

# Higher-Dimensional Affine Registration and Vision Applications

S.M. Shahed Nejhum, Yu-Tseh Chi, Jeffrey Ho, *Member, IEEE*, and  
Ming-Hsuan Yang, *Senior Member, IEEE*

**Abstract**—Affine registration has a long and venerable history in computer vision literature, and in particular, extensive work has been done for affine registration in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . This paper studies affine registration in  $\mathbb{R}^m$  with  $m$  typically ranging from 4 to 12. To justify breaking of this dimension barrier, the first part of the paper describes three novel matching problems that can be formulated and solved as affine point-set registration problems in dimensions greater than three: stereo correspondence under motion, image set matching, and covariant point-set matching, problems that are not only interesting in their own right but also have potential for important vision applications. Unfortunately, most of the existing affine registration algorithms do not generalize easily to higher dimensions due to their inefficiency. Therefore, the second part of this paper develops a novel algorithm for estimating the affine transform between two point sets in  $\mathbb{R}^m$ . Specifically, the algorithm follows the common approach of iteratively solving the correspondences and transform. The initial correspondences are determined using the novel notion of local spectral features, features constructed from local distance matrices. Unlike many correspondence-based methods, the proposed algorithm is capable of registering point sets of different size, and the use of local features provides some degree of robustness against noise and outliers. The proposed algorithm is validated on a variety of synthetic point sets in different dimensions with varying degrees of deformation and noise, and the paper also shows experimentally that several instances of the aforementioned three matching problems can indeed be solved satisfactorily using the proposed affine registration algorithm.

**Index Terms**—Affine registration, point matching, stereo correspondence.

## 1 INTRODUCTION

**M**ATCHING points, particularly in low-dimensional settings such as 2D and 3D, have been a classical problem in computer vision. The problem can be formulated in a variety of ways, depending on the allowable and desired deformations. For instance, the orthogonal and affine cases have been studied quite intensively in the literature, e.g., [23], [41], and recent research activities have been focused on nonrigid deformations, particularly those that can be locally modeled by a family of well-known analytic functions such as splines, e.g., [9]. In this paper,<sup>1</sup> we study the more classical problem of matching point sets related by affine transforms. The novel viewpoint taken here is the emphasis on affine registrations in  $\mathbb{R}^m$  for  $4 \leq m \leq 12$ , and this differs substantially from the past literature on this subject, which has been overwhelmingly devoted to registration problems in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , e.g., [28], [42], [43], [47], [50].

To justify breaking this dimension barrier, the first part of the paper will describe three broad and interesting classes of applications in computer vision that can be formulated and solved as affine registration problems in  $\mathbb{R}^m$  with  $m > 3$ : stereo correspondences, image set matching, and covariant point-set matching (see Fig. 1). In stereo correspondence under motion, two video cameras are observing an object undergoing some motion (rigid or nonrigid), and a set of  $k$  points on the object is tracked consistently in each view. The problem is to match the tracked feature points across two views so that we can identify and locate the  $k$  feature points correctly and consistently across all frames. For image set matching, two sets of images are given such that the images are related through a linear transform  $\mathcal{F} : \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}$  between two high-dimensional image spaces. Interesting examples of such transform  $\mathcal{F}$  include standard image operations such as image rotation, smoothing, downsizing, and pixel permutation, all of which can be modeled as linear transforms between image spaces. The task here is to compute the correspondences between images directly from the sets of images: The correspondences are determined in a global and systematic way by exploiting the geometric relations between images without using local image features (e.g., SIFT). Finally, in many vision applications, features associated to points are often vectorial quantities such that a spatial (affine) transform of the points also induces a linear transform of their feature vectors. Familiar examples include image gradients, structure tensors, high-order jets, and image moments. In particular, the presence of associated feature vectors means that the qualitative

1. An early version of this paper was presented in [8].

- S.M. Shahed Nejhum, Y.-T. Chi, and J. Ho are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32607. E-mail: {smshahed, ychi, jho}@cise.ufl.edu.
- M.-H. Yang is with the Department of Electrical Engineering and Computer Science, University of California at Merced, Merced, CA 95344. E-mail: mhyang@ucmerced.edu.

Manuscript received 25 Mar. 2010; revised 9 Aug. 2010; accepted 10 Sept. 2010; published online 30 Nov. 2010.

Recommended for acceptance by S. Sclaroff.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-03-0221.

Digital Object Identifier no. 10.1109/TPAMI.2010.219.

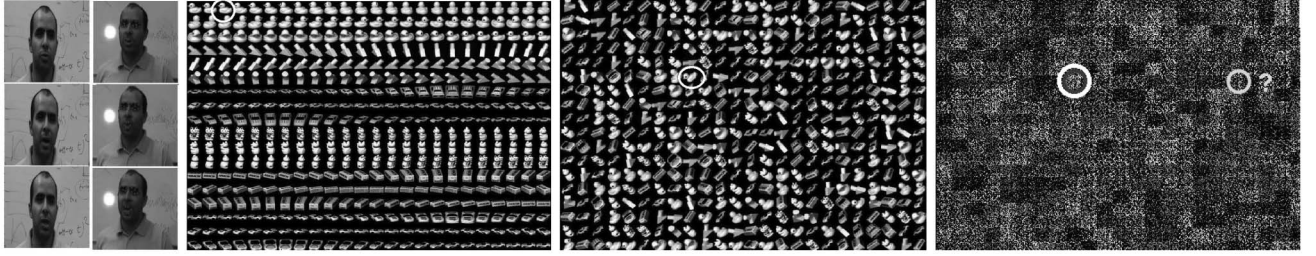


Fig. 1. Left: Stereo correspondence under motion. A talking head is observed by two (affine) cameras. Feature points are tracked separately on each camera and the problem is to compute the correspondences between observed feature points across views. Center left: A collection of 432 images from COIL image database showing six objects. Center right: Images obtained by rotating and downsizing the images on the left. Right: Images obtained by permuting pixels of the images on the left. All images use the same pixel permutation. Without computing local image features, image set matching aims to discover the correspondences (circles) from the sets directly by exploiting the geometry of the images considered as points in some  $\mathbb{R}^m$ . Local image features may be useful for solving the matching problem between the first two image sets; however, it is completely inappropriate for the third set.

measure of a registration result should depend on both the spatial and feature matchings, an important point that seems to have been overlooked in the literature. In covariant point-set registration, we develop a new and more general type of matching problems that incorporate both the spatial and feature components.

While the three problems superficially look disparate, we will show that a principled and unified approach to solve them is indeed possible provided that their underlying geometric contexts are understood. Specifically, we will show that the data transforms underlying the three matching (registration) problems can be modeled by affine transforms in some appropriate dimensions. Instead of computing the correspondences directly, we will focus on determining the affine transform from the input data points, with the correspondences recovered from the affine transform using nearest neighbors. In particular, an important idea of this paper is to reformulate the matching problem in a correct dimension that is often different from the one containing the input data. The choice of the new dimension, which can be higher or lower than the original feature space dimension, will depend on the problem. This dimensionality modification is supported by the following two paradigms that underlie the three matching problems:

1. Dimensionality Reduction: Matching problems in high-dimensional image space can be approximated by matching problems in its low-dimensional approximating subspace.
2. Feature Augmentation: Points in nonrigid motion in  $\mathbb{R}^3$  can be considered as projections of some higher-dimensional feature points. Therefore, the matching problem can be reformulated as a matching problem in the higher-dimensional feature space.

Geometrically, the situation is best illustrated by the following commutative diagram between two pairs of feature spaces  $\mathbb{R}^m$  and  $\mathbb{R}^k$ :

$$\begin{array}{ccc} \mathbb{R}^m & \xrightarrow{g} & \mathbb{R}^m \\ \pi_1 \downarrow & & \downarrow \pi_2 \\ \mathbb{R}^k & \xrightarrow{f} & \mathbb{R}^k, \end{array} \quad (1)$$

where  $m > k$ . The problem-dependent projection maps  $\pi_1, \pi_2$  relate each pair of feature spaces  $\mathbb{R}^m, \mathbb{R}^k$ , and their respective maps  $f, g$ . For each problem, the input point sets

reside in  $\mathbb{R}^m$  (or  $\mathbb{R}^k$ ), and the problem ostensibly requires the determination of  $g$  (or  $f$ ) that registers the point sets. However, the reformulated problem takes place in the other feature space  $\mathbb{R}^k$  ( $\mathbb{R}^m$ ), and the resulting matching problem then requires the determination of  $f$  (or  $g$ ). In particular, in this general context, the registration problem mostly occurs in dimensions greater than three, and in our applications, the dimension typically ranges from 4 to 12, dimensions that are higher than the usual dimension of three but cannot be considered as high-dimensional. Therefore, we term the type of affine registration problems studied in this paper as higher-dimensional affine registration.

Unfortunately, most of the existing affine registration algorithms are tailored for applications in  $\mathbb{R}^2$ , and they are not suitable for registering higher-dimensional point sets. Therefore, in the second part of the paper, we will develop a novel affine registration algorithm. The iterative closest point (ICP) algorithm [3], [57] can be easily generalized to handle higher-dimensional point sets. The resulting affine-ICP algorithm iteratively solves for the correspondences and affine transform. However, the main challenge is to produce good initial correspondences and affine transformation that can be used to start the affine-ICP algorithm. For dimensions two and three, this is already a major problem and the difficulty increases significantly with dimension. We propose a simple algorithm that can estimate the affine transformation (and hence, the correspondences) directly from the point sets. The algorithm first reduces the affine registration to an orthogonal case in which only orthogonal transforms are involved [45]. The orthogonal registration problem is then solved using local spectral features (LSFs), features formed by the eigenvalues of a symmetric matrix constructed from local distance matrix. We show that with noiseless data and generic point sets, the algorithm will recover the exact affine transform. With the noisy data, it will provide a good initialization for the affine-ICP algorithm. The algorithm is validated using a variety of synthetic point sets in different dimensions with varying degrees of deformation and noise, and we provide experimental results demonstrating that the algorithm can indeed provide efficient and accurate solutions to the three aforementioned matching problems.

## 2 AFFINE REGISTRATION AND VISION APPLICATIONS

This section presents the formulations of the three aforementioned matching problems as affine registration problems in appropriate dimensions. An important contribution of this paper is to show that these matching problems can indeed be solved in a systematic and principled way using the proposed affine registration algorithm described later in the paper.

### 2.1 Stereo Correspondences under Motion

A scene (or an object) undergoing some motion is observed by two stationary video cameras  $C_1, C_2$ . On each camera, a tracking algorithm is applied to consistently track  $k$  scene points over  $t$  frames. Let  $(x_{ij}^f, y_{ij}^f)$ ,  $i = 1, 2$ ,  $j = 1, \dots, k$ , denote the image coordinates of the  $j$ th scene point taken with camera  $i$  in frame  $f$ . The problem of stereo correspondences under motion is to compute correspondences across two cameras (views) between the two sets of image points  $(x_{1j}^f, y_{1j}^f), (x_{2j}^f, y_{2j}^f)$ , and the correspondence is defined such that  $(x_{1j}^f, y_{1j}^f) \leftrightarrow (x_{2j}^f, y_{2j}^f)$  if they are two image projections of the same scene point.

