Object Recognition by Implicit Invariants

Jan Flusser[1], Jaroslav Kautsky[2], and Filip Šroubek[1]

[1] Institute of Information Theory and Automation, AS CR
Pod vodárenskou věží 4, 182 08, Prague 8, Czech Republic
[2] The Flinders University of South Australia, Adelaide, Australia

Abstract. The use of traditional moment invariants is limited to a certain set of simple geometric transforms, such as rotation, scaling and affine transform. This paper presents a novel concept of so-called implicit moment invariants, which enable us to recognize objects under a broader set of geometric deformations.

1 Introduction

Recognition of objects and patterns that are deformed in various ways has been a goal of much recent research. There are basically three major approaches to this problem – full search, image normalization, and invariant descriptors. The approach using invariant descriptors appears to be the most promising one and has been used extensively. Its basic idea is to describe the object by a set of features which are not sensitive to particular deformations and which provide enough discrimination power to distinguish among objects belonging to different classes.

In 2D object recognition, various moment invariants have become classical and frequently used shape descriptors during last forty years. Even if they suffer from some intrinsic limitations (the most important of which is their globalness, which prevents them from being used for recognition of occluded objects), they often serve as the "first-choice descriptors" and as a reference method for evaluation of the performance of other shape descriptors.

All moment invariants ever studied (see for instance [1,2,3,4]) are so-called *explicit* invariants. An explicit invariant is a functional (let us denote it as E) acting on the space of image functions which does not change its value if the image f undergoes certain deformation τ from the set of admissible deformations, i.e. which satisfies the condition $E(f) = E(\tau(f))$ for any image f. There have been described many systems of explicit moment invariants with respect to rotation, scaling, affine transform, contrast changes, and linear filtering. However, there are several classes of image deformations which occur frequently in practice but explicit moment invariants with respect to them are not known or even have been proven they cannot exist. Such typical examples are projective transform, cylindrical and spherical projections, quadratic transform, and other polynomial transforms of the image coordinates.

To overcome this, we propose in this paper a new concept of *implicit* invariants. Implicit invariant I is a functional defined on image pairs such that

$I(f, \tau(f)) = 0$ for any image f and deformation τ. According to this definition, explicit invariants are just particular cases of implicit invariants. Clearly, if an explicit invariant exists, we can set $I(f, g) = E(f) - E(g)$. As we show later on in the paper, there are many types of image deformations where explicit moment invariants do not exist while implicit moment invariants do. In those cases, implicit invariants can be used as features for object recognition.

Unlike explicit invariants, implicit invariants do not provide description of a single image because they are always defined for a pair of images. For recognition purposes this is not a drawback. We consider $I(f, g)$ to be a "distance measure" (even if it does not exhibit all properties of a metric) between f and g factorized by τ and we can, for each database template g_i, calculate the value of $I(f, g_i)$ and then to classify f according to the minimum.

2   General Moments

Definition 1. *Let* $p_0, p_1, \ldots, p_{n-1}, \ldots$ *be some basis functions defined on a bounded* $D \subset I\!\!R^N$ *and let* f *be an image function having a finite integral. By a general moment* of f *we understand the functional*

$$\mu_j(f) = \int_D f(x) p_j(x) dx.$$

If $N = 1$ and $p_j(x) = x^j$, we speak about *standard* moments.

Using a matrix notation we can write

$$\boldsymbol{p}(x) = \begin{pmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{n-1}(x) \end{pmatrix} \quad \text{and} \quad \boldsymbol{\mu}(f) = \begin{pmatrix} \mu_0(f) \\ \mu_1(f) \\ \vdots \\ \mu_{n-1}(f) \end{pmatrix}. \tag{1}$$

Let $r : D \to \tilde{D}$ be a transformation of the domain D into $\tilde{D}$ and let $\tilde{f} : \tilde{D} \to I\!\!R$ be another image function which satisfies

$$\tilde{f}(r(x)) = f(x) \tag{2}$$

for $x \in D$ and $f(\tilde{x}) = 0$ for $\tilde{x} \in \tilde{D}/r(D)$. (This means that image $\tilde{f}$ is a spatially deformed version of f.)

