq}} = c_{qp} = c^*_{pq}.$$

Thus, it holds for any basic invariant $\Phi(p, q)$

$$\overline{\Phi(p, q)} = \overline{c_{pq} c^{p-q}_{q_0 p_0}} = c^*_{pq} \cdot (c^*_{q_0 p_0})^{p-q} = \Phi^*(p, q).$$

This proves that the real parts of the basic invariants are true invariants. On the other hand, the imaginary parts of them change their signs under reflection and hence are pseudoinvariants. This brings a consequence for the recognition of objects with axial symmetry. The pseudoinvariants cannot differentiate among any axially symmetric objects because their values must always be zero.

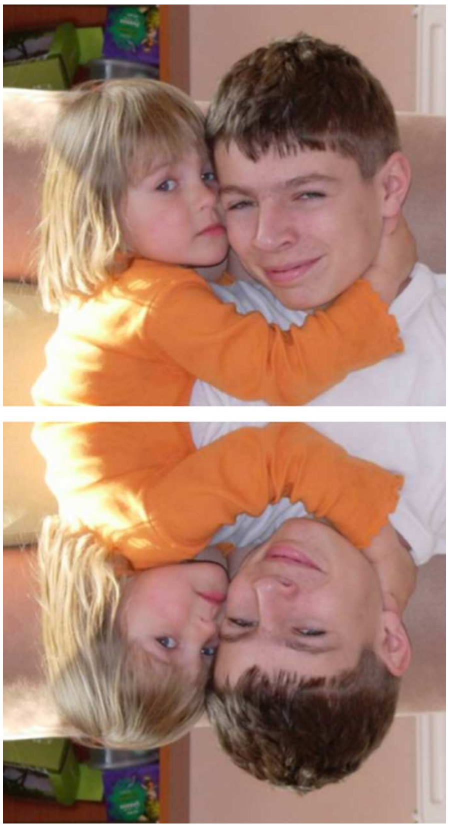


Figure 3.7: The test image and its mirrored version. Basic invariants of the mirrored image are complex conjugates of those of the original.

3.6 Combined invariants to TRS and contrast stretching

So far we have considered invariants to spatial transformations only. However, if the features used in a recognition system are calculated from a graylevel/color image and not just from a binary image, they should be invariant also to various changes of intensity. These changes may be caused by the change of illumination conditions, by the camera setup, by the influence of the environment, and by many other factors. In Chapter 6 we describe invariants to graylevel degradations caused by linear filtering. In this section we consider *contrast stretching* only, which is a very simple graylevel transformation given as

$$f'(x,y) = a \cdot f(x,y),$$

where a is a positive stretching factor. This model approximately describes the change of the illumination of the scene between two acquisitions (in case of digital images, this model may not be valid everywhere because $f'(x,y)$ might overflow the 8-bit intensity range at some places, but we ignore this issue here for simplicity). The invariants to contrast stretching in connection with Hu's rotation invariants were firstly proposed by Maitra [17] and later improved by Hupkens [18]. Here we will demonstrate how to include the invariance to contrast stretching into the TRS invariants described earlier in this chapter.

Since $\mu'_{pq} = a \cdot \mu_{pq}$ and $c'_{pq} = a \cdot c_{pq}$, pure contrast (and translation) invariance and can be achieved simply by normalizing each central or complex moment by μ_{00}. Now consider a contrast stretching *and* a rotation together. It holds for any basic invariant $\Phi(p,q)$

$$\Phi'(p,q) = a^{p-q+1}\Phi(p,q),$$

and it is sufficient to use the normalization

$$\frac{\Phi(p,q)}{\mu_{00}^{p-q+1}}.$$

This approach unfortunately cannot be further extended to scaling. If μ_{00} has been used for normalization to contrast, it cannot be at the same time used for scaling normalization. We have to normalize by some other invariant, preferably by c_{11} for the sake of simplicity.

Let $\widetilde{c_{pq}}$ be defined as follows

$$\widetilde{c_{pq}} = \frac{c_{pq}\mu_{00}^{\frac{p+q}{2}-1}}{c_{11}^{\frac{p+q}{2}}}.$$

Then $\widetilde{c_{pq}}$ is *both* contrast and scaling normalized moment, which can be used for constructing rotation invariants in a usual way as

$$\Psi(p,q) = \widetilde{c_{pq}}\widetilde{c_{q_0p_0}}^{p-q}.$$

This is possible because c_{11} itself is a rotation invariant. Hence, $\Psi(p,q)$ is a combined TRS and contrast invariant. Note that $\Psi(1,1) = 1$ for any object due to the normalization.

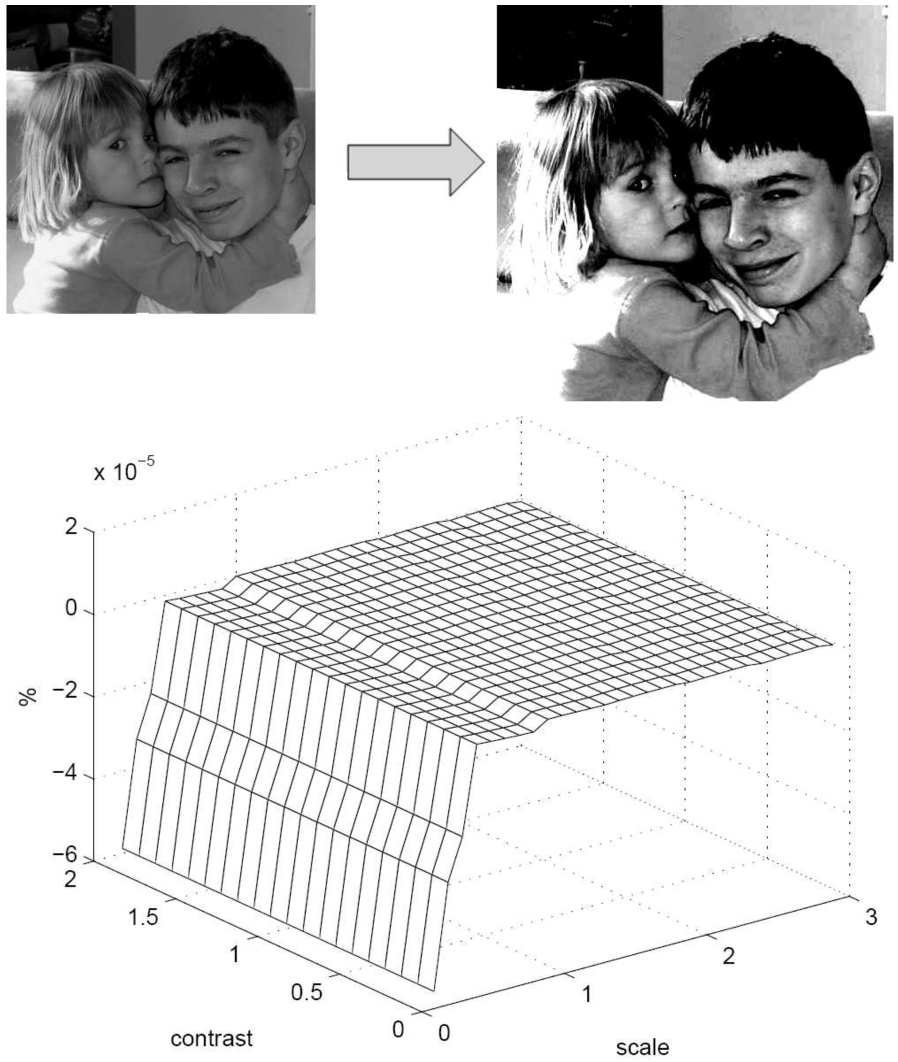


Figure 3.8: Numerical test of the contrast and TRS invariant $\Psi(2,0)$ for $p_0 = 2$ and $q_0 = 1$. Computer-generated scaling of the test image ranged from $s = 0.2$ to $s = 3$, and the contrast stretching factor ranged from $a = 0.1$ to $a = 2$. Horizontal axes: scaling factor s and contrast stretching factor a, respectively. Vertical axis – relative deviation (in %) between $\mathcal{R}e(\Psi(2,0))$ of the original and that of the scaled and stretched image. The test proves the invariance of $\mathcal{R}e(\Psi(2,0))$ with respect to both factors. However, for down-scaling with $s = 0.2$ and $s = 0.3$, the resampling effect leads to higher relative errors.

3.7 Rotation invariants for recognition of symmetric objects

In many applied tasks, we want to classify man-made objects or natural shapes from their silhouettes (i.e., from binary images) only. In most cases such shapes have some kind of symmetry. Different classes may have the same symmetry, which makes the task even more difficult. While humans can use the symmetry as a cue that helps them to recognize objects, moment-based classifiers suffer from the fact that some moments of symmetric shapes are zero and corresponding invariants do not provide any discrimination power. This is why we must pay attention to these situations and why it is necessary to design special invariants for each type of symmetry [19].

For example, all odd-order moments (geometric as well as complex) of a centrosymmetric object are identically equal to zero. If an object is circularly symmetric, all its complex moments, whose indices are different, vanish. Thus, Theorem 3.8 either cannot be applied at all, or many rotation invariants might be useless. Let us imagine an illustrative example. We want to recognize three shapes – a square, a cross, and a circle – independently of their orientation. Because of the symmetry, all complex moments of the second and third orders except c_{11} are zero. If the shapes are appropriately scaled, c_{11} can be the same for all of them. Consequently, neither the Hu invariants nor the basic invariants from Theorem 3.8 provide any discrimination power, even if the shapes can be readily recognized visually. Appropriate invariants in this case would be $c_{22}, c_{40}c_{04}, c_{51}c_{04}, c_{33}, c_{80}c_{04}^2, c_{62}c_{04}, c_{44}$, etc. The above simple example shows the necessity of having different systems of invariants for objects with different types of symmetry.

We have only two types of object symmetry which are relevant to consider in this context[13] – *N-fold rotation symmetry* (N-FRS) and *N-fold dihedral symmetry* (N-FDS).

Object f is said to have N-FRS if it is "rotationally periodic", that is if it repeats itself when it rotates around the origin (which is supposed to be shifted into the object centroid) by $\alpha_j = 2\pi j/N$ for all $j = 1, \ldots, N$.

[13] In the theory of point groups in 2D, it is well known that there exist only cyclic groups C_N, dihedral groups D_N, and translational symmetry groups which are irrelevant here.

