

A Grassmannian Graph Approach to Affine Invariant Feature Matching

Mark Moyou[†], Anand Rangarajan[‡], John Corring[‡] and Adrian M. Peter[†]

[†]Department of CISE, University of Florida, Gainesville, FL, USA

[‡]Department of Engineering Systems, Florida Institute of Technology, Melbourne, FL, USA

Abstract—In this work, we present a novel, theoretical approach to address one of the longstanding problems in computer vision: 2D and 3D affine invariant feature matching. Our proposed Grassmannian Graph (GrassGraph) framework employs a two stage procedure that is capable of robustly recovering correspondences between two unorganized, affinely related feature (point) sets. In the ideal case, the first stage maps the feature sets to an affine invariant Grassmannian representation, where the features are mapped into the same subspace. It turns out that coordinate representations extracted from the Grassmannian differ by an arbitrary orthonormal matrix. In the second stage, by approximating the Laplace-Beltrami operator (LBO) on these coordinates, this extra orthonormal factor is nullified, providing true affine invariant coordinates which we then utilize to recover correspondences via simple nearest neighbor relations. Our validation benchmarks use large number of experimental trials performed on 2D and 3D datasets. Experimental results show that the proposed GrassGraph method successfully recovers a large affine transformations.

I. INTRODUCTION

Feature matching has been a vital component of computer vision since Fischler and Elschlager's seminal work in 1973 [14]. Since that beginning, there have been torrents of work in this area making it almost impossible to characterize or synthesize. Given this voluminous previous work, it's understandable if one adopts a perspective that the bar is too high for new ideas. Our goal in the present work is to believe that opinion, clearly demonstrating a novel approach which is simultaneously new (to the best of our knowledge) and easy to implement.

Graph representations abound in computer vision. The very first matching work featured a relational graph representation [14], invariant under rigid shape transformations. Within this subfield, there again exist innumerable works with extensions, new formulations, algorithms and the like. Graph representations lead to graph matching [8], [9]—which despite being NP-hard [55]—has attracted a huge amount of attention over the decades. Graph representations have also been extended to include invariances under similarity transformations [46] and lately have surfaced in non-rigid matching situations as well [56].

Invariants in computer vision have seen better days. While they were the flavor *du jour* in the early '90s, with work ranging from geometric hashing [24] to generalized Hough transforms [1], lately this work has not seen much development. Invariants were not robust to missing features and they could not easily be extended to non-rigid matching situations.



Figure 1. GrassGraph correspondence recovery of SIFT features on real image pairs related by a large affine transformation. Our affine invariant algorithm recovers dense correspondences on unstructured pointsets no matter how large the affine transformation. (For visualization purposes, only a fraction of the full correspondences are shown.)

We speculate, however, that an important additional reason why invariants did not see wide adoption was due to the absence of invariant feature *coordinate systems*. It is *exactly* this aspect of invariance which has seen a resurgence in the past few years [42].

In recent years that it has become commonplace to see new coordinates built out of the discrete Laplace-Beltrami Operator (LBO) eigenvectors [21], [43]. The basic idea is extremely straightforward: construct a weighted graph representation from a set of features and then use the principal component eigenvectors of the graph (interpreted as a linear operator) to complete the construction. These new coordinates can then be pressed into service in matching and indexing applications. However, note the fundamental limitation to similarity transformations: Can this approach be extended to affine invariance while remaining somewhat robust in the face of noise and outliers?

