

Two-dimensional Object Recognition Based on the Method of Moving Frame

Mahmoud Elgamal*

Abstract— Invariant features play a key role in object and pattern recognition studies. Features that are invariant to geometrical transformations offer succinct representations of underlying objects so that they can be reliably identified. In this paper, a family of novel invariant features is introduced based on Cartan's theory of moving frames. These new features is called summation invariants. Compared to existing invariant features, summation invariants are inherently numerically stable, and do not require computationally complex numerical integrations or analytical representations of underlying data. A robust methods for extracting summation invariants from sampled 2D contours introduced. Then, these new invariant features are applied to 2D object recognition and compared to other methods, e.g. wavelet and found to be more efficient.

Keywords: Object recognition, Method of moving frames, invariant features.

1 Introduction

Object recognition is a major goal of many computer vision applications, but many unsolved issues characterize the development of practical recognition systems. A typical object recognition problem can be described as follow. An object can be observed from a different point of view, and hence, the appearance of the same object will vary. For the objects we want to recognize, their templates, called gallery images, are stored in a database. The task is to match an unknown object observed from an unknown point of view, called the probe image, against all the templates in a database. Now, let us discuss a little bit how an object recognition system works. Given an image of an object, one wants to use an algorithm that produces a unique feature from the image. Given another image of the same object, the algorithm should produce exactly the same feature. To achieve this, the algorithm has to cancel the effect of shape transformations which is the result of changing viewing angles. There are many approaches by which one can deal with shape transformations between images. An intuitive ap-

proach is simply to perform every possible transformation of a template to see whether any of the templates transformed versions match the probe image. For example, if a probe image and a template differ only by translation [2], then one can shift the template, pixel by pixel, until the best match is reached. Obviously, this kind of method is computationally intensive. Also, for common transformations in computer vision, such as affine transformation and projective transformation, the search space becomes overwhelmingly large. Another method is to store images of an object from all possible viewing angles in the database. By doing this, one can reduce the computational complexity significantly, but at the price of increasing the storage requirement. Once again, for the common transformations in computer vision, this approach is not an efficient solution and probably not feasible. Due to the inherent drawbacks of the previous two approaches, it is necessary to come up with a more practical solution for real world applications. One can use the concept of geometric invariance to tackle this challenging problem. Ideally, the invariant descriptors are independent of view-points. By using this kind of shape descriptor, one can match two images directly without any prior knowledge of the position and the orientation of the camera. Therefore, the invariant-based method is neither expensive on computation nor storage. For an invariant descriptor, another important property is that it should extract all the distinctive information about a shape. Thus, one can identify different shapes by using an invariant feature. In this paper, geometric transformations used in developing summation invariants, these invariant features are robust in the presence of noise and quantization error. The paper is organized as follow, in section 2 a brief introduction is given for geometric invariants, e.g., Lie group, orbit, invariant functions, jet space, group action prolongation, and the method of moving frames. In section 3, the differential and integral invariants is presented. In section 4, the summation invariants are addressed, and in section 5, the error analysis is discussed. In section 6, a simulation example is demonstrated. In section 7, an application experiment for the method is conducted.

*PhD, Japan. Email: mahmoud_40@yahoo.com

2 Preliminaries

a.) A *Transformation group* acting on a manifold M is a Lie group G which satisfies

$$e \circ x = x, \quad g \circ (h \circ x) = (g \circ h) \circ x, \quad \forall x \in M, g \in G,$$

where e is an identity element of the Lie group G .

b.) *Orbits and invariant functions*: Let G be a Lie group acting on a manifold M . The orbit of a point $x \in M$ is the set of all images of x under group action. To be precise, the orbit of x is given by:

$$\mathcal{O}_x = \{g \circ x : g \in G\}.$$

An invariant for the transformation group is a function $\eta : M \rightarrow \mathbb{R}$ which satisfies

$$\eta(g \circ x) = \eta(x), \forall g \in G, x \in M.$$

c.) *Jet spaces and Prolongations*[13]: unfortunately, it is quite common that the dimensions of an orbit is greater than or equal to the dimension of the manifold. One way to fix this problem is to create a larger space so that we can find an invariant function there. For this purpose we will introduce a suitable space, called *jet space*. Then define how group actions are *prolonged* so that coordinates of jet space are appropriately transformed. Traditionally, a jet space is coordinated by independent variables, dependent variables and derivatives of dependent variables. Hence, it is called the *derivative jet space*. Consider a smooth function $u = f(x)$ which involves p independent variables $x = (x_1, \dots, x_p)$ and q dependent variables $u = (u_1, \dots, u_q)$. This allows us to define the *derivative jet space* J^n , whose coordinates consist of p independent variables, q dependent variables and all the partial derivatives of order up to n . A point in the derivative jet space J^n is denoted by (x, u^n) , where u^n contains dependent variables and partial derivatives up to order n . This action of group G on J^n is called the n^{th} *prolongation* and it is denoted by $pr^n G$. This prolonged group action is defined so that the derivatives of function $u = f(x)$ are mapped to corresponding derivatives of transformed function $\bar{u} = f(\bar{x})$. Specifically, for any point $(x_0, u_0(n)) \in J^n$, the prolonged group action is defined by

$$pr^n \circ (x_0, u_0(n)) = (\bar{x}_0, \bar{u}_0(n))$$

In other words, the transformed derivatives are found by evaluating the derivatives of transformed function $f(\bar{x})$ at point \bar{x}_0 .

d.) *Method of moving frames*[13]: The invariant function of a group G acting on a jet space J^n is defined as

follows. An *invariant* for a group G is a function $\eta : J^n \rightarrow \mathbb{R}$ such that

$$\eta(pr^n g \circ (x, u^{(n)})) = \eta((x, u^{(n)})), \\ \forall pr^n g \in pr^n G \text{ and } (x, u^{(n)}) \in J^n.$$

In the next, a systematic way to produce the invariants of a Lie group G acting on a manifold M will be introduced. The *method of moving frames* provides a powerful and algorithmic tool to find invariant functions. But in practical applications, one is forced to *differential invariants* by a discrete numerical approximation and thus they are sensitive to noise.

3 Approximation of Differential Invariants

In order to reduce sensitivity to noise, *semi-differential invariants* were introduced by Van Gool et al. [4, 10]. In such an approach, a higher order differential is approximated by a joint invariant depending on lower order derivatives evaluated at several points on a curve. Thus, the computation of high order derivatives can be avoided. In [14], a robust differential technique has been developed. A *canonical* coordinate system which is independent of a given world coordinate system is defined so that all quantities defined on are invariants. This method involves smoothing a discretized curve to make the result reasonably reliable. Human judgments are required to determine appropriate coefficients for smoothing. Calabi et al. [3] constructed numerical approximations of differential invariants by using successive points on a curve. They discussed the cases of planar curves under Euclidean and affine transformations. Their numerical approximation of Euclidean curvature depends on Euclidean distance between successive points and that of affine curvature depends on triangular areas of successive points; it is similar to avoiding computation of high order derivatives.

