

A Unified Framework for Image-Derived Invariants

Yuan-Fang Wang and Ronald-Bryan O. Alferéz
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract: We propose a general framework for computing invariant features from images. The proposed approach is based on a simple concept of basis expansion. It is widely applicable to many popular basis representations, such as wavelets, short-time Fourier transform, and splines.

1 Introduction

Image features and shape descriptors that capture the essential traits of an object and are insensitive to environmental changes are ideal for recognition. The search for invariants (e.g., algebraic and projective invariants) is a classical problem in mathematics dating back to the 18th century. The need for invariant image descriptors has long been recognized in computer vision [3, 6]. Invariant features form a compact, intrinsic description of an object, and can be used to design recognition algorithms that are potentially more efficient than, say, the aspect-based approaches. Hence, it was even argued that object recognition is the search for invariants [6]. Invariant features can be designed based on many different methods and made invariant to rigid motion, general affine transform, scene illumination, occlusion, and projection. See [3, 6] for a comprehensive survey of the subject of invariants.

The proposed framework exploits both global and local information about shape and color, and is neither strictly global nor local. It has the advantage of tolerating a certain degree of occlusion (unlike global analysis) and does not require estimating high-order derivatives in computing invariants (unlike local analysis), whence is more robust. Furthermore, it enables a quasi-localized, hierarchical shape analysis which is unique among known invariant techniques. Unlike some current research on image invariants which concentrates on either geometry or illumination invariants, the proposed framework is very general and produces invariants which are insensitive to rigid motion, general affine transform, changes of parameterization and scene illumination, noise, occlusion, and perspective transform and view point change. Finally, we introduce the rational basis functions [4] to facilitate the analysis of invariants under perspective transform. Though rational basis functions, such as NURBS, are widely known in the computer graphics community, to the best of our knowledge they have not been widely used in computer vision.

2 A Framework of Image-Derived Invariants

We will illustrate the mathematical frameworks using a specific scenario of invariants for planar curves. The particular basis functions we use will be the wavelet bases and spline functions. Though the same framework can be easily extended to other bases such as the short-time Fourier analysis.

A word on the notational convention: matrices and vectors will be represented by bold-face characters, such as \mathbf{M} and \mathbf{V} , while scalar quantities by plain-face characters such as S . 2D quantities will be in small letters while 3D quantities in capital letters. Hence, a 3D coordinate will be denoted as $[X, Y, Z]^T$ while a 2D coordinate as $[x, y]^T$.

We have considered variation in an object's image induced by rigid-body motion, general affine transform, changes in parameterization and illumination, and perspective transform. Due to the page limit, we will review briefly only the formulations for affine, luminance, and perspective invariants and present some preliminary results. Interested readers are referred to [5] for other invariants expressions and more results.

Rigid-Body Motion and Affine Transform Consider a 2D curve, $\mathbf{c}(t) = [x(t), y(t)]^T$, where t denotes the *affine* arc length (invariant under affine transform [2]), and its expansion onto the wavelet basis $\psi_{a,b} = \frac{1}{\sqrt{a}}g(\frac{t-b}{a})$ (where $g(t)$ is the mother wavelet [1]) as

$$\mathbf{u}_{a,b} = \int \mathbf{c}\psi_{a,b}dt, \quad \text{or} \quad \mathbf{c}(t) = \int_a \int_b \mathbf{u}_{a,b}\psi_{a,b}dadb.$$