We answer in the affirmative. Figure 1, shows the GrassGraph algorithm recovering dense correspondences (of SIFT features) on real image pairs related by a large affine transformation. In this work, we begin with the original feature set and first construct the Grassmannian. The Grassmannian [4] is a geometric object with the property that all linear combinations of the feature coordinates remain in the same Grassmannian subspace. Therefore a *single* element of the Grassmannian can be considered to be an affine invariant equivalence class of feature sets. Further, Grassmannians are homeomorphic to orthogonal projection matrices with the latter available for the construction of new affine invariant coordinates. Unfortunately,

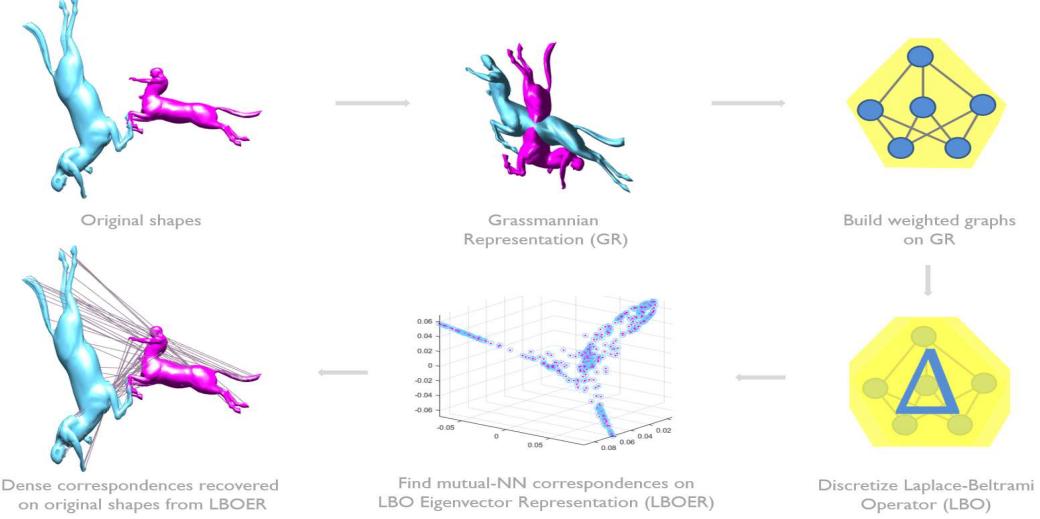


Figure 2. Overview of the GrassGraph framework. The original shape (purple) and its affine transformed version (cyan) are shown on the top left. First, the Grassmannian representation (GR) of the shapes are computed—under this representation the affine transformation is nullified up to an arbitrary rotation. To be fully affine invariant, the rotation must be removed by first constructing a weighted graph on the GR of each shape. Next, the eigenvectors of the graph Laplacian (discretization of the Laplace-Beltrami Operator) computed on the weighted graph are used to obtain affine invariant coordinates. By using the LBO eigenvectors, the arbitrary rotation is removed and dense correspondences (see Figure 3) can be retrieved using mutual nearest neighbor (NN) search. The LBO eigenvector representation of the two pointsets are shown as purple dots (corresponding to the purple centaur) aligned with cyan circles (corresponding to the cyan centaur). Note, that the GG points for both shapes are the same in this embedding space due to the affine invariance property of our method. (In this paradigm we work with the 2D and 3D pointsets directly, but the meshes are shown for visualization; only a subset of correspondences are shown.).

it turns out that two factorizations of projection matrices can differ by an unknown orthonormal matrix. To circumvent this problem, in a second stage, we construct a weighted graph which is invariant to this additional orthonormal factor and then use LBO principal components (as outlined above) to obtain new affine invariant coordinates. Given two sets of features, finding good correspondences is considerably easier in this representation since the affine invariant coordinates lead to efficient nearest neighbor correspondences. The twin strands of research married in our approach are therefore (i) affine invariant Grassmannian representations and (ii) LBO-based weighted graphs resulting in the *GrassGraph algorithm*¹ for affine invariant feature matching. A visual representation of the algorithm is shown in Figure 2. Our successful recovery of dense correspondences in the presence of large affine transformation and ease of implementation should pave the way for the Grassmannian graph representation to be widely used in feature matching and indexing applications.

II. RELATED WORK

As mentioned previously, the corpus related to affine invariance and graph matching is quite vast (see surveys [47], [51]). Here we highlight the most relevant and pioneering works that have steered the course of this research. The works span the timeline from the late 80’s until present, which suggests that the affine invariance problem is still relevant and worthy of further investigation.