In [12], Sato and Cipolla introduce a new framework of integral invariants under transformation group action based on *invariant parameterization*. They work on an important case in many computer vision problems, affine transformation acting on \mathbb{R}^2 . An affine quasi-invariant parameterization is derived. The idea behind it is to approximate affine arc-length by using low order derivatives. It is obvious that the distance and area are unaffected by Euclidean transformation. Based on this fact, Manay et al. [9] propose two integral invariants with respect to Euclidean transformation. Given a curve in \mathbb{R}^2 , they define a disk centered at a point on that curve. On the disk, the *integral distance invariant* computes distance metrics and the *integral area invariant* computes area metrics. By moving the disk and computing these two invariants at each point, they can plot the *integral invariant signature*. Unlike the method of moving frames, this method

can not systematically produce invariants. Also, the generalization of their method to other transformation group is not trivial.

3.1 Integral Invariant

Hann and Hickman [5] observe that the root of the problem lies in the derivation of invariants. The traditional approach prolongs group actions to derivatives and thus the resulting invariants will depend on derivatives. They define a new jet space, which is coordinatized by integrals. Thus, the resulting invariants will depend on integrals rather than derivatives and so will not be sensitive to noise. Compared with the previous two approaches, their method does not use any derivatives and can generate invariants systematically. They open a new window for the equivalence problem of shapes under transformation group actions. Since their work is particularly important, it will be discussed in some details.

3.2 Extending a Lie Group Action on \mathbb{R} to Potentials

Consider a Lie group G acting on \mathbb{R} defined by

$$g \circ (u, x) = (\bar{x}, \bar{u}), \quad g \in G$$

The conventional way of deriving invariants is to prolong the group action to *derivative jet space* J . For example $J = (x, u, u_x, u_{xx})$. For integral invariants, the goal is to extend group action to potentials rather than derivatives. The definition of potentials and potential jet space are defined as follow:

Definition 3.2.1 *The Potential V^{ij} of order k is given by*

$$V^{ij} = \int x^i u^j dx, \quad \text{where } j \neq 0 \text{ and } i + j = k. \quad (1)$$

Definition 3.2.2 *The potential jet space J_p^n is coordinatized by $(x_1, u_1, x_0, u_0, V^{(n)})$, (x_0, u_0) and (x_1, u_1) denote the initial and end points of a curve, respectively, $V^{(n)}$ consists of potentials up to n^{th} order (e.g., $J_p^1 = (x_1, u_1, x_0, u_0, V^{0,1})$, $J_p^2 = (x_1, u_1, x_0, u_0, V^{0,1}, V^{1,1}, V^{0,2})$).*

3.3 Affine Invariants

Consider the affine group on \mathbb{R}^2 ,

$$g \circ (x, u) = (ax + bu + c, dx + eu + f), \quad \det \begin{pmatrix} a & b \\ d & e \end{pmatrix} \neq 0$$

To find invariants, it is first solved for the *moving frame*, $\{a, b, c, d, e, f\}$, by setting

$$(\bar{x}_1, \bar{u}_1, \bar{x}_0, \bar{u}_0, \bar{V}^{0,1}, \bar{V}^{1,1}) = (0, 0, 1, 1, 0, 0)$$

Then, an integral invariant $\eta_{integral}$ can be found by substituting $\{a, b, c, d, e, f\}$ into $\bar{V}^{0,2}$.

$$\begin{aligned} \eta_{integral} = \{ & -3(x_1 - x_0)V^{0,2} + 6(u_1 - u_0)V^{1,1} + 4(V^{0,1})^2 \\ & - 2V^{0,1}(2x_1u_1 - 2x_0u_0 + x_0u_1 - x_1u_0) \\ & + u_0u_1(x_1 - x_0)^2 \} / \{ ((x_1 - x_0)(u_1 + u_0) \\ & - 2V^{0,1})^2 \} \end{aligned} \quad (2)$$

Given a curve $u = f(x)$ through (x_0, u_0) , $\eta_{integral}$ can be computed at each point (x, u) . Note that if two curves C and \bar{C} are related by an affine transformation, a point $(x_0, u_0) \in C$ must be identified with the corresponding point $(\bar{x}_0, \bar{u}_0) \in \bar{C}$ to evaluate $\eta_{integral}$. Integral invariants are different from differential invariants in the sense that differential invariants are defined locally. They are also different from the moment invariants [6], which are defined globally. Integral invariants are defined semi-locally because the range of integration can be specified by varying (x_0, u_0) and (x_1, u_1) . In other words, integral invariants can extract local characteristics like differential invariants do. Meanwhile, unlike moment invariants, they do not suffer from occlusion problems. This property gives us more flexibility in dealing with practical problems. In practical object recognition problems, we are usually given sets of discrete points. Each set of discrete points represents a shape and we want to classify these shapes. From Hann and Hickmans point of view, they assume there are continuous shapes behind the discrete data, which allows them to compute the integral invariants. Their idea has one obvious drawback. It will highly depend on the sampling resolution. The higher the sampling rate is, the more accurate the integral will be.

4 Summation Invariants

A reliable scheme developed here to find invariants, which are based on the summation operation of discrete data, called *Summation invariants*. They are independent of sampling rate and maintain all the nice properties of integral invariants, such as systematic production of invariants and high noise immunity. The transformation groups used to illustrate these novel techniques are Euclidean and affine groups, with emphasize on the cases of curves and surfaces in two and three-dimensional space. In the next section definitions of *potential* and *potential jet space* are given. Then, summation invariants for curves in \mathbb{R}^2 and for the surfaces in \mathbb{R}^3 are

explicitly derived. Also a variety of fundamental geometric transformations are discussed in detail.

4.1 Summation Invariants for Curves in \mathbb{R}^2

Suppose that the boundary of a planar object is given, which is parameterized as $\{(x_n, y_n) : n = 1, 2, \dots, N\}$. Note that the parameterization is not necessarily equally spaced. Consider the transformation group G acting on \mathbb{R}^2 defined by

$$g \circ (x_n, y_n) = (\bar{x}_n, \bar{y}_n), \quad g \in G$$

The *potential* and *potential jet space* are defined as follows.