If the curve is allowed a general affine transform with the transformed curve denoted by:

$$\mathbf{c}'(t) = \mathbf{m}\mathbf{c} + \mathbf{t} = \mathbf{m} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \mathbf{t},$$

where \mathbf{m} is any non-singular 2×2 matrix and \mathbf{t} represents the translational motion, then we have

$$\begin{aligned} \mathbf{u}'_{a,b} &= \int \mathbf{c}'\psi_{a,b}dt \\ &= \int (\mathbf{m}\mathbf{c} + \mathbf{t})\psi_{a,b}dt = \int \mathbf{m}\mathbf{c}\psi_{a,b}dt + \int \mathbf{t}\psi_{a,b}dt \\ &= \mathbf{m} \int \mathbf{c}\psi_{a,b}dt + \mathbf{t} \int \psi_{a,b}dt = \mathbf{m}\mathbf{u}_{a,b}, \quad \text{or} \end{aligned} \quad (1)$$

$$\mathbf{c}'(t) = \int_a \int_b \mathbf{u}'_{a,b}\psi_{a,b}dadb.$$

Note that we use wavelet property $\int \psi_{a,b}dt = 0$ to simplify the second term in Eq. 1. Hence, the transformed curve can be generated using the transformed coefficients and the same wavelet bases, instead of transforming the curve point-by-point. This is an observation made in the computer graphics community on curves generated by the spline functions and associated control vertices [4]. In that sense, $\mathbf{u}_{a,b}$'s are equivalent to the control vertices in a spline curve.

If \mathbf{m} represents a rotation (or the affine transform is a rigid-body motion of a translation plus a rotation), it is easily seen that invariant features can be derived using the ratio expression

$$\frac{|\mathbf{u}'_{a,b}|}{|\mathbf{u}'_{c,d}|} = \frac{|\mathbf{m}\mathbf{u}_{a,b}|}{|\mathbf{m}\mathbf{u}_{c,d}|} = \frac{|\mathbf{u}_{a,b}|}{|\mathbf{u}_{c,d}|}.$$

If the second term in Eq. 1 is not zero, but is a constant (e.g., for spline functions, the area under a spline basis integrates to a constant 1 for a uniformly spaced

knot vector [4]), then invariant expressions can still be derived, albeit in a slightly more complicated form:

$$\frac{|\mathbf{u}'_{a,b} - \mathbf{u}'_{c,d}|}{|\mathbf{u}'_{e,f} - \mathbf{u}'_{g,h}|} = \frac{|(\mathbf{m}\mathbf{u}_{a,b} + \mathbf{v}) - (\mathbf{m}\mathbf{u}_{c,d} + \mathbf{v})|}{|(\mathbf{m}\mathbf{u}_{e,f} + \mathbf{v}) - (\mathbf{m}\mathbf{u}_{g,h} + \mathbf{v})|} = \frac{|\mathbf{m}(\mathbf{u}_{a,b} - \mathbf{u}_{c,d})|}{|\mathbf{m}(\mathbf{u}_{e,f} - \mathbf{u}_{g,h})|} = \frac{|(\mathbf{u}_{a,b} - \mathbf{u}_{c,d})|}{|(\mathbf{u}_{e,f} - \mathbf{u}_{g,h})|},$$

where \mathbf{v} denotes the constant second term in Eq. 1.

For invariants under general affine transform, many forms using ratios, cross ratios, and ratios of ratios have already been derived [3, 6]. For example, it is known that the cross ratio of four co-linear points are invariant under an affine transform, and the area of the triangle formed by any three $\mathbf{u}_{a,b}$ changes linearly in an affine transform (an invariant of weight 1 [3, 6]). Hence, an absolute invariance can be generated by using the ratio of two triangles:¹

$$\frac{\begin{vmatrix} \mathbf{u}'_{a,b} & \mathbf{u}'_{c,d} & \mathbf{u}'_{e,f} \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} \mathbf{u}'_{g,h} & \mathbf{u}'_{i,j} & \mathbf{u}'_{k,l} \\ 1 & 1 & 1 \end{vmatrix}} = \frac{\begin{vmatrix} \mathbf{u}_{a,b} & \mathbf{u}_{c,d} & \mathbf{u}_{e,f} \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} \mathbf{u}_{g,h} & \mathbf{u}_{i,j} & \mathbf{u}_{k,l} \\ 1 & 1 & 1 \end{vmatrix}}. \quad (2)$$