¹The GrassGraph Algorithm code is available at <https://github.com/MarkMoTrin/GrassGraphs.git> under the GNU GPLv3 license.

1) *Spectral Methods:* We mentioned graph matching as a well-worn route to encoding invariance. One popular approach to solve the graph matching problem is to use spectral representations of adjacency matrices or equivalently embed the vertices into a Euclidean space, where linear or nonlinear matching algorithms can be employed. Umeyama [50] pioneered this approach for the weighted graph matching problem, while further methods based on spectral representations were proposed in [5], [6], [7], [13], [28], [33], [44], [45].

For 2D pointsets, Leordeanu and Hebert [26] presented an efficient spectral solution to the correspondence problem using pairwise constraints between candidate assignments. They build a unique adjacency matrix M of a graph whose nodes represent the potential correspondences and the weights represent pairwise agreement between potential correspondences. Correct assignments are recovered using the principal eigenvector of M with flexible mapping constraints deployed. In [19], the Laplacian embedding is used to embed 3D meshes using a global geodesic distance where these embeddings are matched using the iterated closest point (ICP) algorithm [54]. Mateus *et al.* [32] used a subset of Laplacian eigenfunctions to perform dense registration between articulated pointsets. Once in the eigenspace, the registration was solved using unsupervised clustering and the EM algorithm. All of these previous methods used an eigenvector decomposition to solve the point matching problem after removing individual transformations: translation, rotation, scale and shear. None are truly invariant to the class of affine transformations as a whole, nor do they produce affine invariant coordinates; in addition, their actual correspondence schemes in the related feature spaces are significantly more complicated than our simple mutual nearest neighbor matching.

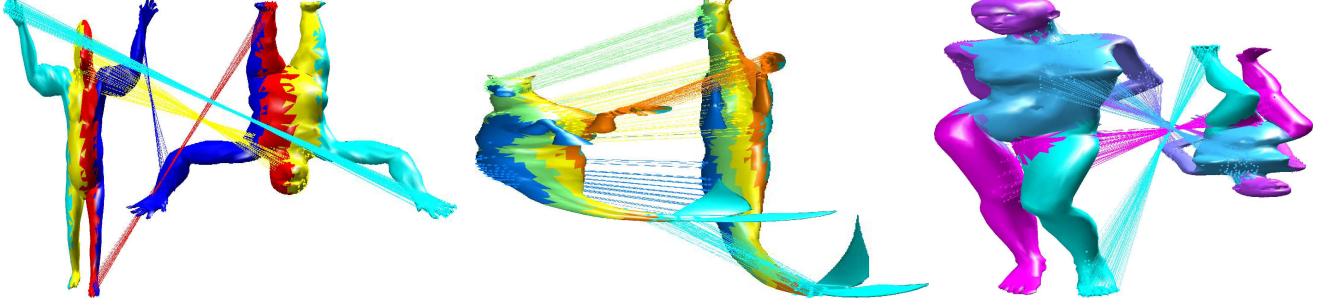


Figure 3. Correspondences on 3D affine shapes from the GrassGraph algorithm. (We recover dense correspondences, but only a subset of the matching points are shown for visualization purposes.)

Two affine invariant works [40], [41] have recently been proposed. In [40] Raviv *et al.* form an equi-affine invariant Laplacian for surfaces, which allows them to perform equi-affine (volume preserving) invariant diffusion geometry. In order to retrieve correspondences, their method uses intrinsic distances on the surface that are invariant to the class of equi-affine transformations. Our entire representation is affine invariant, whereas this method gives a metric on the shape that is affine invariant. In fact, they explicitly parametrize the mesh and work with local metric tensors. This approach is not readily transferable when working with discrete feature sets, is limited to 3D mesh shapes, and considerably more complicated than the proposed GG framework.

