Geometric and Illumination Invariants for Object Recognition

Ronald Alferez and Yuan-Fang Wang
Department of Computer Science
University of California
Santa Barbara, CA 93106
E-mail: {ronald,yfwang}@cs.ucsb.edu

Abstract

We propose invariant formulations that can potentially be combined into a single system. In particular, we describe a framework for computing invariant features which are insensitive to rigid motion, affine transform, changes of parameterization and scene illumination, perspective transform, and view point change. This is unlike most current research on image invariants which concentrates on either geometric or illumination invariants exclusively. The formulations are widely applicable to many popular basis representations, such as wavelets [3, 4, 24, 25], short-time Fourier analysis [13, 35], and splines [2, 5, 37]. Exploiting formulations that examine information about shape and color at different resolution levels, the new approach is neither strictly global nor local. It enables a quasi-localized, hierarchical shape analysis which is rarely found in other known invariant techniques, such as global invariants. Furthermore, it does not require estimating high-order derivatives in computing invariants (unlike local invariants), whence is more robust. We provide results of numerous experiments on both synthetic and real data to demonstrate the validity and flexibility of the proposed framework.

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 [7, 21, 36]. The need for invariant image descriptors has long been recognized in computer vision [36, 47]. 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, aspect-based approaches [6, 8, 9]. Hence, it was even argued that object recognition is the search for invariants [47].

Image invariants can be designed to fit the needs of specific systems. Some require only that it be non-discriminating to an object's geometric pose or orientation. Others may be only interested in it being insensitive to the change of illumination. More complex systems, however, demand that it be insensitive to a combination of several environmental changes. Clearly, the latter case is more difficult to achieve.

Furthermore, invariant features can be designed based on many different methods. It can be computed either globally, which requires knowledge of the shape as a whole, or locally, which are based on local properties such as curvature and arc length. Global invariants suffer when some parts of the image data are unavailable (*i.e.*, occlusion). On the other hand, most local invariants have difficulties tolerating noise because its computation usually involves solving for high order derivatives.

Most current research has focused almost exclusively on single aspects of the problem, concentrating on a few geometric transformations, or on illumination invariants only. Sec. 2 reviews research done in the past which reflects this trend. Unfortunately, some formulations are difficult, if not impossible, to extend to handle a wider scope of transformations.

The proposed framework builds upon past research on image invariants. It allows for the design of invariants that are insensitive to rigid motion, general affine transform, changes of parameterization and scene illumination, perspective transform, and view point change. Furthermore, it tolerates a relatively large degree of noise.

Exploiting formulations that examine information about shape and color at different resolution scales, the new approach is neither strictly global nor local. We feel that the proposed framework has the following salient features:

1.) It is an invariant technique that enables a quasi-localized, hierarchical shape analysis. The additional degree of freedom in designing a basis function—be it the scale in wavelet, the frequency in short-time Fourier analysis, or the polynomial degree in spline (Fig. 1)—gives the technique an additional descriptive power that is rarely found in other invariant techniques. The result is an invariant framework which is more flexible and tolerant to noise.

2.) Unlike most current research which concentrates exclusively on either geometric invariants or illumination invariants, the proposed framework is very general and produces invariants insensitive to rigid motion, affine transform, changes of parameterization and scene illumination, noise, and perspective transform.

3.) The proposed framework is applicable to many basis functions. We propose to use the framework with wavelet, short-time Fourier analysis, and spline bases, which have been widely used in signal and speech processing, image analysis, computer vision, and computer graphics [3, 4, 5, 24, 25, 35, 37].

4.) It ameliorates some difficulties encountered in computing global or local image invariants. We employ basis functions of a compact support (wavelets, short-time Fourier analysis, and splines). Although the invariant features computed capture local shape traits, it does not require estimating high order derivatives unlike in the case of invariants strictly using local analysis. Whence, the new method is more robust.

5.) We introduce the use of rational basis functions to facilitate the analysis of invariants under perspective transform. Rational basis functions, such as NURBS, have been widely used in the computer graphics community [2, 5, 10, 33, 44]. However, their usage in perspective invariants is novel.

The remainder of this paper is organized as follows: Sec. 2 reviews related work done in the past, Sec. 3 presents the framework of image-derived invariants. Sec. 4 presents some experimental results, and finally, Sec. 5 contains concluding remarks.

2 Review of Related Literature

Geometric invariants such as those insensitive to affine and perspective transformations have been studied extensively. Similarly, many have focused their research on illumination invariants. A few have attempted to combine the two.

Invariants under affine transformations have been studied by [20] using Hough-based methods,

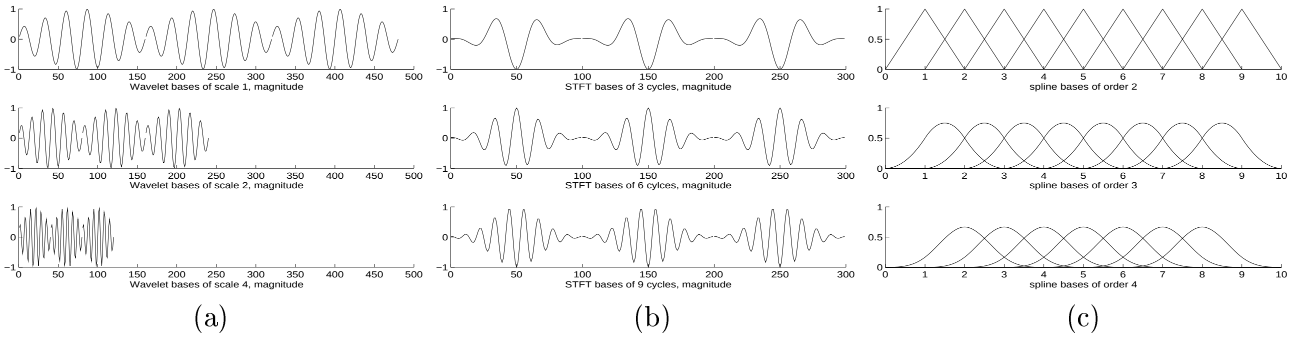


Figure 1: (a) Wavelet, (b) short-time Fourier analysis, and (c) b-spline bases showing two orthogonal dimensions.

by [1, 30] using Fourier descriptors, and by [43] using wavelets. In these cases, affine invariants were used to recognize planar objects in 3D space. Orthographic projection was used to approximate perspective projection, and the "shear" effect in affine transformations modeled perspective distortion. Hence, the assumption is that the size of the observed object is small relative to its distance from the camera, *i.e.*, a weak perspective.

Under large perspective distortion, however, a more rigorous treatment of perspective invariants is needed. Lei [22] demonstrated how cross ratios can be used to recognize planar objects in 3D space. In this case, "true" perspective invariants were formulated. However, objects were restricted to polygons and required accurate identification of vertex positions.

Illumination invariants have also been studied extensively in [12, 16, 15, 27, 28, 29, 40, 41, 46]. These invariants allowed for changes that may include altering the position and number of light sources, the brightness and contrast, and even hue. Illumination invariants have been applied to recognize textures [17], 3D objects [39], and 3D textures [19]. A physics-based approach is used in [29], to produce illumination invariants from infrared imagery. Many, including our proposed technique, assumes a Lambertian surface model [11] for simplicity.

There has been limited success in combining geometric and illumination invariants. For instance, Slater and Healey [39] used local color invariants to recognize 3D objects. In this study, they derived invariants of local color pixel distributions, which were independent of the position and orientation of an object's surface. Recognition of the object's actual position and pose, however, was achieved using a technique that is similar to template matching. The position was estimated by sliding a set of circular windows over the entire image, while object distance was estimated by trying all possible sizes of the circular window. The advantage here is that segmentation is not necessary. The disadvantage, however, is apparent. Computing invariants for each region in the image is necessary for recognition. Furthermore, the allowable range of the object's distance from the camera must be pre-determined, with each window size corresponding to one possible distance measure. This comes close to estimating geometric features by brute-force methods. Their later study [41] describes a recognition system invariant to illumination, rotation and scale. Scale invariance was limited to regions that were locally radially invariant, *i.e.*, the circular image region appears the same, regardless of radius.

Another study [48] proposed an algorithm for classifying textures invariant to rotation and gray-scale transformation. It used *spiral resampling*, subband decomposition, and Hidden Markov Model. Two-dimensional texture images were converted to 1-D signals in a spiral fashion to achieve rotation invariance. This, however, meant that it was limited to recognizing regions that looked similar, independent of its position in the image. While this might be suitable for recognizing textures, extending it to objects in general is nontrivial.

A significant feature of our proposed technique is its ability to perform hierarchical shape analysis. This is possible because of the additional degree of freedom in designing a basis function *e.g.*, the specification of which scale to use in wavelets. Recent research has exploited the properties of wavelets to formulate invariants that allow analysis to be performed at different resolution levels. Dyadic wavelets were used in [42] to decompose object contours into several components at different resolution levels. The resolution levels that were to be used for matching were pre-determined, by selecting the levels where most of its energy was concentrated. Data was analyzed in the frequency domain to limit the effect of noise, but at the same time, spatial information was preserved to establish point correspondence. Hence, the advantages of spatial and frequency domain methods were combined. The result was a curve representation invariant to translation, rotation, and scaling. They used a similar technique in [43] to formulate affine invariants to recognize planar objects in 3D space. However, only a weak perspective was assumed.

Recently, it has become popular to use wavelets (or quadrature mirror filters QMF's) in decomposing and representing signals at multiple scales [38, 26]. Indeed, we use wavelets to achieve multi-resolution analysis in our invariant formulation. However, one caveat in doing this is that orthogonal wavelets are critically sampled. They achieve representation through scaling and translation. They are not invariant to translation, i.e., the content of wavelet subbands is unstable under a translation of input signals. The result is that even though basis functions at a scale are translated versions of each other, it does not imply that the transform coefficients behave in the same way when the input signal undergoes a simple translation. One remedy was proposed by Simoncelli, et. al. [38], where they define a shiftable transform in which the information represented within a subband remains in the same subband as the signal is translated. Another approach is described by Mallat [26], wherein a signal uses the local extreme in its wavelet transform domain to make it invariant to time shifts. Properties of the wavelet transform are discussed in [45].

A comprehensive survey on the subject of invariants in general is presented by [36]. A review of geometric invariants is presented in [31, 47]. Other approaches to invariants and recognition are discussed in [34, 31, 23]. Numerous papers on invariance, with emphasis in their applications in computer vision, can be found in [32].

3 Technical Rationale

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) while a 2D coordinate as (x, y), and coordinates (bold for vector quantities) of a 2D curve (small letter for 2D quantities) will be denoted by $\mathbf{c}$.

We will illustrate the mathematical frameworks using specific scenarios where invariants for curves are sought. For shape invariants, these directly apply to the silhouette (contour) of imaged objects. For illumination invariants, the same technique applies by linearizing internal regions by a characteristic sampling curve and computing invariant color signatures along the characteristic curve. In both cases, the invariant signatures produced can be examined at different resolution scales, making the invariant features both flexible and noise tolerant. The particular basis functions we will use in the illustration are the wavelet bases and spline functions. However, the same framework can be easily extended to other bases such as the short-time Fourier analysis, and for 3D surfaces.

We first examine a proper parameterization under affine transformations. We then consider variation in an object's image induced by rigid motion, general affine transform, changes in parameterization and scene illumination, and perspective projection. In each case, we give examples of how invariants can be designed based on simple basis expansion. It is assumed that the object contours have been properly extracted (*i.e.*, the background can easily be distinguished from the object.) Segmentation is a difficult problem in itself, and is beyond the scope of this study.

Each formulation can be used alone, or in conjunction with others. For example, the formulation for perspective invariants can be used solely, or it may be combined with the formulation for illumination invariants, so that it becomes insensitive to both perspective and illumination transformations. Furthermore, invariant signatures can be examined separately, at different resolution levels. This hierarchical approach makes the invariant features both flexible and noise tolerant.

Affine Invariant Parameterization When defining parameterized curves $\mathbf{c}(t) = [x(t), y(t)]^T$, most prefer to use the intrinsic arc length parameter, t, because of its simplicity. Intrinsic arc length transforms linearly under any rigid-body transformation. Translation and rotation do not affect the arc length, and scaling only scales the parameter uniformly. However, under affine transformation, the arc length parameter is nonlinearly transformed [1]. A more suitable parameterization is thus required. We describe two parameterizations which are linear under an affine transformation.

The first, called *affine arc length*, is defined [14] as:

$$\tau = \int_a^b \sqrt[3]{\dot{x}\ddot{y} - \ddot{x}\dot{y}}\ dt$$

where $\dot{x}, \dot{y}$ are the first and $\ddot{x}, \ddot{y}$ are the second derivatives with respect to any parameter t (possibly the arc length), and (a, b) is the path along a segment of the curve.

The affine arc length parameter is proportional to the curvature of the contour. Hence, on high (low) curvature segments of the curve, the parameter covers less (more) distance, producing a shorter (longer) projected segment. This parameter transforms linearly under a general affine transform. It can easily be made an absolute invariant by normalizing it with respect to the the affine arc length of the entire curve.

Because the parameterization involves second derivatives, it becomes susceptible to noise effects. If the points along the curve is discretized (say into pixels), the curve becomes a polyline (or polygon if enclosed). The effect is a parameterization which is zero along the sides of the polyline, and infinite at the vertices. To avoid this, [1] using a first order form, defined a second parameter which some later called the *enclosed area parameter*:

$$\sigma = \frac{1}{2} \int_a^b |x\dot{y} - y\dot{x}|\ dt.$$

The drawback here is that this parameter is not invariant to translation, and requires a closed contour. This can easily be remedied by moving the object's coordinate system to its centroid. The curve can be forced to close by drawing a line between the first and last vertices of the curve.

One can interpret the enclosed area parameter as the area of the triangular region enclosed by the two line segments from the centroid to two points a and b on the contour, respectively. Since the affine transform linearly changes area, a parameterization that sweeps a constant area will be an invariant of weight 1 [36, 47]. By normalizing this parameter with respect to the total enclosed area of the contour, it too can be made completely invariant to affine transform.

Of course, these parameters assume knowledge of a one-point correspondence (*i.e.*, the starting point), and knowledge of the direction of the contour (*i.e.*, clockwise or counter-clockwise). It can be easily shown that the invariant signatures of two contours, differing only by the starting point, is just a phase-shifted version of each other. Similarly, two contours parameterized in opposing directions is just a mirror image of each other. Hence, a match can be chosen that maximizes the cross-correlation between two signatures.

Allowing an arbitrary change of origin and traversal direction, together with the use of an affine invariant parameterization, implies that *no point correspondence is required*, when computing affine invariants.

In the case of large perspective distortion, however, these affine invariant parameterizations will not be able to produce exact point correspondence. In fact, for the enclosed area parameter, the centroid of one contour may not accurately correspond to the centroid of the other contour. However, our experience indicated that these affine invariant parameters still provide a good initial estimate for finding the correct point correspondence. In the *perspective transform section*, we provide a method for adjusting the position of each corresponding point to account for perspective foreshortening. To increase its accuracy, the centroid may also be adjusted over the local vicinity iteratively, and choosing the one which maximizes perspective invariance. Furthermore, geometrical cues, such as points of discontinuity (*i.e.*, sharp turns), can be used as a rough guide for accurate point correspondence.

Rigid Motion and Affine Transform Consider a 2D curve, where t denotes a parameterization which is invariant under affine transform (as described above),

$$\mathbf{c}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

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 [4]) as

$$\mathbf{u}_{a,b} = \int \mathbf{c}\psi_{a,b} dt \ .$$

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

$$\mathbf{c}'(t) = \mathbf{m}\mathbf{c}(t') + \mathbf{t} = \mathbf{m}\mathbf{c}(\pm t + t_0) + \mathbf{t} = \mathbf{m} \begin{bmatrix} x(\pm t + t_0) \\ y(\pm t + t_0) \end{bmatrix} + \mathbf{t} \ , \tag{1}$$

where $\mathbf{m}$ is any nonsingular 2×2 matrix, $\mathbf{t}$ represents the translational motion, t_0 represents a change of the origin in traversal, and $\pm$ represents the possibility of traversing the curve either counterclockwise or clockwise [1]. It follows that:

$$\begin{aligned}
\mathbf{u}'_{a,b} &= \int \mathbf{c}'\psi_{a,b}dt \\
&= \int(\mathbf{mc}(\pm t + t_0) + \mathbf{t})\psi_{a,b}dt & &= \int \mathbf{mc}(\pm t + t_0)\psi_{a,b}dt + \int \mathbf{t}\psi_{a,b}dt \\
&= \mathbf{m}\int \mathbf{c}(\pm t + t_0)\psi_{a,b}dt + \mathbf{t}\int \psi_{a,b}dt & &= \int \mathbf{mc}(t')\tfrac{1}{\sqrt{a}}g(\tfrac{\mp(t'-t_0)-b}{a})dt' + \int \mathbf{t}\psi_{a,b}dt \qquad (2) \\
&= \mathbf{m}\int \mathbf{c}(t')\tfrac{1}{\sqrt{a}}g(\tfrac{t'-(\pm b+t_0)}{a})dt' & &= \mathbf{m}\int \mathbf{c}(t')\psi(t')_{a,\pm b+t_0}dt' \\
&= \mathbf{mu}_{a,\pm b+t_0} \ .
\end{aligned}$$

Note that we use the wavelet property $\int \psi_{a,b}dt = 0$ to simplify the second term in Eq. 2. Hence, the transformed curve can be generated using the transformed wavelet coefficients and the same wavelet bases, instead of transforming the curve point-by-point. This is an observation which is commonly made in the computer graphics community about curves generated by the spline functions and associated control vertices [2, 5, 37]. In that sense, $\mathbf{u}_{a,b}$'s function the same way as the control vertices in a spline curve.

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

$$\frac{\left|\mathbf{u}'_{a,b}\right|}{\left|\mathbf{u}'_{c,d}\right|} = \frac{|\mathbf{mu}_{a,\pm b+t_0}|}{|\mathbf{mu}_{c,\pm d+t_0}|} = \frac{|\mathbf{u}_{a,\pm b+t_0}|}{|\mathbf{u}_{c,\pm d+t_0}|} \ . \qquad (3)$$

The resulting wavelet coefficients $\mathbf{u}'_{a,b}$ and $\mathbf{u}_{a,\pm b+t_0}$ are functions of two variables: the scale a and the displacement b and $\pm b + t_0$. If we hold the scale a constant, by taking the same number of sample points in each curve, we can construct expressions based on correlation coefficients to cancel out the effect of a different traversal starting point (t_0) and direction ($\pm t$). Therefore, *no point correspondence information is required* in the analysis.