Variation in Lighting Condition Another possible variation in the appearance is due to lighting: that objects can be illuminated by light sources of different numbers and types. To simplify the notation, in the following derivation we will consider three spectral bands of red, green, and blue. Generalizing to an n -band illumination model is straightforward. Let $\mathbf{L}(t)$ denote the perceived image color distribution along a curve, we have:

$$\mathbf{L}(t) = \begin{bmatrix} r(t) \\ g(t) \\ b(t) \end{bmatrix} = \int \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} s(\lambda, t) d\lambda,$$

where λ denotes the wavelength, and $f^r(\lambda)$ the sensitivity of the red sensor (similar interpretations for the green and blue channels). Using a Lambertian model, $s(\lambda, t)$ is

$$s(\lambda, t) = \left(\sum_{i=1}^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i \right) \rho(\lambda, t) + a(\lambda), \quad (3)$$

where n is the number of light sources used, $l_i(\lambda)$ the source luminance spectral distribution, \mathbf{N} the surface normal, \mathbf{N}_i the incident direction for source i , $\rho(\lambda, t)$ the surface reflectivity, and $a(\lambda)$ the ambient light luminance.

When the lighting condition changes, because lights are moved, turned on, or turned off, or the ambient light intensity changes, we have

$$\begin{aligned} s'(\lambda, t) &= \left(\sum_j^m l'_j(\lambda) \mathbf{N} \cdot \mathbf{N}'_j \right) \rho(\lambda, t) + a'(\lambda), \\ &= \frac{\sum_j^m l'_j(\lambda) \mathbf{N} \cdot \mathbf{N}'_j}{\sum_i^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i} \left[\left(\sum_i^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i \right) \rho(\lambda, t) + a(\lambda) \right] + a'(\lambda) - \frac{\sum_j^m l'_j(\lambda) \mathbf{N} \cdot \mathbf{N}'_j}{\sum_i^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i} a(\lambda) \\ &= c_1(\lambda) s(\lambda, t) + c_2(\lambda), \quad \text{where} \end{aligned}$$

¹ Note that there are many valid expressions for affine invariants. Some may require a smaller number of coefficients than that in Eq. 2. For example, when wavelet bases are used where $\int \psi_{a,b} dt = 0$, Eq. 2 can be simplified as

$$\frac{\begin{vmatrix} \mathbf{u}'_{a,b} & \mathbf{u}'_{c,d} \\ \mathbf{u}'_{e,f} & \mathbf{u}'_{g,h} \end{vmatrix}}{\begin{vmatrix} \mathbf{u}_{a,b} & \mathbf{u}_{c,d} \\ \mathbf{u}_{e,f} & \mathbf{u}_{g,h} \end{vmatrix}},$$

where only four coefficients are needed.

$$c_1(\lambda) = \frac{\sum_j^m l'_j(\lambda) \mathbf{N} \cdot \mathbf{N}'_j}{\sum_i^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i} \quad c_2(\lambda) = a'(\lambda) - \frac{\sum_j^m l'_j(\lambda) \mathbf{N} \cdot \mathbf{N}'_j}{\sum_i^n l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i} a(\lambda)$$

capture the changes in the gain and offset in the two different lighting conditions. Following a path similar to that adopted by several researchers, we assume that surface reflectance functions are modeled as a linear combination of a small number of basis functions $s_k(\lambda)$, whence,

$$s(\lambda, t) = \sum_k \alpha_k(t) s_k(\lambda),$$

where $s_k(\lambda)$ denotes the k -th basis function for representing the surface reflectance properties, and $\alpha_k(t)$ is the space varying expansion coefficients. Then using an analysis which is similar to that employed in the affine case, we have