The problem context is quite general and many vision applications, particularly those related to surveillance, require computing correspondences from image sequences. One possible solution is to apply stereopsis and use local intensity patterns to estimate the correspondences from an image pair as is common in many stereo-based algorithm. However, there are three reasons for not favoring this type of solution. First, for a dynamic scene, motions of the scene points affect their appearance and external illumination can complicate matching using brightness constancy assumption. Second, since the stereo algorithm works only with image pairs, it requires the selection of the correct frames from the two cameras. If the two sequences are not temporally aligned or come with some synchronization information, such as time stamps, it is difficult to pick out the correct pair of frames. Furthermore, it is also difficult to determine which pair would be the optimal frame pair to use. Third, using only a pair of frames to compute the correspondences does not utilize all of the input data. In particular, a principled solution should consider all of the input data simultaneously and compute the correspondences without bias to any particular frame pair.

Assuming the cameras are affine [23], we show that it is possible to solve the problem in a principled and unified manner. The basic idea is straightforward: For each camera, we can apply the factorization algorithm (e.g., [48]) to reconstruct the observed 3D points, and the two reconstruction results differ by an unknown linear transform. For clarity of presentation, we will first discuss the rigid motion. Recall that for an affine camera model, the relation between a 3D point  $X_j$  and its 2D image point  $x_j$  is given by

$$x_j = CX_j + t,$$

where  $C$  is a general  $2 \times 3$  matrix with linearly independent rows and  $t$  is the translation. As it can be dealt with by centering the image points, we will ignore the translation  $t$

in the following discussion. For the observed  $k$  scene points, the position matrix  $X$  is

$$X = [X_1 \dots X_k], \quad (2)$$

with columns the 3D coordinates of the scene points. For each camera  $i = 1, 2$ , we can stack the image coordinates of one tracked scene point  $X_j$  over  $T$  frames vertically into a  $2T$ -dimensional vector:

$$p_j = (x_{1j}^1 \ y_{1j}^1 \dots x_{1j}^T \ y_{1j}^T)^\top \quad q_j = (x_{2j}^1 \ y_{2j}^1 \dots x_{2j}^T \ y_{2j}^T)^\top.$$

These vectors can be stacked horizontally to form the data matrices  $M_1, M_2$ :

$$M_1 = [p_1 \dots p_k], \quad M_2 = [q_1 \dots q_k].$$

Similarly, the camera matrices  $C_1^f, C_2^f$  over  $T$  frames can be put together to form<sup>2</sup>

$$C_1 = [C_1^1; \dots; C_1^T], \quad C_2 = [C_2^1; \dots; C_2^T].$$

Equation (2) then implies that

$$M_1 = C_1 X, \quad M_2 = C_2 X. \quad (3)$$

Therefore, the 3D points  $X_j$  can be computed by factorizing the data matrices  $M_1, M_2$  [48]. Evidently the (least-squares) solution for  $X$  is unique only up to a nonsingular linear transform. Therefore, when comparing the reconstruction results  $X_1, X_2$  from the two cameras, there is a nonsingular matrix  $A$  such that

$$X_2 = A X_1.$$

That is, the correspondences can be solved as an affine registration problem between the point sets  $X_1, X_2$ .

The above discussion generalizes immediately to nonrigid motions that can be modeled (or approximated) using linear shape basis [23], [5], [4] since the underlying linear algebra for both rigid and nonrigid motions is the same, except nonrigid motions require points in  $\mathbb{R}^m$  with  $m > 3$ . Choosing a world frame since the target is undergoing a nonrigid motion, the 3D coordinates of the scene points are no longer constant in time and, accordingly, the 3D position matrix

$$X^\top = [X_1^\top \dots X_k^\top]$$

now depends on frame (time)  $t$ . For  $k$  feature points, a shape basis element  $B_l$  is a  $3 \times k$  matrix, and for a motion model that employs  $m$  linear shape basis elements, the position matrix  $X^\top$  at frame  $t$  is assumed to be a linear combination of these shape basis elements:

$$X^\top = [X_1^\top \dots X_k^\top] = \sum_{l=1}^m a_l^\top B_l, \quad (4)$$

with coefficients  $a_l^\top$ . The crucial result in the above discussion is (3) that relates the frame-dependent data and camera matrices to the frame-independent 3D position matrix. For the nonrigid case, an analogous equation can be obtained by cleverly arranging  $B_l, a_l^\top$ , and the camera matrices  $C_i^\top$  [5]. The idea is to move the time-dependent coefficients  $a_l^\top$  in (4)

2. Using MATLAB syntax.



Fig. 2. Image set matching. Left: Images of four objects from the COIL database. The images have been downsized from the original to  $32 \times 32$ . Center: Images obtained by rotating and smoothing the images on the left. The rotation is 45 degrees and the smoothing kernel is the uniform  $3 \times 3$  kernel. Right: Images obtained by randomly permuting the pixels in the images on the left. The four images are subject to the same pixel permutation scheme.

to the time-dependent camera matrix  $C_i^\top$ . Specifically, we define the new  $3m$ -D position matrix  $\mathbf{X}$  as

$$\mathbf{X} = [\mathbf{B}_1; \dots; \mathbf{B}_m],$$

which is a  $3m \times k$  matrix. The camera matrix is then augmented to have  $3m$  columns,

$$\mathbf{C}_i = [a_1^\top C_i^\top \dots a_m^\top C_i^\top; \dots; a_1^\top C_i^\top \dots a_m^\top C_i^\top],$$

to incorporate all of the frame-dependent coefficients  $a_l^\top$ . Using the new camera and position matrices and (4), we again arrive at (3) with the same data matrices  $\mathbf{M}_1, \mathbf{M}_2$ . In particular, this shows that once the position matrix  $\mathbf{X}$  is recovered from each camera, the two results differ by a nonsingular linear transformation in  $\mathbb{R}^{3m}$ , and the correspondences can again be solved as an affine registration problem, albeit in dimension higher than three.

## 2.2 Image Set Matching

In image set matching, two sets of images  $\mathbb{P} = \{I_1, \dots, I_k\} \subset \mathbb{R}^m$ ,  $\mathbb{Q} = \{I'_1, \dots, I'_k\} \subset \mathbb{R}^{m'}$  are given, where  $\mathbb{R}^m, \mathbb{R}^{m'}$  are the image spaces. The relation between the images is given by a linear transformation  $\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$  between the two high-dimensional image spaces such that

$$I'_k = \mathcal{F}(I_k).$$

Interesting examples of the linear transformation  $\mathcal{F}$  are easy to come by. Standard image processing operations, such as image rotation, downsampling, image smoothing, and pixel permutation, all can be modeled as (or approximated by) a linear transform  $\mathcal{F}$  between two image spaces. We have already seen several such image sets in Fig. 1, and Fig. 2 displays four close-up examples.

Given the image sets  $\mathbb{P}, \mathbb{Q}$ , the problem is to recover the correspondences  $I_k \leftrightarrow I'_k$ . Local image features such as SIFT could be useful for discovering the correspondences from the rotated or smoothed images. However, computing local image features can be expensive, and for a large number of images, this approach may not be efficient. Furthermore, for images obtained by randomly permuting the pixels, this method is completely ineffective as there is no meaningful feature that can be extracted from these images. On the other hand, the problem cannot be solved by estimating the linear transform  $\mathcal{F}$  from the images in  $\mathbb{P}, \mathbb{Q}$  as the dimension of the image space  $m$  is typically greater than the number of images. Therefore, without regularization, the problem is ill-posed.

Fortunately, many interesting image sets can be approximated well by low-dimensional linear subspaces in the image space, and these low-dimensional subspaces can be readily computed from the images using Principal Component Analysis (PCA). Let  $L_{\mathbb{P}}, L_{\mathbb{Q}}$  denote two such low-dimensional linear subspaces approximating  $\mathbb{P}, \mathbb{Q}$ , respectively, and we will use the same notations  $\mathbb{P}, \mathbb{Q}$  to denote their projections onto the subspace  $L_{\mathbb{P}}, L_{\mathbb{Q}}$ . A natural question to ask is how are the (projected) point sets  $\mathbb{P}, \mathbb{Q}$  related? Suppose  $\mathcal{F}$  is orthogonal and  $L_{\mathbb{P}}, L_{\mathbb{Q}}$  are the principle subspaces of the same dimension. If the data is “noiseless”, i.e.,  $I'_k = \mathcal{F}(I_k)$ , it is easy to show that  $\mathbb{P}, \mathbb{Q}$  are then related by an orthogonal transformation. In general,  $\mathcal{F}$  may not be orthogonal and data points are noisy. The point sets  $\mathbb{P}, \mathbb{Q}$  are related by a transformation of the form  $\mathbf{T} = \mathbf{A} + \mathbf{r}$ , which is a sum of an affine transformation  $\mathbf{A}$  and a nonrigid deformation  $\mathbf{r}$ . If the nonrigid part is small, we can recover the correspondences by affine registering the two (projected) point sets  $\mathbb{P}, \mathbb{Q}$ . For the image operations listed above, pixel permutation can be modeled by an orthogonal transform (permutation matrix) in the image space. Due to discretization which requires intensity interpolation, image rotation and downsampling cannot be modeled exactly using orthogonal transforms; nevertheless, they can be approximated well by similarity transforms (orthogonal plus uniform scaling). Smoothing, in general, can only be modeled by a general linear transform in the image space.

Therefore, while estimating the correspondences from the original image spaces is intractable, registering points on the subspaces  $L_{\mathbb{P}}, L_{\mathbb{Q}}$ , which have smaller dimensions, are tractable. This gives an algorithm for computing the correspondences without explicitly using the image contents, i.e., there is no feature extraction. Instead, it computes the affine transform directly from the image sets considered as point sets in the subspaces  $L_{\mathbb{P}}$  and  $L_{\mathbb{Q}}$ .

## 2.3 Covariant Point-Set Matching

In many vision applications, points extracted from images (or other data) often can be associated with feature vectors that also transform linearly under the application of a planar affine transform. For example, the  $2 \times 2$  structure tensor  $\mathcal{T}_p$  of an image  $\mathbf{I}$  at an image point  $p$ ,

$$\mathcal{T}_p = \begin{pmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \mathbf{I}_y \\ \mathbf{I}_y \mathbf{I}_x & \mathbf{I}_y^2 \end{pmatrix}, \quad (5)$$

transforms as a rank-two symmetric tensor under an affine transform  $\mathbf{A}$  of the image

$$\mathcal{T}_p \longrightarrow \mathbf{A}^\top \mathcal{T}_p \mathbf{A}.$$

In other words, the affine transform  $\mathbf{A}$  in the spatial domain  $\mathbb{R}^2$  induces a linear transform  $\mathbf{A}^*$  of the structure tensors. Therefore, if structure tensors are used in image registration, the registration algorithm not only has to match the points on the images but also their associated feature vectors. In particular, the 2D registration problem becomes a 5D registration problem as the points now acquire the dimension of their feature vectors,  $\hat{p} = (p, \mathcal{T}_p)$  in  $\mathbb{R}^{2+3=5}$  (the space of  $2 \times 2$  symmetric matrices has dimension three).