Let us define

$$f(x) = \frac{|\mathbf{u}_{a,x}|}{|\mathbf{u}_{a,x+x_0}|} \qquad (4)$$

and

$$f'(x) = \frac{\left|\mathbf{u}'_{a,x}\right|}{\left|\mathbf{u}'_{a,x+x_0}\right|} = \frac{|\mathbf{mu}_{a,\pm x+t_0}|}{\left|\mathbf{mu}_{a,\pm(x+x_0)+t_0}\right|} = \frac{|\mathbf{u}_{a,\pm x+t_0}|}{\left|\mathbf{u}_{a,\pm(x+x_0)+t_0}\right|} \ , \qquad (5)$$

where x_0 represents a constant value separating two indices. Thus, $f(x)$ and $f'(x)$ represent the signatures of the original curve and the transformed curve, respectively. Then we have two cases:

Case 1: *The direction of traversal is the same for both contours.*

$$f'(x) = \frac{|\mathbf{u}_{a,x+t_0}|}{|\mathbf{u}_{a,x+x_0+t_0}|} = f(x + t_0)$$

We can compute the correlation coefficient as

$$R_{f(x)f'(x)}(\tau) = \frac{\int f(x)f'(x+\tau)dx}{\|f\| \cdot \|f'\|} = \frac{\int f(x)f(x+\tau+t_0)dx}{\|f\| \cdot \|f'\|}$$

[1] In the implementation, the parameter is computed in a modular manner over a closed contour.

which is maximized when $\tau = -t_0$.
Case 2: *The direction of traversal is reversed.*

$$f'(x) = \frac{|\mathbf{u}_{a,-x+t_0}|}{|\mathbf{u}_{a,-x-x_0+t_0}|} = \frac{1}{f(-x-x_0+t_0)}$$

We can compute the correlation coefficient as

$$R_{f(x)\frac{1}{f'(-x)}}(\tau) = \frac{\int f(x)\frac{1}{f'(-x+\tau)}dx}{\|f\| \cdot \|\frac{1}{f'}\|} = \frac{\int f(x)f(x-\tau-x_0+t_0)dx}{\|f\| \cdot \|\frac{1}{f'}\|}$$

which is maximized when $\tau = t_0 - x_0$.

Hence, we take the larger of the two $R(\tau)$'s to obtain the invariant measure:

$$I(f,f') = max_{\tau,\tau'}\{R_{f(x)f'(x)}(\tau), R_{f(x)\frac{1}{f'(-x)}}(\tau')\} \ . \tag{6}$$

Other invariant features based on ratios may still be derived where the same technique can be employed to measure similarity, making it independent of the parameterization used. For simplicity, we only show the invariant expressions from this point on.

If the second term in Eq. 2 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 [37]), then invariant expressions can still be derived, albeit in a slightly more complicated form:

$$\begin{aligned} \frac{\left|\mathbf{u}'_{a,b}-\mathbf{u}'_{c,d}\right|}{\left|\mathbf{u}'_{e,f}-\mathbf{u}'_{g,h}\right|} &= \frac{\left|(\mathbf{m}\mathbf{u}_{a,\pm b+t_0}+\mathbf{v})-(\mathbf{m}\mathbf{u}_{c,\pm d+t_0}+\mathbf{v})\right|}{\left|(\mathbf{m}\mathbf{u}_{e,\pm f+t_0}+\mathbf{v})-(\mathbf{m}\mathbf{u}_{g,\pm h+t_0}+\mathbf{v})\right|} \\ &= \frac{\left|\mathbf{m}(\mathbf{u}_{a,\pm b+t_0}-\mathbf{u}_{c,\pm d+t_0})\right|}{\left|\mathbf{m}(\mathbf{u}_{e,\pm f+t_0}-\mathbf{u}_{g,\pm h+t_0})\right|} \\ &= \frac{\left|(\mathbf{u}_{a,\pm b+t_0}-\mathbf{u}_{c,\pm d+t_0})\right|}{\left|(\mathbf{u}_{e,\pm f+t_0}-\mathbf{u}_{g,\pm h+t_0})\right|} \ , \end{aligned}$$

where $\mathbf{v}$ denotes the constant second term in Eq. 2.

For invariants under general affine transform, many forms using ratios, cross ratios, and ratios of ratios have already been derived [36, 47]. For example, it is known that the cross ratio of four collinear points are invariant under the 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 [36, 47]). So we have:

$$\begin{vmatrix} \mathbf{u'_{a,b}} & \mathbf{u'_{c,d}} & \mathbf{u'_{e,f}} \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \mathbf{u_{a,\pm b+t_0}} & \mathbf{u_{c,\pm d+t_0}} & \mathbf{u_{e,\pm f+t_0}} \\ 1 & 1 & 1 \end{vmatrix} \ .$$

Hence, we have the following invariants [2] :

$$\frac{\begin{vmatrix} \mathbf{u'_{a,b}} & \mathbf{u'_{c,d}} & \mathbf{u'_{e,f}} \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} \mathbf{u'_{g,h}} & \mathbf{u'_{i,j}} & \mathbf{u'_{k,l}} \\ 1 & 1 & 1 \end{vmatrix}} = \frac{\begin{vmatrix} \mathbf{u_{a,\pm b+t_0}} & \mathbf{u_{c,\pm d+t_0}} & \mathbf{u_{e,\pm f+t_0}} \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} \mathbf{u_{g,\pm h+t_0}} & \mathbf{u_{i,\pm j+t_0}} & \mathbf{u_{k,\pm l+t_0}} \\ 1 & 1 & 1 \end{vmatrix}} \ . \tag{7}$$

[2] Note that there are many valid expressions for affine invariants. Some may require a smaller number of coefficients than that in Eq. 7. For example, when wavelet bases are used where $\int \psi_{a,b}dt = 0$, Eq. 7 can be simplified as

$$\frac{\begin{vmatrix} \mathbf{u'_{a,b}} & \mathbf{u'_{c,d}} \end{vmatrix}}{\begin{vmatrix} \mathbf{u'_{e,f}} & \mathbf{u'_{g,h}} \end{vmatrix}} = \frac{\begin{vmatrix} \mathbf{u_{a,\pm b+t_0}} & \mathbf{u_{c,\pm d+t_0}} \end{vmatrix}}{\begin{vmatrix} \mathbf{u_{e,\pm f+t_0}} & \mathbf{u_{g,\pm h+t_0}} \end{vmatrix}} \ ,$$

where only four coefficients are needed.

Perspective Transform The affine invariant derived above assumes a planar curve under parallel projection. Distortion caused by perspective foreshortening was not addressed. Allowing perspective transform with an arbitrary view point and large perspective distortion makes the problem much harder as the projection is a non-linear process, involving a division in computing 2D coordinates. Extending the curve to 3D makes it even more difficult.

A simplified model is possible, using a parallel or quasi-perspective model. In this case, the size of the object is assumed to be much smaller than the average distance of the object to the camera. Then the projection can be linearized by replacing the division by varying Z coordinates with a multiplication of a constant magnification factor. In essence this reduces the problem of *perspective* invariants to one of *affine* invariants, which is linear. However, such simplification holds only to a certain degree under a small perspective distortion.

Fortunately, a more rigorous treatment of perspective invariants is possible. The projection process can be linearized using a tool which is well-established in computer graphics, the *rational* form of a basis function [2, 5, 10, 33, 44]. The most famous of such an expression is probably NURBS (Non-Uniform Rational B-Spline), which was adopted as a standard for IGES (Initial Graphics Exchange Specification) [18]. By using a rational basis form, we will show that perspective invariance can be verified efficiently and in a linear manner.

We will use NURBS for illustration. In a nutshell, a b-spline function is a polynomial of a finite support. Non-rational b-spline functions of order k (or a polynomial of degree $k-1$) are generated by the Cox-deBoor recursion formulas [37]:

$$N_{i,1}(t) = \begin{cases} 1 & if\ x_i \le t < x_{i+1} \\ 0 & otherwise \end{cases} \quad , \quad \text{and}$$

$$N_{i,k}(t) = \frac{(t-x_i)N_{i,k-1}(t)}{x_{i+k-1}-x_i} + \frac{(x_{i+k}-t)N_{i+1,k-1}(t)}{x_{i+k}-x_{i+1}} \ .$$

The values of x_i are elements of a knot vector satisfying the relations $x_i \le x_{i+1}$. The rational form of a b-spline function in 2D (3D) is the projection of a non-rational b-spline function in 3D (4D). Specifically, let

$$\mathbf{C}(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \sum_i \mathbf{P}_i N_{i,k}(t) = \sum_i \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} N_{i,k}(t)$$

represent a non-rational curve in 3D with $\mathbf{P}_i$'s as its control vertices. Its projection in 2D will be:

$$\begin{aligned} \mathbf{c}(t) &= \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \frac{X(t)}{Z(t)} \\ \frac{Y(t)}{Z(t)} \end{bmatrix} = \textstyle\sum_i \mathbf{p}_i R_{i,k}(t) \\ &= \textstyle\sum_i \begin{bmatrix} x_i \\ y_i \end{bmatrix} R_{i,k}(t) = \sum_i \begin{bmatrix} \frac{X_i}{Z_i} \\ \frac{Y_i}{Z_i} \end{bmatrix} R_{i,k}(t) \ , \quad \text{where} \end{aligned} \tag{8}$$

$$R_{i,k}(t) = \frac{Z_i N_{i,k}(t)}{\sum_j Z_j N_{j,k}(t)} \ , \tag{9}$$

and $\mathbf{p}_i$'s represent the projected control vertices in 2D. We assume the focal length of the camera is 1.

To illustrate, Fig. 2 shows the rational b-spline bases of order four (third-degree polynomial) for an open knot vector $[X] = [000012222]$ [37]. By varying the Z coordinates of the associated

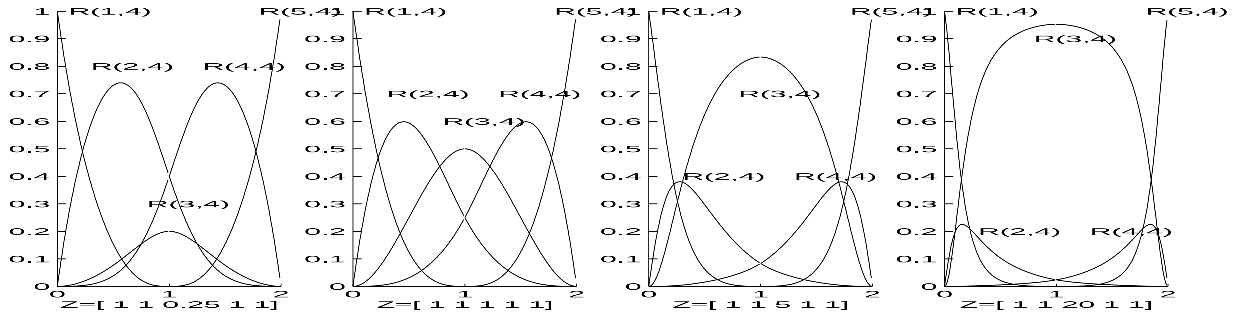


Figure 2: Various rational b-spline bases of order 4 with an open knot vector $[X] = [000012222]$ and five control vertices. The middle control vertex was placed at different depths ranging from 0.5, 1, 5, to 20 from left to right. All others are at a depth of 1.

control vertices, the shape of the bases adapts accordingly. This adaptation correctly accounts for the perspective distortion in image formation, and a control vertex can exert greater (or smaller) influence on the shape and traversal speed of the image curve given the particular perspective. When all the Z's are equal, the rational bases reduce to the non-rational bases as expected in Eq. 9.

Fig. 3 shows sample b-spline curves using both periodical and open knot vectors [37]. Column (a) in Fig. 3 shows curves generated using non-rational spline bases in space. Columns (b) and (c) show the projections of the 3D curves using two different methods: One is generating points along the 3D curves in Fig. 3(a), then projecting them one by one. The other is projecting only the control vertices and then using the rational spline bases to interpolate them in 2D. The two methods produce identical results in Fig. 3.

We can now formulate the problem of finding perspective invariants as a curve fitting problem. Intuitively, 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.)

In more details, if we let the "canonical" view of a general, 3D space curve be represented as in Eq. 3, then for all other views we have:

$$\mathbf{P'}_i = \begin{bmatrix} X'_i \\ Y'_i \\ Z'_i \end{bmatrix} = \mathbf{MP}_i + \mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} r_{11}X_i + r_{12}Y_i + r_{13}Z_i + T_x \\ r_{21}X_i + r_{22}Y_i + r_{23}Z_i + T_y \\ r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_z \end{bmatrix} , \qquad (10)$$

where $\mathbf{M}$ and $\mathbf{T}$ represent the rotation and translation in between the two views, respectively. Hence perspective projection gives

$$\mathbf{p}'_i = \begin{bmatrix} \frac{X'_i}{Z'_i} \\ \frac{Y'_i}{Z'_i} \end{bmatrix} = \begin{bmatrix} \frac{r_{11}X_i + r_{12}Y_i + r_{13}Z_i + T_x}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_z} \\ \frac{r_{21}X_i + r_{22}Y_i + r_{23}Z_i + T_y}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_z} \end{bmatrix} \qquad (11)$$

$$R'_{i,k} = \frac{(r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_z)N_{i,k(t)}}{\sum_j (r_{31}X_j + r_{32}Y_j + r_{33}Z_j + T_z)N_{j,k(t)}} \; . \qquad (12)$$

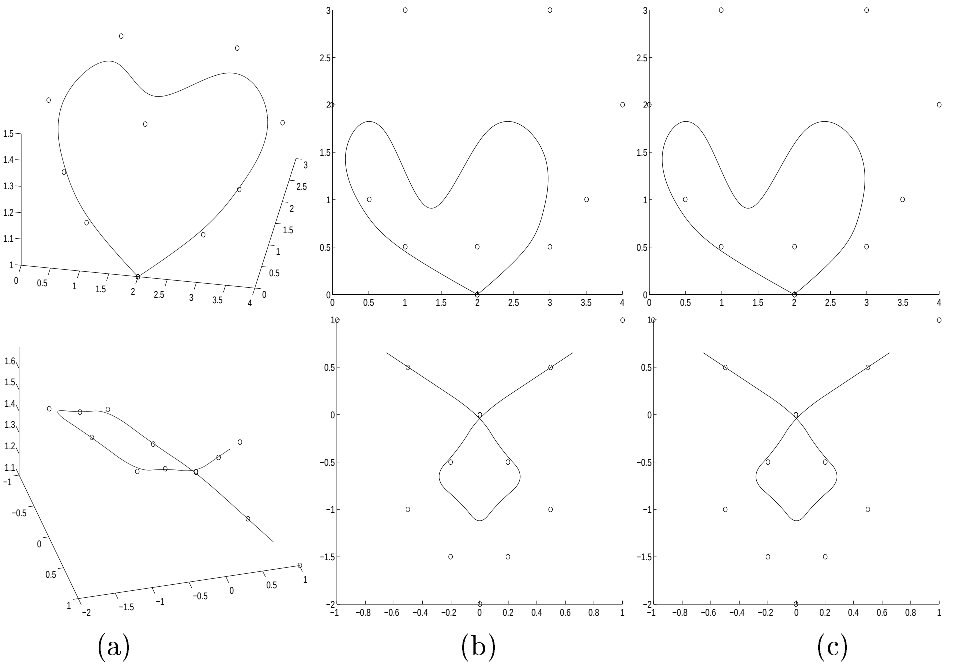


Figure 3: (a) 3D b-spline curves generated using non-rational bases, (b) 2D b-spline curves generated by projecting points on the 3D curves one by one, and (c) 2D b-spline curves generated by projecting the control vertices and interpolating them with the rational bases in 2D. "o" marks a control vertex location.

Image invariant defined by the goodness of fitting is:

$$I = \int_t (\mathbf{d}(t) - \sum_i \mathbf{p}'_i R'_{i,k}(t))^2 \ , \tag{13}$$

where $\mathbf{d}(t)$ denotes the observed image curve. There are a total of twelve unknowns in Eqs. 11 and 12, nine for rotation and three for translation. The number of rotation unknowns can be reduced to three which specify the direction of the rotation axis and the rotation angle around the axis. However, computing such an invariant can still be very expensive in a general scenario where the camera undergoes an arbitrary motion and a search in a six dimensional space is required.

By using rational bases, it is possible to drastically reduces the search effort in verifying the perspective invariance in Eq. 13 while maintaining the linearity of the formulation. Observe that in Eq. 8 the shape of a 2D curve is determined by the *projected control vertices* and the *rational* spline bases, both of which are unknown. To estimate both the locations of the projected control vertices and the rational spline bases can be a highly nonlinear process. Instead, *our approach minimizes I by a two-step gradient descent which maintains the linearity of the whole formulation.* The algorithm comprises the following two steps:

STEP 0: Initialization Specification of the rational bases requires knowledge of the depth of the control vertices Z_i, which is not known. As a first approximation we can assume that all Z_i's are equal. This is equivalent to approximating the *rational* bases using the corresponding *non-rational* bases, which do *not* require knowledge of the object depth. Using the non-rational bases in place of the rational bases in Eq. 8, we estimate the 2D control vertex positions.

Furthermore, affine invariant parameters can be used as an initial estimate for point correspondence. Since affine transformations assume a weak perspective model, we will not obtain exact correspondence. However, the points will be adjusted in the succeeding steps to account for perspective foreshortening.

STEP 1: Iterative Update Observe from Eq. 13

$$dI = \sum_i (\frac{\partial I}{\partial \mathbf{p_i}} d\mathbf{p_i} + \frac{\partial I}{\partial R_{i,k}} dR_{i,k}) \ ,$$

and we drop the prime symbol to simplify the notation from now on. This suggests that minimization can be broken into two stages: that of updating 2D control vertex positions ($d\mathbf{p}_i$) and that of updating rational bases ($dR_{i,k}$).

Updating rational bases: Continuing from Step 0, the estimated 2D control vertex positions are used to constrain the unknown rotation and translation parameters using Eq. 11. A *linear* formulation results using at least six 2D control vertices estimated from Eq. 8. Note that even though there are twelve motion parameters, one of them contributes to a scale change and cannot be determined uniquely. Furthermore, in the case where the 3D curve is planar (or $Z_i = 0$ in Eq. 10), four 2D control vertex positions will suffice. The motion parameters allow $R_{i,k}$'s to be updated using Eq. 12.

Updating 2D control vertices: The updated $R_{i,k}$'s allow a better prediction of the appearance of the curve in images, and any discrepancy in the predicted and actual appearance of the curve is used in a gradient search to further verify the consistency. The prediction involves updating the parameterization t and the 2D control vertex positions $\mathbf{p}_i$ (using Eq. 13).