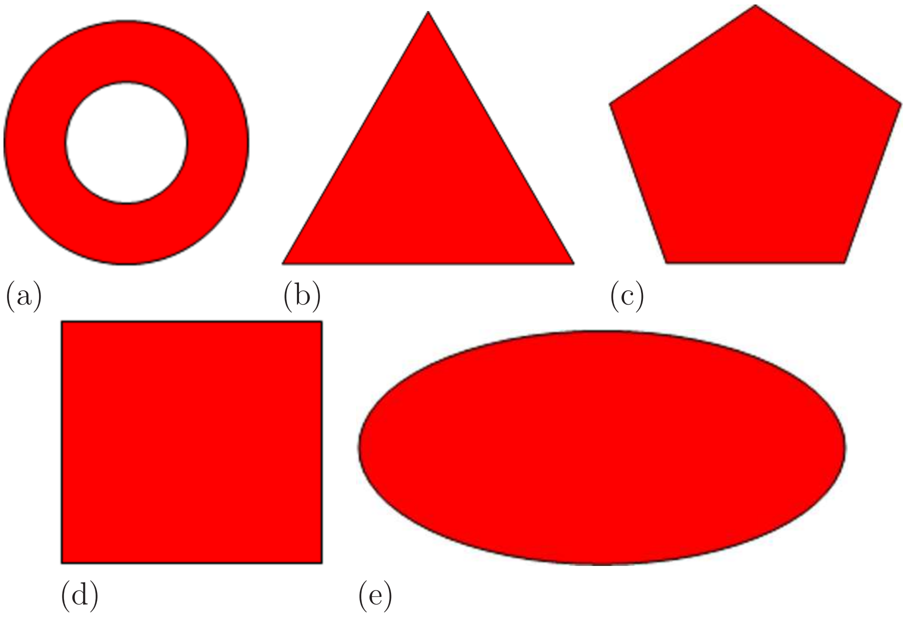


Figure 3.9: Sample objects with an N-fold rotation symmetry. From (a) to (e): $N = \infty$, 3, 5, 4, and 2, respectively. All depicted cases have also an axial symmetry; however this is not a rule.

In polar coordinates this means that

$$f(r, \theta) = f(r, \theta + \alpha_j) \qquad j = 1, \ldots, N.$$

In particular, $N = 1$ means no symmetry in a common sense, and $N = 2$ denotes the central symmetry $f(-x, -y) = f(x, y)$. We use this definition not only for finite N but also for $N = \infty$. Thus, in our terminology, the objects with a *circular symmetry* $f(r, \theta) = f(r)$ are said to have an ∞-FRS.

Object f is said to have N-FDS if it has N-FRS and at least one symmetry axis. The number of symmetry axes cannot be arbitrary – the object with N-FRS may only have N symmetry axes or no axis at all. On the other hand, if an object has N axes of symmetry ($N > 0$), then it is also rotationally symmetric and N is exactly its number of folds [20]. In case of $N = \infty$, the rotation and dihedral symmetries coincide. Examples of N-fold symmetric shapes are shown in Figure 3.9.

We show that both rotation and dihedral symmetries contribute to vanishing of some moments and invariants.

Lemma 3.9: If $f(x, y)$ has an N-fold rotation symmetry (N finite) and if $(p - q)/N$ is not an integer, then $c_{pq} = 0$.

Proof: Let us rotate f around the origin by $2\pi/N$. Due to its symmetry, the rotated object f' must be the same as the original. In particular, it must hold $c'_{pq} = c_{pq}$ for any p and q. On the other hand, it follows from eq. (3.25) that

$$c'_{pq} = e^{-2\pi i(p-q)/N} \cdot c_{pq}.$$

Since $(p - q)/N$ is assumed not to be an integer, this equation can be fulfilled only if $c_{pq} = 0$.[14] □

Lemma 3.9a: If $f(x, y)$ has ∞-fold rotation symmetry and if $p \neq q$, then $c_{pq} = 0$.

Proof: Let us rotate f around the origin by an arbitrary angle α. The rotated version f' must be the same as the original for any α, and, consequently, its moments cannot change under rotation. Eq. (3.25) implies

$$c'_{pq} = e^{-i(p-q)\alpha} \cdot c_{pq}.$$

Since $p \neq q$ and α may be arbitrary, this equation can be fulfilled only if $c_{pq} = 0$. □

The above lemma shows that the matrix of the complex moments $\mathbf{C}_{ij} = c_{i-1,j-1}$ of an N-fold symmetric object always has a specific structure – it is a sparse *multidiagonal* (and of course Hermitian) matrix with non-zero entries on the major diagonal and on the minor diagonals with the equidistant spacing of N (see Figure 3.10).

The other elements are always zero, so we have to avoid their usage as the factors in the invariants. In order to derive invariants for recognition of objects with N-fold rotation symmetry, Theorem 3.8 can be generalized in the following way.

Theorem 3.10: Let $N \geq 1$ be a finite integer. Let us consider complex moments up to the order $r \geq N$. Let a set of rotation invariants $\mathcal{B}_N$ be constructed as follows:

$$\mathcal{B}_N = \{\Phi_N(p, q) \equiv c_{pq} c_{q_0 p_0}^k | p \geq q \wedge p+q \leq r \wedge k \equiv (p-q)/N \quad \text{is an integer}\},$$

[14] This lemma can be generalized for any circular moment; the moment vanishes if $\xi(p, q)/N$ is not an integer.

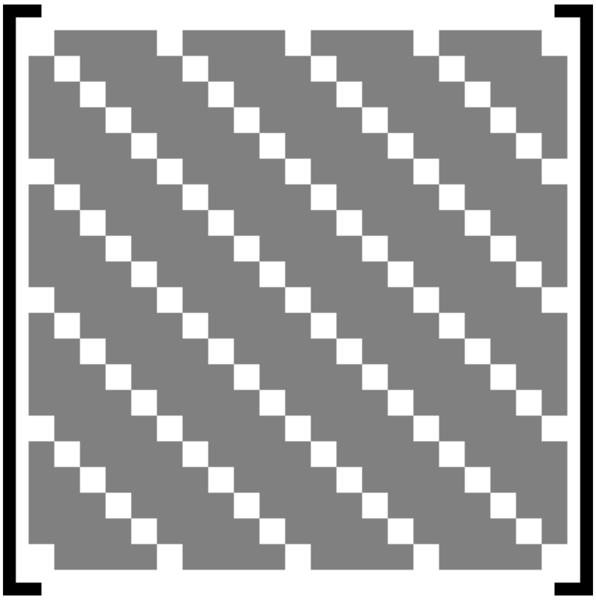


Figure 3.10: The matrix of the complex moments of an N-fold symmetric object. The gray elements are always zero. The distance between neighboring non-zero diagonals is N.

where p_0 and q_0 are arbitrary indices such that $p_0 + q_0 \leq r$, $p_0 - q_0 = N$, and $c_{p_0 q_0} \neq 0$ for all admissible objects. Then $\mathcal{B}_N$ is a basis of all non-trivial rotation invariants for objects with N-FRS, created from the moments up to the order r.

Proof: Rotation invariance of all elements of $\mathcal{B}_N$ follows immediately from Theorem 3.4. The independence and completeness of $\mathcal{B}_N$ can be proven exactly in the same way as in Theorem 3.8, when only non-zero moments are recovered. □

If $2 \leq r < N$, no such $c_{p_0 q_0}$ exists, and the basis is

$$\mathcal{B}_N = \{c_{pp} \,|\, p \leq r/2\}.$$

For $N = 1$, which means no rotation symmetry, Theorem 3.10 is reduced exactly to Theorem 3.8; $\mathcal{B}_1 = \mathcal{B}$ and $\Phi_1(p,q) = \Phi(p,q)$. The following modification of Theorem 3.10 deals with the case of $N = \infty$.

Theorem 3.10a: Let us consider complex moments up to the order $r \geq 2$. Then the basis $\mathcal{B}_\infty$ of all non-trivial rotation invariants for objects with ∞-FRS is

$$\mathcal{B}_\infty = \{c_{pp} \,|\, p \le r/2\}.$$

The proof of Theorem 3.10a follows immediately from Theorem 3.4 and Lemma 3.9a. Note that matrix $\mathbf{C}$ is diagonal for $N = \infty$.

Theorems 3.10 and 3.10a have several interesting consequences. Some of them are summarized in the following lemma.

Lemma 3.11: Let us denote all rotation invariants that can be expressed by means of elements of basis $\mathcal{B}$ as $\langle \mathcal{B} \rangle$. Then it holds for any order r

1. If M and N are finite and L is their least common multiple, then

   $$\langle \mathcal{B}_M \rangle \cap \langle \mathcal{B}_N \rangle = \langle \mathcal{B}_L \rangle.$$

   In particular, if M/N is an integer, then $\langle \mathcal{B}_M \rangle \subset \langle \mathcal{B}_N \rangle$.

2. $$\bigcap_{N=1}^{\infty} \langle \mathcal{B}_N \rangle = \langle \mathcal{B}_\infty \rangle.$$

3. If N is finite, the number of elements of $\mathcal{B}_N$ is

   $$|\mathcal{B}_N| = \sum_{j=0}^{[r/N]} \left[\frac{r - jN + 2}{2} \right],$$

   where $[a]$ means an integer part of a. For $N = \infty$ it holds

   $$|\mathcal{B}_\infty| = \left[\frac{r+2}{2} \right].$$

From the last point of Lemma 3.11 we can see that the higher the number of folds, the fewer non-trivial invariants exist. The number of them ranges from about $r^2/4$ for non-symmetric objects to about $r/2$ for circularly symmetric ones.

If an object has an N-FDS, then its moment matrix has the same multidiagonal structure, but the axial symmetry introduces additional dependence between the real and imaginary parts of each moment. The actual relationship between them depends on the orientation of the symmetry axis. If we, however, consider the basic rotation invariants from Theorem 3.10, the dependence on the axis orientation disappears; the imaginary part of each invariant is zero (we already showed this property in Section 3.5).

In practical pattern recognition experiments, the number of folds N may not be known *a priori*. In that case we can apply a fold detector (see [21, 22, 23, 24] for sample algorithms[15] detecting the number of folds) to all elements of the training set before we choose an appropriate system of moment invariants. In case of equal fold numbers of all classes, the proper invariants can be chosen directly according to Theorem 3.10 or 3.10a. Usually we start selecting from the lowest orders, and the appropriate number of the invariants is determined such that they should provide a sufficient discriminability of the training set.

If there are shape classes with different numbers of folds, the previous theory does not provide a universal solution to this problem. If we want to find such invariants which are non-trivial on *all* classes, we cannot simply choose one of the fold numbers detected as the appropriate N for constructing invariant basis according to Theorem 3.10 (although one could intuitively expect the highest number of folds to be a good choice, this is not that case). A more sophisticated choice is to take the least common multiple of all finite fold numbers and then to apply Theorem 3.10. Unfortunately, taking the least common multiple often leads to high-order instable invariants. This is why in practice one may prefer to decompose the problem into two steps – first, pre-classification into "groups of classes" according to the number of folds is performed, and then final classification is done by means of moment invariants, which are defined separately within each group. This decomposition can be performed explicitly in a separate pre-classification stage or implicitly during the classification. The word "implicitly" here means that the number of folds of an unknown object is not explicitly tested, however, at the beginning we must test the numbers of folds in the training set. We explain the latter version.