The above example is a special case of a broad class of point-set registration problems that we call Covariant Point-Set Matching: Each point  $p$  in the spatial domain  $\mathbb{R}^k$  has an associated feature vector  $\mathbf{v}_p$  in the feature space  $\mathbb{R}^d$  that also transforms linearly under an affine transform  $\mathbf{A} \in \mathbf{GL}(k)$  in  $\mathbb{R}^k$ . That is, the feature vectors are not invariant under the action of  $\mathbf{A}$  but are covariant with respect to it. This means that there is a representation of  $\mathbf{GL}(k)$  via a homomorphism  $\rho$ :

$$\rho : \mathbf{GL}(k) \longrightarrow \mathbf{GL}(d)$$

such that the induced action of an affine transform  $\mathbf{A} \in \mathbf{GL}(k)$  in  $\mathbb{R}^d$  is given by  $\mathbf{A}^* = \rho(\mathbf{A})$ . If we consider each pair  $(p, \mathbf{v}_p)$  as a point  $\hat{p} \in \mathbb{R}^k \oplus \mathbb{R}^d$ , then the transforms  $\mathbf{A}, \mathbf{A}^*$  can be combined to yield an affine transform  $\mathbf{A} \oplus \mathbf{A}^*$  in  $\mathbb{R}^{k+d}$  with matrix representation given as

$$\mathbf{A} \oplus \mathbf{A}^* = \begin{pmatrix} \mathbf{A} & 0 \\ 0 & \rho(\mathbf{A}) \end{pmatrix}. \quad (6)$$

Since the spatial and feature components use different units, we use the weighted  $\mathbf{L}^2$ -norm ( $\lambda > 0$ )

$$\|\hat{p} - \hat{q}\|_\lambda^2 = \|p - q\|_{\mathbb{R}^k}^2 + \lambda \|\mathbf{v}_p - \mathbf{v}_q\|_{\mathbb{R}^d}^2 \quad (7)$$

to define the distance between two points  $\hat{p} = (p, \mathbf{v}_p)$ ,  $\hat{q} = (q, \mathbf{v}_q)$  in  $\mathbb{R}^{k+d}$ . This distance measure will be used to define the matching cost function  $\mathbf{E}(\mathbf{A})$  for registering point sets in  $\mathbb{R}^{k+d}$ . We remark that the usual point-set matching problem can naturally be considered as a special case of the covariant point-set matching if each point is assigned a trivial feature vector  $\mathbf{0} \in \mathbb{R}^d$ , and the map  $\rho$  is the identity map,  $\rho(\mathbf{A}) = \mathbf{I}_{d \times d}$ . The linear transform  $\mathbf{A}$  has no effect on the feature vectors. In this case, the distance  $\|\hat{p} - \hat{q}\|_\lambda$  reduces to the usual  $\mathbf{L}^2$ -distance of the points  $p, q$  in  $\mathbb{R}^k$ .

The affine registration problem formulated above in  $\mathbb{R}^{k+d}$  is for the transforms in  $\mathbf{GL}(k+d)$  with matrix representations given in (6). However, we can relax the problem and consider the affine registration using all transforms in  $\mathbf{GL}(k+d)$ . The relaxation can be justified because if the registration using transforms in (6) has a good solution  $\mathbf{A}$  (i.e., one with small registration error), it should be close to the solution  $\tilde{\mathbf{A}}$  for the  $\mathbf{GL}(k+d)$ -registration problem, and the solution  $\tilde{\mathbf{A}}$  can be used in two different ways to solve the original registration problem in  $\mathbb{R}^k$ . First, we can transfer the correspondences  $\hat{p}_i \leftrightarrow \hat{q}_j$  determined by  $\tilde{\mathbf{A}}$  in  $\mathbb{R}^{k+d}$  to the correspondences  $p_i \leftrightarrow q_j$  in  $\mathbb{R}^k$ . Second, we can define a new transform  $\mathbf{T}$  using  $\tilde{\mathbf{A}}$ :

$$\mathbf{T}(p) = \mathbf{P}_k \tilde{\mathbf{A}}(\hat{p}), \quad (8)$$

where  $\mathbf{P}_k : \mathbb{R}^{k+d} \rightarrow \mathbb{R}^k$  is the orthogonal projection. The transform  $\mathbf{T}$  so defined is, in general, not just an affine transform  $\mathbf{A}$  in  $\mathbb{R}^k$ , but coupled with a nonrigid deformation vector  $\mathbf{r}_p$ ,

$$\mathbf{T}(p) = \mathbf{A}(p) + \mathbf{r}_p,$$

that depends on the feature vector  $\mathbf{v}_p$ . This is clear because the matrix representation for  $\tilde{\mathbf{A}}$  has the block form

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{k \times k} & \mathbf{B}_{k \times d} \\ \mathbf{C}_{d \times k} & \mathbf{D}_{d \times d} \end{pmatrix}, \quad (9)$$

and the deformation vector  $\mathbf{r}_p = \mathbf{B}\mathbf{v}_p$ . Because of this added nonrigidity, the matching provided by  $\mathbf{T}$  typically produces smaller registration error than using only affine transform  $\mathbf{A}$ .

### 2.3.1 Registration of Massive Point Sets

One interesting application of the covariant matching is the registration of point sets containing large numbers of points (say in the tens or hundreds of thousands). The problem can be difficult if the registration algorithm does not scale well with respect to the size of the input. A typical remedy for this is to downsize the point sets, and then register the resulting smaller sets of points that are supposed to have captured a good portion of the original geometries. The simplification procedure can be both subtle and elaborate, and there is a large body of work in computer graphics devoted to mesh and point-set simplifications (e.g., [17], [34]). The downsized points are usually determined as the weighted means of their associated points in the original point sets, and in our viewpoint, working solely with these mean points may have discarded some useful geometric information that could be useful for matching. However, if we can associate certain feature vector to each downsized point that captures the local geometry of the point set, covariant point-set matching will readily provide the solution. In essence, covariant matching allows us to trade complexity for dimension as it transforms a larger problem in low dimension into a smaller one in a higher dimension. The trade is beneficial because the smaller problem can be solved more efficiently and accurately as the reduction in problem complexity usually more than compensates for the moderate increase in dimension.

Discrete polynomial moments provide a natural family of such local features. Recall that the moments are the expected values of monomials with respect to a probability density function. Working in  $\mathbb{R}^3$ , let  $P(x, y, z)$  denote a monomial in three variables. Using uniform discrete density, the discrete polynomial  $P$ -moment [27] for a collection of points  $p_1, \dots, p_s$  with respect to the origin is the sum

$$\mathbf{m}_P = \frac{1}{s} \sum_{i=1}^s P(x_i, y_i, z_i),$$

where  $p_i = (x_i, y_i, z_i)$ . For a given degree  $n$ , let  $P_1, \dots, P_d(n)$  denote the  $d(n)$  basis monomials with  $d(n) = \frac{(n+2)(n+1)}{2}$ . The moment feature vector for  $p_1, \dots, p_s$  is defined as

$$\mathbf{v}_n = [\mathbf{m}_{P_1}, \dots, \mathbf{m}_{P_{d(n)}}]^\top.$$

For example, for  $n = 2$ , the quadratic moment vector is given as

$$\mathbf{v}_2 = \frac{1}{s} \left[ \sum x_i^2, \sum y_i^2, \sum z_i^2, \sum x_i y_i, \sum x_i z_i, \sum y_i z_i \right]^\top, \quad (10)$$

and with  $n = 1$ , the linear moment vector simply gives the mean point

$$\mathbf{v}_1 = \frac{1}{s} \left[ \sum x_i, \sum y_i, \sum z_i \right]^\top.$$

In all of the applications shown in the experiment section, we will only work with quadratic moment ( $n = 2$ ) and no other higher degree moments will be used.

### 3 PREVIOUS WORK

Affine registration for point sets in  $\mathbb{R}^2$  has been investigated quite extensively in the literature, with important applications in image registration, object recognition, and shape analysis. There is a substantial body of literature devoted to this subject, and the overwhelming majority of them deal with registration problems in  $\mathbb{R}^2$ . Therefore, in this necessarily incomplete and brief survey of existing literature, we will focus on a few major approaches and discuss their possible extensions to handle higher-dimensional registration problems.

Correspondence-based algorithms solve the registration problem by computing the correspondences. In particular, the invariant-based approach uses moment invariants that are invariant under affine transforms [27]. For example, affine moment invariants for images have been proposed in [35], [16], and in [55], cross-weighted moment invariants are used for alignment and recognition. Seusse et al. [47], [50] introduce the idea of discrete affine moments, and the normalization technique used in these papers, although in a different context, is the same as our orthogonal reduction step. Recently, [25] proposed a set of affine moment descriptors that use complex moments in the definition, and the resulting descriptors can be converted to affine moment invariants. As moments are the expected values of monomials and the number of monomials of a fixed degree grows exponentially with dimension, the main difficulty of generalizing moment-based approach to higher dimension is the complexity of the algebra involved. In particular, each dimension requires a different formula to compute invariants, and the large number of moments makes the process extremely inefficient.

Inefficiency is also a major problem for generalizing other methods to higher dimensions. For example, [39] contains perhaps the most straightforward method that computes the correspondence between the two point sets using all possible sets of four points. This method is similar to an earlier method [32] that uses affine invariant representation of points based on a triplet of basis points. All possible triplets are considered and the affine invariants of the remaining points are computed and stored in a hash table, which is then used in the matching stage. In [51], a global correlation for affine transforms is used to align affine transformed gray-level images. For similarity transforms, [46] introduces an algorithm that determines both the transformation and correspondences between two images. In this method, transformation parameters for all possible point pairings are calculated and then clustered in a parameter space, and the transformation is determined by the largest cluster. This approach is related to the concept of geometric hashing used in [53]. Unfortunately, these methods do not generalize well to higher dimensions because the size of their data structures (e.g., hash table) grow exponentially with dimension.

Hausdorff distances for image and point-set registrations were first introduced for similarity transforms in [29], and it

was later extended by [36] to the more general affine transforms for locating objects in a scene. In this method, a grid is introduced on the parameter space of allowable affine transforms, and the algorithm's complexity depends on the size of the grid (required precision). In [22], a hierarchical top-down approach is used to estimate the best affine transform among all possible transforms. As the dimension of  $\mathbf{GL}(n)$  depends quadratically on  $n$ , direct generalizations of these methods to higher dimensions would require too much memory to be practical. In another direction, the edges of the point set's convex hull can serve as important features for point matching, as was first noted in [20] for similarity transforms, and the vertices of the convex hull were later used for affine point-set matching in [56], [19]. In [28], affine invariant representations of point sets are obtained by using distance ratios defined by quadruples of feature points, and the convex hull of each point-set is utilized to select some reference points. Again, these techniques are difficult to generalize because convex hull in higher dimensions cannot be determined or represented as easily and efficiently as in  $\mathbb{R}^2$ .

Finally, spectral methods [7], [42], [43] form another important category of registration algorithms. The well-known algorithm of [42] solves the registration problem by aligning the (unit) eigenvectors of the covariance matrices of the two point sets. It has been extended and generalized in several different ways for matching points and shapes in  $\mathbb{R}^2$  [7], [11], [40]. These methods are easier to generalize as their complexity typically does not depend on the dimension but on the number of input points. Our approach can be considered as an extension of these methods, with the novel features of using only local spectral information to establish correspondences. Other types of local features used in matching include shape context [2], spin images [30], and shape descriptors [18]. These features are usually in the form of a histogram encoding the local geometry of a point using orientations, heights, and other geometric quantities. While these features work well in 2D and 3D, it is not immediately clear how they can be extended to higher dimensions. For example, computing the histograms in shape context is reasonably straightforward in  $\mathbb{R}^2$  because directions in  $\mathbb{R}^2$  can be parameterized by a circle and the bins in the histogram can be defined relatively easily. Moving into higher dimensions such as  $\mathbb{R}^6$  or  $\mathbb{R}^{10}$ , the directions in these spaces are parameterized by higher-dimensional spheres, and the definition of the histogram is no longer as straightforward as in  $\mathbb{R}^2$ . Very recently, [52] proposed an elegant algorithm based entirely on linear algebra for computing correspondences that can be applied to any dimension. Unfortunately, the algorithm requires an equal number of points for the input point sets, and at this moment, it is not clear how the presence of noise would affect the algorithm's performance.