We are interested in the relation between the moments $\boldsymbol{\mu}(f)$ and the moments

$$\tilde{\boldsymbol{\mu}}(\tilde{f}) = \int_{r(D)} \tilde{f}(\tilde{x}) \tilde{\boldsymbol{p}}(\tilde{x}) d\tilde{x} = \int_{\tilde{D}} \tilde{f}(\tilde{x}) \tilde{\boldsymbol{p}}(\tilde{x}) d\tilde{x}$$

of the transformed function with respect to some other $\tilde{n}$ basis functions

$$\tilde{\boldsymbol{p}}(\tilde{x}) = \begin{pmatrix} \tilde{p}_0(\tilde{x}) & \tilde{p}_1(\tilde{x}) & \ldots & \tilde{p}_{\tilde{n}-1}(\tilde{x}) \end{pmatrix}^T$$

defined on $\tilde{D}$. We can now formulate the following Theorem.

Theorem 1. *Denote by $J_r(x)$ the Jacobian of the transform function r. If*

$$\tilde{\boldsymbol{p}}(r(x))|J_r(x)| = A\boldsymbol{p}(x) \tag{3}$$

for some $\tilde{n} \times n$ matrix A then

$$\tilde{\boldsymbol{\mu}} = A\boldsymbol{\mu} . \tag{4}$$

The power of this theorem depends on our ability to choose the basis functions so that we can, for a given transform r, express the left-hand side of (3) in terms of the basis functions $\boldsymbol{p}$ and thus construct the matrix A. This is always possible for a polynomial r by choosing polynomial bases $\boldsymbol{p}(x)$ and $\tilde{\boldsymbol{p}}(\tilde{x})$.

3   Implicit Moment Invariants

Let us assume that the transformation r depends on a finite number, say m, $m < \tilde{n}$, parameters $a = (a_1, \ldots, a_m)$. Traditional explicit moment invariants with respect to r can be obtained in two steps.

(a) Eliminate $a = (a_1, \ldots, a_m)$ from the system (4). This leaves us $\tilde{n} - m$ equations which depend only on the two sets of general moments (and on the choice of basis functions, of course). We call it a *reduced system*.
(b) Re-write these equations equivalently in the form

$$q_j(\tilde{\boldsymbol{\mu}}(\tilde{f})) = q_j(\boldsymbol{\mu}(f)) , \quad j = 1, \ldots, \tilde{n} - m \tag{5}$$

for some functions q_j. Then the explicit moment invariants are $E(f) = q_j(\boldsymbol{\mu}(f))$.

However, for some transforms (quadratic, cubic, etc.) we may not be able to perform the second step – finding the explicit forms q_j. Introducing implicit invariants can overcome this drawback.

The reduced system in step (a) is independent of the particular transformation. For classifying of an object, we traditionally compare the values of its descriptors (explicit moment invariants) with those of the database images, that is we look for such database image, which satisfy equations (5). However, it is equivalent to checking for which database image the above reduced system is satisfied.

So we can, in case we are not able to find explicit moment invariants in the form (5), use this system as a set of implicit invariants. In other words, the images are classified according to the error with which the system is satisfied.

We will demonstrate the above idea of implicit moment invariants on a 1D example using standard powers as basis functions. Consider the transform

$$r(x) = x + ax^2,$$

where $a \in (0, 1/2\rangle$, which maps interval $D = \langle -1, 1\rangle$ on interval $\tilde{D} = \langle a-1, a+1\rangle$. Let us show two implicit invariants. Since $m = 1$ (one-parameter transform), we

need $\tilde{n} = 3$ and $n = 6$. The Jacobian is $J_r(x) = 1 + 2ax$ and for standard powers for both $\boldsymbol{p}$ and $\tilde{\boldsymbol{p}}$ we would get

$$A = \begin{pmatrix} 1 & 2a & 0 & 0 & 0 & 0 \\ 0 & 1 & 3a & 2a^2 & 0 & 0 \\ 0 & 0 & 1 & 4a & 5a^2 & 2a^3 \end{pmatrix} .$$