[15] These fold detectors look for the angular periodicity of the image, either directly by calculating the circular autocorrelation function and searching for its peaks or by investigating circular Fourier transformation and looking for dominant frequency. Alternatively, some methods estimate the fold number directly from the zero values of certain complex moments or other circular moments. Such approach may however be misleading in some cases because the moment values could be small even if the object is not symmetric.

Let us have C classes altogether such that C_k classes have N_k folds of symmetry; $k = 1, \ldots, K$; $N_1 > N_2 > \ldots N_K$. The set of proper invariants can be chosen as follows. Starting from the highest symmetry, we iteratively select those invariants providing (jointly with the invariants that have been already selected) a sufficient discriminability between the classes with the fold numbers N_F and higher, but which may equal zero for some (or all) other classes. Note that for some F the algorithm need not to select any invariant because the discriminability can be assured by the invariants selected previously or because $C_F = 1$.

In order to illustrate the importance of a careful choice of the invariants in pattern recognition tasks, we carried out the following experimental studies.

3.7.1 Logo recognition

In the first experiment, we tested the capability of recognizing objects having the same number of folds, particularly $N = 3$. As a test set we used three logos of major companies (Mercedes-Benz, Mitsubishi, and Fischer) and two commonly used symbols ("recycling" and "woolen product"). All logos were downloaded from the respective web-sites, resampled to 128×128 pixels and binarized. We decided to use logos as the test objects because most logos have a certain degree of symmetry. This experiment is also relevant to the state of the art because several commercial logo/trademark recognition systems reported in the literature [25, 26, 27, 28] used rotation moment invariants as features, and all of them face the problem of symmetry.

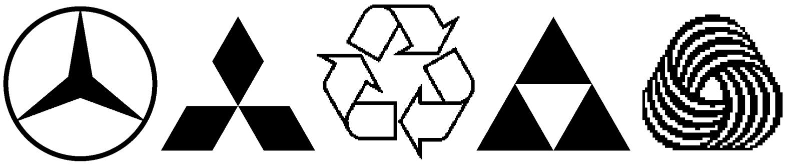


Figure 3.11: The test logos (from left to right): Mercedes-Benz, Mitsubishi, Recycling, Fischer, and Woolen product.

As can be seen in Figure 3.11, all our test logos have three-fold rotation symmetry. Each logo was rotated ten times by randomly generated angles. Since the spatial resolution of the images was relatively high, the sampling effect was insignificant. Moment invariants from Theorem 3.10

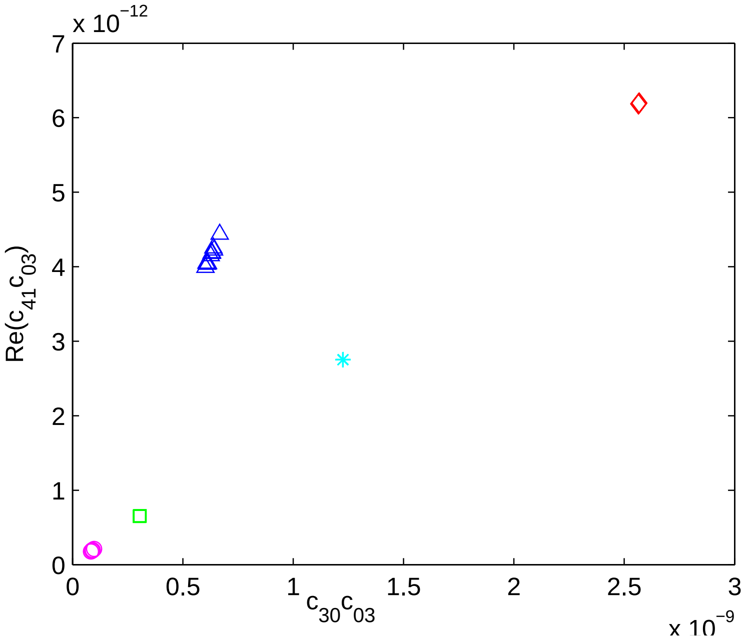


Figure 3.12: The logo positions in the space of two invariants $c_{30}c_{03}$ and $\mathcal{R}e(c_{41}c_{03})$ showing good discrimination power. The symbols: □– Mercedes-Benz, ◇– Mitsubishi, △– Recycling, ∗– Fischer, and ○– Woolen product. Each logo was randomly rotated ten times.

($N = 3$, $p_0 = 3$, and $q_0 = 0$) provide an excellent discrimination power even if only the two simplest ones have been used (see Figure 3.12), while the invariants from Theorem 3.8 are not able to distinguish the logos at all (see Figure 3.13).

3.7.2 Recognition of shapes with different fold numbers

In the second experiment, we used nine simple binary patterns with various numbers of folds: capitals F and L ($N = 1$), rectangle and diamond ($N = 2$), equilateral triangle and tripod ($N = 3$), cross ($N = 4$), and circle and ring ($N = \infty$) (see Figure 3.14). As in the previous case, each pattern was rotated ten times by random angles.

First, we applied rotation invariants according to Theorem 3.8 choosing $p_0 = 2$ and $q_0 = 1$. The positions of our test patterns in the feature space are plotted in Figure 3.15. Although only a 2-D subspace showing

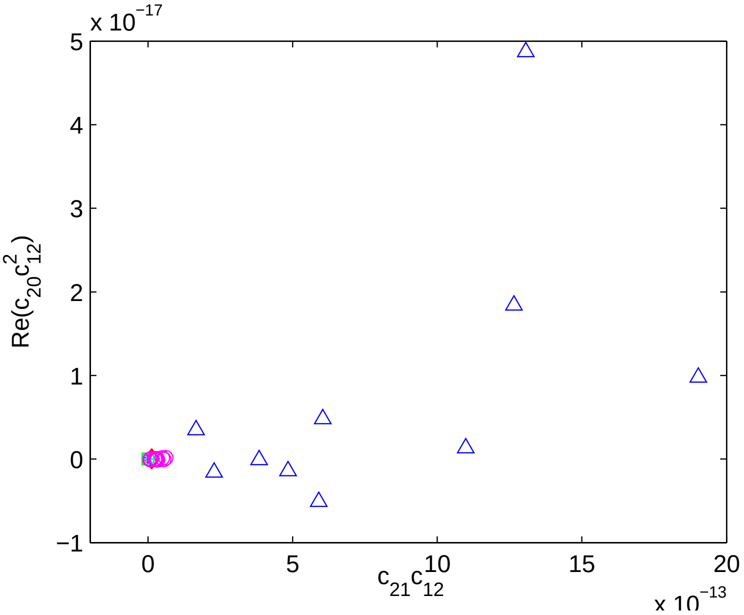


Figure 3.13: The logo positions in the space of two invariants $c_{21}c_{12}$ and $\mathcal{R}e(c_{20}c_{12}^2)$ introduced in Theorem 3.8. These invariants have no discrimination power with respect to this logo set. The symbols: □– Mercedes-Benz, ◇– Mitsubishi, △– Recycling, ∗– Fischer, and ○– Woolen product.

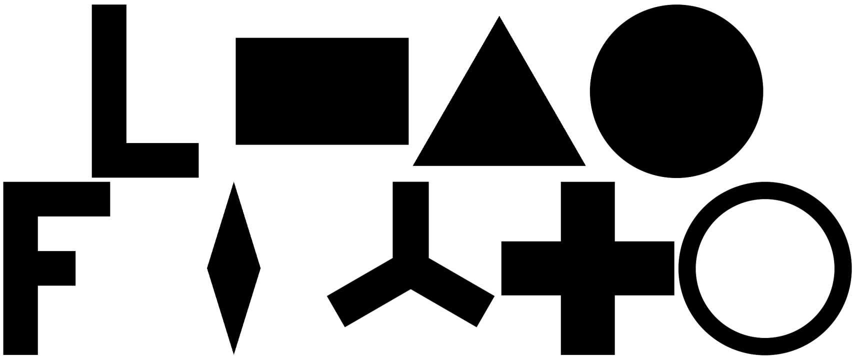


Figure 3.14: The test patterns: capital L, rectangle, equilateral triangle, circle, capital F, diamond, tripod, cross, and ring.

the invariants $c_{21}c_{12}$ and $\mathcal{R}e(c_{20}c_{12}^2)$ is visualized there, we can easily observe that the patterns form a single dense cluster around the origin (the only exception is the tripod, which is slightly biased because of its non-

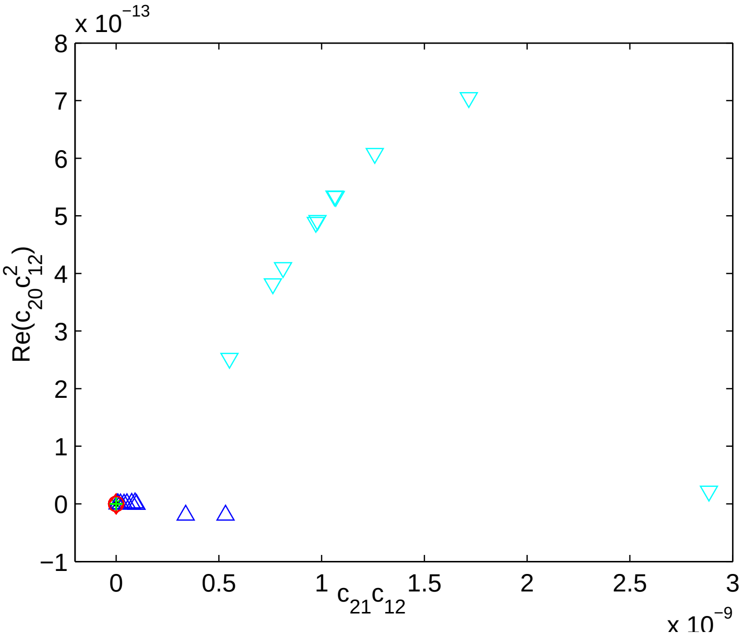


Figure 3.15: The space of two invariants $c_{21}c_{12}$ and $\mathcal{R}e(c_{20}c_{12}^2)$ introduced in Theorem 3.8. The symbols: × – rectangle, ◇ – diamond, △ – equilateral triangle, ▽ – tripod + – cross, • – circle, and ○ – ring. The discriminability is very poor.

symmetry caused by the sampling effect). Two non-symmetric objects – letters F and L – are far from the origin, beyond the displayed area. The only source of non-zero variance of the cluster are spatial sampling errors. All other invariants of the form $c_{pq}c_{12}^{p-q}$ behave in the same way. Thus, according to our theoretical expectation, we cannot discriminate among symmetric objects (even if they are very different) by means of the invariants defined in Theorem 3.8.