### 4 HIGHER-DIMENSIONAL AFFINE REGISTRATION

The above discussion provides the motivation and justification for studying affine registration in  $\mathbb{R}^m$  for  $m > 3$ . An affine transform  $\mathbf{A}$  in  $\mathbb{R}^m$  is represented by a pair  $(\mathbf{A}, \mathbf{t})$ , its linear and translational components such that for  $\mathbf{x} \in \mathbb{R}^m$ ,

$$\mathbf{A}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{t},$$

where  $\mathbf{A} \in \mathbf{GL}(m)$  and  $\mathbf{t} \in \mathbb{R}^m$ . In the following discussion, we will often use the same notation (as above)  $\mathbf{A}$  to denote an affine transform and its linear component as the translational component is usually not the main focus. Let  $\mathbb{P} = \{p_1, \dots, p_k\}$  and  $\mathbb{Q} = \{q_1, \dots, q_l\}$  denote two point sets in  $\mathbb{R}^m$ . We formulate the affine registration problem for  $\mathbb{P}, \mathbb{Q}$  as an optimization problem using the matching cost function  $\mathbf{E}$ ,

$$\mathbf{E}(\mathbf{A}) = \Gamma_{\mathcal{H}}(\mathbf{A}(\mathbb{P}), \mathbb{Q}), \quad (11)$$

that uses Hausdorff distance to measure the quality of the registration given by  $\mathbf{A}$ . Recall that the (averaged) Hausdorff distance [29] between two point sets  $\mathbb{P}, \mathbb{Q}$  is defined as

$$\Gamma_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) = \frac{1}{k} \sum_{p \in \mathbb{P}} \min_{q \in \mathbb{Q}} |p - q| + \frac{1}{l} \sum_{q \in \mathbb{Q}} \min_{p \in \mathbb{P}} |q - p|, \quad (12)$$

and in (11), we have used  $\mathbf{A}(\mathbb{P})$  to denote the transformed point set of  $\mathbb{P}$ . Unfortunately, direct minimization of  $\mathbf{E}$  is difficult, and our strategy is to use an affine variant of the ICP algorithm to obtain an affine transformation with small  $\mathbf{E}(\mathbf{A})$ . While the global minimum of  $\mathbf{E}$  cannot be guaranteed in general, the value  $\mathbf{E}(\mathbf{A})$  serves as a good qualitative measure on the matching provided by the solution.

#### 4.1 Affine ICP

The ICP algorithm is a very general point registration algorithm that has become popular in vision applications, e.g., [1], [12], [15], [21], [37], [44], [57]. While these works are concerned exclusively with rigid transformations in  $\mathbb{R}^2, \mathbb{R}^3$ , it is not difficult to extend the ICP algorithm to incorporate affine transformations. The affine-ICP algorithm iteratively solves for the correspondences and affine transformation as in the usual ICP, and the two steps are easily adopted for higher-dimensional point sets. Specifically, given an assignment (correspondences),  $\pi: \{1, \dots, k\} \rightarrow \{1, \dots, l\}$ , the optimal affine transformation  $\mathbf{A}$  in the least-squares sense can be solved by minimizing

$$\mathcal{E}(\mathbf{A}, \mathbf{t}, \pi) = \sum_{i=1}^k |\mathbf{A}p_i + \mathbf{t} - q_{\pi(i)}|^2. \quad (13)$$

Solving  $\mathbf{A}, \mathbf{t}$  separately, this cost function gives a quadratic programming problem in the entries of  $\mathbf{A}$ , and the optimal solution can be computed readily by solving a linear system. With a fixed  $\mathbf{A}$ ,  $\mathbf{t}$  can be solved immediately. On the hand, given an affine transformation  $\mathbf{A}$ , a new assignment  $\pi$  can be defined using nearest neighbors

$$\pi(i) = \arg \min_{1 \leq j \leq l} |\mathbf{A}p_i + \mathbf{t} - q_j|^2.$$

The iterative process requires an initialization that is typically provided as an initial affine transform  $\mathbf{A}$ . While the affine ICP is easy to implement and very efficient, it suffers from two major shortcomings. First, to the best of our knowledge, there is no proof of convergence for any ICP-like algorithm, although ample empirical results have suggested that if a good initialization is given, the algorithm usually does converge to a reasonable solution, e.g., [15], [57]. Second, the algorithm's performance depends significantly on the initial affine transform, and with a poor choice, it almost always converges to an undesirable solution. Therefore,

determining a good initialization is crucial for the algorithm's success. A direct way to generate an initial affine transformation (disregarding  $\mathbf{t}$ ) is to randomly pick  $m$  pairs of points from  $\mathbb{P}, \mathbb{Q}$ ,  $\{(x_1, y_1), \dots, (x_m, y_m)\}$ ,  $x_i \in \mathbb{P}$ ,  $y_i \in \mathbb{Q}$ , and define  $\mathbf{A}$  as  $y_i = \mathbf{A}(x_i)$ . It is easy to see that the probability of picking a good set of  $m$  pairs that will yield good initialization is roughly in the order of  $1/C(k, m)$ . For small dimensions  $m = 2, 3$  and medium-size point sets ( $k$  in the order of hundreds), it is possible to exhaustively sample all collections of such  $m$  pairs and evaluate their corresponding affine transforms. However, as  $C(k, m)$  depends exponentially on the dimension  $m$ , this approach becomes impractical once  $m > 3$ .

Our approach to generate a good initial affine transform is to first reduce the problem to an orthogonal matching problem. The latter problem is solved by computing correspondences using local spectral features, and the resulting orthogonal transform is used to obtain an affine transform. For generic point sets without noise, it is straightforward to show that the affine transform so determined is the exact affine transform, and this suggests that (with experimental support) for noisy point sets, it should be a good candidate for the affine ICP initialization.

#### 4.2 Orthogonal Reduction

Given the point sets  $\mathbb{P}, \mathbb{Q}$ , their respective normalized discrete linear moments  $\mathbf{m}_p, \mathbf{m}_q$  are given as [50], [27]

$$\mathbf{m}_p = \frac{1}{k} \sum_{i=1}^k p_i, \quad \mathbf{m}_q = \frac{1}{l} \sum_{i=1}^l q_i.$$

Similarly, their normalized quadratic moments (covariance matrices) are given as

$$\mathbf{S}_p = \frac{1}{k} \sum_{i=1}^k (p_i - \mathbf{m}_p)(p_i - \mathbf{m}_p)^\top, \quad (14)$$

$$\mathbf{S}_q = \frac{1}{l} \sum_{i=1}^l (q_i - \mathbf{m}_q)(q_i - \mathbf{m}_q)^\top. \quad (15)$$

For the moment, we will work with noiseless assumption:  $\mathbb{P}, \mathbb{Q}$  have the same number of points and the affine transform  $\mathbf{A}$  is exact, i.e.,<sup>3</sup>

$$q_i = \mathbf{A} p_i + \mathbf{t}, \quad (16)$$

for  $1 \leq i \leq k$ . Furthermore, we will assume that the point sets  $\mathbb{P}, \mathbb{Q}$  are nondegenerated so that they are not properly contained in a linear subspace of  $\mathbb{R}^m$ . It then follows that  $\mathbf{S}_p, \mathbf{S}_q$  have full rank and their inverses exist. Equation (16) implies that  $\mathbf{A}$  has to satisfy the following two equations:

$$\mathbf{S}_q = \mathbf{A} \mathbf{S}_p \mathbf{A}^\top, \quad \mathbf{t} = \mathbf{m}_q - \mathbf{A} \mathbf{m}_p.$$

Therefore,  $\mathbf{t}$  can be determined once the linear component  $\mathbf{A}$  is known, and the first constraint on  $\mathbf{A}$  implies that

$$\mathbf{A} = \mathbf{S}_q^{-\frac{1}{2}} \mathbf{R} \mathbf{S}_p^{-\frac{1}{2}} \quad (17)$$

3. We have used  $\mathbf{A}$  to denote both the affine transform and its linear component.

for some orthogonal transform  $\mathbf{R} \in \mathbf{O}(m)$ . This essentially reduces the problem of determining  $\mathbf{A}$  from  $\mathbf{GL}(m)$  to  $\mathbf{O}(m)$  because  $\mathbf{R}$  has to match the transformed point sets exactly, with the transformations given by

$$p_i \rightarrow \mathbf{S}_{\mathbb{P}}^{-\frac{1}{2}}(p_i - \mathbf{m}_{\mathbb{P}}), \quad q_i \rightarrow \mathbf{S}_{\mathbb{Q}}^{-\frac{1}{2}}(q_i - \mathbf{m}_{\mathbb{Q}}). \quad (18)$$

In other words, after centering each point set ( $p_i \rightarrow p_i - \mathbf{m}_{\mathbb{P}}, q_i \rightarrow q_i - \mathbf{m}_{\mathbb{Q}}$ ), we have the following commutative diagram:

$$\begin{array}{ccc} \mathbb{R}^m & \xrightarrow{\mathbf{A}} & \mathbb{R}^m \\ \mathbf{S}_{\mathbb{P}}^{-\frac{1}{2}} \downarrow & & \downarrow \mathbf{S}_{\mathbb{Q}}^{-\frac{1}{2}} \\ \mathbb{R}^m & \xrightarrow{\mathbf{R}} & \mathbb{R}^m, \end{array}$$

and the original affine registration in the upper part of the diagram can be transferred to the orthogonal registration in the lower part of the diagram using  $\mathbf{S}_{\mathbb{P}}^{-\frac{1}{2}}, \mathbf{S}_{\mathbb{Q}}^{-\frac{1}{2}}$ . Commutativity of the diagram implies that  $\mathbf{A}$  can be reconstituted as in (17) once  $\mathbf{R}$  is determined. We remark that the same reduction, although presented in a different way but with essentially the same mathematical content, appeared in [45].

### 4.3 Determine $\mathbf{R}$

We will use the same notation  $\mathbb{P}, \mathbb{Q}$  to denote the transformed point sets of  $\mathbb{P}, \mathbb{Q}$  using (18). Since the transformed point sets  $\mathbb{P}, \mathbb{Q}$  have unit covariance matrices, the invariant approach in [41], which can be easily generalized to higher dimensions, cannot be applied to determine the orthogonal transform  $\mathbf{R}$ . Nevertheless, there is a very general way to produce a large number of useful local invariants that can be used to estimate the correspondences. Specifically, we aim to define invariant feature vectors  $\mathbf{u}_i, \mathbf{v}_i$  for  $p_i, q_i$ , respectively, such that  $q_i = \mathbf{R}p_i$  implies  $|\mathbf{u}_i - \mathbf{v}_i| = 0$ . These feature vectors will provide a way to determine possible correspondences between points in  $\mathbb{P}$  and  $\mathbb{Q}$ , and if sufficiently many correspondences ( $\geq m$ ) are known,  $\mathbf{R}$  can be recovered immediately.