2) *Multi-step Affine Invariant Methods*: Staying within the theme of multi-step affine invariant approaches, Ho *et al.* [17] proposed an elegant noniterative algorithm for 2D affine registration by searching for the roots of the associated polynomials, but their method does not perform well under large rotations as shown in Figure 4. Qu *et al.* [38] extended the work in [17] to three dimensions using quaternions. Although both of these works [17], [38] are devoid of iterative optimization components in their frameworks (similar to ours), we still differ significantly because of our subspace approach (which addresses all transformations as a whole). Both these methods fall under the stepwise transformation removal paradigm of invariance.

3) *Subspace Methods* : Moving away from the stepwise invariance approach above, the work of Begelfor and Werman [2] popularized the use of Grassmannian subspaces as an affine invariant.. Their work focused on developing clustering algorithms on the Grassmann manifold, but this was not extended to shape registration—which opened the door to our contribution of using this subspace to create affine invariant coordinate representations. A method that did try to address the registration problem through subspace invariance was [52], [53]. Here, Wang and Xiao proposed a compact, arbitrary-dimension, affine invariant matching algorithm that found correspondences using QR factorizations of rank-deficient orthogonal projection matrices. Though the works [52], [53] propose an elegant algorithm based on a similar Grassmannian principle, our method is able to produce a truly affine invariant coordinate representation—recovering correspondences with a simple nearest neighbor scheme versus using column pivoting of QR matrices.

4) *Non-rigid Methods*: Non-rigid pointset registration subsumes affine pointset registration, therefore, in theory, non-rigid methods should perform better at the affine invariance problem. The Robust Point Matching (RPM) algorithm introduced by Gold *et al.* [15], and its variants [10], [11], [39] use soft assignment to assign probability weights between all combinations of points. In [11] Chui and Rangarajan modeled the registration problem as a joint optimization over the transformation parameters and correspondence matrix; the non-rigid deformation is modeled by thin-plate splines (TPS) and interpreted in an expectation maximization (EM) framework. Coherent Point Drift (CPD) by Myronenko *et al.* [35] is similar to [11] but they used Gaussian radial basis functions instead of TPS to model the non-rigid deformation. Ma *et al.* [29] extend RPM [15] with RPM- L_2E by iteratively establishing correspondences using shape contexts [3] as a feature representation and estimating the transformation using an L_2E estimator that minimizes the L_2 distance between densities (visual registration results shown in Figure 7 as RPM-L2E).

Tsin and Kanade [49] proposed a correlation-based approach using a measure called kernel correlation (KC) which is proportional to the correlation of two kernel density estimates. Jian and Vemuri [20] improved [49] by introducing GMMReg, a registration framework using Gaussian mixture models (GMMs). The pointset registration problem is treated as that of aligning two Gaussian mixtures by minimizing the discrepancy between the two mixtures.. Our invariance to rotations in the GrassGraph approach is in sharp contrast to CPD, GMMReg and KC which are well known to be susceptible to large rotations as shown in Figure 4.

A probabilistic approach that uses a GMM based technique for affine and non-rigid registration is [37] where the transformation between the pointsets, noise and outliers are all modeled with separate GMMs. An iterative two stage process involving regression and clustering is used to estimate the transformation and establish correspondences, respectively. Even though noise and missing points are modeled with anisotropic covariance matrices, this method relies on the initialization of 14 parameters for good performance. Figure 7 shows that the method is still susceptible to error in large affine transformation situations.

Ma *et al.* [31] provide a non-rigid correspondence method based on interpolation of a vector field between two point