The updated control vertex positions are then used to estimate the unknown motion parameters through Eq. 11. Hence a recursive process results to refine the positions of the 2D control vertices, the shapes of the rational spline functions, the parameterization, and the 3D motion parameters, until a convergence is achieved. The procedure is summarized in the following pseudo-code program:

```
• Step 0 Initialize
    – the iteration number to 0, or $n \longleftarrow 0$
    – $Z_i^{(0)} \longleftarrow 1$ and $R_{i,k}^{(0)} \longleftarrow N_{i,k}$ for all $i$
    – $r_{i,j}^{(0)} \longleftarrow 0$ and $T_k^{(0)} \longleftarrow 0$, for $i = 1,2,3$, $j = 1,2,3$, and $k = x,y$.
    – Solve $\begin{bmatrix} x_l^{(0)} \\ y_l^{(0)} \end{bmatrix}$ using $\mathbf{p}(t)$ and the current estimate of $R_{i,k}^{(0)}$ based on Eq. 8.
• Step 1 Repeat
    Step 1.a Use $\begin{bmatrix} x_l^{(n)} \\ y_l^{(n)} \end{bmatrix}$ to estimate $r_{i,j}^{(n+1)}$ and $T_k^{(n+1)}$, $i = 1,2,3$, $j = 1,2,3$, and $k = x,y$, based on Eq. 11.
    Step 1.b Update $R_{i,k}^{(n+1)}$ using Eq. 12.
    Step 1.c Use $r_{i,j}^{(n+1)}$ and $T_k^{(n+1)}$, $i = 1,2,3$, $j = 1,2,3$, and $k = x,y$, and $R_{i,k}^{(n+1)}$ to compute $\mathbf{p}_i^{(n+1)}$ and $\mathbf{c}^{(n+1)}(t)$ based on Eqs. 8 and 11.
    Step 1.d Update the parameterization of the predicted curve for a better correspondence with that of the observed image curve.
    Step 1.e Compute the deviation between $\mathbf{d}(t)$ and $\mathbf{c}^{(n+1)}(t)$ to generate a gradient direction $\delta\mathbf{p}_i^{(n+1)}$ for updating 2D control vertex positions.
```

Step 1.f Update 2D control vertices $\mathbf{p}_i^{(n+1)}$ by $\mathbf{p}_i^{(n+1)'} = \mathbf{p}_i^{(n+1)} - \alpha\delta\mathbf{p}_i^{(n+1)}$ where α is the update rate.
Step 1.g If $|r_{i,j}^{(n+1)} - r_{i,j}^{(n)}|$ and $|T_k^{(n+1)} - T_k^{(n)}|$, $i = 1,2,3$, $j = 1,2,3$, and $k = x,y$, are smaller than a preset threshold, then stop. Otherwise $n \longleftarrow n+1$, go back to **Step 1**.

For **Step 0** above, assume that J points, $\mathbf{d}(t_j) = [x(t_j), y(t_j)]^T, j = 0 \cdots J-1$ are sampled from the image curve. Then it is readily shown that $\mathbf{p}_i$ can be estimated by:

$$\begin{bmatrix} R_{0,k}^{(0)}(t_0) & R_{1,k}^{(0)}(t_0) & \cdots & R_{I-1,k}^{(0)}(t_0) \\ R_{0,k}^{(0)}(t_1) & R_{1,k}^{(0)}(t_1) & \cdots & R_{I-1,k}^{(0)}(t_1) \\ \cdots & \cdots & \cdots & \cdots \\ R_{0,k}^{(0)}(t_{J-1}) & R_{1,k}^{(0)}(t_{J-1}) & \cdots & R_{I-1,k}^{(0)}(t_{J-1}) \end{bmatrix} \begin{bmatrix} x_0^{(0)} & y_0^{(0)} \\ x_1^{(0)} & y_1^{(0)} \\ \cdots & \cdots \\ x_{I-1}^{(0)} & y_{I-1}^{(0)} \end{bmatrix} = \begin{bmatrix} x(t_0) & y(t_0) \\ x(t_1) & y(t_1) \\ \cdots & \cdots \\ x(t_{J-1}) & y(t_{J-1}) \end{bmatrix}, \quad (14)$$

where t_j is approximated by an affine invariant parameter.

For **Step 1.a**, one can easily show that

$$\begin{bmatrix} X_0 & Y_0 & Z_0 & 0 & 0 & 0 & -X_0x_0^{(n)} & -Y_0x_0^{(n)} & -Z_0x_0^{(n)} & 1 & 0 \\ 0 & 0 & 0 & X_0 & Y_0 & Z_0 & -X_0y_0^{(n)} & -Y_0y_0^{(n)} & -Z_0y_0^{(n)} & 0 & 1 \\ X_1 & Y_1 & Z_1 & 0 & 0 & 0 & -X_1x_1^{(n)} & -Y_1x_1^{(n)} & -Z_1x_1^{(n)} & 1 & 0 \\ 0 & 0 & 0 & X_1 & Y_1 & Z_1 & -X_1y_1^{(n)} & -Y_1y_1^{(n)} & -Z_1y_1^{(n)} & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & \\ X_{I-1} & Y_{I-1} & Z_{I-1} & 0 & 0 & 0 & -X_{I-1}x_{I-1}^{(n)} & -Y_{I-1}x_{I-1}^{(n)} & -Z_{I-1}x_{I-1}^{(n)} & 1 & 0 \\ 0 & 0 & 0 & X_{I-1} & Y_{I-1} & Z_{I-1} & -X_{I-1}y_{I-1}^{(n)} & -Y_{I-1}y_{I-1}^{(n)} & -Z_{I-1}y_{I-1}^{(n)} & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ T_x \\ T_y \end{bmatrix} = \begin{bmatrix} x_0^{(n)} \\ y_0^{(n)} \\ x_1^{(n)} \\ y_1^{(n)} \\ \cdots \\ \cdots \\ x_{I-1}^{(n)} \\ y_{I-1}^{(n)} \end{bmatrix}, \quad (15)$$

by normalizing T_z to 1. The above equations are linear and can be easily solved.

For **Step 1.d**, once the Z coordinate of a control point is updated, the shape of the associated rational basis function changes accordingly (See Fig. 2). Such a change affects the "speed" of traversal of the curve in the image plane (i.e., even when the curve is traversed with a uniform speed in space, it may sweep across different image distances over unit time due to perspective distortion). To establish a better correspondence of the predicted and the observed curves, we update the predicted curve positions at $t_j, j = 1, \cdots, J-1$, which are specified by the data points on the observed image curve. Or

$$t_j = \frac{\sum_{k=1}^{j} |\mathbf{d}(t_k) - \mathbf{d}(t_{k-1})|}{\sum_{k=1}^{J-1} |\mathbf{d}(t_k) - \mathbf{d}(t_{k-1})|} = \frac{\int_{\tau=0}^{t} |\sum_i \mathbf{p}_i \dot{R}_{i,k}(\tau)| d\tau}{\int_{\tau=0}^{t_{max}} |\sum_i \mathbf{p}_i \dot{R}_{i,k}(\tau)| d\tau}$$

where t_{max} is the maximum parameter range used in defining the curve. We solve for t in the above equation and generate the corresponding point on the predicted curve at t_j.

For **Step 1.e** and **Step 1.f** above, define

$$E = \sum_j (\sum_i \mathbf{p}_i R_{i,k}(t_j) - \mathbf{d}(t_j))^2 \ .$$

By holding $R_{i,k}$ constant, we have:

$$\delta \mathbf{p} = \begin{bmatrix} \delta \mathbf{p}_0 \\ \delta \mathbf{p}_1 \\ \cdots \\ \delta \mathbf{p}_{I-1} \end{bmatrix} = \begin{bmatrix} \frac{\partial E}{\partial \mathbf{p}_0} \\ \frac{\partial E}{\partial \mathbf{p}_1} \\ \cdots \\ \frac{\partial E}{\partial \mathbf{p}_{I-1}} \end{bmatrix} = \begin{bmatrix} -\sum_j (\sum_i \mathbf{p}_i R_{i,k}(t_j) - \mathbf{d}(t_j)) R_{1,k}(t_j) \\ -\sum_j (\sum_i \mathbf{p}_i R_{i,k}(t_j) - \mathbf{d}(t_j)) R_{2,k}(t_j) \\ \cdots \\ -\sum_j (\sum_i \mathbf{p}_i R_{i,k}(t_j) - \mathbf{d}(t_j)) R_{I-1,k}(t_j) \end{bmatrix}$$

which can be readily computed from the current estimate of $\mathbf{p}_i$ and $R_{i,k}$.

Variation in Lighting Condition So far, all the discussions deal with changes in the geometry, or the shape, of a curve. Another possible variation in the appearance is due to different lighting: By this we mean 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.

Assuming two 2D images differ only by scene illumination (*i.e.*, no geometrical changes), we can linearize *interesting (or important)* 2D regions by well-known techniques. We can then treat the problem as an illumination invariance problem for points along a characteristic curve. In fact, the image regions can be linearized in many different ways, each producing its own signature. The signature for the whole region will then be just a concatenation of all the individual signatures.

Alternatively, we can include the affine or perspective case, to produce an invariant which is insensitive to both geometric (affine or perspective) and illumination changes. By solving for the deformation and translation parameters from the affine or perspective invariants (Eq. 15), we can reconstruct the same transformation for any point or curve between two images. Hence, any characteristic curve constructed from one image can be matched, point by point, to its corresponding characteristic curve in the transformed image. Illumination invariants for curves can then be applied, to verify if the two image regions, as the defined by the characteristic curves, are the same. The choice of the length and shape of the characteristic curve is arbitrary, and may be different for each model in a database. Normally, characteristic curves should be chosen that highlight particular regions of interest in the image, while avoiding those regions that could be ambiguous. For example, in recognizing airplanes, it would be ideal for the characteristic curve to pass through any unique pattern imprinted in the body of the airplane that would distinguish it from others, such as a logo, a flag, a name, or maybe the color pattern in which the plane was painted.

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). Based on a Lambertian model [11], $s(\lambda, t)$ is

$$s(\lambda, t) = (\sum_{i=1}^{n} l_i(\lambda) \mathbf{N} \cdot \mathbf{N}_i + a_i(\lambda)) \rho(\lambda, t) \ ,$$

where n is the number of light sources used to illuminate the scene, $l_i(\lambda)$ the source luminance spectral distribution, N the surface normal, N_i the incident direction for source i, $\rho(\lambda,t)$ the surface reflectivity, and a_i 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

$$\begin{aligned} s'(\lambda,t) &= (\textstyle\sum_j^m l'_j(\lambda)\mathbf{N}\cdot\mathbf{N}'_j + a_j(\lambda))\rho(\lambda,\pm t+t_0)\ , \\ &= \frac{\sum_j^m l'_j(\lambda)\mathbf{N}\cdot\mathbf{N}'_j+a'_j(\lambda)}{\sum_i^n l_i(\lambda)\mathbf{N}\cdot\mathbf{N}_i+a_i(\lambda)}[(\textstyle\sum_i^n l_i(\lambda)\mathbf{N}\cdot\mathbf{N}_i + a_i(\lambda))\rho(\lambda,\pm t+t_0)] \\ &= c(\lambda)s(\lambda,\pm t+t_0)\ , \end{aligned}$$

where

$$c(\lambda) = \frac{\sum_j^m l'_j(\lambda)\mathbf{N}\cdot\mathbf{N}'_j+a'_j(\lambda)}{\sum_i^n l_i(\lambda)\mathbf{N}\cdot\mathbf{N}_i+a_i(\lambda)}$$

capture the changes in the two different lighting conditions. As with Eq. 1, t_0 represents a change of the origin in traversal, and ± represents the possibility of traversing the curve either counterclockwise or clockwise. Following a path similar to that adopted by several researchers [17, 27, 28, 39], we assume that reflected radiance 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 reflected radiance 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

$$\begin{aligned} \mathbf{u}_{a,b} &= \textstyle\int \mathbf{L}\psi_{a,b}dt \\ &= \textstyle\int_t\int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} s(\lambda,t)d\lambda\psi_{a,b}dt \\ &= \textstyle\int_t\int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} (\sum_k \alpha_k(t)s_k(\lambda))d\lambda\psi_{a,b}dt \\ &= \textstyle\sum_k(\int_\lambda \begin{bmatrix} f^r(\lambda)s_k(\lambda) \\ f^g(\lambda)s_k(\lambda) \\ f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda)(\int_t \alpha_k(t)\psi_{a,b}dt) \\ &= \textstyle\sum_k \begin{bmatrix} L_k^r \\ L_k^g \\ L_k^b \end{bmatrix} v_{a,b}^k \\ &= \begin{bmatrix} L_1^r & L_2^r & \cdots & L_k^r \\ L_1^g & L_2^g & \cdots & L_k^g \\ L_1^b & L_2^b & \cdots & L_k^b \end{bmatrix} \begin{bmatrix} v_{a,b}^1 \\ v_{a,b}^2 \\ \cdots \\ v_{a,b}^k \end{bmatrix} \\ &= \mathbf{L}_{rgb}\mathbf{v}_{a,b}\ , \end{aligned}$$

where

$$\begin{array}{rcl} \begin{bmatrix} L_k^r \\ L_k^g \\ L_k^b \end{bmatrix} & = & \int_\lambda \begin{bmatrix} f^r(\lambda)s_k(\lambda) \\ f^g(\lambda)s_k(\lambda) \\ f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda \\ v_{a,b}^k & = & \int_t \alpha_k(t)\psi_{a,b}dt \ . \end{array}$$

Similarly,

$$\begin{array}{rcl} \mathbf{u}'_{a,b} & = & \int \mathbf{L}'\psi_{a,b}dt \\ & = & \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} \left[c(\lambda)s(\lambda, \pm t + t_0)\right] d\lambda\psi_{a,b}dt \\ & = & \int_t \int_\lambda \begin{bmatrix} f^r(\lambda) \\ f^g(\lambda) \\ f^b(\lambda) \end{bmatrix} \left[c(\lambda)(\sum_k \alpha_k(\pm t + t_0)s_k(\lambda))\right] d\lambda\psi_{a,b}dt \\ & = & \sum_k \left[(\int_\lambda \begin{bmatrix} c(\lambda)f^r(\lambda)s_k(\lambda) \\ c(\lambda)f^g(\lambda)s_k(\lambda) \\ c(\lambda)f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda)(\int_t \alpha_k(\pm t + t_0)\psi_{a,b}dt) \right] \\ & = & \sum_k \left[(\int_\lambda \begin{bmatrix} c(\lambda)f^r(\lambda)s_k(\lambda) \\ c(\lambda)f^g(\lambda)s_k(\lambda) \\ c(\lambda)f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda)(\int_t \alpha_k(t')\frac{1}{\sqrt{a}}g(\frac{\pm(t'-t_0)-b}{a})dt') \right] \\ & = & \sum_k \left[(\int_\lambda \begin{bmatrix} c(\lambda)f^r(\lambda)s_k(\lambda) \\ c(\lambda)f^g(\lambda)s_k(\lambda) \\ c(\lambda)f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda)(\int_t \alpha_k(t')\frac{1}{\sqrt{a}}g(\frac{t'-(\pm b+t_0)}{a})dt') \right] \\ & = & \sum_k \left[(\int_\lambda \begin{bmatrix} c(\lambda)f^r(\lambda)s_k(\lambda) \\ c(\lambda)f^g(\lambda)s_k(\lambda) \\ c(\lambda)f^b(\lambda)s_k(\lambda) \end{bmatrix} d\lambda)(\int_t \alpha_k(t')\psi(t')_{a,\pm b+t_0}dt') \right] \\ & = & \sum_k \begin{bmatrix} L_k^{r'} \\ L_k^{g'} \\ L_k^{b'} \end{bmatrix} v^k_{a,\pm b+t_0} \\ & = & \begin{bmatrix} L_1^{r'} & L_2^{r'} & \cdots & L_k^{r'} \\ L_1^{g'} & L_2^{g'} & \cdots & L_k^{g'} \\ L_1^{b'} & L_2^{b'} & \cdots & L_k^{b'} \end{bmatrix} \begin{bmatrix} v^1_{a,\pm b+t_0} \\ v^2_{a,\pm b+t_0} \\ \cdots \\ v^k_{a,\pm b+t_0} \end{bmatrix} \\ & = & \mathbf{L}'_{rgb}\mathbf{v}_{a,\pm b+t_0} \ . \end{array}$$

Then it is easily shown that the following expression is invariant under different lighting conditions (similar to Eq. 7):

$$\frac{\left|\left[\mathbf{u}'_{a_1,b_1}\mathbf{u}'_{a_2,b_2}\cdots\ \mathbf{u}'_{a_k,b_k}\right]^T\left[\mathbf{u}'_{a_1,b_1}\mathbf{u}'_{a_2,b_2}\cdots\ \mathbf{u}'_{a_k,b_k}\right]\right|}{\left|\left[\mathbf{u}'_{c_1,d_1}\mathbf{u}'_{c_2,d_2}\cdots\ \mathbf{u}'_{c_k,d_k}\right]^T\left[\mathbf{u}'_{c_1,d_1}\mathbf{u}'_{c_2,d_2}\cdots\ \mathbf{u}'_{c_k,d_k}\right]\right|}$$

$$
\begin{aligned}
&= \frac{\left|\left[\mathbf{v}_{a_1,\pm b_1+t_0}\mathbf{v}_{a_2,\pm b_2+t_0}\cdots\mathbf{v}_{a_k,\pm b_k+t_0}\right]^T \mathbf{L'}^T_{rgb}\mathbf{L'}_{rgb}\left[\mathbf{v}_{a_1,\pm b_1+t_0}\mathbf{v}_{a_2,\pm b_2+t_0}\cdots\mathbf{v}_{a_k,\pm b_k+t_0}\right]\right|}{\left|\left[\mathbf{v}_{c_1,\pm d_1+t_0}\mathbf{v}_{c_2,\pm d_2+t_0}\cdots\mathbf{v}_{c_k,\pm d_k+t_0}\right]^T \mathbf{L'}^T_{rgb}\mathbf{L'}_{rgb}\left[\mathbf{v}_{c_1,\pm d_1+t_0}\mathbf{v}_{c_2,\pm d_2+t_0}\cdots\mathbf{v}_{c_k,\pm d_k+t_0}\right]\right|} \\
&= \frac{\left|\left[\mathbf{v}_{a_1,\pm b_1+t_0}\mathbf{v}_{a_2,\pm b_2+t_0}\cdots\mathbf{v}_{a_k,\pm b_k+t_0}\right]^T \mathbf{L}^T_{rgb}\mathbf{L}_{rgb}\left[\mathbf{v}_{a_1,\pm b_1+t_0}\mathbf{v}_{a_2,\pm b_2+t_0}\cdots\mathbf{v}_{a_k,\pm b_k+t_0}\right]\right|}{\left|\left[\mathbf{v}_{c_1,\pm d_1+t_0}\mathbf{v}_{c_2,\pm d_2+t_0}\cdots\mathbf{v}_{c_k,\pm d_k+t_0}\right]^T \mathbf{L}^T_{rgb}\mathbf{L}_{rgb}\left[\mathbf{v}_{c_1,\pm d_1+t_0}\mathbf{v}_{c_2,\pm d_2+t_0}\cdots\mathbf{v}_{c_k,\pm d_k+t_0}\right]\right|} \\
&= \frac{\left|\left[\mathbf{u}_{a_1,\pm b_1+t_0}\mathbf{u}_{a_2,\pm b_2+t_0}\cdots\mathbf{u}_{a_k,\pm b_k+t_0}\right]^T \left[\mathbf{u}_{a_1,\pm b_1+t_0}\mathbf{u}_{a_2,\pm b_2+t_0}\cdots\mathbf{u}_{a_k,\pm b_k+t_0}\right]\right|}{\left|\left[\mathbf{u}_{c_1,\pm d_1+t_0}\mathbf{u}_{c_2,\pm d_2+t_0}\cdots\mathbf{u}_{c_k,\pm d_k+t_0}\right]^T \left[\mathbf{u}_{c_1,\pm d_1+t_0}\mathbf{u}_{c_2,\pm d_2+t_0}\cdots\mathbf{u}_{c_k,\pm d_k+t_0}\right]\right|}\,.
\end{aligned}
\tag{16}
$$

This derivation is in spirit similar to that of [15, 17]. By using a ratio expression, we obtain a much simpler and computationally efficient form of invariants which does not require computing the color correlation matrix and the singular value decomposition of such a matrix [15, 17].