However, now we have to evaluate the moments of the transformed signal over the domain $\tilde{D}$ which depends on the unknown parameter a. This problem is resolved by choosing a shifted power basis

$$\tilde{p}_j(\tilde{x}) = (\tilde{x} - a)^j , \qquad j = 0, 1, \ldots \tilde{n} - 1$$

as we have then, after the shift of variable $\tilde{x} = \hat{x} + a$,

$$\tilde{\mu}_j(\tilde{f}) = \int_{a-1}^{a+1} \tilde{f}(\tilde{x})(\tilde{x} - a)^j d\tilde{x} = \int_{-1}^{1} \tilde{f}(\hat{x} + a)\hat{x}^j d\hat{x}$$

which is now independent of a as $\hat{f}(\hat{x}) = \tilde{f}(\hat{x} + a)$ has domain $\langle -1, 1 \rangle$. For this basis $\tilde{\boldsymbol{p}}$ we obtain a different transform matrix

$$A = \begin{pmatrix} 1 & 2a & 0 & 0 & 0 & 0 \\ -a & 1 - 2a & 3a & 2a^2 & 0 & 0 \\ a^2 & 2a(a^2 - 1) & 1 - 6a^2 & 4a(1 - a^2) & 5a^2 & 2a^3 \end{pmatrix}$$

and the first of equations (4) gives

$$a = \frac{\tilde{\mu}_0 - \mu_0}{2\mu_1}$$

while the two reduced equations rewrite, after substitution, as

$$\begin{aligned} 2\mu_1^2(\tilde{\mu}_1 - \mu_1) &= \mu_1(3\mu_2 - \tilde{\mu}_0)(\tilde{\mu}_0 - \mu_0) + \mu_3(\tilde{\mu}_0 - \mu_0)^2 \\ 4\mu_1^3(\tilde{\mu}_2 - \mu_2) &= 4\mu_1^2(2\mu_3 - \mu_1)(\tilde{\mu}_0 - \mu_0) + \mu_1(5\mu_4 + \tilde{\mu}_0 - 6\mu_2)(\tilde{\mu}_0 - \mu_0)^2 \\ &\quad + (\mu_5 - 2\mu_3)(\tilde{\mu}_0 - \mu_0)^3 . \end{aligned} \qquad (6)$$

In the example above it was straightforward to derive the transform matrix A for simple transformation r and a small number of invariants. For numerical reasons, this intuitive approach cannot be used for higher-order polynomial transform r and/or for more invariants. To obtain numerically stable method it is important to use suitable polynomial bases, such as orthogonal polynomials, without using their expansions into standard (monomial) powers. Our implementation is based on the representation of polynomial bases by matrices with a special structure [5]. This representation allows to evaluate the polynomials efficiently by means of recurrent relations.

4   Implementation of the Implicit Invariants

Depending on r, the elimination of the m parameters of the transformation function from the $\tilde{n}$ equations of (4) to obtain a parameter-free reduced system

may require numerical solving of nonlinear equations. This may be undesirable or impossible. Even the simple transform r used in the experimental section would lead to cubic equations in terms of its parameters. Obtaining a neat reduced system may be very difficult. Furthermore, even if successful, we create an unbalanced method – we have demanded some of the equations in (4) to hold exactly and use the accuracy in the resulting system as a matching criterion to find the transformed image. We therefore propose another implementation of the implicit invariants. Instead of eliminating the parameters, we calculate the "uniform best fit" from all equations in (4). For a given set of values of the moments $\boldsymbol{\mu}$ and $\tilde{\boldsymbol{\mu}}$, we find values of the m parameters to satisfy (4) as best as possible in ℓ_2 norm; the error of this fit then becomes the value of the respective implicit invariant.

Our actual implementation of the recognition by implicit invariants can be described as follows.