$$\begin{aligned} \mathbf{u}_{a,b} &= \int \mathbf{L} \psi_{a,b} dt \\ &= \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} s(\lambda, t) d\lambda \psi_{a,b} dt = \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} \left(\sum_k \alpha_k(t) s_k(\lambda) \right) d\lambda \psi_{a,b} dt \\ &= \sum_k \left(\int_\lambda \begin{bmatrix} f^r(\lambda) s_k(\lambda) \\ f^g(\lambda) s_k(\lambda) \\ f^b(\lambda) s_k(\lambda) \end{bmatrix} d\lambda \right) \left(\int_t \alpha_k(t) \psi_{a,b} dt \right) = \sum_k \begin{bmatrix} L_k^r \\ L_k^g \\ L_k^b \end{bmatrix} v_{a,b}^k \\ &= \begin{bmatrix} L_1^r & L_2^r & \dots & L_k^r \\ L_1^g & L_2^g & \dots & L_k^g \\ L_1^b & L_2^b & \dots & L_k^b \end{bmatrix} \begin{bmatrix} v_{a,b}^1 \\ v_{a,b}^2 \\ \dots \\ v_{a,b}^k \end{bmatrix} = \mathbf{L}_{rgb} \mathbf{v}_{a,b}, \quad \text{where} \\ &\quad \begin{bmatrix} L_k^r \\ L_k^g \\ L_k^b \end{bmatrix} = \int_\lambda \begin{bmatrix} f^r(\lambda) s_k(\lambda) \\ f^g(\lambda) s_k(\lambda) \\ f^b(\lambda) s_k(\lambda) \end{bmatrix} d\lambda \quad \text{Similarly,} \\ &\quad v_{a,b}^k = \int_t \alpha_k(t) \psi_{a,b} dt. \\ \mathbf{u}'_{a,b} &= \int \mathbf{L}' \psi_{a,b} dt \\ &= \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} [c_1(\lambda) s(\lambda, t) + c_2(\lambda)] d\lambda \psi_{a,b} dt = \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} [c_1(\lambda) (\sum_k \alpha_k(t) s_k(\lambda)) + c_2(\lambda)] d\lambda \psi_{a,b} dt \\ &= \sum_k \left\{ \left(\int_\lambda \begin{bmatrix} c_1(\lambda) f^r(\lambda) s_k(\lambda) \\ c_1(\lambda) f^g(\lambda) s_k(\lambda) \\ c_1(\lambda) f^b(\lambda) s_k(\lambda) \end{bmatrix} d\lambda \right) \left(\int_t \alpha_k(t) \psi_{a,b} dt \right) + \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} c_2(\lambda) d\lambda \psi_{a,b} dt \right\} \\ &= \sum_k \begin{bmatrix} L_{k'}^r \\ L_{k'}^g \\ L_{k'}^b \end{bmatrix} v_{a,b}^k = \begin{bmatrix} L_1^{r'} & L_2^{r'} & \dots & L_{k'}^{r'} \\ L_1^{g'} & L_2^{g'} & \dots & L_{k'}^{g'} \\ L_1^{b'} & L_2^{b'} & \dots & L_{k'}^{b'} \end{bmatrix} \begin{bmatrix} v_{a,b}^1 \\ v_{a,b}^2 \\ \dots \\ v_{a,b}^k \end{bmatrix} \\ &= \mathbf{L}'_{rgb} \mathbf{v}_{a,b}. \end{aligned}$$

Then it is easily shown that the following expression is invariant:

$$\begin{aligned} &\frac{\left| \begin{bmatrix} \mathbf{u}'_{a_1, b_1} \mathbf{u}'_{a_2, b_2} \dots \mathbf{u}'_{a_k, b_k} \end{bmatrix}^T \begin{bmatrix} \mathbf{u}'_{a_1, b_1} \mathbf{u}'_{a_2, b_2} \dots \mathbf{u}'_{a_k, b_k} \end{bmatrix} \right|}{\left| \begin{bmatrix} \mathbf{u}'_{c_1, d_1} \mathbf{u}'_{c_2, d_2} \dots \mathbf{u}'_{c_k, d_k} \end{bmatrix}^T \begin{bmatrix} \mathbf{u}'_{c_1, d_1} \mathbf{u}'_{c_2, d_2} \dots \mathbf{u}'_{c_k, d_k} \end{bmatrix} \right|} \\ &= \frac{\left| \begin{bmatrix} \mathbf{v}_{a_1, b_1} \mathbf{v}_{a_2, b_2} \dots \mathbf{v}_{a_k, b_k} \end{bmatrix}^T \mathbf{L}'_{rgb} \mathbf{L}'_{rgb} \begin{bmatrix} \mathbf{v}_{a_1, b_1} \mathbf{v}_{a_2, b_2} \dots \mathbf{v}_{a_k, b_k} \end{bmatrix} \right|}{\left| \begin{bmatrix} \mathbf{v}_{c_1, d_1} \mathbf{v}_{c_2, d_2} \dots \mathbf{v}_{c_k, d_k} \end{bmatrix}^T \mathbf{L}'_{rgb} \mathbf{L}'_{rgb} \begin{bmatrix} \mathbf{v}_{c_1, d_1} \mathbf{v}_{c_2, d_2} \dots \mathbf{v}_{c_k, d_k} \end{bmatrix} \right|} \\ &= \frac{\left| \begin{bmatrix} \mathbf{v}_{a_1, b_1} \mathbf{v}_{a_2, b_2} \dots \mathbf{v}_{a_k, b_k} \end{bmatrix}^T \mathbf{L}_{rgb} \mathbf{L}_{rgb} \begin{bmatrix} \mathbf{v}_{a_1, b_1} \mathbf{v}_{a_2, b_2} \dots \mathbf{v}_{a_k, b_k} \end{bmatrix} \right|}{\left| \begin{bmatrix} \mathbf{v}_{c_1, d_1} \mathbf{v}_{c_2, d_2} \dots \mathbf{v}_{c_k, d_k} \end{bmatrix}^T \mathbf{L}_{rgb} \mathbf{L}_{rgb} \begin{bmatrix} \mathbf{v}_{c_1, d_1} \mathbf{v}_{c_2, d_2} \dots \mathbf{v}_{c_k, d_k} \end{bmatrix} \right|} \\ &= \frac{\left| \begin{bmatrix} \mathbf{u}_{a_1, b_1} \mathbf{u}_{a_2, b_2} \dots \mathbf{u}_{a_k, b_k} \end{bmatrix}^T \begin{bmatrix} \mathbf{u}_{a_1, b_1} \mathbf{u}_{a_2, b_2} \dots \mathbf{u}_{a_k, b_k} \end{bmatrix} \right|}{\left| \begin{bmatrix} \mathbf{u}_{c_1, d_1} \mathbf{u}_{c_2, d_2} \dots \mathbf{u}_{c_k, d_k} \end{bmatrix}^T \begin{bmatrix} \mathbf{u}_{c_1, d_1} \mathbf{u}_{c_2, d_2} \dots \mathbf{u}_{c_k, d_k} \end{bmatrix} \right|}. \end{aligned} \quad (4)$$

By using a ratio expression, we obtain a much simpler and computationally efficient form of invariant which does not require computing the color correlation matrix and the SVD of such a matrix as in some previously reported technique.

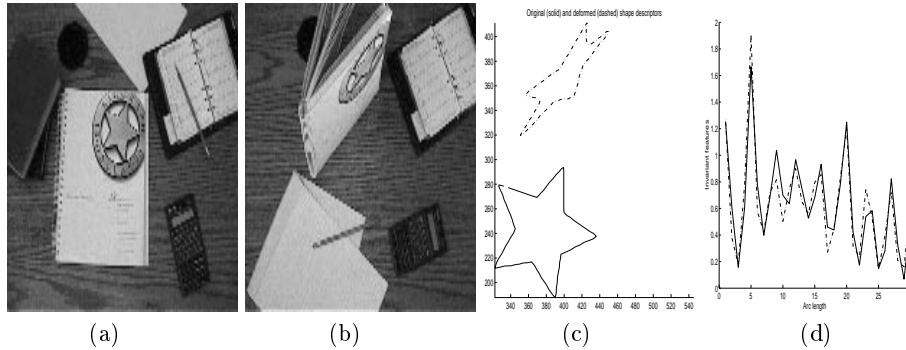


Fig. 1. (a) Original image, (b) comparison image, (c) original and deformed patterns after undergoing a rigid-body motion, and (d) invariant signatures of the original (solid) and transformed (dashed) patterns plotted along the contours.