Definition 4.1.1 The potential $P_{i,j}$ of order k is given by

$$P_{i,j} = \sum_{n=1}^N x_n^i y_n^j, \quad \text{where } i + j = k$$

Definition 4.1.2 The potential jet space J^k is the Euclidean space with coordinates

$$(x_1, y_1, x_N, y_N, P_{(k)})$$

where $P_{(k)}$ consists of potentials up to k^{th} order.

For example,

$$J^1 = (x_1, y_1, x_N, y_N, \sum_{n=1}^N x_n, \sum_{n=1}^N y_n)$$

In the following two sections, we will address two important transformation groups in computer vision, namely Euclidean and affine transformations, and derive their summation invariants.

4.2 Euclidean Summation Invariant of Curves

Given the sampled points on a curve $\{(x_n, y_n) : n = 1, 2, \dots, N\}$, let (\bar{x}_n, \bar{y}_n) denote that they are transformed by Euclidean transformation.

$$\begin{bmatrix} \bar{x}_n \\ \bar{y}_n \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (3)$$

where $a, b, \theta \in \mathbb{R}$. One can construct a family of invariant functions η_{ij} by applying a *moving frame* to potentials P_{ij} . For example, a moving frame can be found by solving the following equations,

$$(\bar{x}_1, \bar{y}_1, \bar{y}_N) = (0, 0, 0) \quad (4)$$

The equations above are called the *normalization equations*. Note that we are free to specify normalization equations as long as they can be solved, i.e. a moving frame can be found. Here, we specify the normalization equations so that the corresponding moving frame can be easily solved. Let \bar{P}_{ij} be the potentials transformed by a moving frame,

$$\bar{P}_{ij} = \sum_{n=1}^N \bar{x}_n^i \bar{y}_n^j \quad (5)$$

where \bar{x} and \bar{y} denote the x and y coordinates transformed by a moving frame. The \bar{P}_{ij} are invariant functions under Euclidean transformation acting on \mathbb{R}^2 [8], i.e., $\eta_{ij} = \bar{P}_{ij}$. The first and second invariant functions, $i + j = 1$ or 2 , were explicitly derived as shown below

$$\begin{aligned} \eta_{1,0} &= P_{1,0}(x_N - x_1) + P_{0,1}(y_N - y_1) \\ &\quad + Nx_1(x_1 - x_N) + Ny_1(y_1 - y_N) \end{aligned} \quad (6)$$

$$\begin{aligned} \eta_{0,1} &= P_{0,1}(y_N - y_1) + P_{0,1}(x_1 - x_N) \\ &\quad + N(x_N y_1 - x_1 y_N) \end{aligned} \quad (7)$$

$$\begin{aligned} \eta_{2,0} &= -2P_{1,0}(x_1 - x_N)(x_1^2 - x_1 x_N + y_1^2 - y_1 y_N) \\ &\quad - 2P_{0,1}(y_1 - y_N)(y_1^2 - y_1 y_N + x_1^2 - x_1 x_N) \\ &\quad + P_{2,0}(x_1 - x_N)^2 + P_{0,2}(y_1 - y_N)^2 \\ &\quad + 2P_{1,1}(x_1 - x_N)(y_1 - y_N) \\ &\quad + N(x_1(x_1 - x_N) + y_1(y_1 - y_N))^2 \end{aligned} \quad (8)$$

$$\begin{aligned} \eta_{1,1} &= P_{1,1}((x_1 - x_N)^2 - (y_1 - y_N)^2) \\ &\quad + P_{1,0}(y_1^3 + 2x_1 x_N y_N - 2y_N x_1^2 \\ &\quad + x_1^2 y_1 - 2y_1^2 y_N + y_1 y_N^2 - x_N^2 y_1) \\ &\quad - P_{0,1}(x_1^3 + 2y_1 y_N x_N - 2x_N y_1^2 \\ &\quad + y_1^2 x_1 - 2x_1^2 x_N + x_1 x_N^2 - y_N^2 x_1) \\ &\quad + (P_{0,2} - P_{2,0})(x_1 - x_N)(y_1 - y_N) \\ &\quad + N(x_N y_1 - x_1 y_N)(x_1(x_N - x_1) + y_1(y_N - y_1)) \end{aligned} \quad (9)$$

$$\begin{aligned} \eta_{0,2} &= 2(x_N y_1 - x_1 y_N)(P_{1,0}(y_N - y_1) - P_{0,1}(x_N - x_1)) \\ &\quad + P_{2,0}(y_1 - y_N)^2 + P_{0,2}(x_1 - x_N)^2 \\ &\quad - 2P_{1,1}(x_1 - x_N)(y_1 - y_N) + N(x_1 y_N - x_N y_1)^2 \end{aligned} \quad (10)$$

4.3 Affine Summation Invariant of Curves

Affine Transformation can be used to approximate the perspective projection of 3D objects on the image plane. Therefore, invariants of affine transformation have significant importance in computer vision. In this section, we explicitly derive the summation invariants of affine transformations. Consider the affine transformation group G

acting on \mathbb{R}^2 given by

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \quad (11)$$

where $\det \begin{bmatrix} a & b \\ d & e \end{bmatrix} \neq 0$ Again, we prolong group action to potentials. The affine-transformed potentials are shown below

$$\bar{P}_{1,0} = \sum_{n=1}^N (ax_n + by_n + c) = a.P_{1,0} + b.P_{0,1} + c.N \quad (12)$$

$$\bar{P}_{0,1} = \sum_{n=1}^N (dx_n + ey_n + f) = d.P_{1,0} + e.P_{0,1} + f.N \quad (13)$$

Next, *moving frame* $\{a, b, c, d, e, f\}$ can be found by solving the *normalization equations*. In this case, the *normalization equations* specified as follows

$$(\bar{x}_1, \bar{y}_1, \bar{x}_N, \bar{y}_N, \bar{P}_{1,0}, \bar{P}_{0,1}) = (0, 0, 1, 1, 0, 0) \quad (14)$$

By applying the *moving frame* $\{a, b, c, d, e, f\}$ to higher order potentials, it will yield the summation invariants under affine group actions. The second order potentials are shown below

$$P_{2,0} = \sum_{n=1}^N x_n^2 \quad (15)$$

$$P_{1,1} = \sum_{n=1}^N x_n y_n \quad (16)$$

$$P_{0,2} = \sum_{n=1}^N y_n^2 \quad (17)$$

They are transformed by affine group action as follows

$$\begin{aligned} \bar{P}_{2,0} = a^2 P_{2,0} + 2ab P_{1,1} + 2ac P_{1,0} + 2bc P_{0,1} \\ + b^2 P_{0,2} + c^2 N \end{aligned} \quad (18)$$

$$\begin{aligned} \bar{P}_{1,1} = ad P_{2,0} + (ae + bd) P_{1,1} + (af + cd) P_{1,0} \\ + (bf + ce) P_{0,1} + be P_{0,2} + cf N \end{aligned} \quad (19)$$

$$\begin{aligned} \bar{P}_{0,2} = d^2 P_{2,0} + 2de P_{1,1} + 2df P_{1,0} + 2ef P_{0,1} \\ + e^2 P_{0,2} + f^2 N \end{aligned} \quad (20)$$