We can follow the same technique used for Eq. 3 in order to measure similarity under different parameterizations (Eqs. 4-refeq-invmeasure).

4 Experimental Results

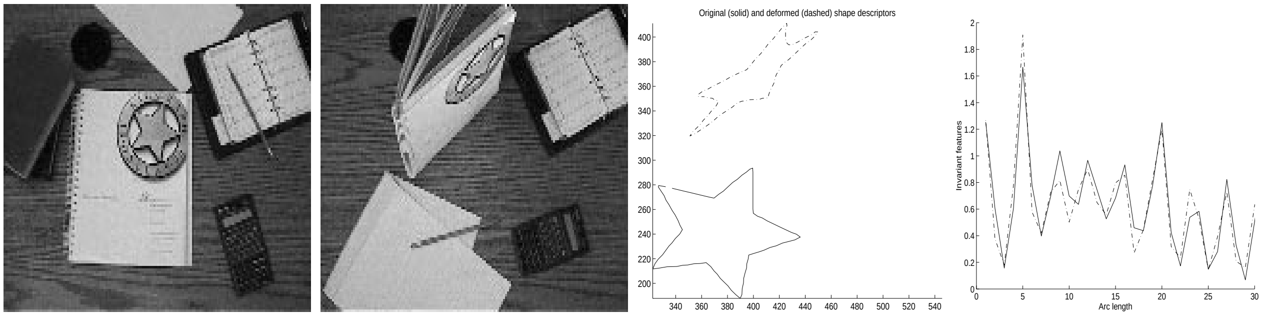
We will first show the results of applying affine, perspective and illumination invariant formulations individually for both real and synthetic images. Then we will put all these invariant formulations together in a real recognition experiments.

General Affine Transform with Change of Parameterization Fig. 4 shows (a) a star pattern on a book cover and (b) the same pattern after undergoing an affine transformation (simulated by a rigid-body transform in 3D - a rotation and translation of the book cover). The extracted patterns are shown in Fig. 4(c) as solid (original pattern) and dashed lines (transformed pattern). We use the second-order b-spline function of a uniform knot vector [37] in the basis expansion. Fig. 4(d) shows the invariant signatures (based on Eq. 7) of the original (solid) and transformed (dashed) curves along the contours. The invariant signatures were aligned by maximizing the cross-correlation. As can be seen from the figure, the invariant signatures are quite consistent.

Figs. 5(a) and (b) show a shirt with a dolphin imprint and a stretched and deformed version of the same imprint. Fig. 4(c) shows the extracted patterns. The invariant signatures are plotted in 4(d), and, again, they are quite consistent.

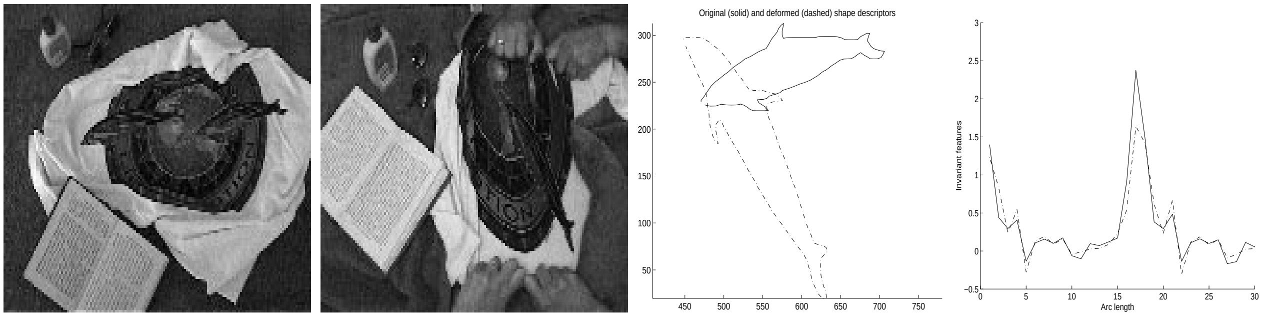
Perspective Transform Our formulation, though recursive in nature, is nonetheless linear and achieves fast convergence in our preliminary experiments. Initially, a good search point was established by using non-rational spline bases to approximate rational spline bases. Subsequently, the number of iterations needed to verify the invariance was small (about 3 to 4) even for large perspective distortion. Furthermore, by using rational spline functions and partitioning the search in two stages, we maintain the linearity of the whole formulation.

Fig. 6 depicts some preliminary results on synthetic images. Both the star and knot shapes have a canonical head-on view. A 2D perspective image was generated by applying an arbitrary rotation and translation to the shape before perspective projection. In Fig. 6, we show in solid lines in the first and third columns the projected 2D shapes and in the second and fourth columns the depth profiles of the control vertices from the particular view point. The search for invariance started by assuming there was no rotation nor translation and the depth was a constant 1. Under this assumption, the predicted 2D shapes (the head-on view) and the depth profiles (a constant 1)



(a) (b) (c) (d)

Figure 4: (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.



(a) (b) (c) (d)

Figure 5: (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.

are shown (in dashed line) in the first row in Fig. 6. As can be seen by comparing the two curves, perspective distortion was quite severe.

We then applied our algorithm of estimating the motion parameters, updating the control vertex positions, revising the shapes of the rational spline bases (using Eqs. 11 and 12), and updating the parameterization. The revised 2D shapes and depth profiles, generated using updated control vertices and rational bases, are displayed in Fig. 6 after 1, 3, and 10 updates. Convergence was fast and most significant changes occurred in the first three iterations. The depth profiles converged only qualitatively as a scaling factor cannot be recovered.

Figs. 7 and 8 show results on real images. To demonstrate the accuracy of the new perspective invariance formulation, we compared our results with the initial estimate, using traditional affine invariants. It can be easily seen that the affine transform models faithfully a planar curve (with $Z = 0$) under a parallel projection, and hence, is linear. Its efficacy in producing an invariant signature by itself, for a general 3D curve under a perspective projection is in doubt.

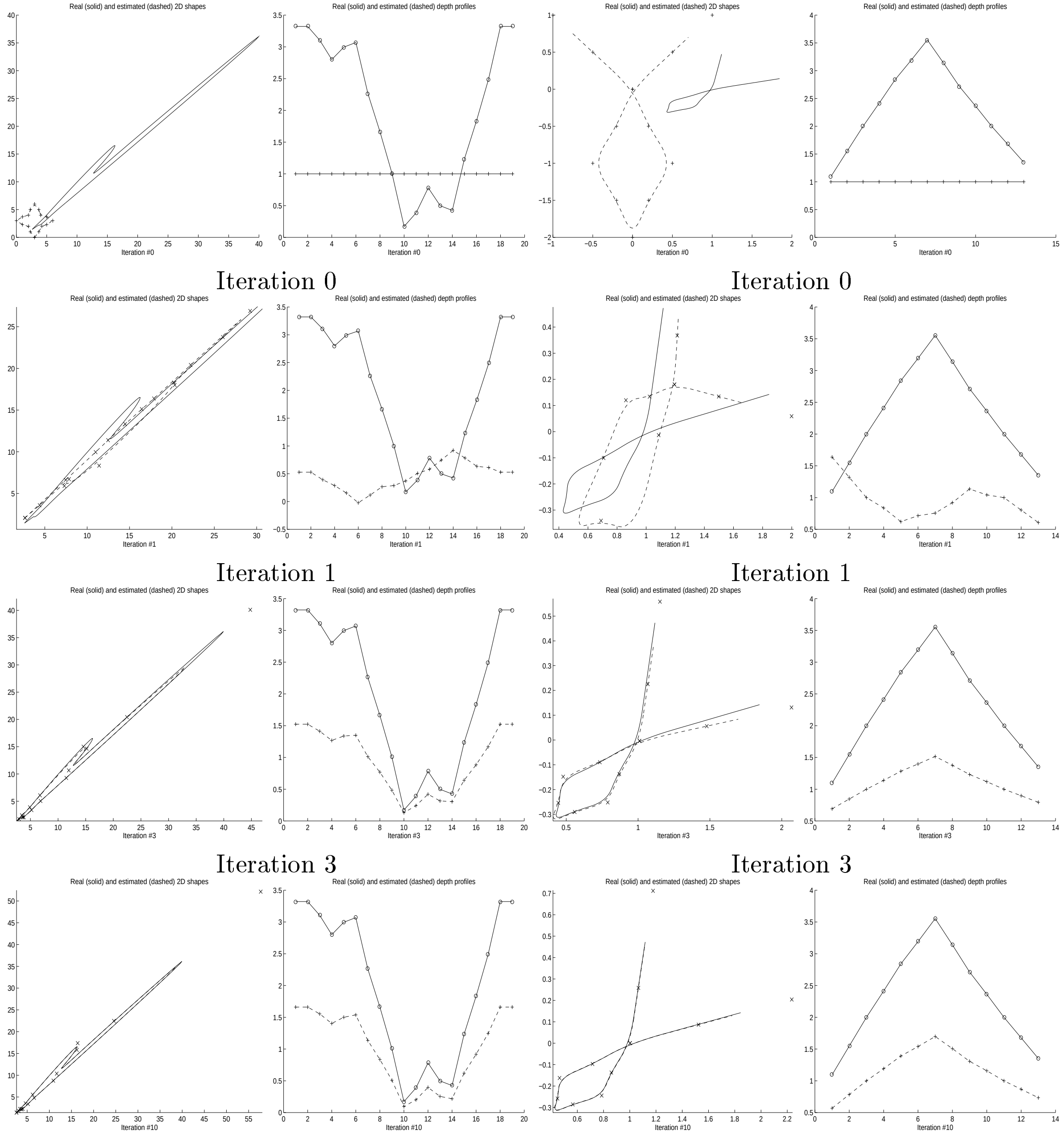
Fig. 7(a) shows the canonical view of a toy duck and Fig. 7(b) shows a perspective view of it. The initial fitting results and invariant signatures, using the traditional affine invariant formulation, are depicted in Figs.7(c) and (d) respectively. Fig. 7(e) depicts the 2D image curve (in solid line) and the curve derived from the perspective invariant fitting after seven iterations (in dashed line). Fig. 7(f) depicts the invariant shape signatures computed along the 2D image curve (solid) and along the curve computed using the perspective invariant fitting after seven iterations (dashed). As can be seen from these figures, the fitting and signature based on perspective invariants are much closer to the true curve and signature. This is because the perspective framework gives a better approximation both in terms of curve fitting and parameterization. The results were obtained with only seven iterations.

Fig. 8 shows another example on real images. Now the curve is embedded on a curved surface (a cylindrical pail) and affine transform does not even apply. Figs. 8(a) and (b) show the canonical view and another perspective, respectively. We extracted the silhouette of the car from both images for verifying the perspective invariance. Curve fitting and invariant signature (after five iterations) thus computed are displayed in Figs. 8(c) and (d) respectively. It can be seen that our invariance framework produces consistent results for general, non-planar 3D curves, all with a small number (in this case five) of iterations.

Change of Illumination To illustrate the correctness of the invariance formulation under illumination changes, we placed different color filters in front of the light sources used to illuminate the scene and verify the similarity of illumination invariant signatures. It should be noted that the particular examples we show here only serve to demonstrate the validity and correctness of the framework. A recognition experiment presented later in this section will illustrate the use of curves of arbitrary shape for illumination invariants.

Two examples are shown below. Fig. 9 shows the same mouse pad under white and blue illumination. For simplicity, we randomly placed three circular curves (the red, green, and blue curves in Fig. 9) and computed the invariant signatures along these three curves for both the images under white and blue illumination. As discussed in the previous section, we can linearize the image to obtain an invariant signature for the *whole* image. Fig. 10(a), (c), and (e) show the r-g-b profiles along the red, green, and blue circles in Fig. 9 under white (solid) and blue (dashed) illumination. Fig. 10(b), (d), and (f) show the invariant profiles computed from the white (solid) and blue (dashed) illumination. As can be seen from the figure, they are quite consistent.

The degree of consistency of illumination invariants is best appreciated by comparing the above results with the following: Tbl. 1 tabulates the RMS deviation of the invariant signatures computed



Iteration 0

Iteration 0

Iteration 1

Iteration 1

Iteration 3

Iteration 3

Iteration 10

Iteration 10

Figure 6: Real (solid) and estimated (dashed) shape and depth profiles for star (columns 1 and 2) and knot (columns 3 and 4) after 0th, 1st, 3rd, and 10th iterations. "o" and "+" denote the projected control vertex positions.

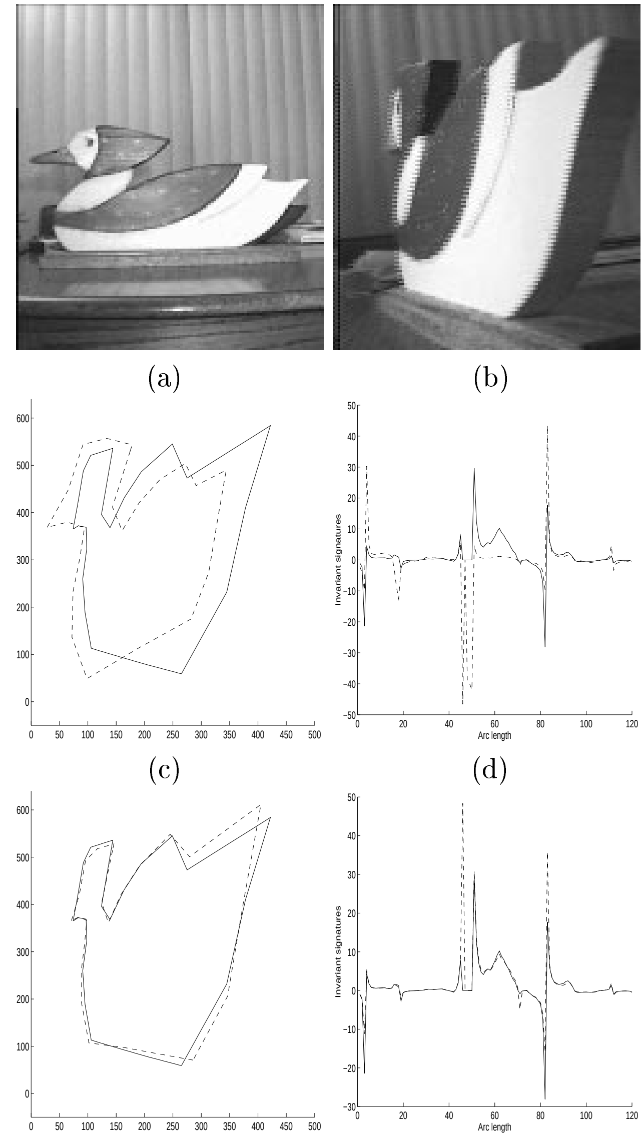


Figure 7: (a) The canonical view, (b) another perspective, (c) 2D image curve (solid) and the curve derived w. perspective invariant fitting (dashed), (d) shape signatures of the 2D image curve (solid) and the curve derived w. perspective invariant fitting (dashed), (e) 2D image curve (solid) and the curve derived w. affine invariant fitting (dashed), and (f) shape signatures of the 2D image curve (solid) and the curve derived w. affine invariant fitting (dashed).