The feature vectors will be defined using the spectral information of local neighborhoods: for each point  $x$ , we will use its  $K$  nearest neighbors to construct a symmetric matrix  $\mathcal{L}_x$ , and its eigenvalues give the associated feature vector  $\mathbf{u}$ . More specifically, let  $x_1, \dots, x_K$  denote the  $K$  nearest neighbors of  $x$  and  $\mathbf{D}_x$  the  $(K+1) \times (K+1)$  distance matrix for these  $K+1$  points:

$$\mathbf{D}_p = \begin{pmatrix} |p-p| & \cdots & |p-p_k| \\ \vdots & \cdots & \vdots \\ |p_k-p| & \cdots & |p_k-p_k| \end{pmatrix}.$$

If  $K > m+2$ , it is known that generically the symmetric distance matrix  $\mathbf{D}_x$  has rank  $m+2$ . Therefore, it has at most  $m+2$  nonzero eigenvalues. To get more eigenvalues, we can define the following  $(K+1) \times (K+1)$  symmetric matrix:

$$\mathcal{L}_x = \mathbf{I}_k - \mu \begin{pmatrix} f(|p-p|) & \cdots & f(|p-p_k|) \\ \vdots & \cdots & \vdots \\ f(|p_k-p|) & \cdots & f(|p_k-p_k|) \end{pmatrix}, \quad (19)$$

where  $\mu$  is a constant,  $\mathbf{I}_k$  is the identity matrix, and the function  $f(t) = \exp(-t^2/\sigma^2)$  is the Gaussian exponential. For most choices of  $\mu$ ,  $\mathcal{L}_x$  has full rank, and it is related to the well-known (unnormalized) discrete Laplacian associated with the point set  $\{x, x_1, \dots, x_K\}$  [10]. As  $\mathcal{L}_x$  is symmetric, its eigenvalues are all real, and we define the feature vector  $\mathbf{u}$  associated to  $x$  as the vector of eigenvalues of  $\mathcal{L}_x$ , arranged in decreasing order. For a given point  $x$ , its  $K$  nearest neighbors form a local discrete neighborhood  $\mathcal{U}$  of  $x$ , whose geometry is encoded by the distance matrix  $\mathbf{D}_x$  as well as  $\mathcal{L}_x$ . In particular, similarly to the discrete Laplacian, the eigenvalues and eigenvectors of  $\mathcal{L}_x$  are determined by the geometry of  $\mathcal{U}$  that is invariant under orthogonal transforms [10]. Therefore, we call  $\mathbf{u}$  the LSF associated to  $x$  as it is completely determined by the spectral content of  $\mathcal{L}_x$  that encodes the geometry of a local discrete neighborhood  $\mathcal{U}$ .

It follows easily from the discussion that  $\mathbf{u}$  has the desired invariance:  $q = \mathbf{R}p$  implies  $\mathbf{u}_p = \mathbf{v}_q$ . This can be easily seen by noting that the two distance matrices  $\mathbf{D}_p, \mathbf{D}_q$  differ only by a permutation matrix  $P$ :

$$\mathbf{D}_q = P\mathbf{D}_pP^T.$$

Likewise,  $\mathcal{L}_p, \mathcal{L}_q$  also differ by the same permutation matrix. In particular,  $\mathcal{L}_p, \mathcal{L}_q$  have the same set of eigenvalues, and thus,  $\mathbf{u}_p = \mathbf{v}_q$  as both are arranged in decreasing order. We note that for a generic point set  $\mathbb{P}$ ,  $\mathbf{u}_p \neq \mathbf{u}_{\tilde{p}}$  for two distinct points  $p, \tilde{p} \in \mathbb{P}$ . Otherwise, we can always add a small perturbation to the point set to ensure this. In particular, this implies that for each  $p \in \mathbb{P}$ , there is one and only one point  $q \in \mathbb{Q}$  such that  $|\mathbf{v}_q - \mathbf{u}_p| = 0$ , and the right correspondences  $p \rightarrow q$  can be discovered by comparing the local spectral features. Thus, we have the following proposition:

**Proposition 4.1.** *Assume no noise.  $\mathbf{R}$  and hence  $\mathbf{A}$  can be determined exactly from a (generic) pair of point sets  $\mathbb{P}, \mathbb{Q}$  using local spectral features.*

### 4.4 Dealing with Noise

So far, we have assumed noiseless condition with equal number of points in  $\mathbb{P}, \mathbb{Q}$  and exact matching under  $\mathbf{A}$  and  $\mathbf{R}$ . Unfortunately, the invariance of local spectral features can be established only under this assumption. With the presence of noise,  $\mathbf{u}_p$  is generally not equal to  $\mathbf{v}_p$ . In particular,  $\mathcal{L}_q$  differs from  $\mathcal{L}_p$  by the permutation matrix  $P$  and an error matrix  $E$ :

$$\mathcal{L}_q = P\mathcal{L}_pP^T + E.$$

Under this perturbation, each eigenvalue  $\hat{\lambda}_i$  of  $\mathcal{L}_q$  can be considered as a shift of its corresponding eigenvalue  $\lambda_i$  of  $\mathcal{L}_p$ ,

$$\hat{\lambda}_i = \lambda_i + \Delta\lambda_i.$$

The standard result from numerical analysis ([24, p. 171]) shows that the change in the eigenvalue for a symmetric matrix under the above perturbation is conveniently bounded by the 2-norm of  $E$  for all  $i$ ,

$$|\Delta\lambda_i| \leq |E|_2,$$

Thus, provided that the noise is not too great, the difference  $|\mathbf{u}_p - \mathbf{v}_q|$  can still serve as a good indicator of plausible (or tentative) correspondence  $p \leftrightarrow q$  between points in  $\mathbb{P}$  and  $\mathbb{Q}$ .



Using local spectral features, the remaining steps of the algorithm that determines  $\mathbf{R}$  are well known [23]. The LSFs will be used to establish tentative correspondences: For each  $p \in \mathbb{P}$ , its plausible correspondence  $q \in \mathbb{Q}$  is the one with smallest  $|\mathbf{v}_p - \mathbf{v}_q|$ . While some of these plausible correspondences thus established are wrong, a good portion of them are expected to be correct. These correspondences  $p \leftrightarrow q$  are ranked using their similarity value  $|\mathbf{v}_p - \mathbf{v}_q|$ , and only the top 10 percent of them is retained as the tentative correspondences. RANSAC [14] is then applied to this collection of tentative correspondences to determine  $\mathbf{R}$  by randomly generating a number of hypotheses, orthogonal matrices obtained from sets of randomly generated  $m$  correspondences. Each orthogonal matrix is evaluated using the matching error function  $\mathbf{E}$ , and  $\mathbf{R}$  is given as the one with smallest matching error.

We conclude this section with a short summary of the three important features of the proposed algorithm. First, using local spectral features allows the algorithm to register point sets  $\mathbb{P}, \mathbb{Q}$  containing different number of points, a feature that is often not applicable to many correspondence-based methods, e.g., [52]. Second, the algorithm uses spectral invariants constructed from local distance matrices. Their definition and underlying geometric motivation are easier to understand than the more traditional methods of using algebraic invariants [27]. Finally, the affine ICP is initialized with an affine transform provided by an algorithm that is guaranteed to produce the exact affine transform in noiseless case. This imparts a certain assurance on the quality of the initialization, and consequently, the final result as well. For clarity, the full algorithm is summarized in Fig. 3.

#### 4.4.1 Managing Outliers

The proposed affine registration algorithm has two main steps: a normalization step followed by a step that uses RANSAC to determine the affine transform. The effect of outliers on the RANSAC step is generally minor compared with the effect on the normalization step since the covariance matrices as computed by (14) and (15) can be significantly affected by the outliers. Therefore, our approach to managing outliers will primarily focus on estimating the covariances matrices (and means) robustly in the presence of outliers. This is clearly related to the robust principal component analysis, and [49], [54] provide two of the first several robust PCA algorithms published in computer vision literature based partially on the earlier work in statistics [6], [31], [38]. While [49] is mostly concerned with PCA in high-dimensional image spaces, the method presented in [54] is ideally suited for the range of dimensions that we are working with. However, minimization of the objective function in [54] is not straightforward as it requires a combination of discrete and continuous optimization that attempts to avoid local minima by using simulated annealing. Therefore, in the experiments reported below, we choose to use an easier method to deal with the outliers.

The basic idea is to treat the points in  $\mathbb{P}$  and  $\mathbb{Q}$  as sample points drawn from some unknown probability density functions  $\mathbf{P}_{\mathbb{P}}, \mathbf{P}_{\mathbb{Q}}$ , respectively, and the covariance matrices (and the means) are determined using the PDFs instead of computing directly as in (14) and (15). Specifically, we fit the

### Affine Registration Algorithm

**Inputs:** Two point sets  $\mathbb{P} = \{p_1, \dots, p_k\}, \mathbb{Q} = \{q_1, \dots, q_l\}$  in  $\mathbb{R}^m$ , an integer  $K > 0$  and a real number  $\sigma > 0$ .

**Output:** An affine transform  $\mathbf{A}$ .

**1. Normalization** Points in  $\mathbb{P}, \mathbb{Q}$  are centered and normalized according to Equation 18, with  $\mathbf{S}_{\mathbb{P}}, \mathbf{S}_{\mathbb{Q}}$  defined in Equations 14 and 15, respectively.  $\mathbb{P}, \mathbb{Q}$  will denote the normalized point sets.

**2. Local Spectral Features** For each point  $x$  in  $\mathbb{P}$  and  $\mathbb{Q}$ , compute  $\mathcal{L}_x$ , a  $(K+1) \times (K+1)$  symmetric matrix, according to Equation 19, using its  $K$  nearest neighbors and  $\sigma$ . The eigenvalues of  $\mathcal{L}_x$ , arranged in the decreasing order, give the feature vector  $\mathbf{v}_x$  for  $x$ .

**3. Tentative Correspondences** For each  $p \in \mathbb{P}$ , its tentative correspondence  $q$  in  $\mathbb{Q}$  is determined as,

$$q = \arg \min_{q_i \in \mathbb{Q}} |\mathbf{v}_{q_i} - \mathbf{v}_p|_W^2.$$

The tentative correspondences  $p \leftrightarrow q$  are ranked using their  $L^2$ -differences,  $|\mathbf{v}_q - \mathbf{v}_p|_W^2$ , and top 10% correspondences with smallest  $L^2$ -differences are kept as tentative correspondences.

**4. RANSAC** Choose a collection  $\mathcal{C}$  of  $m$ -point correspondences from the pool of tentative correspondences determined in 3. For each  $m$ -point correspondence  $\tau$ , compute a rotation  $R_h \in \mathbf{O}(m)$ , and the optimal rotation is determined by

$$\mathbf{R} = \arg \min_{R_h, h \in \mathcal{C}} \mathbf{E}(R_h).$$

**5. Initial Affine transform**  $\mathbf{A} = (\mathbf{A}, \mathbf{t})$  is determined as

$$\mathbf{A} = \mathbf{S}_{\mathbb{Q}}^{\frac{1}{2}} \mathbf{R} \mathbf{S}_{\mathbb{P}}^{-\frac{1}{2}}, \quad \mathbf{t} = \mathbf{m}_{\mathbb{Q}} - \mathbf{A} \mathbf{m}_{\mathbb{P}}.$$

**6. Affine ICP**  $\mathbf{A} = (\mathbf{A}, \mathbf{t})$  as the initial affine transform

**Repeat Until Convergence** (Or the number of iterations exceeds a predefined value)

a. Determine Correspondences  $\pi$

$$\pi(i) = \arg \min_j \|\mathbf{A}p_i + \mathbf{t} - q_j\|^2$$

b. Determine Affine Transform  $\mathbf{A}_n$

$$(\mathbf{A}_n, \mathbf{t}_n) = \arg \min_{\mathbf{A}, \mathbf{t}} \|\mathbf{A}p_i + \mathbf{t} - q_{\pi(i)}\|^2$$

**If**  $\mathbf{E}(\mathbf{A}) < \mathbf{E}(\mathbf{A}_n)$

return  $\mathbf{A}$

**Else**

$\mathbf{A} = \mathbf{A}_n$

**End**

**End**

The registration error  $\mathbf{E}(\mathbf{A})$  is defined as the Hausdorff distance between  $\mathbf{A}(\mathbb{P}), \mathbb{Q}$  (Equation 11).