Perspective Transform Perspective transform is a non-linear process involving a division in computing 2D coordinates. The projection process can be linearized by using the *rational* form of a basis function, such as NURBS [4]. Briefly, we represent a general 3D curve by decomposing it onto pre-selected bases. We will call the projection coefficients control vertices, following the convention used in computer graphics. The projection of a 3D curve is then represented by the projected control vertices and the *rational* bases. The problem of finding projective invariants is then a curve fitting problem: If a 2D curve results from the projection of a 3D curve, then it should be possible to interpolate the observed 2D curve using the projected control vertices and the rational spline bases and obtain a good fit. If that is not the case, then the curve probably does not come from the projection of the particular 3D curve. Hence, the error in curve fitting is a measure of invariance (In the ideal case, the error should be zero).

3 Experimental Results

Here we report some preliminary experimental results. Interested readers are referred to [5] for more detailed descriptions of the algorithms and results.

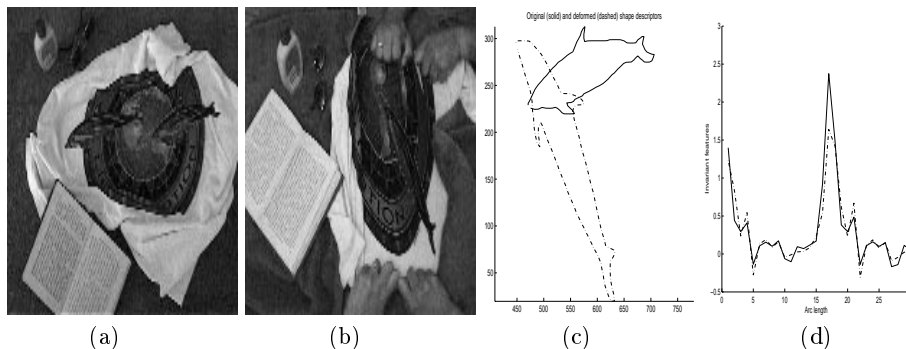


Fig. 2. (a) Original image, (b) comparison image, (c) original and deformed patterns after undergoing an affine transform, and (d) invariant signatures of the original (solid) and transformed (dashed) patterns plotted along the contours.



Fig. 3. The same mouse pad under (a) white and (b) blue illumination.

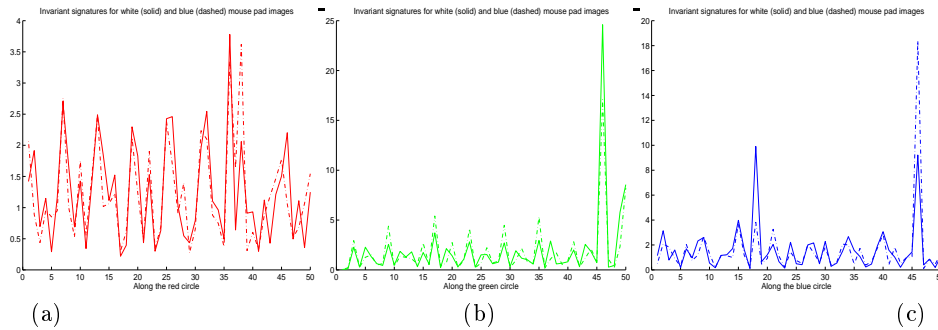


Fig. 4. R-g-b invariant signatures computed for the mouse pad under white (solid) and blue (dashed) illumination. Invariant signatures were computed on the red (a), green (b), and blue (c) circles in Fig. 3.