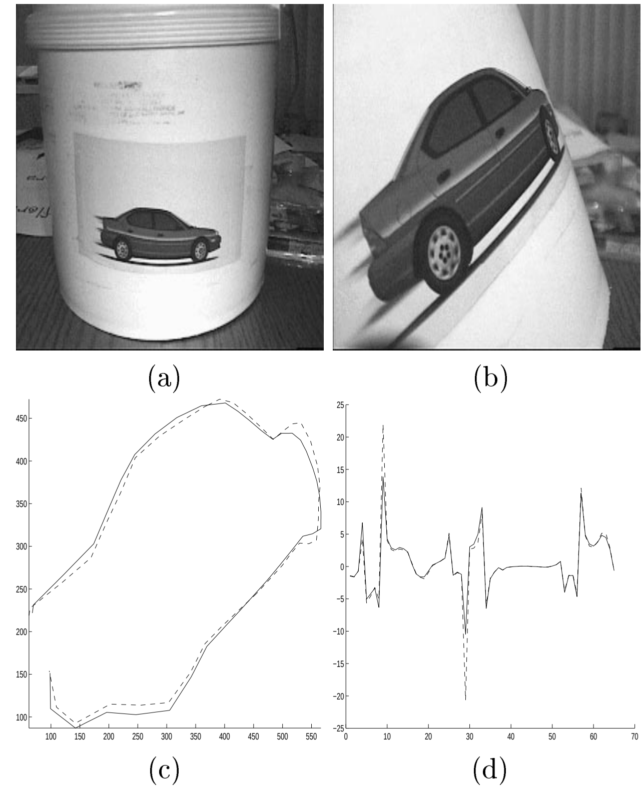


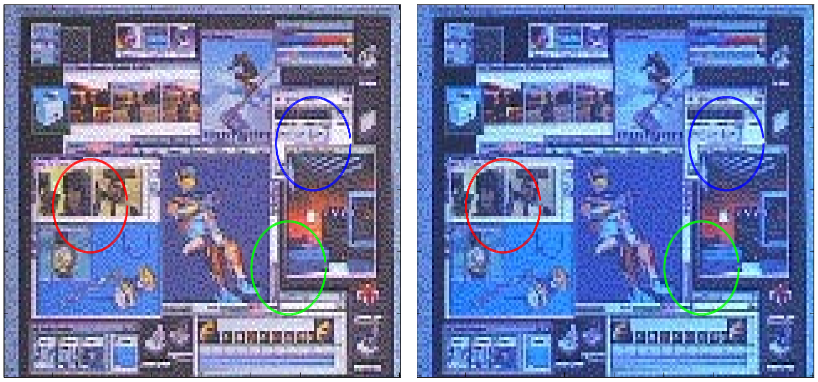
Figure 8: (a) The canonical view, (b) another perspective, (c) 2D image curve (solid) and the curve derived w. perspective invariant fitting (dashed), and (d) shape signatures of the 2D image curve (solid) and the curve derived w. perspective invariant fitting (dashed)

red-red curve	green-green curve	blue-blue curve	red-green curve	green-blue curve	blue-red curve
0.3392	0.3544	0.5465	0.8831	0.8039	0.8983

Table 1: RMS deviation in invariant signatures computed based on Eq. 16 for Fig. 9

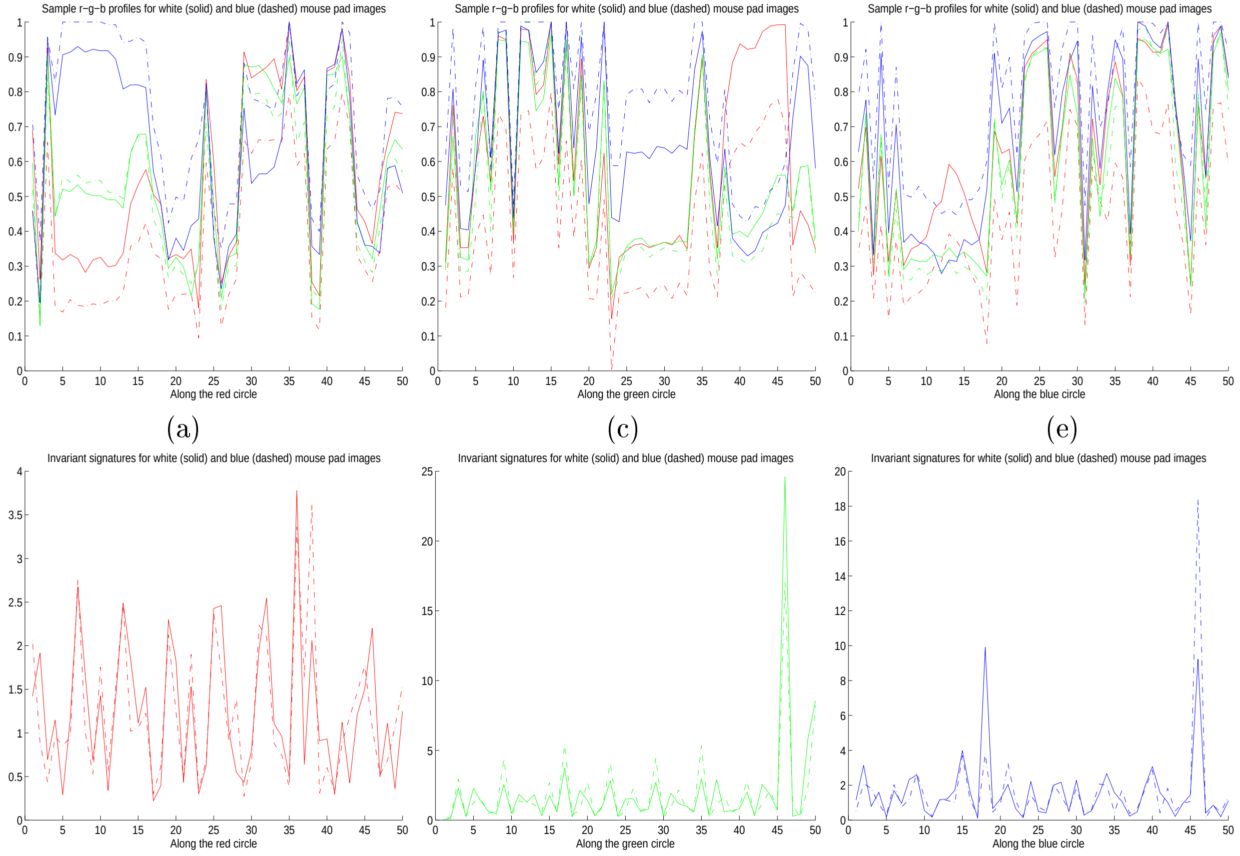
on the *same* curves vs. those computed on *different* curves in Fig. 9. As can be seen, the results are much less consistent when invariant signatures are compared on different curves. In fact, the RMS deviation of the invariant features computed from different curves is 147% to 265% those computed from the same curves.

Hierarchical Invariant Analysis The additional degree of freedom in designing the basis function enables a hierarchical shape analysis. To illustrate, Fig. 13 shows the original (solid lines) and deformed (dashed lines) shapes with different amounts of noise added. The noise level in Fig. 13(b) is 15 times higher than that in Fig. 13(a). As can be seen from Fig. 14, our approach, which analyzes the shape at different scales locally, will eventually discover the similarity, even though the similarity may manifest at different levels of details. This is the case in Fig. 14, where shape similarity is apparent from scale 3 onward for small noise case and from scale 7 onward for large noise case. Traditional analysis relying on a single scale or requiring high-order derivatives of the contour function will have difficulty handling both cases.



(a) (b)

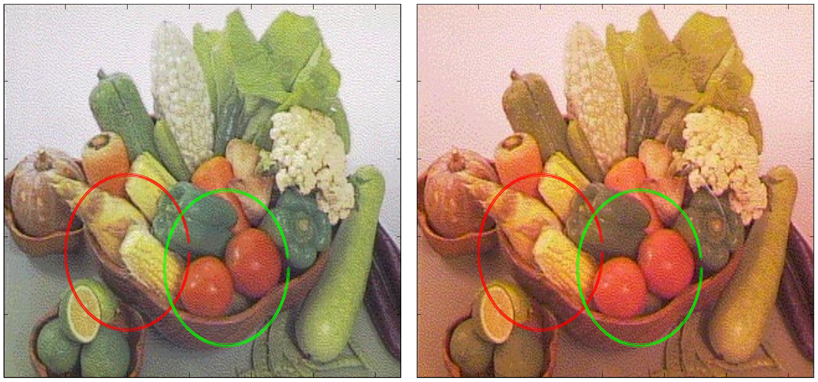
Figure 9: The same mouse pad under (a) white and (b) blue illumination.



(a) (c) (e)

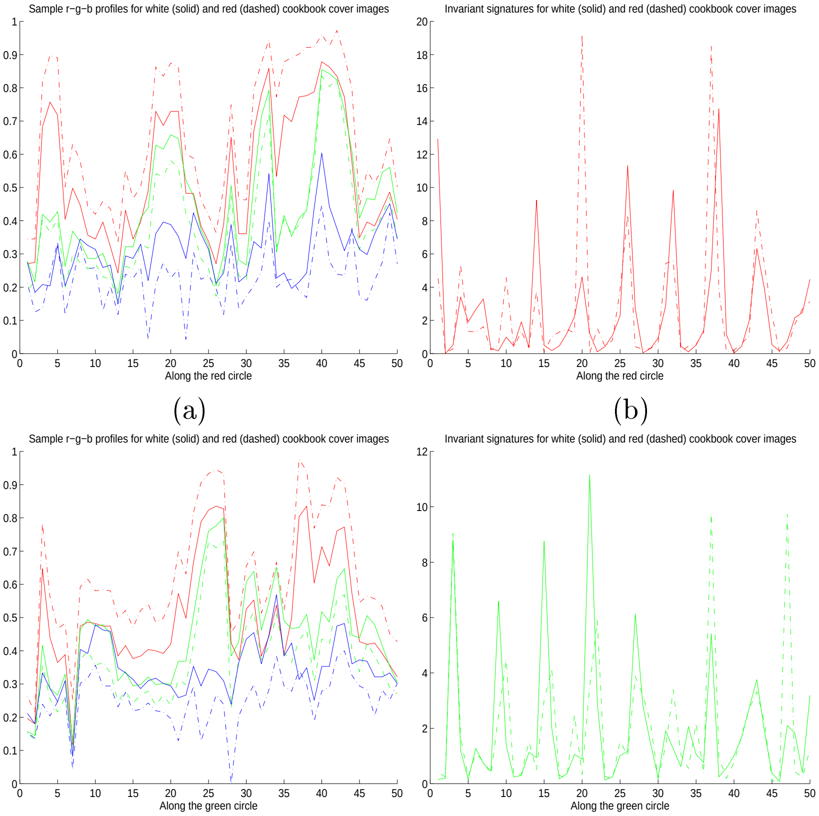
(b) (d) (f)

Figure 10: R-g-b profiles and invariant signatures computed for the mouse pad under white (solid) and blue (dashed) illumination. Profiles and invariant signatures were computed on the red (a) and (b), green (c) and (d), and blue (e) and (f) circles in Fig. 9.



(a) (b)

Figure 11: The same cookbook cover under (a) white and (b) red illumination.

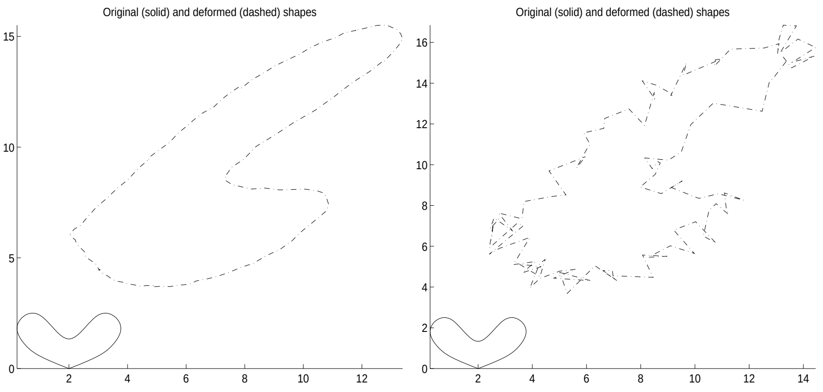


(a) (b)

(c) (d)

Figure 12: R-g-b profiles and invariant signatures computed for the cookbook cover under white (solid) and red (dashed) illumination. Profiles and invariant signatures were both computed on the red circle in Fig. 11 for (a) and (b), and on the green circle in Fig. 11 for (c) and (d)

.



(a) (b)

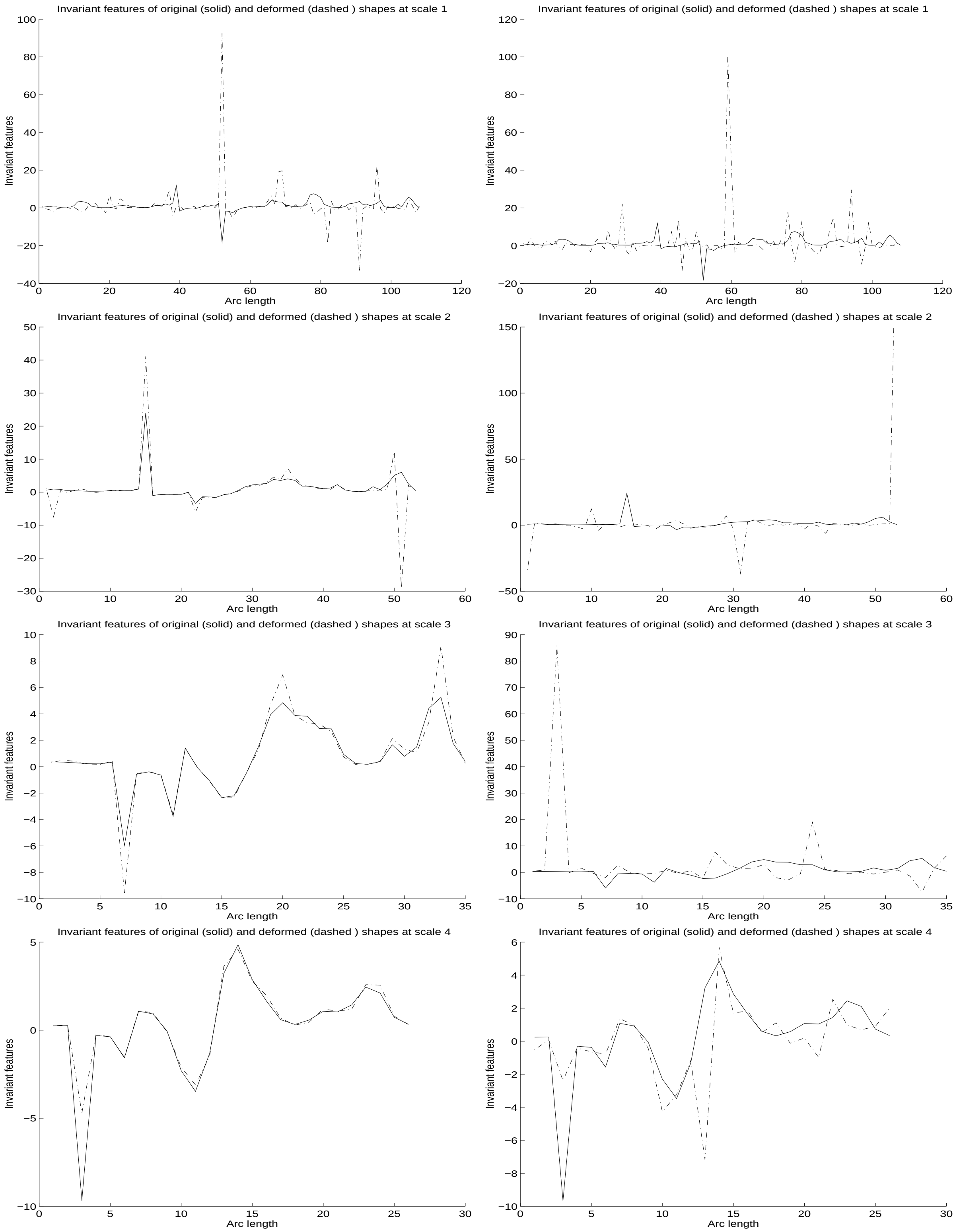
Figure 13: Original and transformed shapes with noise added. The noise level in (b) is 15 times that in (a). Solid lines for original shapes and dashed lines for transformed and noise-corrupted shapes.

A Recognition Experiment To further illustrate the use of these invariant expressions, we conducted a recognition experiment on real images. The scenario is that of a digital library application. The database comprises a collection of sixteen airplane models in canonical (top) view. The airplanes were automatically extracted from the images and invariant shape and color signatures computed off-line. The query images were photographed from a different perspective and under varying illumination. The airplanes in the query images were extracted using a semi-automated process possibly with user intervention. We then used a two-stage approach where features invariant to affine and perspective projection were first used to match the silhouette of the query airplane with the silhouettes of those in the database. We then employed the illumination invariant features to verify the match and disambiguate among models with similar shape but different colors. The results show that we were able to achieve 100% accuracy for a database comprising very similar models, presented with query images of large perspective shape distortion and illumination change. This is in contrast with most image digital library retrieval schemes which can perform between-class (e.g., airplanes vs. cars) retrievals but not within-c-class (e.g., different types of airplanes) retrievals.

Furthermore, we exploit both geometric properties and color to produce a more robust recognition strategy. The integration of geometric and illumination invariants is a novel approach which has produced only limited success in object recognition so far [39, 41, 48]. Our integration of geometric and illumination invariants has produced promising results.

We first demonstrate the performance of affine invariants. It will be shown that affine invariants work well with objects that are relatively far from the camera, but fails miserably with objects that are near the camera or otherwise showing large perspective distortion. We then present the results of using perspective invariants, which correctly identified all sample images. Finally, we show the results of applying illumination invariants to further verify a correct match.