(a) Given is a library (database) of images $g_j(x, y)$, $j = 1, \ldots, L$, and a deformed image $\tilde{f}(\tilde{x}, \tilde{y})$ which is assumed to have been obtained by a transform of a known polynomial form $r(x, y, \boldsymbol{a})$ with unknown values of m parameters $\boldsymbol{a}$.

(b) Choose the appropriate domains, polynomial bases $\boldsymbol{p}$ and $\tilde{\boldsymbol{p}}$, and the recurrence matrices for evaluation of the polynomials.

(c) Derive a program to evaluate the matrix $A(\boldsymbol{a})$. This critical error-prone step is performed by a symbolic algorithmic procedure which produces the program used then in numerical calculations. This step is performed only once for the given task. (It has to be repeated only if we change the polynomial bases or the form of transform $r(x, y, \boldsymbol{a})$, which basically means only if we move to another application).

(d) Calculate the moments $\boldsymbol{\mu}(g_j)$ of all library images $g_j(x, y)$.

(e) Calculate the moments $\tilde{\boldsymbol{\mu}}(\tilde{f})$ of the deformed image $\tilde{f}(\tilde{x}, \tilde{y})$.

(f) For all $j = 1, \ldots, L$ calculate, using an optimizer, the values of the implicit invariant

$$I(\tilde{f}, g_j) = \min_{\boldsymbol{a}} \|\tilde{\boldsymbol{\mu}}(\tilde{f}) - A(\boldsymbol{a})\boldsymbol{\mu}(g_j)\|$$

and denote

$$M = \min_j I(\tilde{f}, g_j).$$

The norm used here should be weighted, for example relatively to the components corresponding to the same degree.

(g) The identified image is g_k for which $I(\tilde{f}, g_k) = M$; the ratios $\frac{I(\tilde{f}, g_j)}{I(\tilde{f}, g_k)}$, $j \neq k$, may be used as confidence measures of the identification.

5   Numerical Experiments

As we have shown earlier, the implicit moment invariants can be constructed for a very broad class of image transforms including all polynomial transforms.

Here we will demonstrate the implementation and the power of the method on images transformed by the following function

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \boldsymbol{r}(x,y) = \begin{pmatrix} ax + by + c(ax + by)^2 \\ -bx + ay \end{pmatrix} \;, \tag{7}$$

which is a rotation with scaling (parameters a and b) followed by a quadratic deformation in the $\tilde{x}$ direction (parameter c). We have chosen this particular transform for our tests for the following reasons:

- It is general enough to approximate many real-life situations, for instance deformations caused by the fact that the photographed object was drawn/ printed on a spherical or cylindrical surfaces like bottles and balls.
- It is sometimes used by web designers to warp images in order to reach desirable visual effect. Very often this is an unauthorized act violating the copyright. It is important for the copyright owners to have a tool how to identify such images.
- Explicit invariants to this kind of transforms cannot exist because they do not preserve the moment orders and do not form a group.

The first experiment was aimed to test the discriminative power of the implicit invariants and to demonstrate that they can be used as shape descriptors for recognition of distorted real objects This test was done on a standard benchmark database ALOI [6].

We took 100 ALOI images and deformed each of them by the warping model (7) (see Fig. 1 for some examples). The coefficients of the deformations were generated randomly; c from a range of admissible values and the rotation angle from $(-40°, 40°)$, both with uniform distribution. Each deformed image was then classified against the undistorted database by three different methods: by implicit invariants according to minimal norm, by the Hu's rotation moment invariants [1] according to minimum distance, and by affine moment invariants (AMI) [2] also using the minimum-distance rule. In all three cases, six invariants were used. The last two methods were selected for a comparison because they are similar to the new technique in their nature (all of them are based on moments) and because they are traditional, well-established reference methods in pattern recognition.