Second, we employed the invariants introduced in Theorem 3.10 choosing $N = 4$ (the highest finite number of folds among the test objects), $p_0 = 4$, and $q_0 = 0$ to resolve the above recognition experiment. The situation in the feature space appears to be different from the previous case (see the plot of the two simplest invariants $c_{40}c_{04}$ and $\mathcal{R}e(c_{51}c_{04})$ in Figure 3.16). Five test patterns formed their own very compact clusters that are well separated from each other. However, the patterns circle, ring, triangle, and tripod still made a mixed cluster around the origin and remained non-separable. This result is fully in accordance with the

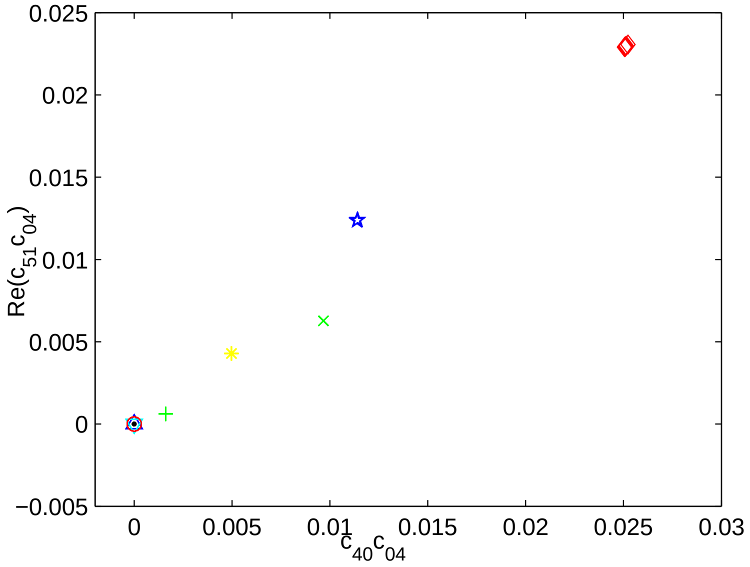


Figure 3.16: The space of two invariants $c_{40}c_{04}$ and $\mathcal{R}e(c_{51}c_{04})$ introduced in Theorem 3.10, $N = 4$. The symbols: × – rectangle, ◇ – diamond, △ – equilateral triangle, ▽ – tripod + – cross, • – circle, and ○ – ring, ✳ – capital F, and ☆ – capital L. Some clusters are well separated.

theory, because the number of folds used here is not optimal for our test set.

Third, we repeated this experiment again with invariants according to Theorem 3.10 but selecting N as the least common multiple of all finite fold numbers involved, that is, $N = 12$. One can learn from Figure 3.17 that now all clusters are well separated (because of the high dynamic range, logarithmic scale was used for visualization purposes). The only exception are two patterns having circular symmetry – the circle and the ring – that still created a mixed cluster. If also these two patterns were to be separated from one another, we could use the invariants c_{pp}. On the other hand, using *only* these invariants for the whole experiment is not a good choice from the practical point of view – since there is only one such invariant for each order, we would be pushed into using high-order noise-sensitive moments.

Finally, we used the algorithm described at the end of Section 3.7. In this case, two invariants $c_{30}c_{03}$ and $c_{40}c_{04}$ are sufficient to separate all classes (of course, again with the exception of the circle and the ring), see Figure 3.18. Compared to the previous case, note less correlation

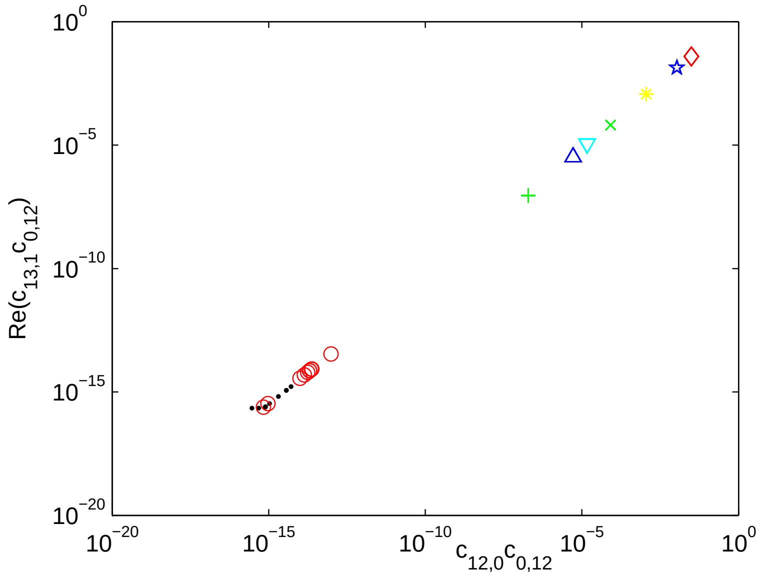


Figure 3.17: The space of two invariants $c_{12,0}c_{0,12}$ and $\mathcal{R}e(c_{13,1}c_{0,12})$ introduced in Theorem 3.10, $N = 12$ (logarithmic scale). The symbols: × − rectangle, ◇ − diamond, △ − equilateral triangle, ▽ − tripod + − cross, • − circle, and ○ − ring, ✳ − capital F, and ☆ − capital L. All clusters except the circle and the ring are separated.

of the invariants, their higher robustness, and lower dynamic range. On the other hand, neither $c_{30}c_{03}$ nor $c_{40}c_{04}$ provide enough discrimination power when used individually while the twelfth-order invariants are able to distinguish all classes.

3.7.3 Experiment with a baby toy

We demonstrate the performance of invariants in an object matching task. We used a popular baby toy (see Figure 3.19) that is also commonly used in testing computer vision algorithms and robotic systems. The toy consists of a hollow sphere with twelve holes and of twelve objects of various shapes. Each object matches up with just one particular hole. The baby (or the algorithm) is supposed to assign the objects to the corresponding holes and insert them into the sphere. The baby can employ both the color and shape information; however, in our experiment we completely disregarded the colors to make the task more challenging.

First, we binarized the pictures of the holes (one picture per each

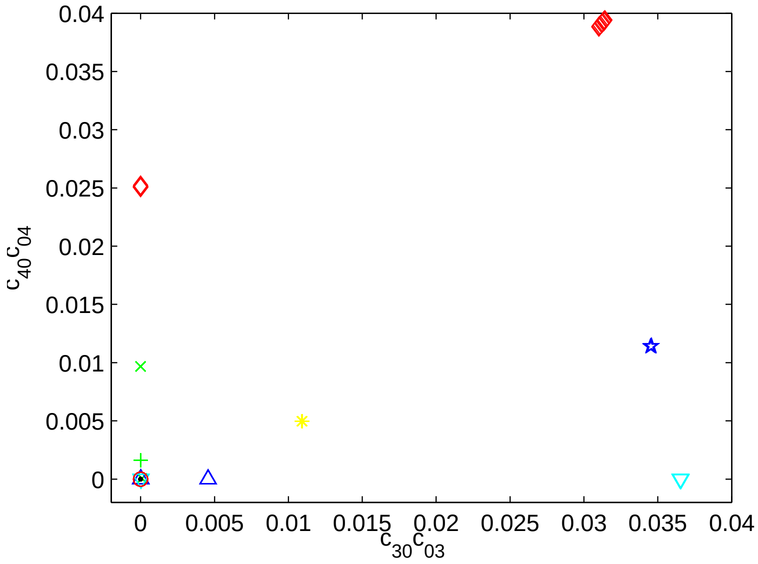


Figure 3.18: The space of two invariants $c_{30}c_{03}$ and $c_{40}c_{04}$. The symbols: × – rectangle, ◇ – diamond, △ – equilateral triangle, ▽ – tripod + – cross, • – circle, and ○ – ring, ✳ – capital F, and ☆ – capital L. All clusters except the circle and the ring are separated. Comparing to Fig. 3.17, note less correlation of the invariants and a lower dynamic range.

hole) by simple thresholding. Binarization was the only pre-processing; no sphere-to-plane corrections were applied.

To select proper invariants, we applied the algorithm from Section 3.7 on the images of the holes. As a discriminability measure we took weighted Euclidean distance, where the weights were set up to normalize the dynamic range of the invariants. As one can observe, the highest finite number of folds is 6. The algorithm terminated after passing three loops and selected the three following invariants: $c_{60}c_{06}$, $c_{50}c_{05}$, and $c_{40}c_{04}$.

Then we took ten pictures of each object with random rotations, binarized them, and run the classification. This task is not so easy as it might appear because the holes are a bit larger than the objects, but this relation is morphological rather than linear and does not preserve the shapes exactly. Fortunately, all 120 unknown objects were recognized correctly and assigned to proper holes. It should be emphasized that only three invariants without any other information yielded 100% recognition rate for twelve classes, which is a very good result even though the shapes are relatively simple.

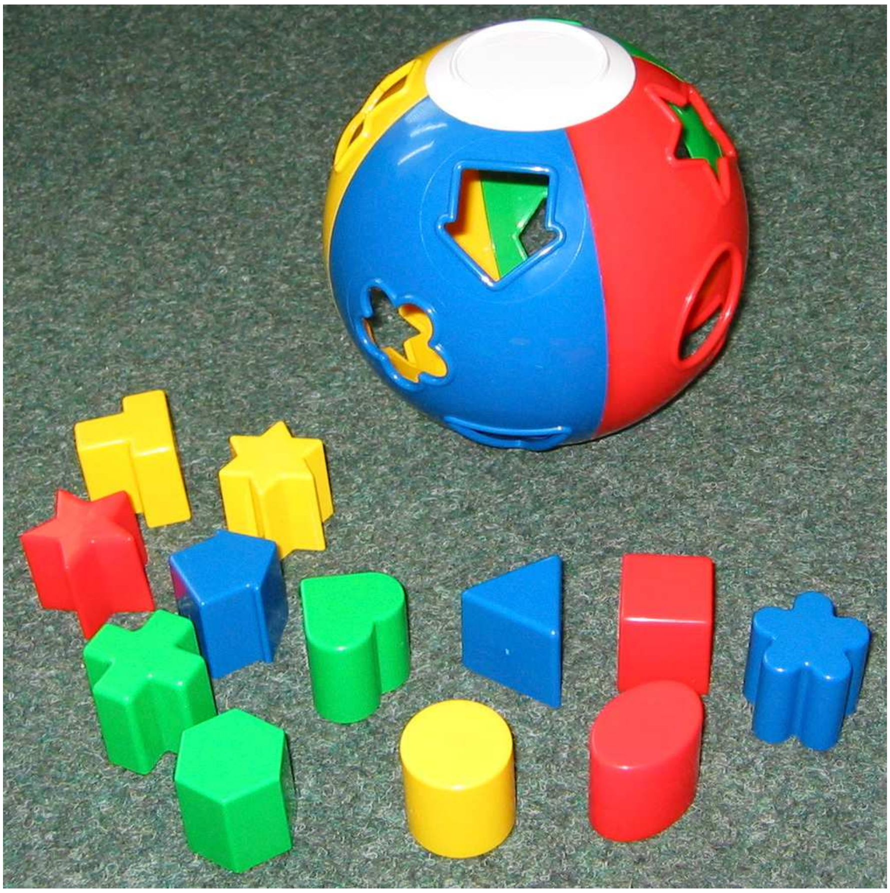


Figure 3.19: The toy set used in the experiment.