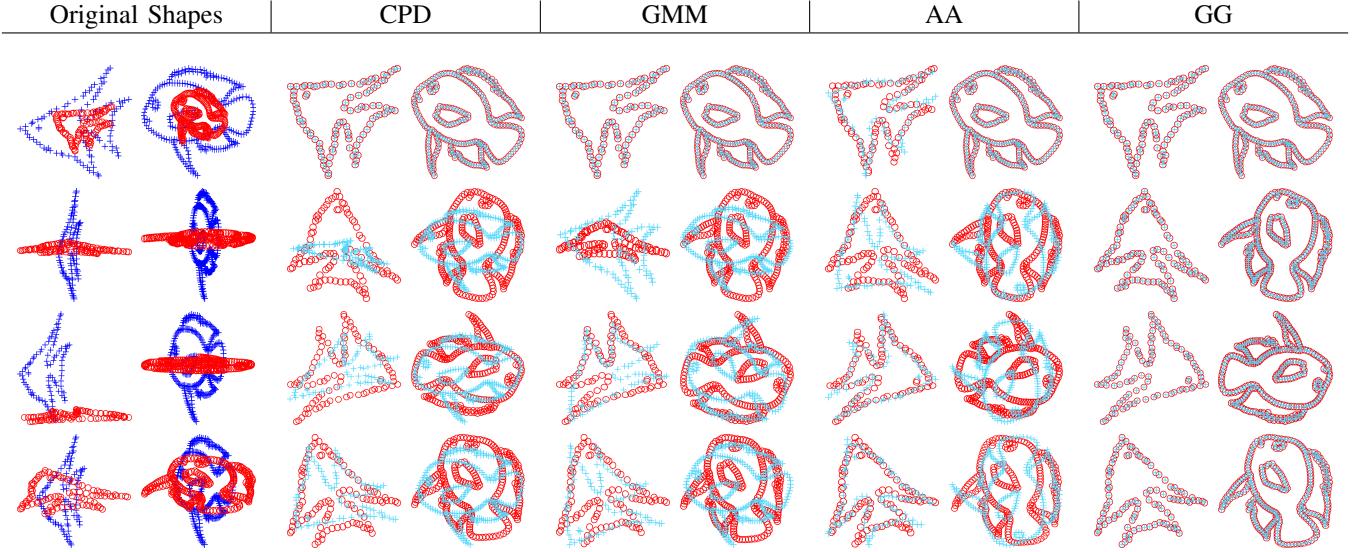


Figure 4. 2D registration results for Coherent Point Drift (CPD) , GMMReg (GMM), Algebraic Affine (AA), and GrassGraphs (GG) algorithms. The column titled “Original Shapes” shows two fishes, the source shapes are in blue and the corresponding affine transformed target shapes are shown in red. The goal is to register the blue (source shape) onto the red (target shape) by recovering the true affine transformation. The images in the remaining columns show the target shape in red and the registered pointset in light blue (The affine transformations shown here only display small translations, due to space constraints). Notice that in the presence of large rotations and uneven scaling, the competing methods break down and are not able to recover the true affine transformation; whereas our GrassGraphs algorithm recovers it perfectly.

sets. First a set of putative correspondences are computed from image features (SIFT, shape contexts, spin image and meshHOG descriptors), where dissimilar feature points are treated as outliers. These putative correspondences are used as an initial vector field which is interpolated in a vector-valued RKHS using Tikhonov regularization. The algorithm simultaneously generates a smoothly interpolated vector field and consensus set by an iterative EM algorithm but struggles in matching far from isometric cases.

Zhou and de la Torre [55] presented the factorized graph matching (FGM) algorithm, in which the affinity matrix is factorized as a Kronecker product of smaller matrices. Although the factorization of affinity matrices makes large graph matching problems more tractable, the method does not use approximations of these affinity matrices thereby making the method very memory intensive. In future, we will investigate the benefit of these factorizations in the GrassGraph framework.

This remainder of this paper is organized as follows: Section III outlines the background on Grassmannians and how they are used to achieve affine invariance. Then we detail the LBO and the intricacies of the GrassGraph framework. Section IV discusses the wide variety of experiments used to evaluate the framework and finally we conclude in Section V with our findings and ideas for future research.

III. GRASSMANNIANS AND AFFINE INVARIANCE

The principal contributions of this work are (i) a Grassmannian representation (GR) of feature vectors and (ii) new affine invariant coordinates in which feature correspondences can be sought. Below, we describe both the intuitive and formal aspects of the new representation.