Fig. 3. Algorithm summary.

point set  $\mathbb{P}$  (and  $\mathbb{Q}$ ) with a Gaussian Mixture Model (GMM) of  $K$  components, where  $K$  is an integer that can be either user specified or determined by the required precision for the fitting [13]. The density function  $\mathbf{P}$  is given in the form

$$\mathbf{P}(x|\Theta) = \sum_{i=1}^K \omega_i \mathbf{N}(x; \mu_i, \Sigma_i),$$

TABLE 1  
Experimental Results I

Dim → Noise ↓	3	3	5	5	10	10
	Matrix Error	Matching Error	Matrix Error	Matching Error	Matrix Error	Matching Error
0%	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1%	0.0 (0.005)	0 (0)	0.0 (0.006)	0 (0)	0.004 (0.0008)	0.0 (0.002)
2%	0.007 (0.015)	0 (0)	0.004 (0.011)	0 (0)	0.013 (0.021)	0.012 (0.023)
5%	0.03 (0.03)	0 (0)	0.05 (0.04)	0 (0)	0.08 (0.12)	0.02 (0.04)
10%	0.08 (0.1)	0.01 (0.019)	0.11 (0.12)	0.03 (0.04)	0.13 (0.14)	0.04 (0.05)

For each dimension and each noisy setting, 100 trials, each with different point sets and matrix  $\mathbf{A}$ , were performed. The averaged relative error and percentage of mismatched points as well as standard deviations (in parenthesis) are shown.

where

$$\mathbf{N}(x; \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{m/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i)\right)$$

is the normal distribution in  $\mathbb{R}^m$  with mean  $\mu_i$  and covariance matrix  $\Sigma_i$ , and  $\omega_i$  are nonnegative weights that sum to 1. Let  $\Theta = \{\mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K, \omega_1, \dots, \omega_K\}$  denote the parameters of the Gaussian Mixture Model, and it is standard to estimate these parameters by maximizing the (log)likelihood

$$\mathbf{L}(\Theta) = \prod_{i=1}^K \mathbf{P}(x_i | \Theta)$$

given the data points  $x_1, \dots, x_K$ . This can be solved using the EM algorithm, and the details are available in many textbooks (e.g., [13]). Once the density function  $\mathbf{P}$  has been estimated, the mean and covariance matrix of the PDF can be computed using the formulas

$$\mu = \sum_{i=1}^K \omega_i \mu_i, \quad (20)$$

$$\Sigma = \sum_{i=1}^K \omega_i \Sigma_i + \sum_{i=1}^K (\mu_i - \mu)(\mu_i - \mu)^\top. \quad (21)$$

Compared with (14) and (15), the effect of outliers has been considerably curtailed in the above equations as the mean and the covariance matrix are no longer computed from the points directly. In addition, we will use the density functions  $\mathbf{P}_{\mathbb{P}}, \mathbf{P}_{\mathbb{Q}}$  to modify the Hausdorff distance (see (12)) that is used to define the matching cost function  $\mathbf{E}(\mathbf{A})$

$$\Gamma_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) = \sum_{p \in \mathbb{P}} \eta_p \min_{q \in \mathbb{Q}} |p - q| + \sum_{q \in \mathbb{Q}} \eta_q \min_{p \in \mathbb{P}} |q - p|, \quad (22)$$

with the weights  $\eta_p, \eta_q$  given by

$$\eta_p = \frac{\mathbf{P}_{\mathbb{P}}(p)}{\sum_{p \in \mathbb{P}} \mathbf{P}_{\mathbb{P}}(p)}, \quad \eta_q = \frac{\mathbf{P}_{\mathbb{Q}}(q)}{\sum_{q \in \mathbb{Q}} \mathbf{P}_{\mathbb{Q}}(q)}.$$

## 5 EXPERIMENTS

In this section, we report four sets of experimental results. First, with synthetic point sets, we show that the proposed affine registration algorithm does indeed recover exact affine transformations and correspondences for noiseless data. Second, we provide two sequences of nonrigid motions and show that the stereo correspondence under motion can be satisfactorily solved using affine registration

in  $\mathbb{R}^9$ . Third, we use images from COIL database to show that the image set matching problem can also be solved using affine registration in  $\mathbb{R}^8$ . And finally, we show two examples of covariant point-set matching. We have implemented the algorithm using MATLAB without any optimization. The sizes of the point sets range from 20 to 600, and on a DELL desktop with single 2.1 GHz processor,<sup>4</sup> each registration takes only a few minutes to complete.

### 5.1 Affine Registration in $\mathbb{R}^m$

In this set of experiments, our aim is to give a qualitative as well as quantitative analysis on the accuracy and robustness of the proposed method. Tables 1 and 2 summarize the experimental results. In Table 1, the algorithm is tested in three dimensions, 3, 5, and 10, and five different noise settings, 0, 1, 2, 5, and 10 percent. For each pair of dimension and noise setting, we ran 100 trials, each with a randomly generated nonsingular matrix  $\mathbf{A}$  and a point set containing 250 points. In trials with  $x$  percent noise setting, we add a uniform random noise ( $\pm x\%$ ) to each coordinate of every point independently. Let  $\mathbf{A}'$  denote the estimated matrix. A point  $p \in \mathbb{P}$  is matched to the point  $q \in \mathbb{Q}$  if  $q = \min_{q_i \in \mathbb{Q}} \text{dist}(\mathbf{A}'p, q_i)$ . For each trial, we report the percentage of mismatched points and the relative error of the estimated matrix  $\mathbf{A}'$ :  $\frac{\|\mathbf{A}' - \mathbf{A}\|}{\|\mathbf{A}\|}$ , using the Frobenius norm for matrices.

The number of RANSAC samples drawn in each trial has been fixed at 800 for the results reported in Table 1. This is the number of samples needed to produce zero mismatch for dimension five with 5 percent noise setting. In general, for lower dimensions, a much smaller number of samples (around 200) would also have produced similar results. In Table 2, we vary the sizes of the point sets and work in  $\mathbb{R}^{10}$ . The setting is similar to that of Table 1 except with fixed 5 percent noise setting for all trials. Note also that, for noiseless point sets, the exact affine transformations are always recovered.

### 5.2 Stereo Correspondences under Nonrigid Motions

In this experiment, we apply the affine registration algorithm to compute correspondences between tracked feature points in two image sequences. We gathered four video sequences from two cameras observing two objects undergoing non-rigid motions (Fig. 4). One is a talking head and the other is a patterned tattoo on a man's belly. A simple correlation-based feature point tracker is used to track 20-60 points for these

4. All running times reported below use this hardware configuration.

TABLE 2  
Experimental Results II

# of Pts → Errors ↓	100 Points	150 Points	200 Points	250 Points	300 Points	400 Points
Matrix Error	0.05 (0.03)	0.08 (0.08)	0.08 (0.09)	0.08 (0.11)	0.08 (0.012)	0.08 (0.11)
Matching Error	0.021 (0.012)	0.023 (0.014)	0.024 (0.011)	0.02 (0.012)	0.021 (0.016)	0.019 (0.014)

Point sets of different sizes with 5 percent noise added. All trials match point sets in  $\mathbb{R}^{10}$  with settings similar to Table 1. Average errors for 100 trials are reported with standard deviations in parenthesis.

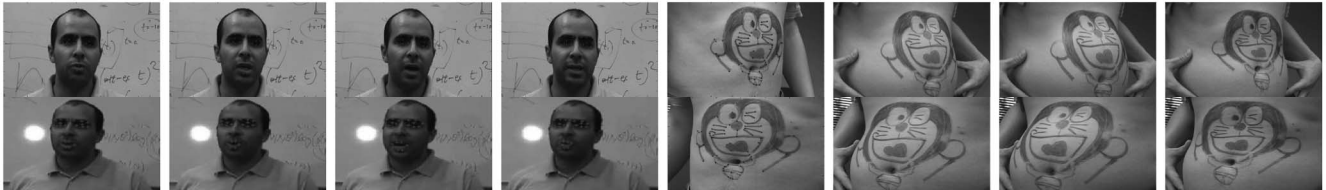


Fig. 4. Top: Sample frames from two video sequences of two objects undergoing nonrigid motions. Bottom: Sample frames from another camera observing the same motions.

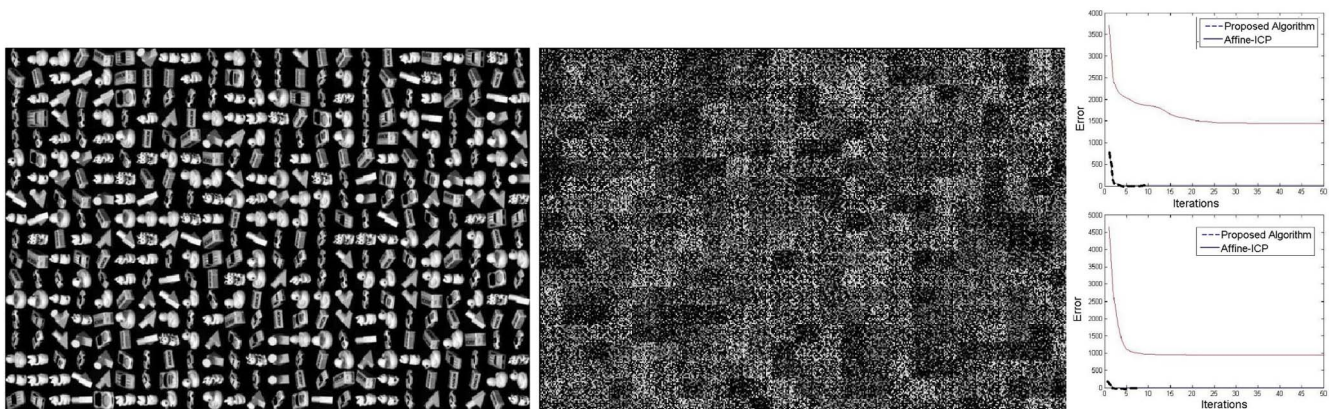


Fig. 5. Image set matching. The original image set  $A$  is shown in Fig. 1. Image sets  $B, C$  are shown above. The plots on the right show the  $L^2$  registration error for each of the 50 iterations of running the affine-ICP algorithm. Using the output of the proposed affine registration as the initial guess, the affine-ICP algorithm converges quickly to the desired transformation (dashed curves) and yields correct correspondences. Using closest points for initial correspondences, the affine-ICP algorithm converges (solid curves) to incorrect solutions in both experiments.

two sequences, respectively. Seventy frames were tracked in both sequences and manual intervention was required in both sequences to correct and adjust tracking results. We use three shape basis for both sequences [5], and to compute the correspondences, we affine register two point sets in  $\mathbb{R}^9$  as discussed before. For the two point sets  $\mathbb{P}, \mathbb{Q} \subset \mathbb{R}^9$ , we applied the proposed algorithm to obtain initial correspondences and affine transformation. This is followed by running an affine-ICP algorithm with 50 iterations. For comparison, the affine-ICP algorithm initialized using closest points<sup>5</sup> is run for 100 iterations. For the talking head sequence, the proposed algorithm recovers all correspondences correctly, while for the tattoo sequence, among the 60 recovered feature point correspondences, nine are incorrect. This may be explained by the fact that in several frames, some of the tracked feature points are occluded and missing and the subsequent factorizations produce relatively noisy point sets in  $\mathbb{R}^9$ . On the other hand, affine ICP with closest point initialization fails poorly for both sequences. In particular, more than 80 percent of the estimated correspondences are incorrect.