We repeated the classification once again with invariants constructed according to Theorem 3.8 setting $p_0 = 2, q_0 = 1$. Only three objects having 1-fold symmetry were recognized correctly, the others were classified randomly.

3.8 Rotation invariants via image normalization

The TRS invariants can also be derived in another way than by means of Theorem 3.8. This alternative method called *image normalization* first brings the image into some "standard" or "canonical" position, which is defined by setting certain image moments to pre-defined values (usually

0 or 1). Then the moments of the normalized image (except those that have been used to define the normalization constraints) can be used as TRS invariants of the original image. It should be noted that no actual geometric transformation and resampling of the image is required; the moments of the normalized image can be calculated directly from the original image by means of normalization constraints.

While normalization to translation and scaling is trivial by setting $m'_{10} = m'_{01} = 0$ and $m'_{00} = 1$, there are many possibilities of defining the normalization constraints to rotation. The traditional one known as the *principal axis* method [3] requires for the normalized image the constraint $\mu'_{11} = 0$.[16] It leads to diagonalization of the second-order moment matrix

$$\mathbf{M} = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix}. \tag{3.35}$$

It is well-known that any symmetric matrix can be diagonalized on the basis of its (orthogonal) eigenvectors. After finding the eigenvalues of $\mathbf{M}$ (they are guaranteed to be real) by solving

$$|\mathbf{M} - \lambda \mathbf{I}| = 0,$$

we obtain the diagonal form

$$\mathbf{M}' = \mathbf{G}^T \mathbf{M} \mathbf{G} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} = \begin{pmatrix} \mu'_{20} & 0 \\ 0 & \mu'_{02} \end{pmatrix},$$

where $\mathbf{G}$ is an orthogonal matrix composed of the eigenvectors of $\mathbf{M}$. It is easy to show that we obtain for the eigenvalues of $\mathbf{M}$

$$\mu'_{20} \equiv \lambda_1 = ((\mu_{20} + \mu_{02}) + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})/2,$$

$$\mu'_{02} \equiv \lambda_2 = ((\mu_{20} + \mu_{02}) - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})/2,$$

and that the angle between the first eigenvector and the x-axis, which is actually the normalization angle, is [17]

$$\alpha = \frac{1}{2} \cdot \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right).$$

[16] The reader may notice a clear analogy with the principal component analysis.

[17] Some authors define the normalization angle with an opposite sign. Both ways are, however, equivalent – we may “rotate” either the image or the coordinates.

Provided that $\lambda_1 > \lambda_2$, there are four different normalization angles which all make $\mu'_{11} = 0$: $\alpha, \alpha + \pi/2, \alpha + \pi$, and $\alpha + 3\pi/2$. To remove the ambiguity between "horizontal" and "vertical" position, we impose the constraint $\mu'_{20} > \mu'_{02}$. Removing the "left-right" ambiguity can be achieved by introducing an additional constraint $\mu'_{30} > 0$ (provided that μ'_{30} is nonzero; a higher-order constraint must be used otherwise).

If $\lambda_1 = \lambda_2$, then *any* nonzero vector is an eigenvector of $\mathbf{M}$, and the principal axis method would yield infinitely many solutions. This is possible only if $\mu_{11} = 0$ and $\mu_{20} = \mu_{02}$; the principal axis normalization cannot be used in that case.

Now each moment of the normalized image μ'_{pq} is a rotation invariant of the original image. For the second-order moments, we can see clear correspondence with the Hu invariants

$$\mu'_{20} = (\phi_1 + \sqrt{\phi_2})/2,$$
$$\mu'_{02} = (\phi_1 - \sqrt{\phi_2})/2.$$

The principal axis normalization is linked with a notion of the *reference ellipse* (see Figure 3.20). There exist two concepts in the literature[18]. The first one, which is suitable namely for binary images, requires the ellipse to have a unit intensity. The other one, which is more general and more convenient for graylevel images, requires a constant-intensity ellipse, but its intensity may be arbitrary. The former approach allows to constraining the first- and the second-order moments of the ellipse such that they equal the moments of the image, but the zero-order moment of the ellipse is generally different from that of the image.

After the principal axis normalization, the reference ellipse is in axial position, and we have the following constraints for its moments,

$$\mu_{11}^{(e)} = 0,$$
$$\mu_{20}^{(e)} \equiv \frac{\pi a^3 b}{4} = \mu'_{20},$$
$$\mu_{02}^{(e)} \equiv \frac{\pi a b^3}{4} = \mu'_{02},$$

where a and b are the length of the major and minor semiaxis, respectively. By the last two constraints, a and b are unambiguously determined.

[18] Whenever one uses the reference ellipse, it must be specified which definition has been applied.

If we apply the latter definition of the reference ellipse, the constraints also contain the ellipse intensity ℓ, which allows us to constraint also the zero-order moment

$$\begin{aligned}\mu_{00}^{(e)} &\equiv \pi\ell ab = \mu'_{00},\\ \mu_{11}^{(e)} &= 0,\\ \mu_{20}^{(e)} &\equiv \frac{\pi\ell a^3 b}{4} = \mu'_{20},\\ \mu_{02}^{(e)} &\equiv \frac{\pi\ell ab^3}{4} = \mu'_{02}.\end{aligned}$$

These constraints allow an unambiguous calculation of a, b, and ℓ.

All images having the same second-order moments have identical reference ellipses. This can also be understood from the opposite side – knowing the second-order image moments, we can reconstruct the reference ellipse only, without any finer image details.

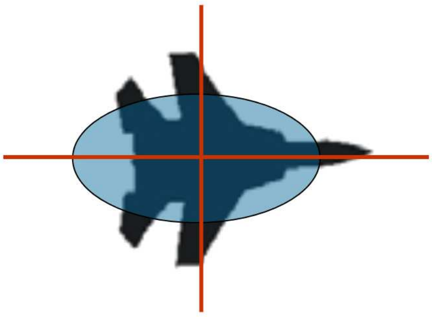


Figure 3.20: Principal axis normalization to rotation – an object in the normalized position along with its reference ellipse superimposed.

Mostafa and Psaltis [29] proposed a general normalization scheme by means of the complex moments. Their normalization constraint requires one complex moment c_{st}, where $s \neq t$, of the image to be real and positive (we could of course use any other well-defined orientation, but this one was chosen for simplicity). As follows from Eq. (3.25), this is always possible to achieve provided that $c_{st} \neq 0$. Then the constraint leads to the normalization angle

$$\alpha = \frac{1}{s-t} \cdot \arctan\left(\frac{\mathcal{I}m(c_{st})}{\mathcal{R}e(c_{st})}\right). \qquad (3.36)$$

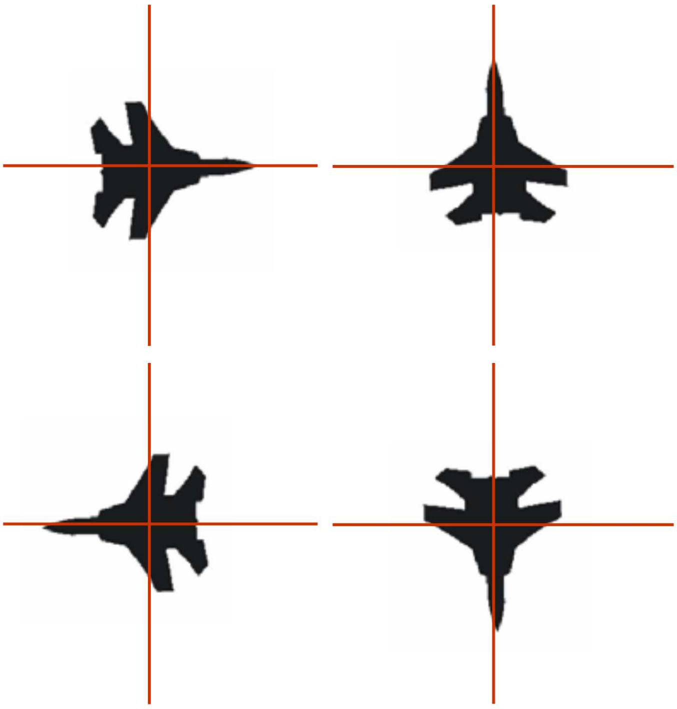


Figure 3.21: Ambiguity of the principal axis normalization. These four positions of the object satisfy $\mu'_{11} = 0$. Additional constraints $\mu'_{20} > \mu'_{02}$ and $\mu'_{30} > 0$ make the normalization unique.

There exist exactly $s-t$ distinct angles satisfying (3.36), see Figure 3.22. This normalization is unambiguous only if $s - t = 1$. For $s - t > 1$, additional constraints are required to remove the ambiguity. If we choose $s = 2$ and $t = 0$, the constraint (3.36) brings us back to the principal axis normalization.

The selection of the normalizing moment c_{st} is a critical step, and in practice it must be done very carefully. If we calculated the normalization angle with an error, this error would propagate and would affect all invariants. Even a very small error in the normalizing angle can influence some invariants significantly. The non-zero constraint is easy to check in a continuous domain, but in a discrete case some moments that are in fact zero due to the object symmetry may appear in certain positions of the object as nonzero because of the sampling effect. We have to identify such moments and avoid their usage in the normalization. At the same time, we want to keep the moment order $s+t$ as low as possible in order

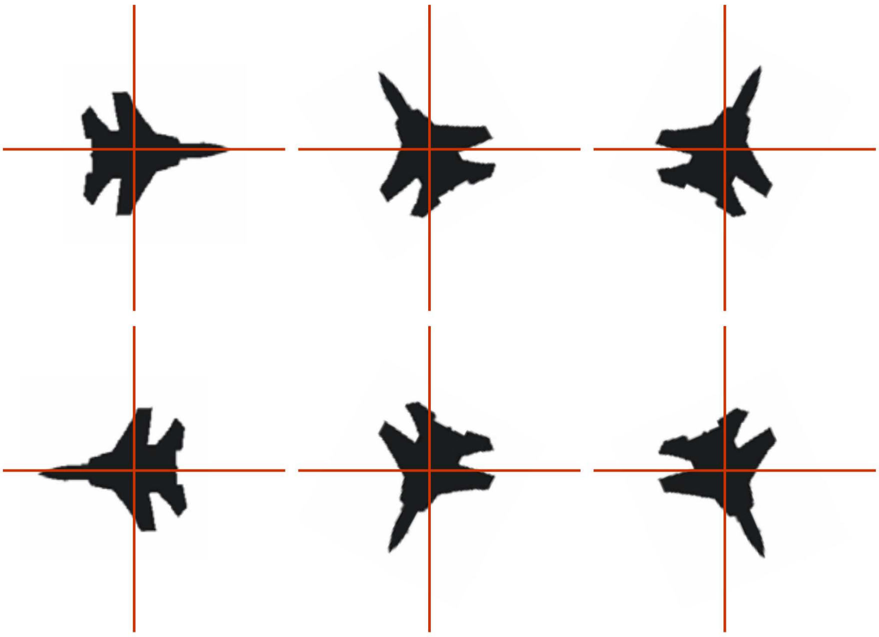


Figure 3.22: An example of the ambiguity of the normalization by complex moments. In all these six positions, c'_{60} is real and positive as required.

to make the normalization robust.