Then, we apply *moving frame* $\{a, b, c, d, e, f\}$ solved from equation (14) to $\bar{P}_{2,0}$. The resulting affine transformation invariant $\xi_{2,0}$ is given by

$$\begin{aligned} \xi_{2,0} = \{P_{2,0}(Ny_1 - P_{0,1})^2 + P_{0,2}(Nx_1 - P_{1,0})^2 \\ - 2P_{1,1}(Nx_1 - P_{1,0})(Ny_1 - P_{0,1}) \\ - N(y_1 P_{1,0} - x_1 P_{0,1})^2\} / \{(N(x_N y_1 - x_1 y_N) \\ + (y_N - y_1)P_{1,0} - (x_N - x_1)P_{0,1})^2\} \end{aligned} \quad (21)$$

Note that one can find infinitely many affine invariant functions by substituting solved values of $\{a, b, c, d, e, f\}$ into higher order transformed potentials. Furthermore, one can show that

$$g \circ \text{denominator}(\xi_{2,0}) = (ae - bd)^2 \text{denominator}(\xi_{2,0}) \quad (22)$$

$$g \circ \text{numerator}(\xi_{2,0}) = (ae - bd)^2 \text{numerator}(\xi_{2,0}) \quad (23)$$

In other words, both denominator and numerator of $\xi_{2,0}$ are relative invariants under affine transformation. Form an application point of view, relative invariants provide enough information for object recognition and there is no need to worry about the situation where the denominator vanishes.

5 Robustness in the Presence of Noise

Without the presence of noise, invariant functions obtained by different approaches all exhibit perfect invariance properties. However, it is always the case in practical situations that the invariance properties will not be exactly maintained and one can only compute invariant features with limited precision. Thus, it is of great importance to analyze the robustness to noise for different kinds of invariant functions. In this section, we will compare the summation invariants and the integral invariants in terms of their numerical robustness. In particular, theoretical analysis relating second-order statistics are derived to evaluate the impact of noise. Suppose one wants to evaluate a function f of several variables X_1, X_2, \dots, X_n . The variables X_1, X_2, \dots, X_n are only available in quantized form, i.e., we only have access to $q(X_1), q(X_2), \dots, q(X_n)$ where the quantized function is defined by

$$q(x) = k\Delta_q, \text{ if } (k - 0.5)\Delta_q \leq x \leq (k + 0.5)\Delta_q \quad (24)$$

Thus, we are only concerned with the error due to quantization of the value of variables and are not concerned with error due to computation or evaluation of functions. In principle, we can compute the moments of Δf if the joint probability distribution of X_1, X_2, \dots, X_n is known. Specifically, the n^{th} moment of Δf is given by

$$M_n = \int \dots \int (\Delta f)^n h_{x_1, x_2, \dots, x_n}(x_1, x_2, \dots, x_n). dx_1 dx_2 \dots dx_n$$

where h_{x_1, x_2, \dots, x_n} is the joint probability density function of X_1, X_2, \dots, X_n . However, in practice we will see that f can be linear or non linear function of its variables and an analytical evaluation of quantization error can be difficult. Some assumptions is usually made to simplify the error analysis. The error analysis of summation invariants and integral invariants will be presented in the

following two sections. For each, its robustness is characterized in terms of the variance of error in the computed value, as described by equation (24).

5.1 Error Analysis on Summation Invariants

In this section, we will give the analytic derivation of the expected squared error for two simple cases, namely $\eta_{0,1}$ and $\eta_{1,0}$. This serves to give an idea of the mathematics involved. For other summation invariants, their error analysis can be done in the same way. The focus is placed on $\eta_{0,1}$ and $\eta_{1,0}$ because they are composed of second order monomials. In the following section, the integral invariants will be composed of second order monomials, too. This leads to a fair comparison between them in term of their numerical stability. First, it is assumed without loss of generality that the quantization errors are independent identically distributed random variables. In order to simplify the analysis, it is further assumed that the mean of x coordinates and the mean of y coordinates are not zero, i.e.

$$m_x = \frac{1}{N} \sum_{n=1}^N x_n \neq 0 \tag{25}$$

$$m_y = \frac{1}{N} \sum_{n=1}^N y_n \neq 0 \tag{26}$$

Then,

$$\begin{aligned} \frac{\sum_{n=1}^N q(x_n)}{\sum_{n=1}^N x_n} &= \frac{\sum_{n=1}^N x_n + \epsilon_{x_n}}{\sum_{n=1}^N x_n} \\ &= 1 + \frac{\sum_{n=1}^N \epsilon_{x_n}}{\sum_{n=1}^N x_n} \\ &= 1 + \frac{\sum_{n=1}^N \epsilon_{x_n}}{Nm_x} \end{aligned} \tag{27}$$

where ϵ is the quantization error associated with x_n . It can be readily shown that

$$\lim_{N \rightarrow \infty} \frac{\sum_{n=1}^N q(x_n)}{\sum_{n=1}^N x_n} = 1 \tag{28}$$

Based on this, we can assume N is large enough so that the potential $P_{0,1}$ and $P_{1,0}$ are relatively insensitive to the quantization error, i.e.

$$\sum_{n=1}^N q(x_n) \approx \sum_{n=1}^N x_n \tag{29}$$

$$\sum_{n=1}^N q(y_n) \approx \sum_{n=1}^N y_n \tag{30}$$

Therefore, the error in the estimate $\bar{\eta}_{0,1}$ is given by

$$\begin{aligned} \Delta\eta_{0,1} &= \bar{\eta}_{0,1} - \eta_{0,1} \\ &\approx -(y_N - y_1) \sum_{n=0}^{N-1} \epsilon_{x_n} - (\epsilon_{y_N} - \epsilon_{y_1}) \sum_{n=0}^{N-1} x_n \\ &\quad + (x_N - x_1) \sum_{n=0}^{N-1} \epsilon_{y_n} + (\epsilon_{x_N} - \epsilon_{x_1}) \sum_{n=0}^{N-1} y_n \\ &\quad - N\{x_N \epsilon_{y_1} + y_1 \epsilon_{x_N} - x_1 \epsilon_{y_N} - y_N \epsilon_{x_1}\} \end{aligned} \tag{31}$$