Rigid-body Motion Fig. 1 shows (a) a star pattern on a book cover and (b) the same pattern after undergoing a rigid-body motion in 3D (a rotation and translation of the book cover). The extracted patterns are shown in Fig. 1(c) as solid (original pattern) and dashed lines (transformed pattern). We use the second-order b-spline function of a uniform knot vector [4] in the basis expansion step. Fig. 1(d) shows the invariant signatures (based on Eq. 2) of the original (solid) and transformed (dashed) curves along the contours. (The starting point and traversal direction were manually picked.) As can be seen from the figure, the invariant signatures are quite consistent.

General Affine Transform Figs. 2(a) and (b) show a shirt with a dolphin imprint and a stretched and deformed version of the same imprint. Fig. 2(c) shows the extracted patterns. The invariant signatures are plotted in Fig. 2(d), and, again, they are quite consistent.

Change of Illumination To illustrate invariance under illumination changes, we placed different color filters in front of the light source. An example is shown below: Fig. 3 shows the same mouse pad under white and blue illumination. We randomly placed three circular curves—the red (left), green (bottom right), and blue (top right) curves in Fig. 3, and computed the invariant signatures along these three curves for both the images under white and blue illumination. Fig. 4(a), (b), and (c) show the invariant profiles computed from the white (solid)

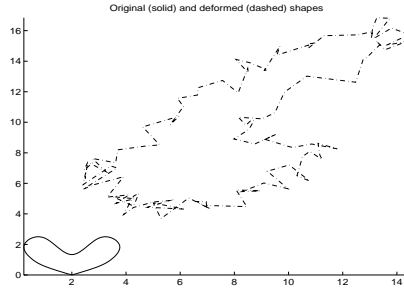


Fig. 5. Original and transformed shapes with noise added. Solid lines for original shapes and dashed lines for transformed and noise-corrupted shapes.

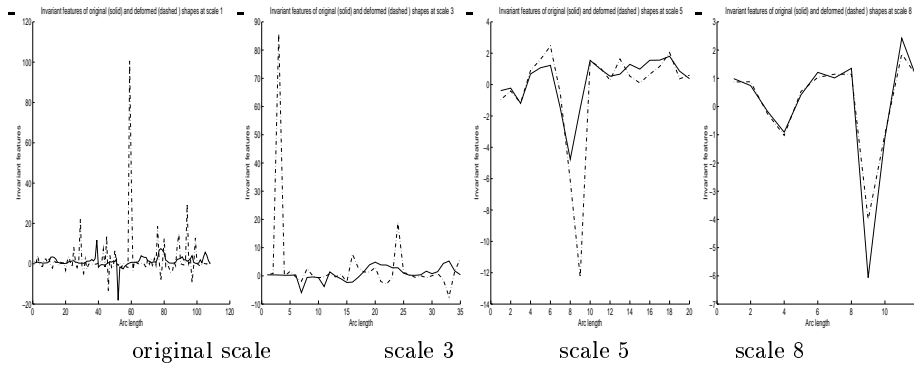


Fig. 6. Invariant shape descriptors at many scales for the original and deformed shapes with noise added. Solid lines for original shapes and shape descriptors and dashed lines for transformed and noise-corrupted shapes and shape descriptors.

and blue (dashed) illumination. As can be seen from the figure, they are quite consistent.

Hierarchical Invariant Analysis The additional degree of freedom in designing the basis function enables a hierarchical shape analysis. To illustrate, Fig. 5 shows the original (solid lines) and deformed (dashed lines) shapes with a significant amount of noise added. Our approach, which analyzes the shape at many scales locally, will discover the similarity which may manifest itself at different levels of detail. This is the case in Fig. 6, where shape similarity may not be apparent in the original shapes and shape descriptors, but eventually manifest itself from scale 5 onward. Traditional analysis relying on a single scale or requiring high-order derivatives of the contour function will have difficulty handling this and similar cases.