The normalizing moment can be found as follows. We sort all complex moments (except those that were already used for translation and scaling normalization) up to the given order r according to the index difference $p-q$ and those moments with the same index differences we sort according to their order. We obtain an ordered sequence c_{21}, c_{32}, c_{43}, $\ldots$, c_{20}, c_{31}, c_{42}, c_{53}, $\ldots$, etc. The first nonzero moment in this sequence is then used for normalization. To test if the moment is nonzero, we compare its value with a user-defined threshold. The choice of the threshold must reflect the magnitudes of the moments in question, and, especially when considering high r, it may be useful to define different thresholds for different moment orders. If all moments in the sequence are zero, we consider the object circularly symmetric, and no normalization is necessary.

One can, of course, use another ordering scheme when looking for the normalizing moment. An alternative is to sort the moments in the other way round – first according to their orders and the moments of the same order according to the index differences. Both methods have certain advantages and drawbacks. For non-symmetric objects there is no difference because we always choose c_{21}. For symmetric and close-to-symmetric objects, when we sort the moments according to their orders first, then the order of the chosen moment is kept as low as possible. This is a favorable property because low-order moments are more robust to noise than higher-order ones. On the other hand, one can find objects where this approach fails. Sorting moments according to the index difference first generally leads to a higher-order normalization constraint that is numerically less stable, but from the theoretical point of view this works for any object.

There have been many discussions on the advantages and drawbacks of both approaches – moment invariants and image normalization. Here we show that from the theoretical point of view they are fully equivalent and that the normalizing moment c_{st} and the basic moment $c_{p_0q_0}$ play very similar roles.

Let us consider the basis of the rotation invariants designed according to Theorem 3.8. We can apply the Mostafa and Psaltis normalization method with $c_{st} = c_{p_0q_0}$. Since each $\Phi(p,q)$ is a rotation invariant, it holds

$$\Phi(p,q) = c'_{pq} c'^{p-q}_{q_0p_0},$$

where c'_{pq} denotes the moments of the normalized image. Consequently, since $c'_{q_0p_0}$ is real and positive, we get

$$\Phi(p,q) = c'_{pq} |c_{p_0q_0}|^{p-q}. \tag{3.37}$$

Eq. (3.37) shows that there is a one-to-one mapping between moment invariants $\Phi(p,q)$ of the original image and complex moments c'_{pq} of the normalized image. An analogous result can be obtained for symmetric objects and the invariants from Theorem 3.10. Thus, normalization approach and invariant-based approach are equivalent on the theoretical level. Computational issues in particular applications might be, however, different.

3.9 Moment invariants of vector fields

A vector-valued image $\mathbf{f}(\mathbf{x})$ can be viewed as a set of m scalar images $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$. Depending on its character, there may be various relationships among individual components f_k, which influence the behavior of $\mathbf{f}$ under spatial transformations and hence influence the design of the invariants. Examples of vector-valued images are conventional color/multispectral images and *vector fields*. The former are not very difficult to handle because their components (even if they are usually correlated) can be treated as independent graylevel images, which allows us to apply the rotation invariants derived earlier in this chapter separately to each component. However, the vector fields behave differently and require a different treatment.

A 2D vector field is a vector-valued image $\mathbf{f}(\mathbf{x})$ with $d = m = 2$. Typical examples of vector fields are gradient fields, particle velocity fields, optical flow fields, fluid flow fields, and others, see Figures 3.23, 3.24, 3.25 and 3.27. A common feature is that at each point (x, y), the value of $\mathbf{f}(x, y)$ shows the orientation and the magnitude of certain vector.

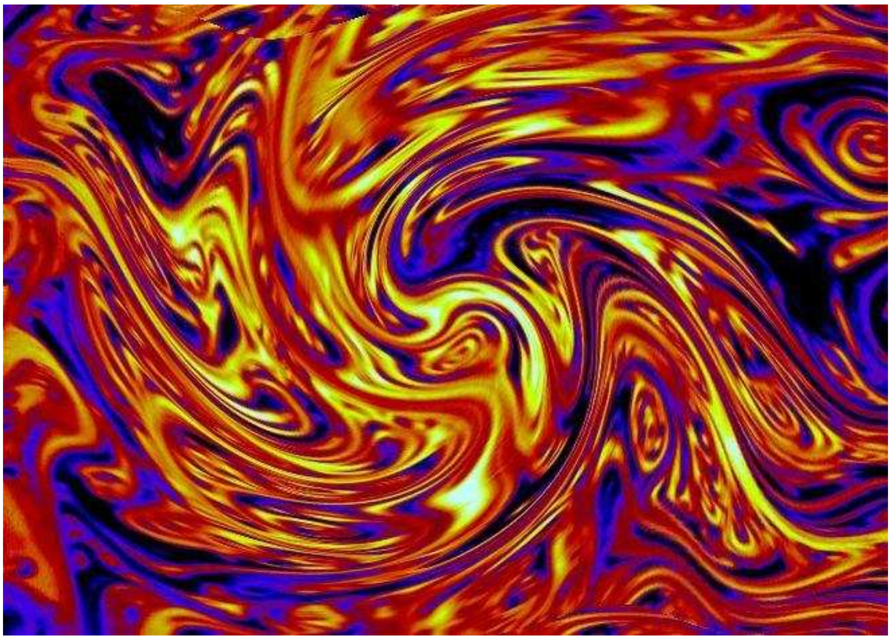


Figure 3.23: Turbulence in a fluid (the colors show the local velocity of the flow; the direction is not displayed).

If a 2D vector-valued image $\mathbf{f}(\mathbf{x})$ is rotated, we have to distinguish four kinds of rotation. *Inner rotation* is a traditional rotation known from scalar images

$$\mathbf{f}'(\mathbf{x}) = \mathbf{f}(\mathbf{R}_{-\alpha}\mathbf{x}).$$

Outer rotation does not affect the spatial coordinates but acts on the

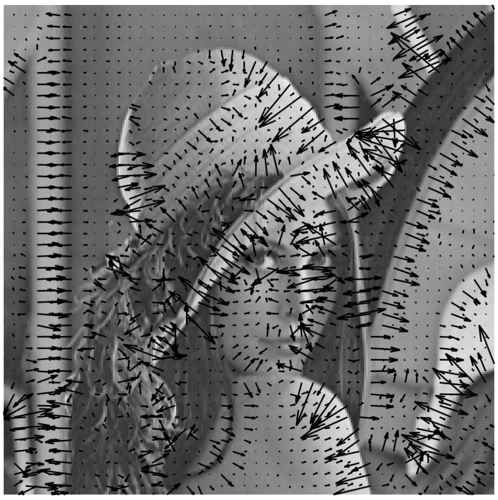


Figure 3.24: Image gradient as a vector field. For visualization purposes, the field is depicted by arrows on a sparse grid and laid over the original Lena image.

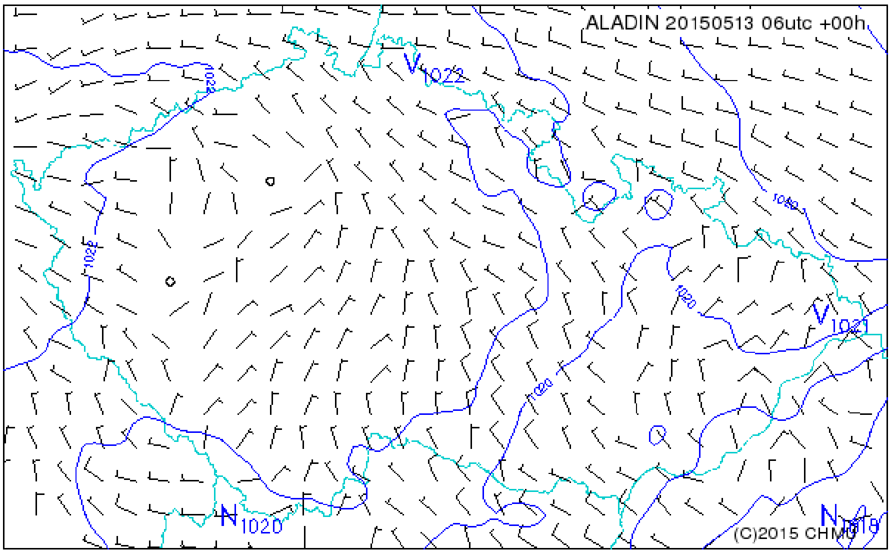


Figure 3.25: The wind velocity forecast for the Czech Republic (courtesy of the Czech Hydrometeorological Institute, the numerical model Aladin). The longer dash is constant part of the wind, the shorter dash expresses perpendicular squalls. The figure actually represents two vector fields.

vector values only

$$\mathbf{f}'(\mathbf{x}) = \mathbf{R}_\alpha \mathbf{f}(\mathbf{x}).$$

It appears for instance in color images when the color space rotates but the picture itself does not move. *Total rotation* is a combination of both inner and outer rotations of the same rotation matrices

$$\mathbf{f}'(\mathbf{x}) = \mathbf{R}_\alpha \mathbf{f}(\mathbf{R}_{-\alpha}\mathbf{x}).$$

Finally, the most general *independent total rotation* is a total rotation for which the outer and inner rotation matrices are different

$$\mathbf{f}'(\mathbf{x}) = \mathbf{R}_\alpha \mathbf{f}(\mathbf{R}_\beta \mathbf{x}).$$

Let us illustrate these terms in Figure 3.26 for $\alpha = 22.5^\circ$. In the case of the inner rotation, each arrow sustains its direction, but it is rotated around the image center to the new position. In the outer rotation, each arrow sustains its position, but its direction is rotated by 22.5°. In the total rotation, each arrow is rotated around the image center to the new position, and its direction is also rotated by the same angle.

Traditional spatial-domain rotation of a color image is inner rotation since the colors do not change. Rotation of a vector field is typically a total rotation because if a vector field is rotated in the coordinate space, then its values must be rotated accordingly such that the vector directions are preserved relative to the image content. An independent total rotation may appear when spatially rotating a color image at the same time that we independently rotate the color space.