Fig. 15 shows 16 models of airplanes from a database, each posed in a canonic view, under standard lighting. It should be noted that many of the airplane models have approximately the same shape (e.g., models 5 and 6, and models 3, 7 and 14), making the recognition problem difficult. Eleven sample images of the same airplanes are shown in Fig. 16, each in a different pose. Moreover, colored filters were placed in front of the light sources (and the number of light sources varied) to simulate changes in illumination. From these images, we extracted the silhouettes of these airplanes


Invariant features of original (solid) and deformed (dashed) shapes at scale 1

Invariant features of original (solid) and deformed (dashed) shapes at scale 1

Invariant features of original (solid) and deformed (dashed) shapes at scale 2

Invariant features of original (solid) and deformed (dashed) shapes at scale 2

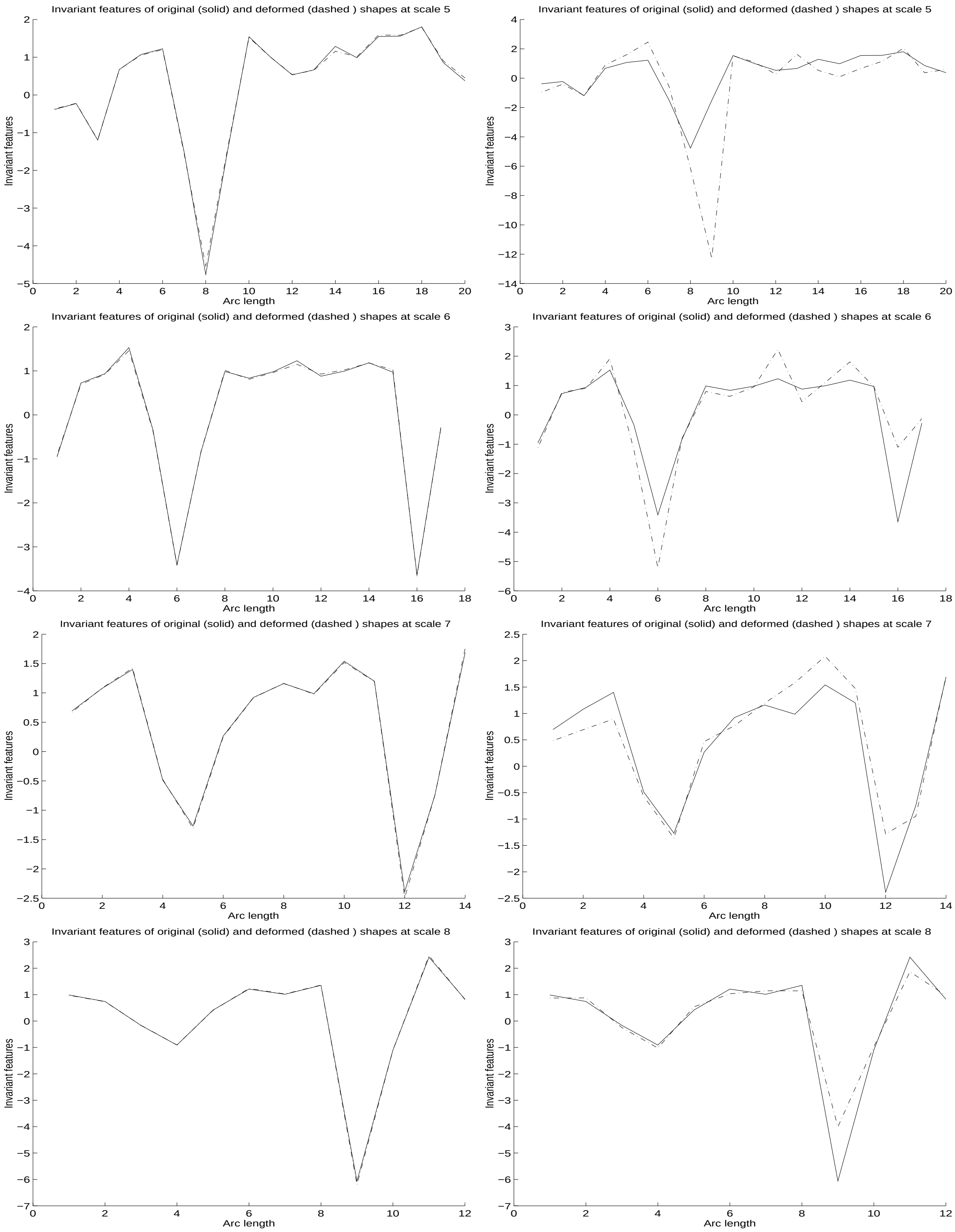
Invariant features of original (solid) and deformed (dashed) shapes at scale 3

Invariant features of original (solid) and deformed (dashed) shapes at scale 3

Invariant features of original (solid) and deformed (dashed) shapes at scale 4

Invariant features of original (solid) and deformed (dashed) shapes at scale 4

Figure 14: Invariant shape descriptors for the original and deformed shapes with noise added at different scales. Solid lines for original shape and shape descriptors and dashed lines for transformed and noise-corrupted shapes and shape descriptors (continued on the next page).


Invariant features of original (solid) and deformed (dashed) shapes at scale 5

Invariant features of original (solid) and deformed (dashed) shapes at scale 5

Invariant features of original (solid) and deformed (dashed) shapes at scale 6

Invariant features of original (solid) and deformed (dashed) shapes at scale 6

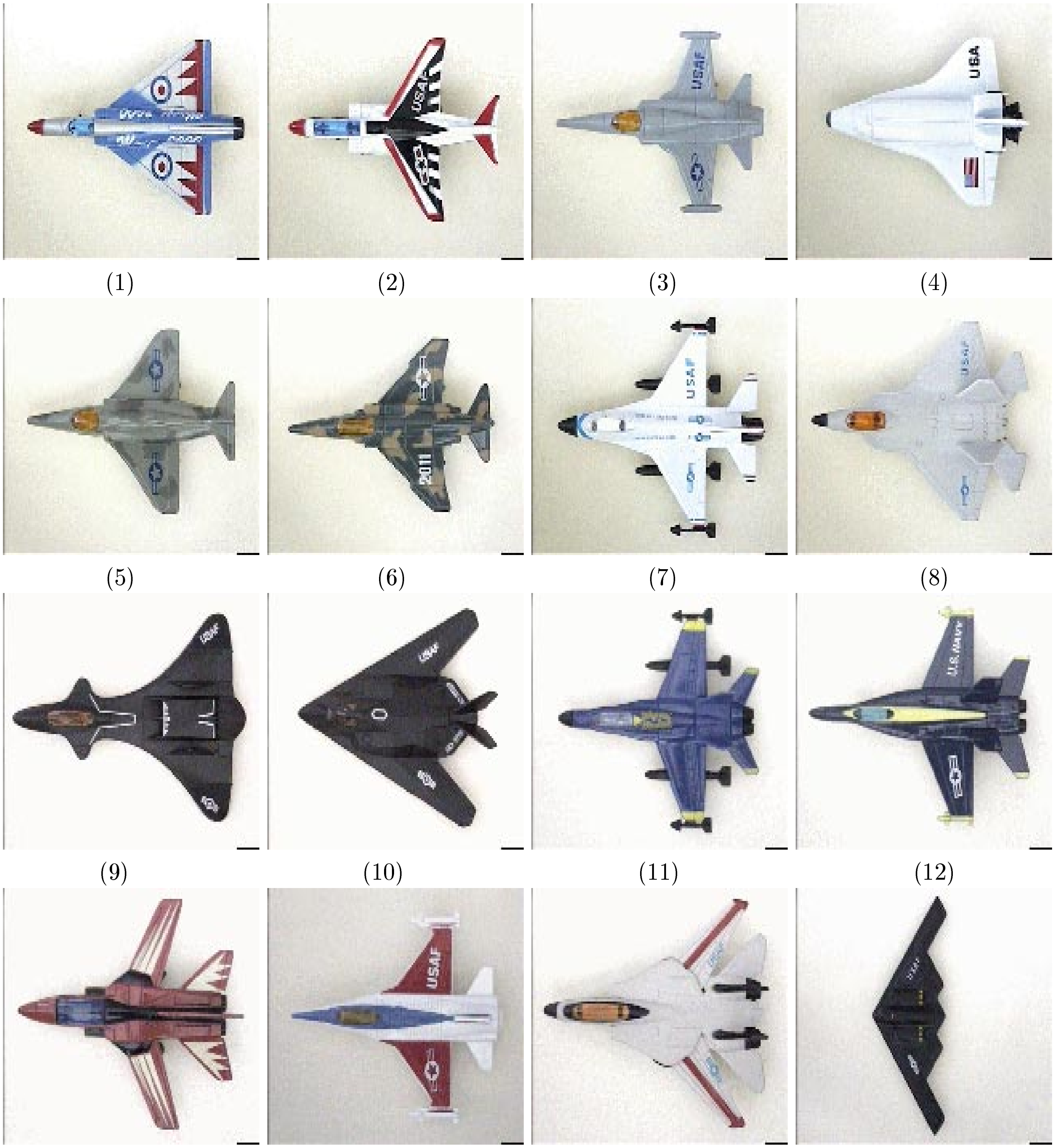
Invariant features of original (solid) and deformed (dashed) shapes at scale 7

Invariant features of original (solid) and deformed (dashed) shapes at scale 7

Invariant features of original (solid) and deformed (dashed) shapes at scale 8

Invariant features of original (solid) and deformed (dashed) shapes at scale 8

Figure: (continued from previous page) Invariant shape descriptors for the original and deformed shapes with noise added at different scales. Solid lines for original shapes and shape descriptors and dashed lines for transformed and noise-corrupted shapes and shape descriptors.



(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11) (12)

(13) (14) (15) (16)

Figure 15: A database of airplane models.

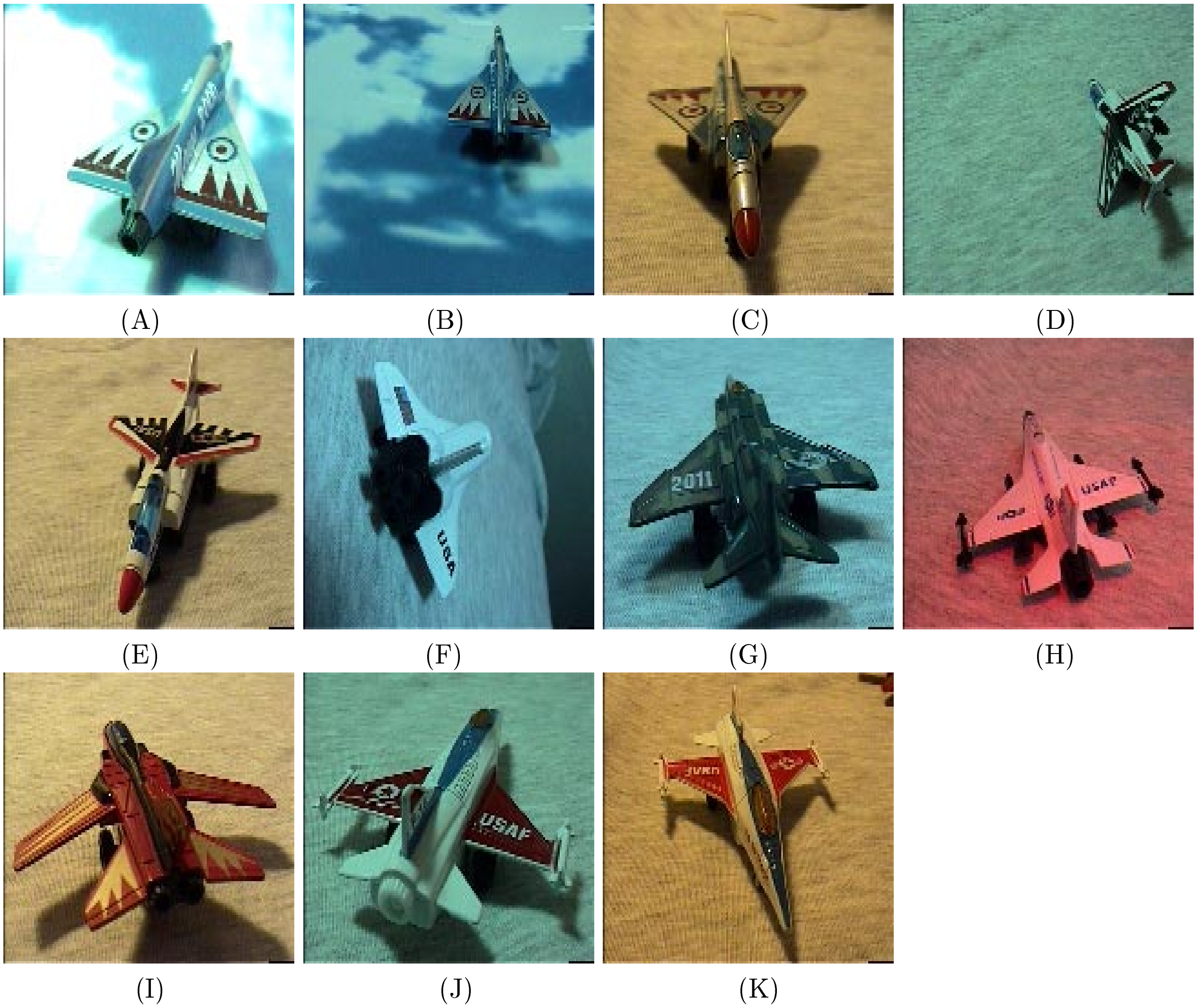


Figure 16: The same airplanes in varying poses and illumination: observed images A-C, D-E, F, G, H, I, and J-K correspond to airplanes models 1, 2, 4, 6, 7, 13, and 14 in the database, respectively.

and used both affine and perspective invariants to verify their similarity.

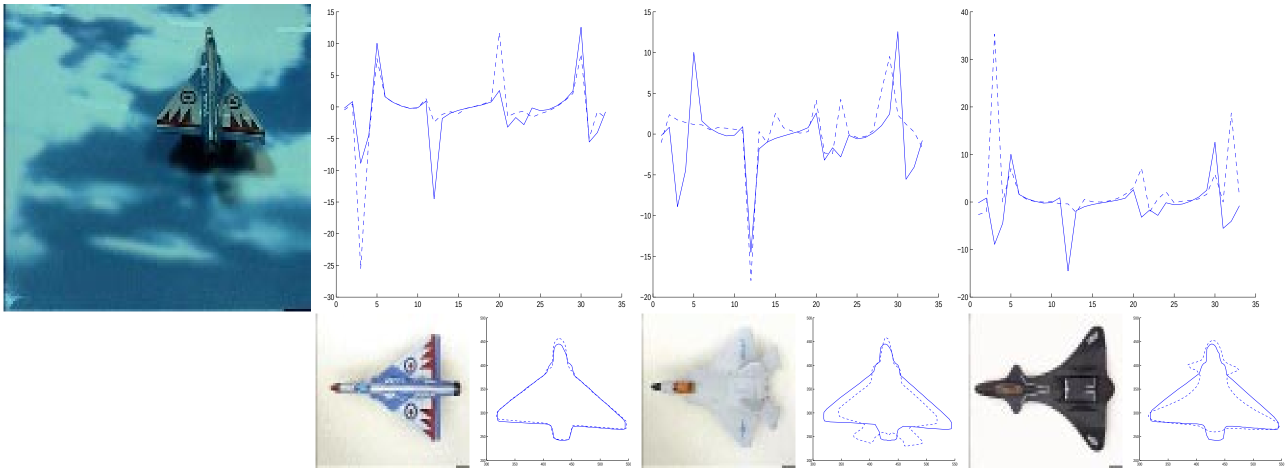
	Rank (using affine invariants)									
Image	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
A	(5) 0.7919	(14) 0.0222	(3) 0.0077	(12) -0.0054	(9) -0.0081	(11) -0.0103	(16) -0.0244	(4) -0.0258	(8) -0.0584	(10) -0.0703
B	**(1)** **0.6772**	(8) 0.2487	(9) 0.1879	(2) 0.1556	(6) 0.1512	(15) 0.1350	(13) 0.1327	(14) 0.1053	(3) 0.0739	(16) 0.0655
C	(16) 0.3473	(14) 0.2944	(6) 0.1457	(2) 0.1046	(13) 0.0781	(3) 0.0758	**(1)** **0.0562**	(8) 0.0528	(5) 0.0432	(12) 0.0278
D	**(2)** **0.7236**	(14) 0.4480	(11) 0.3666	(6) 0.3650	(3) 0.2065	(16) 0.1739	(8) 0.0486	(15) 0.0421	(10) 0.0370	(9) -0.0600
E	(6) 0.3014	(12) 0.2680	(9) 0.2555	**(2)** **0.2541**	(14) 0.2262	(7) 0.1981	(15) 0.1404	(13) 0.1224	(3) 0.0887	(8) 0.0365
F	(15) 0.3350	(16) 0.2513	(9) 0.2492	(14) 0.2490	(8) 0.1634	(6) 0.1535	(13) 0.1335	(2) 0.1087	(3) 0.0794	**(4)** **0.0633**
G	(12) 0.6979	(14) 0.4114	(2) 0.2093	(3) 0.1452	(5) 0.0531	(9) 0.0007	(4) -0.0140	**(6)** **-0.0194**	(10) -0.0356	(1) -0.0518
H	(14) 0.5241	(13) 0.4595	(6) 0.4225	(2) 0.3987	(10) 0.2601	(12) 0.2424	(11) 0.1617	(15) 0.0281	(16) 0.0161	(9) 0.0100
I	(15) 0.3693	(2) 0.3445	(7) 0.1674	(3) 0.1644	(1) 0.0487	(10) 0.0340	(9) -0.0186	(8) -0.0306	(16) -0.0545	(6) -0.0549
J	**(14)** **0.4709**	(6) 0.3821	(2) 0.3416	(11) 0.3314	(8) 0.2884	(3) 0.2849	(7) 0.2573	(16) 0.0699	(5) 0.0003	(9) -0.0504
K	(7) 0.7407	(4) 0.5313	(11) 0.3247	(9) 0.0994	(8) 0.0372	(1) -0.0175	(5) -0.0306	(15) -0.0345	(10) -0.0489	(2) -0.0917

Table 2: Top matches between affine invariant signatures of observed images and models in the database. Numbers in parentheses indicate the airplane model selected. The value beneath it is the correlation coefficient of the two signatures. The expected (correct) airplane model is in boldface. Each row corresponds to an observed image. The columns are arranged left to right, from the best match to worse.

Table 2 shows both the effectiveness and limitations of using affine invariants (Eq. 7), when dealing with objects under perspective transformations. Affine invariant signatures of the airplane models were stored in a database (performed off-line). The resolution scale used[3] for recognition was pre-determined by computing the energy of the models' signatures at different scales. The scale where energy appeared to be concentrated, was selected. (*i.e*, Large values of energy implies more information). Scale 3 was selected for this database. For each observed image (A through K), the affine invariant signature was computed, and compared with the signatures of all models in the database. Correlation coefficients were used to determine the similarity between each pair of signatures. Each row in Table 2 refers to an observed image. Each of the ten columns represents the rank given to each airplane model from the database (shown in parentheses). The columns are ordered from left to right, with the leftmost column being the best match found. Only the top ten matches are shown. The values (not in parentheses) are the correlation coefficients. Entries printed in boldface are the expected (correct) matches.

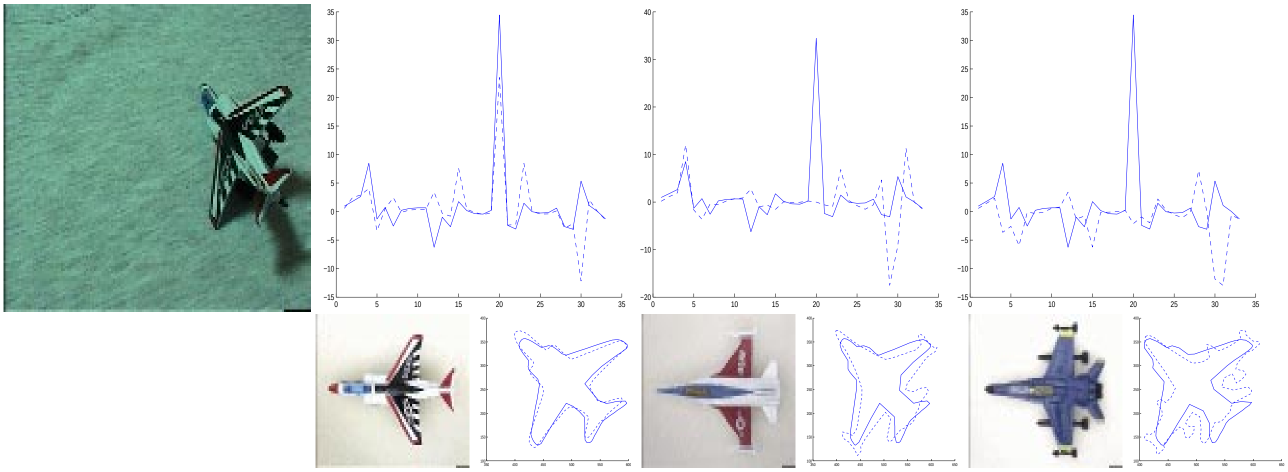
It is clear from Table 2 that the affine invariant works well in cases where the object is far from the camera relative to its size. The observed images B and D are consistent with this scenario. For observed images B and D, the correlation between their signatures and that of the correct model in the database is significantly higher (both rank the 1^{st}), relative to that for the rest of the models. Fig. 17 shows the affine invariant signature of airplane B (solid) correctly matched with that of

[3] It is possible to select a range of scales. We simply use some statistical measure to get a scalar value.