Perspective Invariants Fig. 7(a) and (b) show the canonical (head-on) view and another perspective of a package box, respectively. We extracted the arc-shape pattern from both images for verifying the perspective invariance. We display the predicted shapes superimposed on Fig. 7(b) after certain iterations of our algorithm. The predicted shape matched the real one well which verifies the perspective invariance.

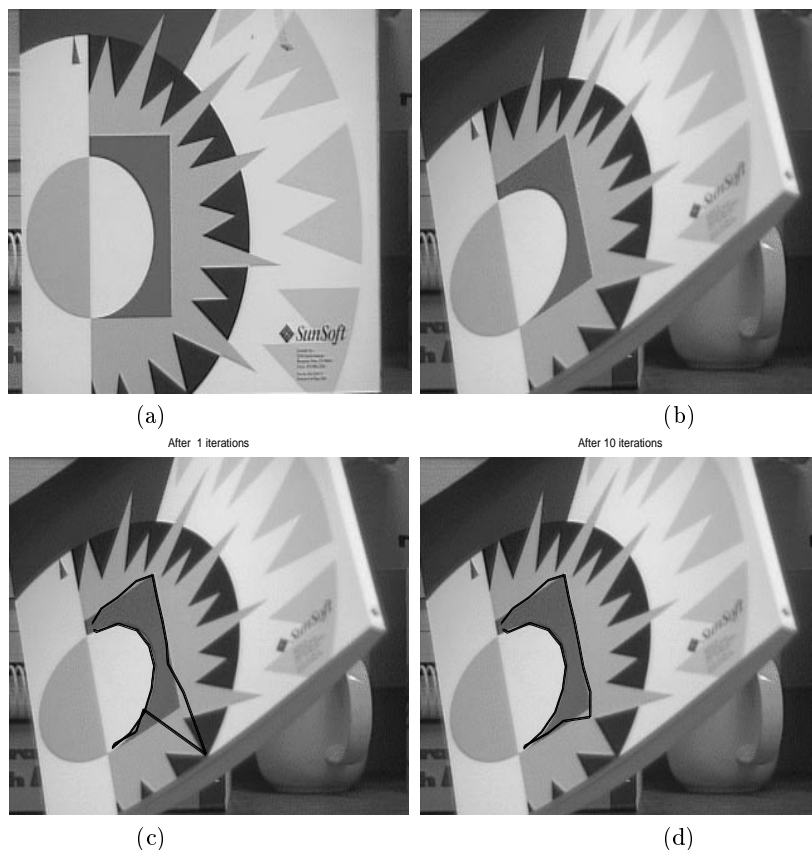


Fig. 7. (a) The canonical view, (b) another perspective, and (c) and (d) the estimated shapes after the 1st and 10th iterations.

4 Conclusion

We present a new framework for computing image invariants. The framework utilizes many desirable properties of wavelet and basis expansion techniques and is neither strictly global nor purely local. Furthermore, it is quite simple and straightforward to implement. Preliminary results on both real and synthetic images are very promising. These results demonstrate the tolerance to noise, change of luminance, and perspective distortion, and the ability for multi-scale analysis.

References

1. I. Daubechies. Orthonormal Bases of Compactly Supported Wavelets. *Commun. Pure Appl. Math.*, 41:909–960, 1988.
2. H. W. Guggenheimer. *Differential Geometry*. McGraw-Hill, New York, 1963.
3. T. H. Reiss. *Recognizing Planar Objects Using Invariant Image Features*. Springer-Verlag, Berlin, 1993.
4. D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics, 2nd Ed.* McGraw-Hill, New York, NY, 1990.
5. Y. F. Wang. A New Framework for Image Invariants using Basis Expansion. Technical Report TRCS97-02, Dept. of Computer Science, UCSB, Feb. 1997.
6. I. Weiss. Geometric Invariants and Object Recognition. *Int. J. Comput. Vision*, 10(3):207–231, 1993.

This article was processed using the \LaTeX macro package with LLNCS style