Analogously to various kinds of rotations, we recognize *inner scaling* $\mathbf{f}'(\mathbf{x}) = \mathbf{f}(\mathbf{x}/s)$, *outer scaling* $\mathbf{f}'(\mathbf{x}) = s\mathbf{f}(\mathbf{x})$, *total scaling* $\mathbf{f}'(\mathbf{x}) = s\mathbf{f}(\mathbf{x}/s)$, and *independent total scaling* $\mathbf{f}'(\mathbf{x}) = s_1\mathbf{f}(\mathbf{x}/s_2)$. For vector fields, typical scaling in practice is the total scaling. We could also analogously define an inner, outer, total, and independent total translations, but it does not make much sense in this context because the translation of a vector field in practice is always the inner one.

Although some vector field images may arrive as the results of image or video processing, such as gradients and optical flow images, we meet vector field images more often in visualization of fluid flow and of particle motion. These images may come from the solution of Navier-Stokes equations as well as from real measurements and may show, for instance, flowing water in a pipe or an air flow around an aircraft wing or around a coachwork. For engineers and designers, it is very important

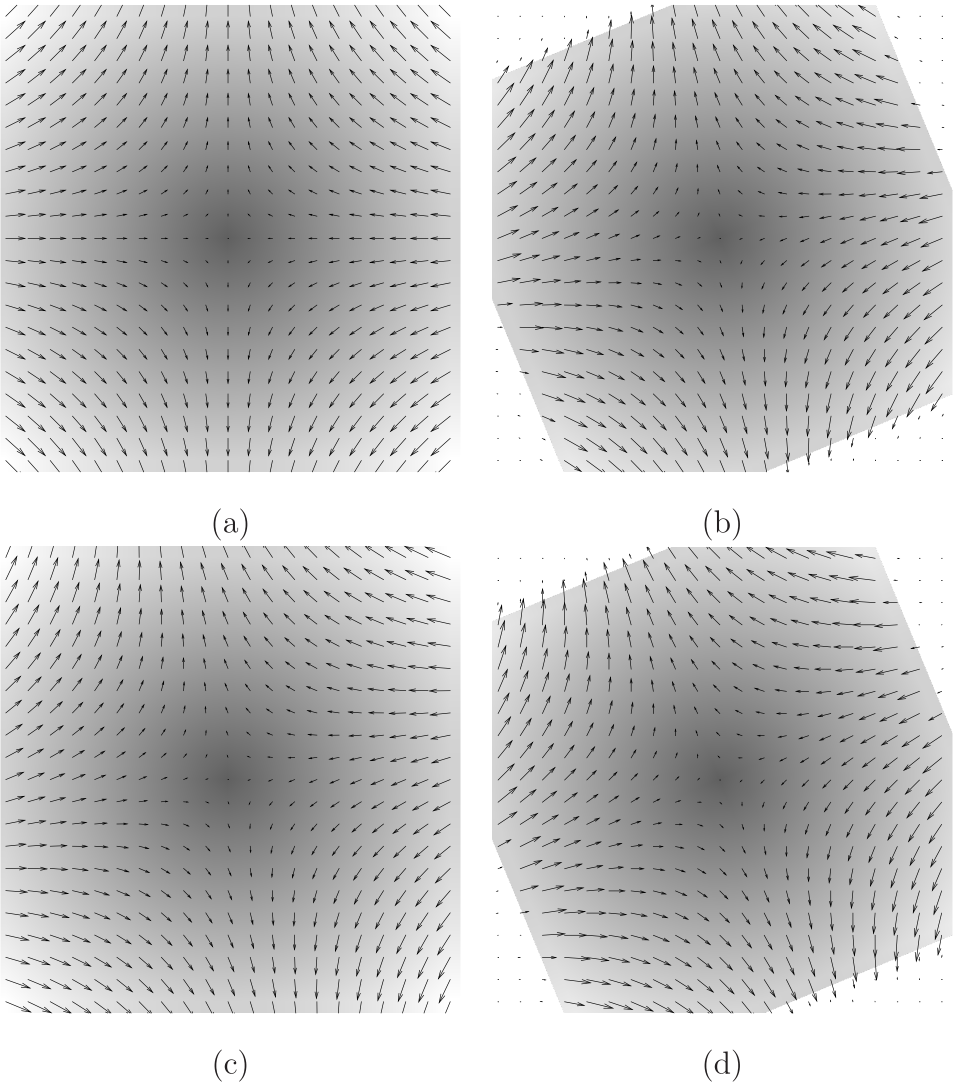


Figure 3.26: Various rotations of a vector field: (a) the original vector field, (b) the inner rotation, (c) the outer rotation, (d) the total rotation. The graylevels corresponds to the vector sizes.

to identify singularities in the flow such as sinks and vortexes because they increase the friction, decrease the speed of the medium, and consequently increase the power and cost which is necessary to transport the medium through the pipe or the object through the air or water. Searching for the singularities is a typical template matching problem – we have a database of template singularities, and we look for these patterns in the vector field. Since the orientation of the template is unknown and irrelevant, we cannot just correlate the templates but need invariants to total rotation.

This problem was addressed for the first time by Schlemmer et al. [30] who realized that the invariants from complex moments can be used here in a similar way as for the graylevel images. If $d = m = 2$, we can treat the vector field as a field of complex numbers

$$\mathbf{f}(x, y) = f_1(x, y) + i f_2(x, y)$$

which allows us to use the standard definition of complex moments. It holds

$$c_{pq}^{(\mathbf{f})} = c_{pq}^{(f_1)} + i c_{pq}^{(f_2)} \ .$$

Unlike the moments of a scalar image, it holds for the vector fields moments in general

$$c_{pq}^{(\mathbf{f})} \neq c_{qp}^{(\mathbf{f})*}.$$

Now let us investigate how $c_{pq}^{(\mathbf{f})}$ is changed under a total rotation by angle α

$$c_{pq}^{(\mathbf{f}')} = e^{-i\alpha} e^{-i(p-q)\alpha} \cdot c_{pq}^{(\mathbf{f})} = e^{-i(p-q+1)\alpha} \cdot c_{pq}^{(\mathbf{f})}. \tag{3.38}$$

By a multiplication of proper powers we can cancel the rotation parameter. Let $\ell \geq 1$ and let $k_i, p_i \geq 0$, and $q_i \geq 0$ $(i = 1, \dots, \ell)$ be arbitrary integers such that

$$\sum_{i=1}^{\ell} k_i (p_i - q_i + 1) = 0.$$

Then

$$I = \prod_{i=1}^{\ell} c_{p_i q_i}^{k_i} \tag{3.39}$$

is invariant to total rotation. To obtain a basis, we apply a construction analogous to that one we used for graylevel images. We set $p_0 - q_0 = 2$ to get the desired effect[19]. The basis invariants for vector fields then have the form

$$\Phi(p, q) \equiv c_{pq} c_{q_0 p_0}^{p-q+1} \ . \tag{3.40}$$

[19] Other choices of p_0, q_0 are also possible, we may for instance choose $p_0 = q_0$. The most general solution has been proposed by Bujack [31], who uses any indices such that $p_0 - q_0 \neq 1$. This choice is the most flexible one. It allows avoiding the use of close-to-zero moments, but leads to rational exponents of $c_{p_0 q_0}$.

There are, however, slight differences from the basis of the scalar invariants. In the case of vector fields, it is meaningful to consider also such $\Phi(p, q)$ where $p < q$ because they are independent. $\Phi(q_0, p_0) = 1$ and should be discarded from the basis. The invariants (3.40) are not complete because $|c_{q_0,p_0}|$ is another independent invariant. If we incorporate it, we obtain a complete system. Note that always $\Phi(p, p+1) = c_{p,p+1}$.

Under a total scaling, the complex moment of a vector field changes as

$$c_{pq}^{(\mathbf{f}')} = s^{(p+q+3)} c_{pq}^{(\mathbf{f})} \tag{3.41}$$

which means we can achieve the scaling invariance when normalizing by means of $c_{00}^{(p+q+3)/3}$.

Note that for vector fields, c_{00} is not invariant w.r.t. a total rotation. To combine total scaling and rotation invariance, it cannot be used as a sole normalization factor because such normalization would violate the rotation invariance. Instead, we have to use some rotation invariant for normalization. We can, for instance, normalize each moment by $(c_{00}c_{02})^{(p+q+3)/8}$, which preserves the rotation invariance after normalization and yields the invariance to total scaling and rotation.

Schlemmer [30] used a few invariants of the lowest orders of the type (3.40) (without an explicit mention of this general formula) to detect specific patterns in a turbulent swirling jet flow. He also noticed a symmetry problem[20] similar to that one we discussed in the Section 2.6. Specific templates we search for are often N-fold symmetric with respect to total rotation. Hence, any vector field moment c_{pq} such that $(p - q + 1)/N$ is not an integer vanishes. We have to take this into account when choosing the basis of the invariants, and we should avoid vanishing invariants. Similarly to the scalar case, the *total dihedral symmetry* of the template makes all imaginary parts of the rotation invariants zero (note that the total axial symmetry is defined as $\mathbf{f}(x, -y) = \mathbf{f}^*(x, y)$).

We present here a very simple experiment, which demonstrates the basic property of rotation invariance of the above-mentioned features. We detected optical flow in a video sequence (see Figure 3.27a) by the method described in [32], using the authors' code [33]. Then we calculated five Schlemmer's invariants c_{01}, $c_{00}c_{02}$, $c_{11}c_{02}$, $c_{10}c_{02}^2$, and $c_{20}c_{02}^3$ of the optical flow vector field. Thereafter we rotated the flow field by 90°

[20] Schlemmer mentioned this issue for the second-order moments only; we describe it here in a general form.