5. Given two point sets in  $\mathbb{R}^9$ , the initial correspondence  $p_i \leftrightarrow q_j$  is computed by taking  $q_j$  to be the point in  $\mathbb{Q}$  closest to  $p_i$ .

### 5.3 Image Set Matching

In this experiment, images from the first six objects in the COIL database are used. They define the source image set  $A$  with 432 images. Two new sets  $B, C$  of images are generated from  $A$ : images in  $B$  are 85 percent downsampled images of  $A$  followed by 45 degree rotation. Images in  $C$  are smoothed images in  $B$  (using a uniform  $7 \times 7$  kernel) followed by a random pixel permutation. Same pixel permutation scheme is applied to all images. The two experiments shown in Fig. 5 match point sets  $A, B$  and  $A, C$ . The original images have size  $128 \times 128$  and the images in the two new sets have size  $110 \times 110$ . An 8D PCA subspace is used to fit each set of images with relative residue smaller than 1 percent. Images in each set are projected down to their respective PCA subspaces, and the correspondences are automatically computed by affine registering the projected point sets. Specifically, a basis is chosen for each PCA subspace, and each projected image is represented as a vector using the basis. The data are nontrivial in the sense that correspondences established using nearest neighbors in the PCA subspaces for both image sets are almost always incorrect (less than 5 percent correct). We apply the proposed affine registration algorithm to obtain an initial estimate on correspondences and affine registration. Since the data are noisy, we follow this with the affine-ICP algorithm running 50 iterations as above. For comparison,

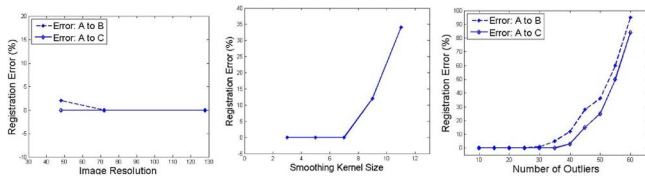


Fig. 6. Left: Image resolution versus registration error (in percent). Three image resolutions are used:  $48 \times 48$ ,  $72 \times 72$ , and  $128 \times 128$ . Errors from matching A to B and A to C are plotted in solid and dashed lines, respectively. Center: Smoothing kernel size versus registration error (in percent). Five different kernel sizes are used in this experiment:  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ , and  $11 \times 11$ . Right: Number of outliers versus average registration errors (in percent). The source image set A has 432 images, and up to a maximum number of 60 outliers are added to B and C in this experiment. Errors from matching A to B and A to B are plotted in solid and dashed lines, respectively.

we apply the affine-ICP algorithm using closest points as initialization. In both experiments, the affine-ICP algorithm, not surprisingly, performs poorly, with substantial registration errors (see (11)) and a large number of incorrect correspondences. The proposed algorithm recovers all correspondences correctly and it yields small registration errors. We remark that images in C are obtained from images in B by smoothing with a uniform kernel and a pixel permutation. Superficially, it would appear that pixel permutation is the main source of difficulty as it gives the appearance of impossibility. However, it is the smoothing that causes the difficulty since its associated image transform  $\mathcal{F}$  is not an orthogonal transform in the image space. In fact, for this example, using smoothing kernels larger than  $7 \times 7$ , the matching problem cannot be solved satisfactorily using affine registration. Pixel permutation, on the other hand, can be represented by a permutation (orthogonal) matrix in the image space; therefore, as an orthogonal transform, it will preserve the PCA subspaces and the correspondences can be recovered exactly by affine registering the points in the PCA subspaces.

Fig. 6 contains the plots from three experiments that evaluate the effects of image resolutions and outliers on the image set matching. In the first plot, we test the algorithm using three different image resolutions,  $48 \times 48$ ,  $72 \times 72$ , and  $128 \times 128$ . For the three resolutions, the image sizes of the target sets B, C are  $42 \times 42$ ,  $62 \times 62$ , and  $110 \times 110$ , respectively, and the sizes of the smoothing kernels are  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$ , respectively. The error rates of the matching results show that they do not depend significantly on the image resolution with the chosen smoothing kernels. In the second plot, we show the effect of varying the size of the smoothing kernel in matching A to C. In this experiment, the image resolution is  $128 \times 128$ , and we evaluate the algorithm using five different kernel sizes,  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ , and  $11 \times 11$ . Matching error emerges when  $9 \times 9$  kernel is used to smooth the images, and the result degrades precipitously with larger kernels. In the third plot, we evaluate the matching algorithm using image sets containing outliers. In this experiment, image resolution for the source image set A is  $128 \times 128$ , with 85 percent reduction for images in B, C and  $7 \times 7$  smoothing kernel as above. The number of images in A is kept fixed in this experiment while we steadily increase the number of images in B, C by adding outliers as follows: Images from the remaining COIL database are randomly

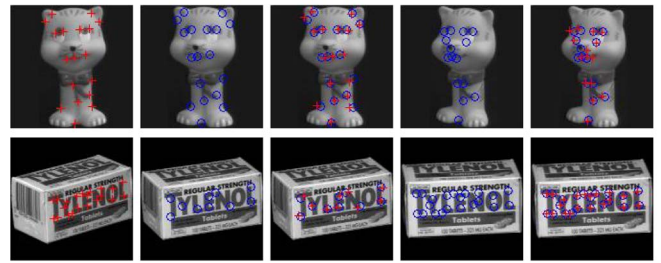


Fig. 7. 2D image registration. First column: Source images (taken from COIL database) with feature points marked in red. Second and fourth column: Target images with feature points marked in blue. Third and fifth column: Target images with corresponding feature points marked in red. The affine transformed points from the source images are marked in red. Images are taken with 15 and 30 degrees differences in viewpoint. The RMS errors for these four experiments (from left to right) 2.4500, 2.502, 1.9834, and 0.6345, respectively.

selected as outliers, and they are rotated and permuted/smoothed as the images in B and C before projecting onto their respective 8D subspaces. The matching result is evaluated using a different number of outliers, up to 15 percent of the number of images in A. For each number of outliers, we run the matching algorithm 20 times using 20 different collections of outliers, and the averaged matching error from these 20 independent matching result is plotted in Fig. 6. The result shows that the algorithm is capable of producing satisfactory results if the number of outliers is less than 8 percent of the number of images. Beyond this point, the result degrades quickly. In this experiment, the number of Gaussian mixture components used is set at  $K = 18$  (see (20) and (21)).

## 5.4 Covariant Point-Set Matching

For covariant point-set matching, we report two experiments.

### 5.4.1 Image Registration

In the first experiment, we register 2D images using covariant point-set matching. It is known that the effect of a small view change on an image can be approximated by a 2D affine transformation of the image [23]. Using images from the COIL database, we manually click feature points on pairs of images with a 15 to 30 degree difference in viewpoint. The feature vector associated to each point is the averaged structure tensor (see (5)) over a  $3 \times 3$  neighborhood, and the image registration is solved as an affine registration in  $\mathbb{R}^5$ . The results for four pairs of images are shown in Fig. 7, and the RMS registration errors (in pixels) for these four examples are 2.4500, 2.502, 1.9834, and 0.6345, respectively. Compared with the 2D affine registration of these points with RMS registration errors of 2.6646, 3.0260, 2.0632, 0.7060, the covariant point-set matching provides about 15 percent improvement in registration error. This is not particularly surprising since, as discussed before, the induced transform in  $\mathbb{R}^2$  of the affine transform in  $\mathbb{R}^5$  has a nonrigid component, which, in this case, accounts for the slight improvement in accuracy.

### 5.4.2 3D Registrations

In this experiment, we use covariant point-set matching to register point sets in  $\mathbb{R}^3$  containing thousands of points. We use the 33,456 points from the Stanford Bunny (Fig. 8) as the



TABLE 3  
Registration Errors

Reduced Point Set Size	Matching Error		Noise Level 500 Points	Matching Error	
	Axis	Angle		Axis	Angle
500(350)	4.32°	4.76°	0%(350)	4.32°	4.76°
400(250)	7.32°	7.76°	6%(350)	9.21°	10.16°
200(140)	13.32°	18.76°	12%(350)	15.52°	19.26°

Left: The number of neighbors  $K$  used in computing local spectral features are shown in the parenthesis, and 500 RANSAC samples are evaluated for each registration. Each error is averaged from 100 independent registrations. Right: Registration Errors with noisy data points. 500 reduced points are used with two noise settings, 6 and 12 percent. All other settings are the same as above.

point set  $\mathbb{P}$ , and  $\mathbb{P}$  is normalized so that it has zero mean and unit variance. The target point set  $\mathbb{Q}$  is obtained by applying a randomly generated rotation  $\mathbf{R}$  to  $\mathbb{P}$ . Since both  $\mathbb{P}, \mathbb{Q}$  have unit variance, efficient spectral methods [42], [43] that match principal directions cannot be applied here. Furthermore, direct methods such as [33] become inefficient because of the large number of points.<sup>6</sup> We register  $\mathbb{P}, \mathbb{Q}$  by simplifying the point sets first: 500 points are randomly selected from  $\mathbb{P}, \mathbb{Q}$  independently to form the reduced point sets  $\overline{\mathbb{P}}, \overline{\mathbb{Q}}$  that are registered using covariant matching. The feature vector associated to each point in  $\overline{\mathbb{P}}, \overline{\mathbb{Q}}$  is the 6D feature vector formed by the local quadratic moments shown in (10). Specifically, a radius  $r$  is specified for all points in  $\overline{\mathbb{P}}, \overline{\mathbb{Q}}$  such that for each  $p \in \overline{\mathbb{P}}$  ( $q \in \overline{\mathbb{Q}}$ ), points  $p_1, \dots, p_l$  ( $q_1, \dots, q_l$ ) in  $\mathbb{P}$  ( $\mathbb{Q}$ ) that are within  $r$ -distance from  $p$  ( $q$ ) are determined. These points are used to compute the feature vector  $\mathbf{v}_p$  ( $\mathbf{v}_q$ ) as in (10), where

$$(x_i, y_i, z_i) = p_i - p.$$

The covariant matching result is used to determine the correspondences, and given the correspondences, the optimal rotation  $\overline{\mathbf{R}}$  is computed using SVD [26]. We compare  $\overline{\mathbf{R}}$  with the ground truth  $\mathbf{R}$  using the angle between their rotation axes and the difference between their rotation angles. We randomly sample 100 rotation matrices  $\mathbf{R}$  as the ground truths and the average errors (rotation axis and angle) are shown in Table 3 (and an example shown in Fig. 8). The result shows that, not surprisingly, using fewer points, the estimated rotation deviated significantly from the ground truth. However, using 500 points, which are about 1.5 percent of the original points, the registration result is quite reasonable with average error in rotation axis less than 5 degrees. Each registration takes less than 4 minutes to complete, and it shows the benefit of trading problem complexity (number of points) with feature dimension.