One can compute the expected value and the variance of $\Delta\eta_{0,1}$

$$E[\Delta\eta_{0,1}] = 0 \tag{32}$$

$$\begin{aligned} var[\Delta\eta_{0,1}] &\approx \sigma^2 \{ (\sum_{n=1}^N y_n - Ny_N)^2 + (\sum_{n=1}^N y_n - Ny_1)^2 \\ &\quad + (\sum_{n=1}^N x_n - Nx_N)^2 + (\sum_{n=1}^N x_n - Nx_1)^2 \\ &\quad + N(y_N - y_1)^2 + N(x_N - x_1)^2 \} \end{aligned} \tag{33}$$

Here, the observations are the following. First, the quantization error does not introduce bias in the estimate $\bar{\eta}_{0,1}$. Second, the variance of the estimate $\bar{\eta}_{0,1}$ is proportional to N^2 . Similarly, the error in the estimate of $\eta_{1,0}$ is given by

$$\begin{aligned} \Delta\eta_{1,0} &= \bar{\eta}_{1,0} - \eta_{1,0} \\ &\approx -(\epsilon_{x_N} - \epsilon_{x_1})P_{1,0} - (\epsilon_{y_N} - \epsilon_{y_1})P_{0,1} \\ &\quad - N[-x_1 \epsilon_{x_N} + \epsilon_{x_1}(2x_1 - x_N) - y_1 \epsilon_{y_N} \\ &\quad \quad + \epsilon_{y_1}(2y_1 - y_N)] \end{aligned} \tag{34}$$

where the products of the quantization errors are regarded as negligible. The mean and variance of $\Delta\eta_{0,1}$ are given by

$$\begin{aligned} E[\Delta\eta_{1,0}] &= 0 \\ var[\Delta\eta_{1,0}] &\approx \sigma^2 \{ (-P_{1,0} + N(2x_1 - x_N))^2 + (P_{1,0} - Nx_1)^2 \\ &\quad + (-P_{0,1} + N(2y_1 - y_N))^2 + (P_{0,1} - Ny_1)^2 \} \end{aligned} \tag{35}$$

Here, the observation are the same as the previous case, i.e., quantization error does not introduce bias and the variance is proportional to N^2 .

5.2 Error Analysis on Integral Invariants

Hann and Hickman [5] introduced the integral invariants and explicitly derive the affine integral invariant equation (2). Unfortunately, it is not easy to give an analytical error analysis on equation (2). Instead we consider

a simpler case, Euclidean/rigid transformation group, and derive the corresponding integral invariants. Then, the mean and variance of the estimate of the Euclidean/rigid integral invariants can be derived analytically. Let $\{(x, y) : y = f(x), x \in [x_a, x_b]\}$ be a curve parameterized by x . In order to find integral invariants, one need to first define a *jet space* J . For Euclidean group acting on \mathbb{R}^2 , the jet space is given by

$$J = \{(x_a, y_a, y_b) : y_a = f(x_a), y_b = f(x_b)\} \quad (36)$$

Then, one can specify normalization equations and find the corresponding moving frame. To make the algebraic manipulation as easy as possible, we choose the normalization equations to be

$$(\bar{x}_a, \bar{y}_a, \bar{y}_b) = (0, 0, 0) \quad (37)$$

where $(\bar{x}_a, \bar{y}_a, \bar{y}_b)$ denotes the results of applying an Euclidean transformation to (x_a, y_a, y_b) . After solving the normalization equations, integral invariants can then be obtained by applying the moving frame to potentials. Recall that the potentials defined by Hann and Hickman [5] are shown in definition (3.2.1). Thus, $V^{0,1}$ is an order 1 monomial potential and $V^{1,1}, V^{0,2}$ are order 2 monomial potentials. The integral invariants shown below is obtained by applying the moving frame to $V^{0,1}$.

$$\mu_{0,1} = V^{0,1} - \frac{(x_b - x_a)(y_b + y_a)}{2} \quad (38)$$

The most straightforward methods for numerical integrations are the left and right Riemann sums. Here, the trapezoidal sums as a numerical approximation used. The trapezoid approximation of $V^{0,1}$ associated with the partition $x_a = x_0 < x_1 < \dots < x_{N-1} = x_b$ is given by

$$\bar{V}^{0,1} = \frac{1}{2} \sum_{n=0}^{N-2} (y_{n+1} + y_n)(x_{n+1} - x_n) \quad (39)$$

The estimate $\bar{\mu}_{0,1}$ is calculated by replacing $V^{0,1}$ in $\mu_{0,1}$ with $\bar{V}^{0,1}$. Note that the estimate $\bar{\mu}_{0,1}$ itself is also an invariant function, i.e.,

$$\bar{\mu}_{0,1}(x, y) = \bar{\mu}_{0,1}(\bar{x}, \bar{y}) \quad (40)$$

where (\bar{x}, \bar{y}) denotes a curve under arbitrary Euclidean/rigid transformations. Put

$$\nu_{0,1} = -2 \cdot \bar{\mu}_{0,1} \quad (41)$$

and it can be easily shown that

$$\nu_{0,1} = \sum_{n=0}^{N-1} (x_n y_{n+1} - x_{n+1} y_n) \quad (42)$$

where the indices of x, y are given modulo N . Following equation (42) and equation (24), the error in the integral

invariants $\nu_{0,1}$ due to the quantization in x and y coordinates is given by

$$\begin{aligned} \Delta \nu_{0,1} &= \sum_{n=0}^{N-1} x_n y_{n+1} - x_{n+1} y_n \\ &\quad - \sum_{n=0}^{N-1} q(x_n) q(y_{n+1}) - q(x_{n+1}) q(y_n) \\ &= \sum_{n=0}^{N-1} x_n y_{n+1} - x_{n+1} y_n - \sum_{n=0}^{N-1} (x_n + \epsilon_{x_n}) \\ &\quad (y_{n+1} + \epsilon_{y_{n+1}}) - (x_{n+1} + \epsilon_{x_{n+1}})(y_n + \epsilon_{y_n}) \end{aligned} \quad (43)$$

It is assumed without loss of generality that the $\epsilon_{x_n}, \epsilon_{y_n}$ are independent and identically distributed, with zero mean and variance σ^2 . Since the products of $\epsilon_{x_n}, \epsilon_{y_n}$ are relatively small, the quantization error $\Delta \nu_{0,1}$ is approximated by its first-order errors, i.e.

$$\Delta \nu_{0,1} \approx \sum_{n=0}^{N-1} x_n \epsilon_{y_{n+1}} + y_{n+1} \epsilon_{x_n} - x_{n+1} \epsilon_{y_n} - y_n \epsilon_{x_{n+1}} \quad (44)$$

Thus, the expected value and the variance of $\Delta \nu_{0,1}$ are given by

$$E[\Delta \nu_{1,0}] = 0 \quad (45)$$

$$\text{var}[\Delta \nu_{1,0}] = \sigma^2 \left(\sum_{n=0}^{N-1} (x_n - x_{n+2k})^2 + (y_n - y_{n+2k})^2 \right) \quad (46)$$

According to equation (44), the quantization error does not introduce bias in the estimate $\bar{\nu}_{0,1}$. Also, the variance of the estimate $\bar{\nu}_{0,1}$ is proportional to N and the quantization error σ .