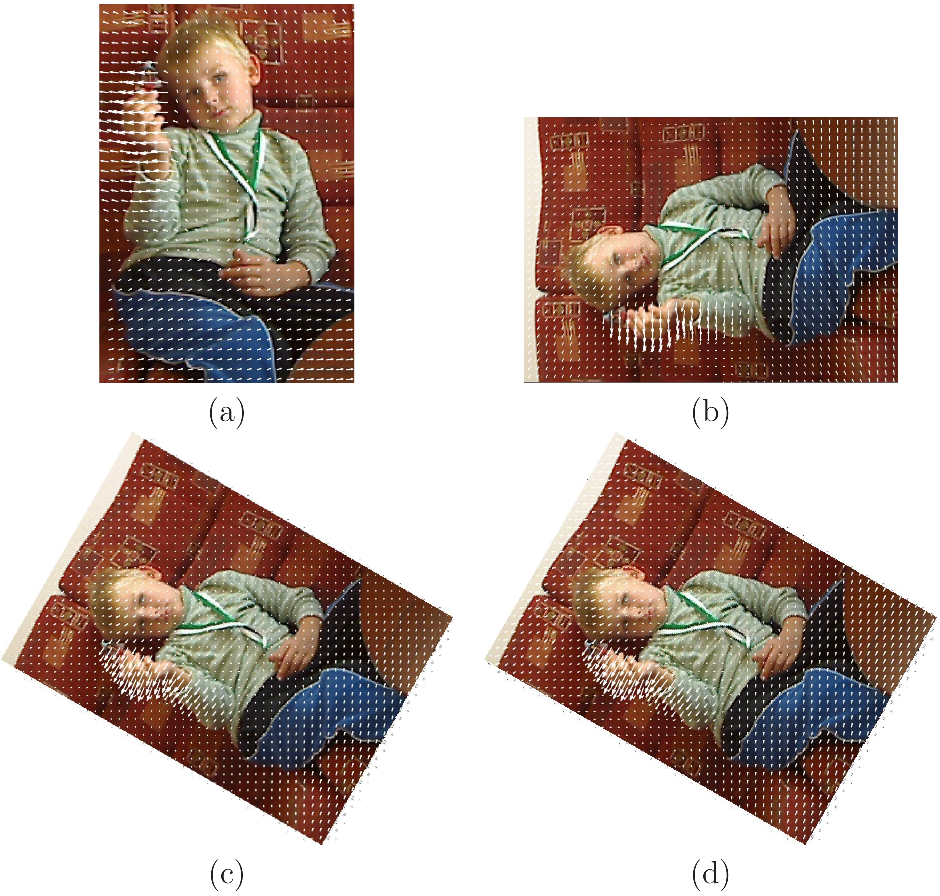


Figure 3.27: Optical flow as a vector field: (a) the original field, (b) the optical flow computed from the video sequence rotated by 90°, (c) the original optical flow field after total rotation by 60°, (d) the optical flow computed from the video sequence rotated by 60°. All rotations are counterclockwise. The arrows show the direction and velocity of the movement between two consecutive frames of the video sequence.

(this angle was chosen to avoid resampling errors) and calculated the invariants again. The average relative error was $2 \cdot 10^{-14}\%$, which shows that the only source of errors is numerical inaccuracy. Another comparison was done such that we rotated the video sequence, then we detected the optical flow again (see Figure 3.27b – there are no visible differences between it and the rotated vector field from previous case) and calculated the invariants. This scenario corresponds to the real situations, and one can expect bigger errors since the two vector fields may be generally different if the optical flow detection is not perfectly isotropic. The average relative error was 2.79% which is still very good.

Finally, we tested the influence of resampling of the field when it has been rotated by 60° (see Figure 3.27c). The average relative error was only 0.0025%. On the other hand, if these two factors – image resampling and independent optical flow detection – are combined together, the relative error of the invariants may range from 10 to 100% depending on the image texture. In our case it was 46.6%, the corresponding vector field is in Figure 3.27d.

The idea of vector field invariants has found several applications. Liu and Ribeiro [34] used it, along with a local approximation of the vector field by a polynomial, to detect singularities on meteorological satellite images where the respective field was a wind velocity map. Basically the same kind of rotation invariants was used by Liu and Yap [35] for indexing and recognition of fingerprint images. The respective vector field was a field of local orientation of the fingerprint ridges and valleys; the directional information was discarded (Liu and Yap call this the *orientation field*).

Similarly to scalar invariants, the vector field invariants can also be alternatively derived via normalization of the vector field w.r.t. total rotation and scaling. This approach has been developed by Bujack et al. [36, 37], who proposed the normalization constraints based on the zero and second-order moments. The authors demonstrated the usage of the normalized moments in template matching, where the template vortexes were searched in the image showing the *Karman vortex street* simulation[21]. They also applied the normalized moments to a segmentation of vector fields via clustering [38].

3.10 Conclusion

In this chapter, we introduced 2D moment invariants with respect to the translation, rotation and scaling. We described a general theory showing how to generate invariants of any orders. We defined the basis of the invariants as the smallest complete subset, we showed its importance and described an explicit method how to find it. As a consequence of this theory, we proved that some traditional invariant sets are dependent

[21] In fluid dynamics, the Karman vortex street is a repeating pattern of swirling vortexes caused by the flow of a fluid around blunt bodies.

and/or incomplete, which explains certain failures reported in the literature. It was shown that the proposed invariants outperform the widely used Hu moment invariants both in discrimination power and dimensionality requirements.

Furthermore, we discussed the difficulties with recognition of symmetric objects, where some moment invariants vanish, and we presented moment invariants suitable for such cases. We briefly reviewed an alternative approach to constructing invariants via normalization and showed that both methods are theoretically equivalent.

At the end, we addressed a recent development on the field of moment invariants – rotation and scale invariants of vector fields.

References

[1] Z. Feng, L. Shang-Qian, W. Da-Bao, and G. Wei, “Aircraft recognition in infrared image using wavelet moment invariants,” *Image and Vision Computing*, vol. 27, no. 4, pp. 313–318, 2009.

[2] S. Dominguez, “Image analysis by moment invariants using a set of step-like basis functions,” *Pattern Recognition Letters*, vol. 34, no. 16, pp. 2065–2070, 2013.

[3] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[4] F. Pan and M. Keane, “A new set of moment invariants for handwritten numeral recognition,” in *Proceedings of the International Conference on Image Processing ICIP'94*, pp. 154–158, IEEE, 1994.

[5] L. Jin and Z. Tianxu, “Fast algorithm for generation of moment invariants,” *Pattern Recognition*, vol. 37, no. 8, pp. 1745–1756, 2004.

[6] M. R. Teague, “Image analysis via the general theory of moments,” *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[7] Å. Wallin and O. Kübler, “Complete sets of complex Zernike moment invariants and the role of the pseudoinvariants,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1106–1110, 1995.

[8] Y. Li, “Reforming the theory of invariant moments for pattern recognition,” *Pattern Recognition*, vol. 25, no. 7, pp. 723–730, 1992.

[9] W.-H. Wong, W.-C. Siu, and K.-M. Lam, “Generation of moment invariants and their uses for character recognition,” *Pattern Recognition Letters*, vol. 16, no. 2, pp. 115–123, 1995.

[10] Y. S. Abu-Mostafa and D. Psaltis, “Recognitive aspects of moment invariants,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 698–706, 1984.

[11] J. Flusser, “On the independence of rotation moment invariants,” *Pattern Recognition*, vol. 33, no. 9, pp. 1405–1410, 2000.

[12] J. Flusser, “On the inverse problem of rotation moment invariants,” *Pattern Recognition*, vol. 35, no. 12, pp. 3015–3017, 2002.

[13] S. Derrode and F. Ghorbel, “Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description,” *Computer Vision and Image Understanding*, vol. 83, no. 1, pp. 57–78, 2001.

[14] B. Yang, J. Flusser, and T. Suk, “Design of high-order rotation invariants from Gaussian-Hermite moments,” *Signal Processing*, vol. 113, no. 1, pp. 61–67, 2015.

[15] D. Bhattacharya and S. Sinha, “Invariance of stereo images via theory of complex moments,” *Pattern Recognition*, vol. 30, no. 9, pp. 1373–1386, 1997.

[16] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis*. Singapore: World Scientific, 1998.

[17] S. Maitra, “Moment invariants,” *Proceedings of the IEEE*, vol. 67, no. 4, pp. 697–699, 1979.

[18] T. M. Hupkens and J. de Clippeleir, “Noise and intensity invariant moments,” *Pattern Recognition Letters*, vol. 16, no. 4, pp. 371–376, 1995.

[19] J. Flusser and T. Suk, “Rotation moment invariants for recognition of symmetric objects,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3784–3790, 2006.

[20] D. Shen, H. H.-S. Ip, and E. K. Teoh, "A novel theorem on symmetries of 2D images," in *Proceedings of the 15th International Conference on Pattern Recognition ICPR'00*, vol. 3, pp. 1014–1017, IEEE, 2000.

[21] J. Lin, W. Tsai, and J. Chen, "Detecting number of folds by a simple mathematical property," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1081–1088, 1994.

[22] J. Lin, "A simplified fold number detector for shapes with monotonic radii," *Pattern Recognition*, vol. 29, no. 6, pp. 997–1005, 1996.

[23] D. Shen, H. H.-S. Ip, K. K. T. Cheung, and E. K. Teoh, "Symmetry detection by generalized complex (GC) moments: A close-form solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 466–476, 1999.

[24] S. Derrode and F. Ghorbel, "Shape analysis and symmetry detection in gray-level objects using the analytical Fourier-Mellin representation," *Signal Processing*, vol. 84, no. 1, pp. 25–39, 2004.

[25] J. Chen, L. Wang, and D. Chen, *Logo Recognition: Theory and Practice.* Boca Raton, FL: CRC Press, 2011.

[26] Y.-S. Kim and W.-Y. Kim, "Content-based trademark retrieval system using a visually salient feature," *Image and Vision Computing*, vol. 16, no. 12–13, pp. 931–939, 1998.

[27] A. Jain and A. Vailaya, "Shape-based retrieval: A case study with trademark image databases," *Pattern Recognition*, vol. 31, no. 9, pp. 1369–1390, 1998.

[28] M. S. Hitam, W. Nural, J. Hj Wan Yussof, and M. M. Deris, "Hybrid Zernike moments and color-spatial technique for content-based trademark retrieval," in *Proceedings of the International Symposium on Management Engineering, ISME'06*, 2006.

[29] Y. S. Abu-Mostafa and D. Psaltis, "Image normalization by complex moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 1, pp. 46–55, 1985.

[30] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen, "Moment invariants for the analysis of 2D flow fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1743–1750, 2007.

[31] R. Bujack, "General basis of rotation invariants of vector fields." private communication (unpublished), 2015.

[32] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Northeastern University, Boston, Massachusetts, USA, 2009.

[33] C. Liu, 2011. `http://people.csail.mit.edu/celiu/OpticalFlow/`.

[34] W. Liu and E. Ribeiro, "Detecting singular patterns in 2-D vector fields using weighted Laurent polynomial," *Pattern Recognition*, vol. 45, no. 11, pp. 3912–3925, 2012.

[35] M. Liu and P.-T. Yap, "Invariant representation of orientation fields for fingerprint indexing," *Pattern Recognition*, vol. 45, no. 7, pp. 2532–2542, 2012.

[36] R. Bujack, I. Hotz, G. Scheuermann, and E. Hitzer, "Moment invariants for 2D flow fields using normalization," in *Pacific Visualization Symposium, PacificVis'14*, pp. 41–48, IEEE, 2014.

[37] R. Bujack, M. Hlawitschka, G. Scheuermann, and E. Hitzer, "Customized TRS invariants for 2D vector fields via moment normalization," *Pattern Recognition Letters*, vol. 46, no. 1, pp. 46–59, 2014.

[38] R. Bujack, J. Kasten, V. Natarajan, G. Scheuermann, and K. Joy, "Clustering moment invariants to identify similarity within 2D flow fields," in *EG / VGTC Conference on Visualization, EuroVis'15* (J. Kennedy and E. Puppo, eds.), pp. 31–35, The Eurographics Association, 2015.

Index