We run the whole experiment several times with different deformation parameters. In each run the recognition rate we achieved was 99 or 100% for the implicit invariants, from 43 to 47% for the rotation invariants, and from 34 to 40% for the AMI's. These results illustrate two important facts. First, the implicit invariants can serve as an efficient tool for object recognition in case when the object deformation corresponds to the assumed model. Secondly, in case of nonlinear distortions the implicit invariants significantly outperform both rotation as well as affine moment invariants, which corresponds to our theoretical expectation. In case when only rotation is present, all three methods are equivalent. To illustrate this, we run the experiment once again but we fixed $c = 0$. Then the recognition rate of all three methods was 100%.

Fig. 1. The original images from the ALOI database (top) and their deformed versions (bottom)

The second experiment was done on real images taken in our lab. It illustrates good performance and high recognition power of the implicit invariants even in the case where theoretical assumptions about the degradation are not fulfilled.

With a standard digital camera (Olympus C-5050), we took a photo of letters printed on a label which was glued to a bottle, see Fig. 2(a). The letters were organized in a 4×3 mesh with "A"s, "B"s, "V"s and "X"s each printed three times in a row. After a simple segmentation, the letters were labeled from left to right A_1, A_2, A_3, B_1, ..., V_1,...,X_1, X_2, and X_3. Due to the curvature of the bottle surface, the letters appear distorted in the horizontal direction and the distortion grows to the right. A_1 does not exhibit any visible distortion while A_3 is the most distorted one and likewise for the other three letters. The task was to recognize (classify) these letters against a database containing the full English alphabet (26 undistorted letters of the same font).

In an "ideal" case when the camera is in infinity the image distortion can be described by orthogonal projection of the cylinder onto a plane, i.e.

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} r\sin(\frac{x}{r}) \\ y \end{pmatrix} \ ,$$

where r is the bottle radius, x, y are the coordinates on the bottle surface and $\tilde{x}, \tilde{y}$ are the coordinates on the acquired images. In our case the object-to-camera distance was finite, so small perspective effect appears in addition to the above model. We assume the actual image deformation can be approximated by a quadratic polynomial in x direction. Although it is clear that such approximation cannot be very accurate, we will demonstrate that it is accurate enough for our purpose.

We classified all deformed letters by means of implicit invariants in the same way as in the previous experiment and also by the Hu's moment invariants. The table in Fig 2(b) summarizes the classification results of both methods. Implicit invariants provided a perfect recognition rate as all deformed letters were classified correctly with high minimum confidence (see the definition in (g), Section 4). It might be a bit surprise if one considers the very rough approximation of the

(a)

	A_1	A_2	A_3	B_1	B_2	B_3	V_1	V_2	V_3	X_1	X_2	X_3
Imp-inv.	A	A	A	B	B	B	V	V	V	X	X	X
confidence	65	29	15	64	186	69	96	62	80	35	98	19
Hu-inv.	A	Y	N	B	S	S	V	G	N	X	X	I

(b)

Fig. 2. (a) Letters captured by a standard digital camera exhibit distortion due to the cylindrical shape of the bottle. (b) Classification of four letters (each having three different degrees of distortion) by implicit invariants (first row) with confidence in the second row, and classification by Hu's invariants (third row).

transformation model made above. It indicates some degree of robustness of the implicit invariants to the type of the image deformation. Hu's moment invariants classified correctly only the letters without any quadratic deformation (A_1, B_1, V_1, X_1) and failed in other cases, which is in agreement with the theory.

References

1. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Information Theory 8, 179–187 (1962)
2. Flusser, J., Suk, T.: Pattern recognition by affine moment invariants. Pattern Recognition 26, 167–174 (1993)
3. Reiss, T.H.: The revised fundamental theorem of moment invariants. IEEE Trans. Pattern Analysis and Machine Intelligence 13, 830–834 (1991)
4. Flusser, J.: On the independence of rotation moment invariants. Pattern Recognition 33(9), 1405–1410 (2000)
5. Kautsky, J., Golub, G.: On the calculation of Jacobi matrices. Linear Algebra Appl. 52/53, 439–455 (1983)
6. Amsterdam Library of Object Images: `http://staff.science.uva.nl/~aloi/`