6 Simulation

A numerical simulation is conducted to evaluate the accuracy of the derived first and second order statistics. The simulation consists of the following steps:

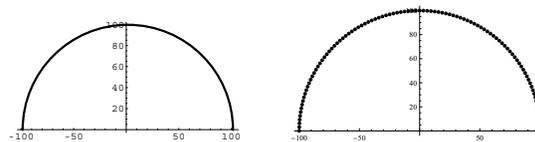


Figure 1: a) Semi-circle with radius 100.
b) Approximation of the semi-circle by 200 points.

- 1.) Generate a semi-circle with radius 100 and perform sampling uniformly along its arc length. The number of sampling points is 200 (figure (1)).
- 2.) Pick up the first N points of the semi-circle and compute its invariants. The initial value of N is 30.
- 3.) The curve with N points is subject to arbitrary Euclidean transformations and then quantization. The parameters of Euclidean transformation are randomly generated. During the quantization, x and y coordinates are rounded to the nearest integer. Then, the invariants computed from the quantized curve. This step is repeated 1000 times.
- 4.) Increase N by 10 and repeat steps 2 and 3.

In practical applications, shapes are always given in quantized format. Hence, there is no way for a user to control the quantization granularity. Reflecting on this fact, the quantization granularity is a fixed value in the simulation. It is also to be pointed out that the value of an invariant feature depends on the shape of an object. Here, the analysis is limited to the cases where quantization errors are relatively small, i.e., $\eta \gg \Delta\eta$. For the shapes not satisfying this condition, the computed invariant values are not reliable and hence have little practical value.

The variance of $\Delta\nu_{0,1}$ equation(46) was computed for the sampled points on the semi-circle *Theoretical estimate* and for the points generated using Euclidean transformation *Numerical approximation*. Figure (2) shows the result. It is observed that the results of simulation are very close to that of the theoretical analysis. Also, the variance is proportional to N as predicted by equation (46). The variance of $\Delta\eta_{0,1}$ are shown in figure (3). Again, the result of the simulation are very close to the theoretical predictions. However, it is also to be pointed out that choosing a large N has an undesirable consequence. In particular, the local characteristics of a shape will be averaged by using a large N . the rule of thumb is to keep N as small as possible while maintaining an acceptable accuracy. According to figure (4), $\eta_{0,1}$ and $\nu_{0,1}$ achieve similar accuracy in the range where N is relatively small. Thus, one may conclude that %1 mean squared relative error can be achieved for curve segments with 30 points by using either $\eta_{0,1}$ or $\nu_{0,1}$. Figure (4) shows the variance of relative errors which is a more descriptive quantity in assessing the precision of an estimate. It is immediately clear that relative error is inversely proportional to N . So, one can increase the accuracy of summation or integral invariants by increasing N . According to the error analysis in the previous section, the variances of $\Delta\nu_{0,1}$ and $\Delta\eta_{0,1}$ are proportional to N and N^2 respectively. Hence, the variance of the relative error $\Delta\nu_{0,1}/\nu_{0,1}$ is expected to drop faster than that of $\Delta\eta_{0,1}/\eta_{0,1}$. This phe-

nomenon is observed in the results of the simulation. It indicates that $\nu_{0,1}$ can reduce error more efficiently than $\eta_{0,1}$ does.

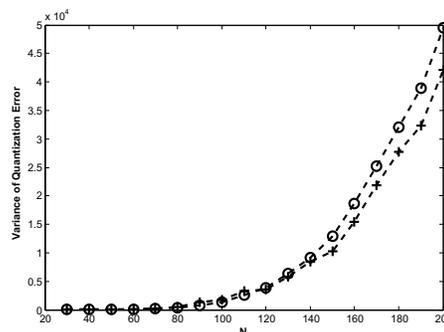


Figure 2: The variance of the quantization error $\Delta\nu_{0,1}$

($\circ \equiv$ Theo.est.(points sampled on the semi circle) and

$+$ \equiv Num. Approx.(points sampled on the Euclidean transformed curve)).

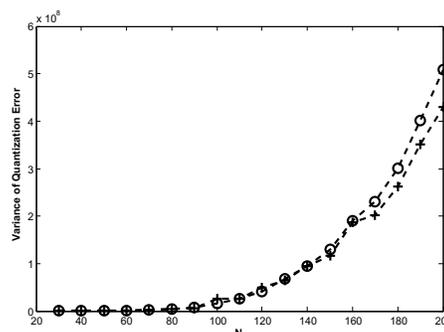


Figure 3: The variance of the quantization error $\Delta\eta_{0,1}$

($\circ \equiv$ Theo.est.(points sampled on the semi circle) and

$+$ \equiv Num. Approx.(points sampled on the Euclidean transformed curve)).

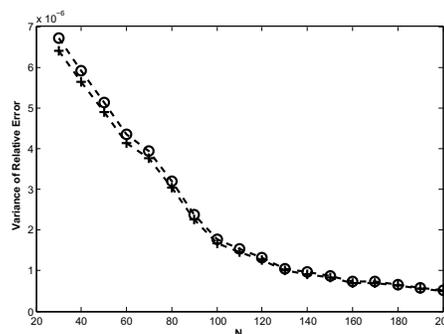


Figure 4: The variance of relative error $\Delta\eta_{0,1}/\eta_{0,1}$ and $\Delta\nu_{0,1}/\nu_{0,1}$ ($\circ \equiv \Delta\eta_{0,1}$ and $+$ $\equiv \Delta\nu_{0,1}$.)

7 Two-dimensional Shape Matching

In this section, a shape matching experimentation in 2D using summation invariants will be presented. As an application of the technique two dimensional affine sum-

mation invariants are utilized for recognizing computer-generated shape contours, i.e. closed curves. Finally, a comparison with other methods will be given.