In this experiment, an important issue is the determination of appropriate  $\lambda$  (see (7)) to use. Currently, we do not have a principled method for determining  $\lambda$ , and we have simply scanned for the best value over a viable range for  $\lambda$ . This experiment shows that for the right values of  $\lambda$ , covariant point-set matching is a viable method for registering point sets containing large number of points. The issue of  $\lambda$  selection will be addressed in the future.

6. Using a C-implementation, [33] reports the running time of 1,200 seconds with 200 points.

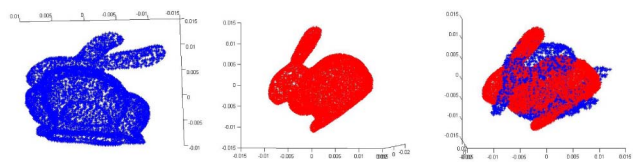


Fig. 8. Left: The normalized Stanford bunny shown with about 6,000 points. Center: A rotated bunny shown with about 6,000 points. Right: Reduced point sets  $\overline{\mathbb{P}}, \overline{\mathbb{Q}}$ . Source (blue) and target (red) point sets shown together.

## 6 CONCLUSION

This paper has described three interesting types of point-set matching problems that can be formulated and solved as affine registration problems in  $\mathbb{R}^m$  with  $m > 3$ : stereo correspondences under motion, image set matching, and covariant point-set matching. To efficiently solve these problems, the paper has proposed a novel affine registration algorithm. The algorithm first computes the correspondences using local spectral features, which are the eigenvalues of certain symmetric matrices determined by local distance matrices. The correspondences then form the initialization for the affine-ICP algorithm that iteratively solves for the correspondences and affine transform. Experimental results using synthetic data points have shown that the proposed algorithm is efficient and accurate for data containing moderate amount of noise. Furthermore, the paper has presented a variety of the aforementioned matching problems that can indeed be solved satisfactorily using affine registration.

## ACKNOWLEDGMENTS

The authors thank one of the reviewers for providing many invaluable comments and suggestions. S.M. Shahed Nejhum, Y.-T. Chi, and J. Ho are partially supported by a US National Science Foundation (NSF) grant IIS-0916001. M.-H. Yang is supported in part by a Google faculty award and University of California Merced faculty start-up fund.

## REFERENCES

- [1] A. Makadia, A.I. Patterson, and K. Daniilidis, "Fully Automatic Registration of 3D Point Clouds," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 1297-1304, 2006.
- [2] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522, Apr. 2002.
- [3] P.J. Besel and H.D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.
- [4] M. Brand, "Morphable 3D Models from Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 456-463, 2001.
- [5] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering Non-Rigid 3D Shape from Image Streams," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2690-2696, 2000.
- [6] N. Campbell, "Multivariate Analysis I: Robust Covariance Estimation," *Applied Statistics*, vol. 29, no. 3, pp. 231-2137, 1980.
- [7] M. Carcassoni and E. Hancock, "Spectral Correspondence for Point Pattern Matching," *Pattern Recognition*, vol. 36, no. 1, pp. 193-204, 2003.
- [8] Y.-T. Chi, N. Shahed, J. Ho, and M.-H. Yang, "Higher Dimensional Affine Registration and Vision Applications," *Proc. European Conf. Computer Vision*, vol. 4, pp. 256-269, 2008.
- [9] H. Chui and A. Rangarajan, "A New Algorithm for Non-Rigid Point Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 44-51, 2000.

- [10] F.R.K. Chung, *Spectral Graph Theory*. Am. Math. Soc., 1997.
- [11] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active Shape Models: Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, 1995.
- [12] S. Du, N. Zheng, S. Ying, and J. Liu, "Affine Iterative Closest Point Algorithm for Point Set Registration," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 791-799, 2010.
- [13] R. Duda and P. Hart, *Pattern Classification*. Wiley-Interscience, 2000.
- [14] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, June 1981.
- [15] A.W. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets," *Computer Vision and Image Understanding*, vol. 2, nos. 13/14, pp. 1145-1153, 2003.
- [16] J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants," *Pattern Recognition*, vol. 1, pp. 167-174, 1993.
- [17] M. Garland and P. Heckbert, "Surface Simplification Using Quadric Error Metrics," *Proc. SIGGRAPH*, pp. 209-216, 1998.
- [18] N. Gelfand, N. Mitra, L. Guibas, and H. Pottmann, "Robust Global Registration," *Proc. Symp. Geometry Processing*, pp. 197-207, 2005.
- [19] C. Gope and N. Kehtarnavaz, "Affine Invariant Comparison of Point-Sets Using Convex Hulls and Hausdorff Distances," *Pattern Recognition*, vol. 40, pp. 309-320, 2007.
- [20] A. Goshtasby and G. Stockman, "Point Pattern Matching Using Convex Hull Edges," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, pp. 631-637, 1985.
- [21] S. Granger and X. Pennec, "Multi-Scale EM-ICP: A Fast and Robust Approach for Surface Registration," *Proc. European Conf. Computer Vision*, vol. 3, pp. 418-432, 2002.
- [22] M. Hagedoorn and R. Velkamp, "Reliable and Efficient Pattern Matching Using an Affine Invariant Metric," *Int'l J. Computer Vision*, vol. 21, nos. 2/3, pp. 203-225, 1999.
- [23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, second ed. Cambridge Univ. Press, 2003.
- [24] M.T. Heath, *Scientific Computing—An Introductory Survey*. McGraw-Hill, 2002.
- [25] J. Heikkilä, "Pattern Matching with Affine Moment Descriptors," *Pattern Recognition*, vol. 37, no. 9, pp. 1825-1834, 2004.
- [26] B. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *J. Optical Soc. Am. A: Optics, Image Science, and Vision*, vol. 4, no. 4, pp. 629-642, 1987.
- [27] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Trans. Information Theory*, vol. IT-8, no. 2, pp. 179-187, Feb. 1962.
- [28] D. Huttenlocher, "Fast Affine Point Matching: An Output-Sensitive Method," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 263-268, 1991.
- [29] D. Huttenlocher, D. Klanderman, and A. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850-863, Sept. 1993.
- [30] A. Johnson and M. Herbert, "Using Spin Images for efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433-449, May 1999.
- [31] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag, 1986.
- [32] Y. Lamdan, J. Schwartz, and H. Wolfson, "Affine Invariant Model-Based Object Recognition," *IEEE Trans. Robotics Automation*, vol. 6, no. 5, pp. 578-589, Oct. 1990.
- [33] H. Li and R. Hartley, "The 3D-3D Registration Problem Revisited," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.
- [34] M. Pauly and M. Gross, "Spectral Processing of Point-Sampled Geometry," *Proc. ACM SIGGRAPH*, pp. 379-386, 2001.
- [35] T. Reiss, "The Revised Fundamental Theorem of Moment Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 830-834, Aug. 1991.
- [36] W. Rucklidge, "Efficiently Locating Objects Using the Hausdorff Distance," *Int'l J. Computer Vision*, vol. 24, no. 3, pp. 251-270, 1997.
- [37] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," *Proc. Third Int'l Conf. 3D Digital Imaging and Modeling*, pp. 145-152, 2001.
- [38] E. Ruymaart, "A Robust Principal Component Analysis," *J. Multivariate Analysis*, vol. 1, pp. 485-497, 1981.
- [39] R. Sarfari, N. Narsimhamurthi, M. Shridhar, and M. Ahmadi, "Affine Point Pattern Matching and Its Application to Form Registration," *Proc. IEEE Symp. Circuits and Systems*, vol. 2, pp. 656-659, 1996.
- [40] S. Sclaroff and A. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545-561, June 1995.
- [41] G. Scott and C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," *Proc. Royal Soc. London*, vol. B244, pp. 21-26, 1991.
- [42] G. Scott and C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," *Proc. Royal Soc. London*, vol. B244, pp. 21-26, 1991.
- [43] L. Shapiro and J. Brady, "Feature-Based Correspondence: An Eigenvector Approach," *Image and Vision Computing*, vol. 10, no. 5, pp. 283-288, 1992.
- [44] G.C. Sharp, S.W. Lee, and D.K. Wehe, "ICP Registration Using Invariant Features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90-102, Jan. 2002.
- [45] J. Sprinzak and M. Werman, "Affine Point Matching," *Pattern Recognition Letters*, vol. 15, pp. 337-339, 1994.
- [46] G. Stockman, "Object Recognition and Localization via Pose Clustering," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 3, pp. 361-387, 1987.
- [47] H. Suesse and W. Ortmann, "Robust Matching of Affinely Transformed Objects," *Proc. Int'l Conf. Image Processing*, vol. 2, pp. 375-378, 2003.
- [48] C. Toamsi and T. Kanade, "Shape and Motion from Image Streams Under Orthography—A Factorization Method," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 137-154, 1992.
- [49] F.D.L. Torre and M. Black, "Robust Principal Component Analysis for Computer Vision," *Proc. Int'l Conf. Computer Vision*, pp. 362-369, 2001.
- [50] K. Voss and H. Suesse, "Affine Point Pattern Matching," *Proc. Pattern Recognition*, pp. 155-162, 2001.
- [51] T. Wakahara, Y. Kimura, and A. Tomono, "Affine-Invariant Recognition of Gray-Scale Characters Using Global Affine Transformation Correlation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 384-395, Apr. 2001.
- [52] Z. Wang and H. Xiao, "Dimension-Free Affine Shape Matching through Subspace Invariance," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
- [53] H. Wolfson and I. Rigoutsos, "Geometric Hashing: An Overview," *IEEE Computational Science and Eng.*, vol. 4, no. 4, pp. 10-21, Oct. 1997.
- [54] L. Xu and A. Yuille, "Robust Principal Component Analysis by Self-Organizing Rules Based on Statistical Physics Approach," *IEEE Trans. Neural Networks*, vol. 6, no. 1, pp. 131-143, Jan. 1995.
- [55] Z. Yang and F. Cohen, "Cross-Weighted Moments and Affine Invariants for Image Registration and Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 804-814, Aug. 1999.
- [56] Z. Yang and F. Cohen, "Image Registration and Object Recognition Using Affine Invariants and Convex Hulls," *IEEE Trans. Image Processing*, vol. 8, no. 7, pp. 934-946, July 1999.
- [57] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves," *Int'l J. Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.



**S.M. Shahed Nejhum** received the BS degree from the Bangladesh University of Engineering and Technology. He is currently working toward the PhD degree in computer and information science and engineering at the University of Florida. His research interests include computer vision and machine learning.



**Yu-Tseh Chi** received the BS degree from the National Taiwan University and the MS degree from The Ohio State University. He is currently working toward the PhD degree in computer and information science and engineering at the University of Florida. His research interests include computer vision and machine learning.



**Jeffrey Ho** received the PhD degree in mathematics and the MS degree in computer science from the University of Illinois, Urbana-Champaign, in 1999 and 2000, respectively. He is an assistant professor at the University of Florida. His current research interests include point-set and image registration, and segmentation and visual tracking. He is a member of the IEEE and the IEEE Computer Society.



**Ming-Hsuan Yang** received the PhD degree in computer science from the University of Illinois, Urbana-Champaign, in 2000. He is an assistant professor in electrical engineering and computer science at the University of California, Merced. He coauthored the book *Face Detection and Gesture Recognition for Human-Computer Interaction* (Kluwer Academic, 2001) and edited a special issue on face recognition for *Computer Vision and Image Understanding* in 2003. His

research interests include computer vision, pattern recognition, artificial intelligence, robotics, and machine learning. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Image and Vision Computing*. He is a senior member of the IEEE and a member of the IEEE Computer Society and the ACM.

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**