(a) (b) (c)

Figure 17: (a) Affine invariant signature of airplane B (solid) correctly matched with that of model 1 (dashed) (best match found), and (b)-(c) the next two best matches.



(a) (b) (c)

Figure 18: (a) Affine invariant signature of airplane D (solid) correctly matched with that of model 2 (dashed) (best match found), and (b)-(c) the next two best matches.

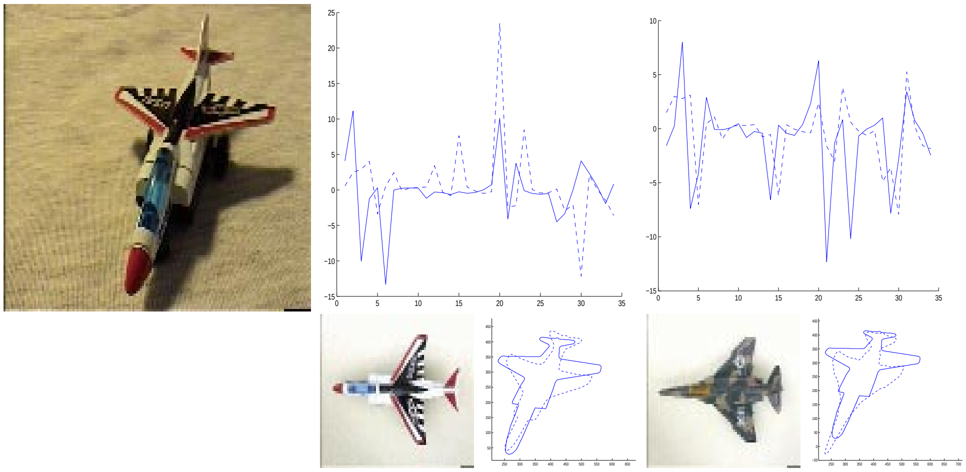
model 1 (dashed) (best match found). The next two best matches (models 8 and 9) are shown in Figs. 17(b) and (c). Similarly for airplane D, Fig. 18(a) is the best (and correct) match (model 2). Fig. 18(b) and (c) are the next two best matches (models 14 and 11).

However, the performance with other images was unsatisfactory. Fig. 19(a) shows the signature match between airplane E and model 2 (supposedly a correct match). Because of large perspective distortion, the signatures are inconsistent, and the match is ranked the 4^{th}. Instead, the match in Fig. 19(b) is (incorrectly) ranked the 1^{st} (model 6). Hence, the affine invariant formulation can be used to approximate perspective invariance only when the object is relatively far from the camera. For our test images, the affine invariant correctly identified only three of them. For four of the images, the correct models were not even among the top ten candidates.

	Rank (using perspective invariants)									
Image	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
A	**(1)** **120.82**	(9) 279.01	(4) 383.86	(6) 503.34	(5) 533.66	(10) 542.16	(2) 596.96	(7) 675.19	(11) 755.74	(14) 761.19
B	**(1)** **47.30**	(9) 146.81	(10) 233.45	(4) 252.14	(6) 337.03	(2) 389.68	(5) 405.70	(15) 463.60	(16) 524.44	(7) 542.40
C	**(1)** **146.16**	(4) 319.36	(2) 347.87	(9) 398.37	(6) 437.71	(5) 464.66	(10) 555.42	(14) 664.10	(7) 690.55	(11) 761.43
D	**(2)** **71.67**	(6) 99.78	(5) 103.76	(4) 182.25	(13) 190.34	(14) 219.87	(1) 226.97	(7) 233.68	(3) 249.75	(12) 256.10
E	**(2)** **77.18**	(5) 225.33	(6) 237.78	(14) 302.50	(12) 383.28	(4) 383.29	(3) 385.40	(13) 409.81	(7) 429.61	(15) 518.69
F	**(4)** **363.12**	(1) 399.78	(9) 418.96	(6) 470.93	(10) 479.52	(14) 494.35	(5) 551.37	(11) 571.67	(2) 596.43	(7) 605.42
G	**(6)** **174.62**	(13) 270.70	(5) 297.42	(4) 338.39	(2) 354.01	(14) 360.44	(12) 371.27	(3) 396.45	(1) 406.96	(7) 436.25
H	**(7)** **125.27**	(14) 144.79	(3) 160.16	(11) 177.42	(13) 215.16	(6) 233.24	(12) 233.71	(5) 271.79	(2) 299.27	(4) 301.98
I	**(13)** **139.15**	(6) 310.96	(3) 343.69	(14) 353.16	(12) 365.75	(5) 389.29	(7) 408.44	(2) 415.15	(15) 422.48	(1) 448.38
J	**(14)** **118.53**	(3) 198.34	(12) 243.56	(13) 248.79	(7) 294.51	(11) 319.46	(6) 349.93	(4) 365.38	(5) 416.17	(15) 428.91
K	**(14)** **122.10**	(3) 144.23	(7) 237.71	(13) 272.76	(12) 272.96	(6) 276.52	(11) 279.11	(2) 349.75	(5) 380.91	(4) 454.06

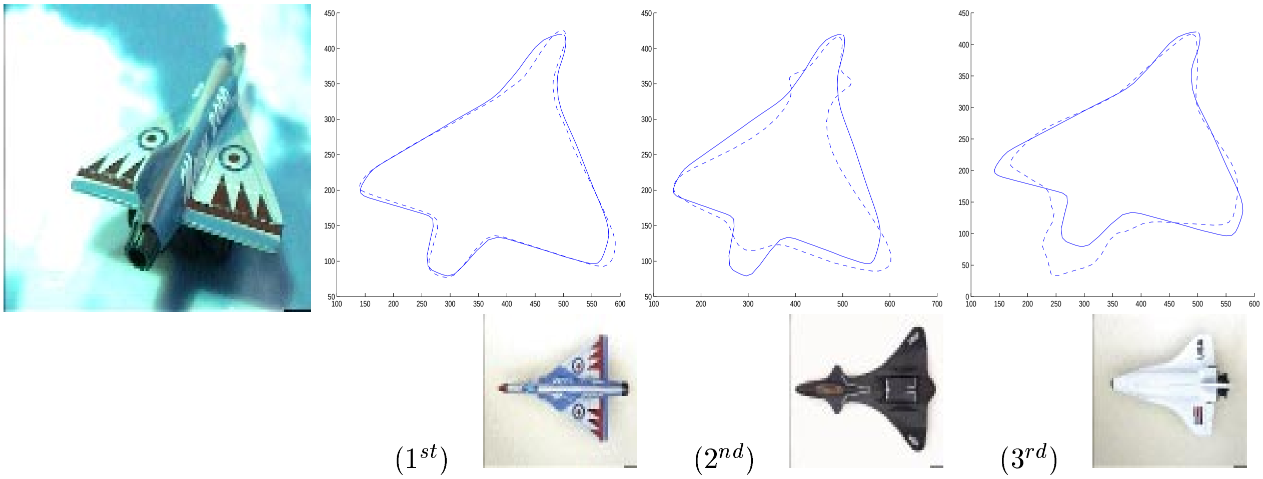
Table 3: Top ten matches between each observed image and models in the database, using perspective invariants. Numbers in parentheses indicate the airplane model selected. The value beneath it is the relative shape deviation (error) between the estimated image and observed image. The expected (correct) airplane model is in boldface. Each row corresponds to an observed image. The columns are arranged left to right, from the best match to worse.

In cases of large perspective distortion, the affine invariant performs poorly as expected. Fortunately, a more suitable invariant can be used. Table 3 shows the results of using the perspective invariants on the same observed images. A polynomial degree of order 2 was used, since we want to treat the observed image points as control points. (*I.e.*, The curve passes through each of the observed image points exactly.) Only the top 10 matches are shown, with the leftmost column as the top match. The values shown are the relative shape deviations (error) between the observed image and the best estimate using a particular model. The numbers in parentheses correspond to the model number from the database. Entries printed in boldface are the expected (correct) matches. As can be seen, all observed images were identified correctly. Figs. 20 to 22 show the results in more detail. In these figures, the leftmost image is the observed image. The top three matches are in the next three columns – the observed image (solid) and estimated image (dashed)



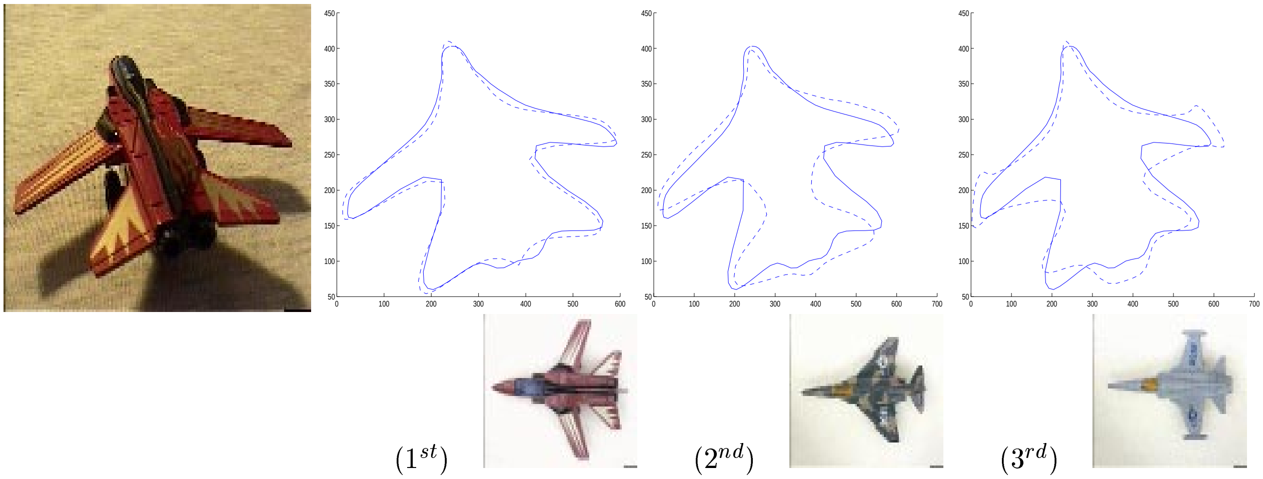
(a) (b)

Figure 19: (a) Affine invariant signature of airplane C (incorrectly) failed to match with that of model 2 (ranked the 4^{th}); (b) instead, model 6 was the top match found.



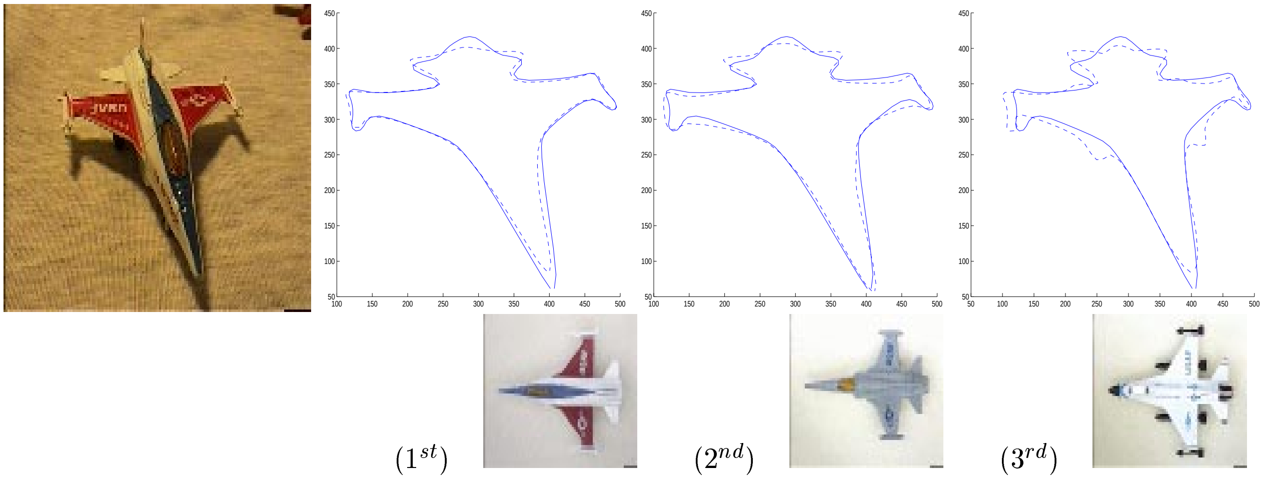
(1^{st}) (2^{nd}) (3^{rd})

Figure 20: Observed image A, with the top three matches from the database, using perspective invariants. (Solid for the observed image, dashed for the estimated image.)



(1st) (2nd) (3rd)

Figure 21: Observed image I, with the top three matches from the database, using perspective invariants. (Solid for the observed image, dashed for the estimated image.)



(1st) (2nd) (3rd)

Figure 22: Observed image K, with the top three matches from the database, using perspective invariants. (Solid for the observed image, dashed for the estimated image.)

with the corresponding database model are shown.

In order to show the superior performance of using perspective invariants over affine invariants, we computed the signatures of two images, *after* obtaining the estimated affine and perspective transformations. In Fig. 23, (a) is the observed image E, (b) is the best match using the affine transformation, and (c) is the signature match for the observed image (solid) and the best affine estimate (dashed). In contrast, (d) is the best estimated image using perspective invariants, and (e) is their signature match. Clearly, using perspective invariants is superior over affine invariants, in cases of large perspective distortion. Fig. 24 show similar results for observed image J.

For this experiment, *all* observed images were correctly matched with the models from the database, using perspective invariants. However, the system is far from perfect. The results for observed image F show that it was correctly identified (model 4). Nonetheless, the relatively large error (possibly due to noise) between the observed image and estimated image may be unacceptable. On the other hand, the difference between the error of two different matches may not be significantly high enough, which decreases the level of confidence in the correctness of the object identification. For instance, the error values of the top two matches for airplane K were very close to each other.

This confidence can be strengthened by testing whether the regions inside the object contours are also consistent. Illumination invariants readily applies here.

For illumination invariants, curves were uniquely defined on the surface of each airplane model in the database (performed off-line), so that its superimposition over the image emphasizes important (or interesting) color patterns in the image. Figs. 25(a) and (d) and Fig. 26(d) show one of many such curves defined for models 7, 14 and 3, respectively. (in this case, a zig-zagging curve over the different colors of the airplane). A different set of curves for other models in the database were also defined. Using the same strategy as that for affine invariants, we determined a resolution level (possibly many) to be used for recognition; in this case, level 7. Computation of the illumination invariant signatures were done off-line.

After computing the perspective invariants, the transformation parameters were easily obtained (Eq. 15). The same parameters were used to transform the curve defined for each model, to its correct pose in the observed image. Hence, the colors defined by the curve in the model should match the colors defined by the transformed curve in the observed image (except for changes due to illumination). Illumination invariant signatures for the observed images were then computed, and compared with the signatures stored in the database. Again, correlation coefficients were used to measure the similarity between the pairs of signatures.

Illumination invariants were computed only for those matches that had a certain level of uncertainty. Here, we only show results of illumination invariant computations where the (perspective invariant) errors of the 1^{st} and 2^{nd} best matches differ by a small amount; in this case, observed images H and K.

In Fig. 25, (a) and (d) show one of many curves superimposed (in red) over the image of models 7 and 14. The transformed curves, shown in (b) and (e), is superimposed (in green) over the observed image H. The estimated image (from perspective invariants) is outlined in blue. Finally, (c) and (f) show the illumination invariant signatures. Clearly, the signatures in (c) is more consistent.

Fig. 26 shows similar results for observed image K, with models 14 and 3.

As a final measure for recognition, the weighted average between perspective invariants and illumination invariants can be computed (depending on their relative importance). For this experiment, any weighting will produce the correct results, since both the perspective invariants and illumination invariants always ranked the correct pair as the best match. It should be noted however, that the success of the illumination invariants relies heavily on the success of the perspective invariants. (*I.e*, If the estimated image does not accurately match the observed image, then the

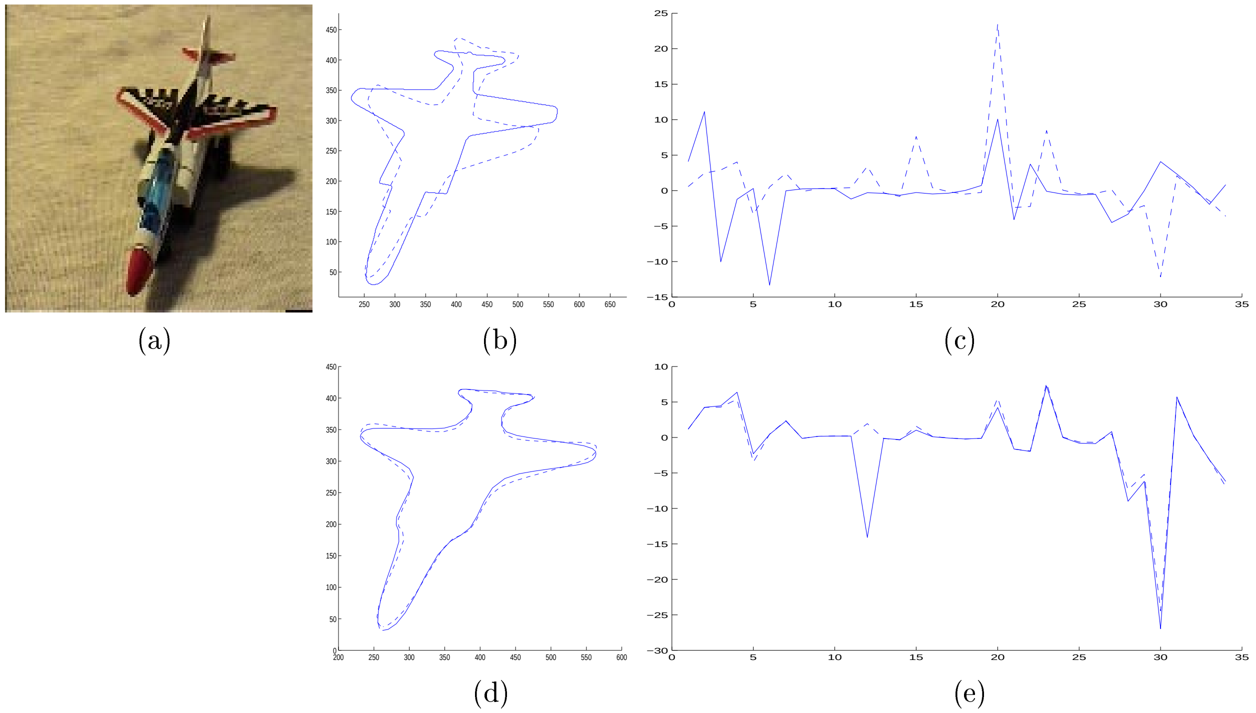


Figure 23: (a) Observed image E (solid), (b) the best affine estimate (dashed), (c) signature match using affine invariants, (d) the best perspective estimate (dashed), and (e) signature match using perspective invariants.