A. Formulation

Let $X \in \mathbb{R}^{N \times D}$ denote a set of N point features in a D dimensional space (with $D = 2$ or 3). We strip the features of their “identity” choosing to merely represent each one as a location in 2D or in 3D. Later, after projection into the space of new coordinates, feature identity can be restored, aiding in correspondence recovery. The application of an affine transformation on the point set X can be written as

$$\tilde{X} = XA + 1T \quad (1)$$

where $A \in \mathbb{R}^{D \times D}$ and $T \in \mathbb{R}^D$ are the multiplicative and additive aspects of the affine transformation. The matrix A comprises global rotation, scale and shear factors and the vector T contains the global translation factors (while the vector $\mathbf{1} \in \mathbb{R}^N$ is the vector of all ones). The new point set \tilde{X} exists in the same $\mathbb{R}^{N \times D}$ space as X . The above notation can be considerably simplified by moving to a homogeneous coordinate representation. We abuse notation to write

$$\tilde{X} = XA \quad (2)$$

where $X \in \mathbb{R}^{N \times (D+1)}$ has an additional (last) column set to $\mathbf{1}$ and $A \in \mathbb{R}^{(D+1) \times (D+1)}$ now subsumes the translation factors. (The operator A is also constrained to have only $(D+1) \times D$ free parameters.

If two point sets X and \tilde{X} (in the homogeneous representation and of the same cardinality) differ by an affine transformation A , the best least-squares estimate of A is obtained by minimizing the objective function