7.1 Summation Invariants Representation of Curves

Given a set of sample points on a curve, one can compute summation globally, i.e. the resulting invariant feature is a scalar number. However, it will be very difficult to perform accurate classification in such a low-dimensional feature space since it does not adequately capture local variations of the underlying objects. To address this problem, a semi-local summation invariant feature that computes the summation invariant over a local interval of a curve is used rather than the entire curve. An important observation is that the lower and upper limits in the definition of the potentials are not functions of group operations. As such, these limits can be changed from all the points in the objects to a local subset of points. Let us use the affine invariants as an example to illustrate this method. For a curve $(x_n, y_n), n \in \{1, 2, \dots, N\}$, the m^{th} semi-local summation affine invariant is given by [8]:

$$\lambda_{affine}[m] = \left(M(x_{m_M}y_{m_1} - x_{m_1}y_{m_M}) + P_{1,0}(y_{m_M} - y_{m_1}) - P_{0,1}(x_{m_M} - x_{m_1}) \right)^2 \quad (47)$$

here

$$\begin{cases} P_{1,0} &= \sum_{n=m}^{m+M-1} x[\text{mod}(n, N)] \\ P_{0,1} &= \sum_{n=m}^{m+M-1} y[\text{mod}(n, N)] \\ m_1 &= m \\ m_M &= \text{mod}(m + M - 1, N) \end{cases}$$

Given N sampled points on a curve, one can compute semi-local summation invariants with non-overlapping or overlapping intervals. Obviously, using overlapping intervals will yield high dimensional features. For simplicity, we will use overlapping intervals. In the following experiment, the semi local summation invariant is computed from a local interval surrounding each sample point so that the resulting representation of a curve is a vector with dimension N . The length of the the local interval, M , which can be adjusted according to the specific application, is empirically determined.

7.2 Shape Matching Application Using Synthesized Data

A synthesized data set of computer-generated shapes(CG-shapes) by using Legendre polynomials

is created for shape matching experiment. These CG-shapes contours are depicted in Figure(5) and are called CG-shapes prototypes. Twenty deformations are generated for each prototype using affine transformations, for example, contour deformation of the first image is shown in figure (6).

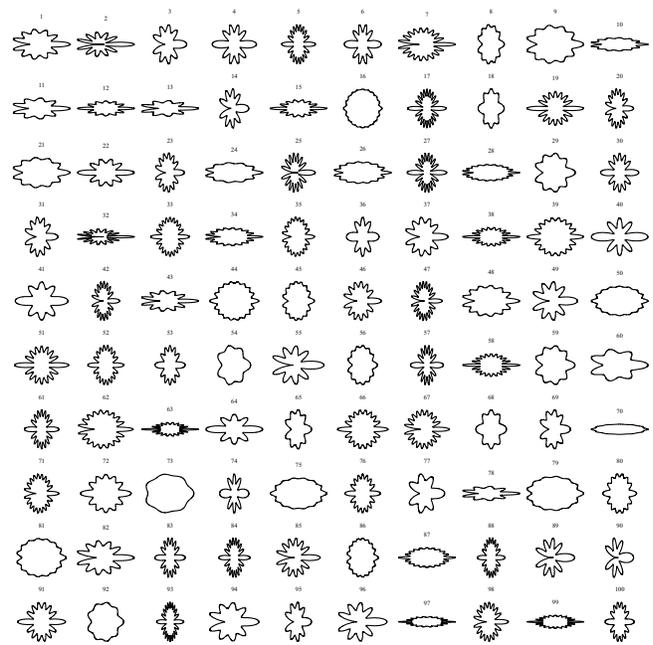


Figure 5: Database contains 100 prototype of CG-shapes contours. Each prototype represents one distinct type of CG-shapes.

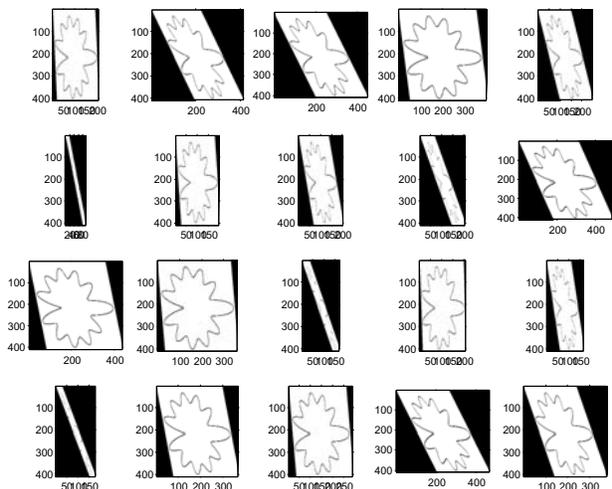


Figure 6: Contour deformations are generated by applying affine transformations on a the first CG-image prototype. Twenty deformations are produced for each prototype.

The affine transform parameters $\{a, b, c, d, e, f\}$ are ran-

domly generated for each deformation.

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \quad (48)$$

The affine transformed coordinates $[\bar{x}, \bar{y}]^T$ will be quantized into integer values. If different points on the original contour are mapped onto the same coordinate after the affine transformation and quantization, the duplicate points are eliminated. Linear interpolation to re-sample each transformed contour is applied so that all re-sampled contours have the same number of sample points. In this experiment, the number of sample points on each re-sampled contour is chosen to be $N_0 = 512$. The relationship between affine-transformed contour $\bar{x}[n]$ and re-sampled contour $\bar{x}'[n]$ described as [8]:

$$\bar{x}'[n] = \begin{cases} \bar{x}[0] & \text{if } n = 0; \\ \bar{x}[\lfloor \frac{Nn}{N_0} \rfloor] (\lceil \frac{Nn}{N_0} \rceil - \frac{Nn}{N_0}) + \bar{x}[\lceil \frac{Nn}{N_0} \rceil] \\ (\frac{Nn}{N_0} - \lfloor \frac{Nn}{N_0} \rfloor) & \text{if } n = 1, 2, \dots, (N_0 - 1). \end{cases} \quad (49)$$

where N is the total number of sample points on an affine-transformed contour $\bar{x}[n]$. To evaluate the semi-local summation invariant features, we choose $M = 51$ as the range of local summation. The resulting representation of the CG-shapes contour is a $N_0 \times 1$ feature vector. Each entry of this vector corresponds to $\lambda_{affine}[m]$, equation(47), calculated from a 51-point shape segment.