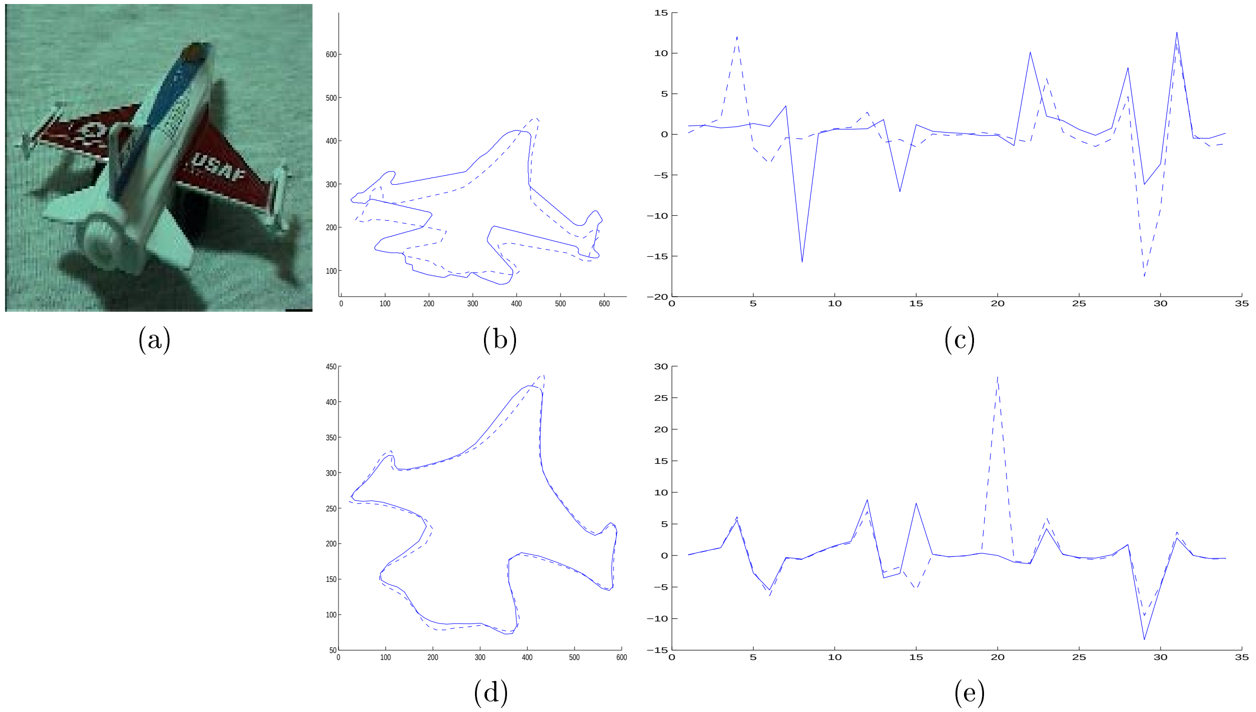
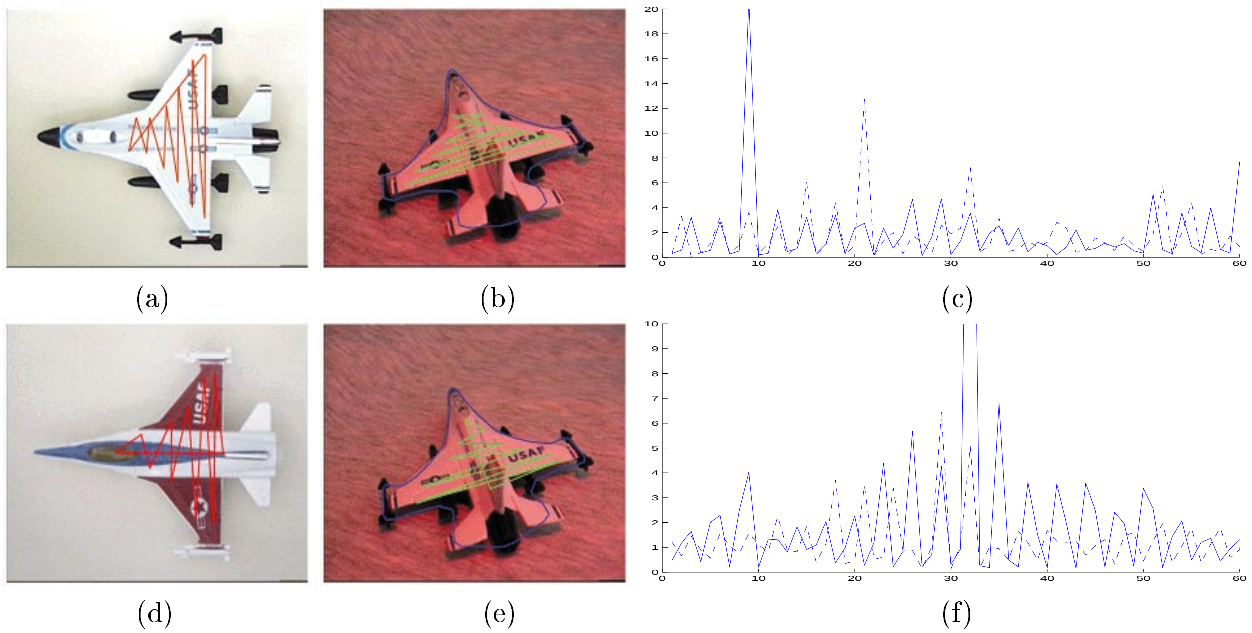


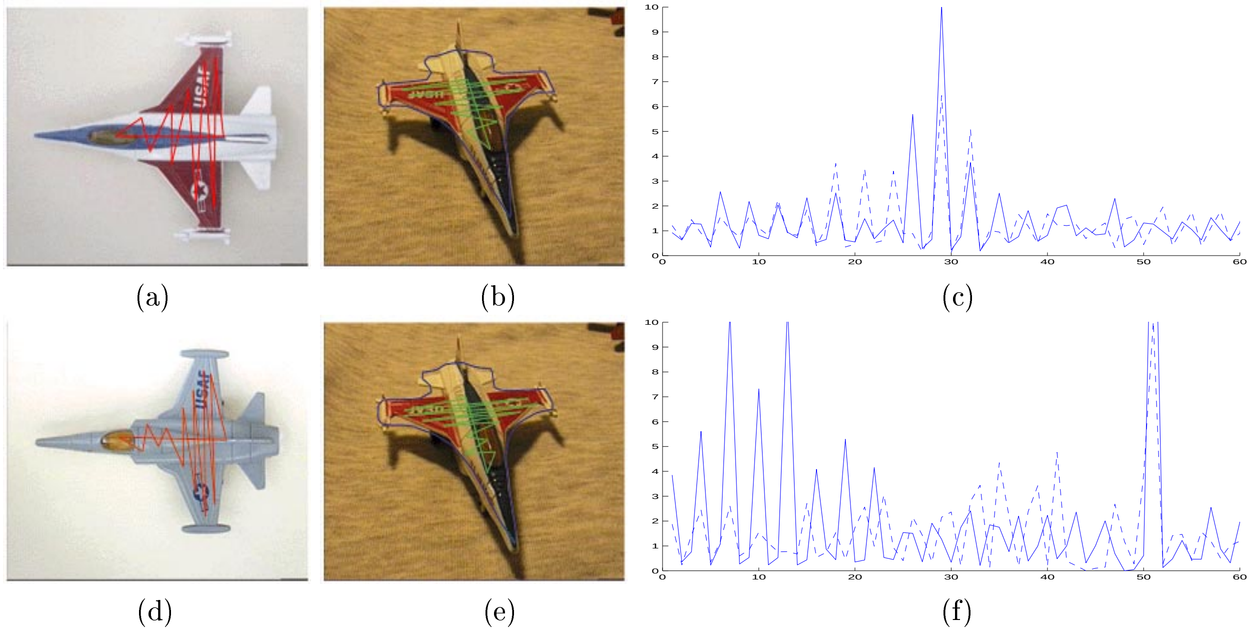
Figure 24: (a) Observed image J (solid), (b) the best affine estimate (dashed), (c) signature match using affine invariants, (d) the best perspective estimate (dashed), and (e) signature match using perspective invariants.



(a) (b) (c)

(d) (e) (f)

Figure 25: (a),(d) Airplane models with one of the defined curves (in red) superimposed, (b),(e) observed image with the transformed curve superimposed (in green). The transformed contour of the model is in blue, and (c),(f) illumination invariant signatures for observed image H (solid) and for models 7 and 14 (dashed).



(a) (b) (c)

(d) (e) (f)

Figure 26: (a),(d) Airplane models with one of the defined curves (in red) superimposed, (b),(e) observed image with the transformed curve superimposed (in green). The transformed contour of the model is in blue, and (c),(f) illumination invariant signatures for observed image K (solid) and for models 14 and 3 (dashed).

illumination invariants will probably not match as well.)

5 Conclusion

In this paper we present a new framework for computing image invariants. The framework utilizes many desirable properties of wavelet and basis expansion techniques, including the ability to analyze the shape and color at different resolution levels. Both geometric and illumination invariants were discussed, including the potential for combining them into one system. Furthermore, the formulations are quite simple and straightforward to implement.

Preliminary results on both real and synthetic images are very promising. These results demonstrate the tolerance to noise, affine transformations, perspective distortion and illumination changes, and the ability for hierarchical shape analysis.

Acknowledgment: We thank Dr. N. Nandhakumar and Professor T.C. Pong for their careful proofreading of and constructive comments on this manuscript. This research was supported in part by NSF/NASA/ARPA through the Alexandria Digital Library Project (under grant NSF IRI94-11330) and in part by a grant from Electroglas Inc.

References

[1] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger. Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects. *IEEE Trans. Pattern Analy. Machine Intell.*, 12:640–647, 1990.

[2] W. Bohm, G. Farin, and J. Kahmann. A Survey of Curve and Surface Methods in CAGD. *Comput. Aided Geometric Des.*, pages 1–60, 1984.

[3] J. M. Combes, A. Grossman, and Ph. Tchamitchian (Eds.). *Wavelets: Time-Frequency Methods and Phase Space, 2nd ed.* Springer-Verlag, Berlin, 1990.

[4] I. Daubechies. Orthonormal Bases of Compactly Supported Wavelets. *Commun. Pure Appl. Math.*, 41:909–960, 1988.

[5] C. deBoor. *A Practical Guide to Spline.* Springer-Verlag, New York/Berlin, 1978.

[6] S. J. Dickinson, A. Pentland, and A. Rosenfeld. 3-D Shape Recovery Using Distributed Aspect Matching. *IEEE Trans. Pattern Analy. Machine Intell.*, 14(2):174–198, 1992.

[7] L. E. Dickson. *Algebraic Invariants.* John-Wiley & Sons, 1914.

[8] D. Eggert and K. Bowyer. Computing the Perspective Projection Aspect Graph of Solids of Revolution. *IEEE Trans. Pattern Analy. Machine Intell.*, 15(2):109–128, 1993.

[9] D. Eggert, K. Bowyer, C. R. Dyer, and H. I. Christensen. The Scale Space Aspect Graph. *IEEE Trans. Pattern Analy. Machine Intell.*, 15(11):1114–1130, 1993.

[10] G. Farin. Algorithms for Rational Bézier Curves. *Comput. Aided Des.*, 15(2):73–77, Mar. 1983.

[11] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice, 2nd ed.* Addison-Wesley, Reading, MA, 1990.

[12] D. Forsyth. A Novel Algorithm for Color Constancy. *Int. J. Comput. Vision*, 5:5–36, 1990.

[13] D. Gabor. Theory of Communication. *J. Inst. Elec. Eng.*, 93:429–457, 1946.

[14] H. W. Guggenheimer. *Differential Geometry.* McGraw-Hill, New York, 1963.

[15] G. Healey and A. Jain. Using Physics-Based Invariant Representations for the Recognition of Regions in Multispectral Images. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pages 750–755, San Francisco, CA, Jun. 96.

[16] G. Healey and D. Slater. Global Color Constancy: Recognition of Objects by Use of Illumination-Invariant Properties of Color Distributions. *Opt. Soc. Am. A*, 11(11):3003–3010, Nov. 1994.

[17] G. Healey and L. Wang. Illumination-Invariant Recognition of Texture in Color Images. *Opt. Soc. Am. A*, 12(9):1877–1883, Sep. 1995.

[18] IGES. *Initial Graphics Exchange Specifications, Ver. 3.0.* Nat. Bur. of Stds., Gaithersburg, MD, 1986.

[19] R. Kondepudy and G. Healey. Use of Invariants for Recognition of Three-Dimensional Color Textures. *Opt. Soc. Am. A*, 11(11):3037–3049, Nov. 1994.

[20] Yehezkel Lamdan, Jacob T. Schwartz, and Haim J. Wolfson. Affine Invariant Model-Based Object Recognition. *IEEE Trans. Robot. and Automat.*, 6(5):578–589, 1990.

[21] E. P. Lane. *Projective Differential Geometry of Curves and Surfaces.* Univ. of Chicago Press, Chicago, IL, 1932.

[22] Guo Lei. Recognition of Planar Objects in 3-D Space from Single Perspective Views Using Cross Ratio. *IEEE Trans. Robot. and Automat.*, 6(4):432–437, 1990.

[23] Simon X. Liao and Miroslaw Pawlak. On Image Analysis by Moments. *IEEE Trans. Pattern Analy. Machine Intell.*, 3:254–266, 1996.

[24] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Analy. Machine Intell.*, 11(7):674–693, 1989.

[25] S. G. Mallat. Multifrequency Channel Decompositions of Images and Wavelet Models. *IEEE Trans. Acoust. Speech Signal Processing*, 37:2091–2110, 1989.

[26] S. G. Mallat. Zero-Crossings of a Wavelet Transform. *IEEE Trans. on Information Theory*, 37(4):1019–1033, 1991.

[27] L. Maloney. Evaluation of Linear Models of Surface Spectral Reflectance with Small Number of Parameters. *Opt. Soc. Am. A*, 3:1673–1683, 1986.

[28] L. Maloney and B. Wandell. Color Constancy: A Method for Recovering Surface Spectral Reflectance. *Opt. Soc. Am. A*, 3:29–33, 1986.

[29] J. Michel, N. Nandhakumar, and V. Velten. Thermophysical Algebraic Invariants from Infrared Imagery for Object Recognition. *IEEE Trans. Pattern Analy. Machine Intell.*, 19:41–51, 1987.

[30] T. Miyatake, T. Matsuyama, and M. Nagao. Affine Transform Invariant Curve Recognition Using Fourier Descriptors (original in Japanese). *Trans. Inform. Processing Soc. Japan*, 24(1):64–67, 1983.

[31] J. Mundy and A. Zisserman (eds.). *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, MA, 1992.

[32] Joseph L. Mundy, Andrew Zisserman, and David Forsyth (Eds.). Proc. Applications of Invariance in Computer Vision. In *Proc. Second Joint European-US Workshop*, Oct. 1993.

[33] L. Peigl and W. Tiller. Curve and Surface Constructions using Rational B-Splines. *Comput. Aided Des.*, 19(9):485–498, 1987.

[34] S.J. Perantonis and P.J. Lisboa. Translation, Rotation, and Scale Invariant Pattern Recognition by High-order Neural Networks and Moment Classifiers. *IEEE Trans. on Information Theory*, 3:241–251, 1992.

[35] L. R. Rabiner and D. W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, Englewood Cliffs, NJ, 1978.

[36] T. H. Reiss. *Recognizing Planar Objects Using Invariant Image Features.* Springer-Verlag, Berlin, 1993.

[37] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics, 2nd Ed.* McGraw-Hill, New York, NY, 1990.

[38] E.P. Simoncelli, E.H. Adelson W.T. Freeman, and D.J. Heeger. Shiftable Multiscale Transforms. *IEEE Trans. on Information Theory*, 38:587–607, 1991.

[39] D. Slater and G. Healey. The Illumination-Invariant Recognition of 3D Objects Using Local Color Invariants. *IEEE Trans. Pattern Analy. Machine Intell.*, 18(2):206–210, Feb. 1996.

[40] D. Slater and G. Healey. Using a Spectral Reflectance Model for the Illumination-Invariant Recognition of Local Image Structure. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pages 770–775, San Francisco, CA, Jun. 1996.

[41] D. Slater and G. Healey. The Illumination-Invariant Matching of Deterministic Local Structure in Color Images. *IEEE Trans. Pattern Analy. Machine Intell.*, 19(10):1146–1151, 1997.

[42] Q. M. Tieng and W. W. Boles. Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation. *IEEE Trans. Pattern Analy. Machine Intell.*, 19(8):910–916, Aug. 1997.

[43] Q. M. Tieng and W. W. Boles. Wavelet-Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space. *IEEE Trans. Pattern Analy. Machine Intell.*, 19(8):846–857, Aug. 1997.

[44] W. Tiller. Rational B-Splines for Curve and Surface Representation. *IEEE Comput. Graphics & App.*, 3(6):61–69, 1987.

[45] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding.* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[46] L. Wang and G. Healey. Illumination and Geometry Invariant Recognition of Texture in Color Images. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pages 419–424, San Francisco, CA, Jun. 1996.

[47] I. Weiss. Geometric Invariants and Object Recognition. *Int. J. Comput. Vision*, 10(3):207–231, 1993.

[48] W-R Wu and S-C Wei. Rotation and Gray-Scale Transformation-Invariant Texture Classification Using Spiral Resampling, Subband Decomposition, and Hidden Markov Model. *IEEE Trans. Image Processing*, 5(10):1423–1434, Oct. 1996.