7.3 Simulation and Result

This experiment is conducted to evaluate the shape matching performance of the summation invariants. Details of this experiment is as follow:

- i.) A set of 100 images of CG-shapes are used as prototypes. For each prototype, 20 deformations are generated by applying affine transformations on the prototype. Hence, a data set containing 2000 shapes of the images is created.
- ii.) The dataset is divided into 5 equal-sized partitions. In each partition, there are 100 prototypes and 4 deformations associated with each prototype. In other words, each partition will contain 400 shapes that correspond to four variations of each of the 100 CG-shape prototypes.
- iii.) In one shape matching experiment, one partition will be used as gallery set and the others will be used as probe set. Since the gallery set and the probe set are mutually exclusive, one could evaluate how well a shape matching algorithm could be generalized to shapes it has never seen before. More specifically, a probe shape will be matched against

all of the gallery shapes. Then, we can pick up the pair of shapes with highest matching score. If these two shapes belong to the same prototype, we treat it as a correct matching result. If these two shapes do not belong to the same prototype, it is regarded as an incorrect match. In this experiments, the matching score, or, *similarity score*, between two shapes is given by [8]:

$$\rho = \frac{\sum_{n=0}^{N-1} v_{gallery}[n] \cdot v_{probe}[n]}{\sqrt{\sum_{n=0}^{N-1} v_{gallery}^2[n] \cdot \sum_{n=0}^{N-1} v_{probe}^2[n]}} \quad (50)$$

- iv.) In a M -way cross-validation method ($M = 5$ in this experiment), one can perform M experiments. In the first experiment, the first partition will be used as gallery set and the others will be used as probe sets. Similarly, in the second experiment, the second partition will be used as gallery set and the others will be used as probe sets. Therefore, given M partitions of the whole data set, one could conduct a total of M experiments. This way, each partition will be used as the gallery set exactly once in a M -way cross-validation.
- v.) For each experiment, the shape matching results are reported by a 100×100 confusion matrix C . c_{ij} denotes the $(i, j)^{th}$ element of C and is the number of probe shapes that actually belong to prototype i , but are considered as prototype j by a shape matching algorithm. The confusion matrices of all M shape matching experiments will be added together to form the final confusion matrix, from which one can perform different kinds of statistical analysis.
- a.) Circularly shift the feature vector from the probe set by one and compute the corresponding similarity score ρ with the one from the gallery set.
- b.) Circularly shift the feature vector from the probe set by one and compute the corresponding similarity score ρ with the one from the gallery set.
- c.) Repeat the previous step N times, where N is the number of points on a probe shape. Choose the maximum as the final similarity score between two shapes.

7.4 Comparison with Other Methods

It is important to know how the summation invariant performs compared with the integral invariant[5]. Like semi-local summation invariant, each CG-shape contour

Table 1: An error based comparison between the three methods of 2D shape matching.

Method	Error	prob. of error	Ratio(%)
Summ. invariant	190	190/8000	2.375%
Integral invariant	1195	1195/8000	14.9375%
Wavelet invariant	711	711/8000	8.8875%

is represented by a 512×1 feature vector. Each element of this feature vector is calculated by using the denominator of $\eta_{integral}$, equation (2). The integration, $V^{0,1}$, is approximated by numerical integration over a shape segment with 51 points. Its also important to know how it performs compared with the state-of-the-art methods. In[7], wavelet-based invariant functions were demonstrated to have superior discrimination power compared to some traditional methods, namely the moment invariant[11] and Fourier descriptor method[1]. We calculate $\eta_{3,4,5,6,7,8}[n], n = 1, \dots, 512$ using Daubechies wavelet of length 4. According to the authors suggestion in[7], only the coarsest scale levels (3, \dots , 8) are used to calculate the affine invariant function. The finest scale levels have been dropped because they are sensitive to noise. As a result, each CG-shape contour is represented by a 512×1 feature vector. In table(1), a comparison of the probability of incorrect matching(error), between the methods are shown with the error ratio in every case.

8 Conclusion

In this paper, the performance of the proposed 2D contour summation invariant features was compared with that of integral invariant features. A performance enhancement achieved by the new summation invariant features was marked. To conclude, a systematic approach for constructing robust geometrically invariant features was introduced. The proposed features provide improved accuracy and are applicable to a wide range of pattern recognition applications. These are versatile features that can be adapted to an engineers choice of transformation group, such as rigid, affine, or similarity transformation group, to mention a few. The invariant features discussed in this paper can be extended to 3D shape recognition application in the future work.

References

- [1] K. Arbter, W. Snyder, H. Burkhardt, and G. Hirzinger, *Application of affine-invariant fourier descriptors to recognition of 3-d objects*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 640–7, 1990.
- [2] D. H. Ballard, and C. M. Brown, , *Computer Vision*, Prentice Hall Professional Technical Reference, 1982.
- [3] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker, *Differential and numerically invariant signature curves applied to object recognition*, Intl. Journal of Computer Vision, vol. 26, no. 2, pp. 107135, 1998.
- [4] L. V. Gool, T. Moons, E. Pauwels, and A. Oosterlinck, *Geometric invariance in computer vision.*, Cambridge: MIT, 1992, ch. Semi-Differential Invariants, pp. 157192.
- [5] C. E. Hann and M. S. Hickman, *Projective curvature and integral invariants*, Acta Applicandae Mathematicae, vol. 74, no. 2, pp. 177193, 2002.
- [6] M. K. Hu, *Visual pattern recognition by moment invariants*, IRE Trans. on Information Theory, vol. IT-8, no. 2, pp. 179187, 1962.
- [7] M. Khalil and M. Bayoumi, *A dyadic wavelet affine invariant function for 2d shape recognition*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1152–1164, 2001.
- [8] W. Y. Lin, *Robust geometrically invariant features for 2d shape matching and 3d face recognition*, PhD dissertation, 2006.
- [9] S. Manay, B.-W. Hong, A. Yezzi, and S. Soatto, *Integral invariant signatures*, ECCV 2004. Proc. (Lecture Notes in Comput. Sci. Vol.3024), vol. 4, pp. 87–99, 2004.
- [10] E. J. Pauwels, T. Moons, L. J. V. Gool, P. Kempenaers, and A. Oosterlinck, *Recognition of planar shapes under affine distortion*, Intl. Journal of Computer Vision, vol. 14, no. 1, pp. 4965, 1995.
- [11] T. Reiss, *The revised fundamental theorem of moment invariants*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 830–4, 1991.
- [12] J. Sato and R. Cipolla, *Affine integral invariants and matching of curves*, in Proc. of 13th Intl. Conf. on Pattern Recognition, vol. 1, 1996, pp. 91519.
- [13] Olver P. J., *Equivalence, Invariants and Symmetry*, Cambridge University Press, 1995.
- [14] I. Weiss, *Geometric invariants and object recognition*, International Journal of Computer Vision, vol. 10, no. 3, pp. 20731, 06 1993.