$$E(A) = \|\tilde{X} - XA\|_F^2, \quad (3)$$

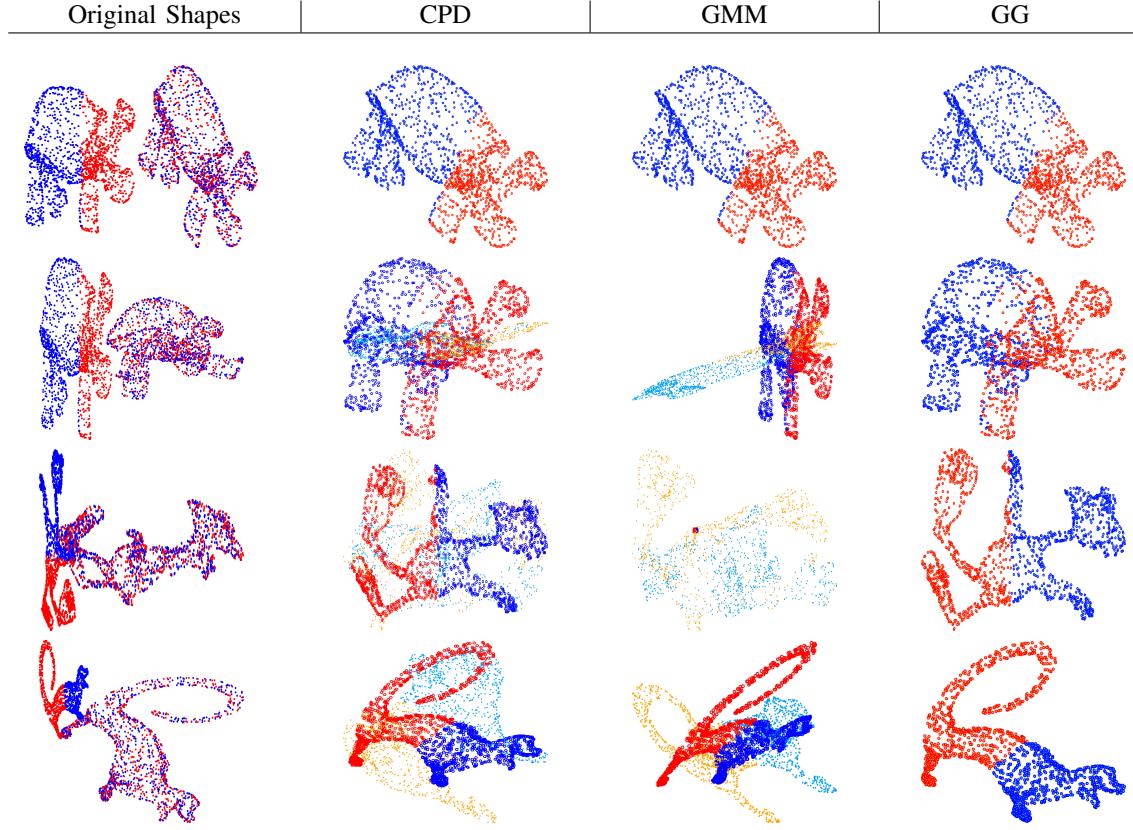


Figure 5. 3D registration results for Coherent Point Drift (CPD) [35], GMMReg (GMM) [20], and GrassGraph (GG) [17] algorithms. The left side of the “Original Shapes” column shows the source shape segmented into a red and blue part. The right side shows the target shape—created by applying an affine transformation to the source shape and then shuffling the points, hence the loss of the segmentation of the source. The remaining columns show the target and registered shapes of the various methods. The target shape is now shown as a segmented red and blue shape with “circles” and the registered shape is shown as a segmented light orange and blue shape with “points”. If the affine transformation is correctly recovered, the light orange and blue points from the registered shape will perfectly align with the red and blue circles of the target shape, respectively (The affine transformations shown here only display small translations, due to space constraints). Notice that in the presence of large rotations and uneven scaling, the competing methods break down and are not able to recover the true affine transformation whereas our method consistently recovers the true transformation.

where $\|\cdot\|_F$ is the Frobenius norm and it is implicitly assumed that the point sets are in correspondence. The solution to eq. (3) is

$$\hat{A} = (X^T X)^{-1} X^T \tilde{X}. \quad (4)$$

In most real-world scenarios, two sets of point features need not be linked by just an affine transformation. The relations can include noise, occlusion, spurious features, unknown correspondence and non-rigid transformations. In this paper, we are mainly concerned with the group of affine transformations and in using equivalence classes of feature sets (under affine transformations) to construct new invariant coordinate representations. Since unknown correspondence is often the key confounding factor in vision applications, we model the relationship between two feature sets with the inclusion of this factor as

$$Y = P X A. \quad (5)$$

In (5), P is a permutation matrix (a square binary valued $N \times N$ matrix with rows and columns summing to one) included to model the loss of correspondence between two sets of features X and Y . While the inclusion of a permutation matrix does not account for occlusion and spurious features,

we show in (numerous) experiments that affine transformation recovery is not adversely hampered provided *strong* correspondences persist.

We now establish the connection to Grassmannians. Ignoring the effect of action by the permutation matrix for the moment, the action of the affine transformation clearly results in a new feature set whose columns are linear combinations of the columns in X . We formalize the relationship between these linear combinations and subspaces.

Lemma 1. Define the set W to be the set of all linear combinations of the columns of X , i.e., $W = \{w | w = \sum_{k=1}^{D+1} \alpha_k x_k, \forall \alpha_k\}$. Then W is a linear subspace of \mathbb{R}^N .

Proof: The $\mathbf{0}$ vector belongs to the subspace since $w = \mathbf{0}$ for $\alpha_k = 0, \forall k$. $u + v \in W$ for any $u, v \in W$ since $u + v = \sum_k \alpha_k x_k + \sum_k \beta_k x_k = \sum_k (\alpha_k + \beta_k) x_k = \sum_k \gamma_k x_k$ for $u = \sum_k \alpha_k x_k$, $v = \sum_k \beta_k x_k$ and $\gamma_k \equiv \alpha_k + \beta_k$. Similarly $c u \in W$ for any $u \in W$, $c \in \mathbb{R}$. Therefore W is a linear subspace of \mathbb{R}^N [25]. ■

Since $\tilde{X} = X A$ (for an affine transformation A) is a linear combination of the columns of X , the columns of \tilde{X} are also in W . If we have multiple point sets $\tilde{X}_j = X A_j$ which differ

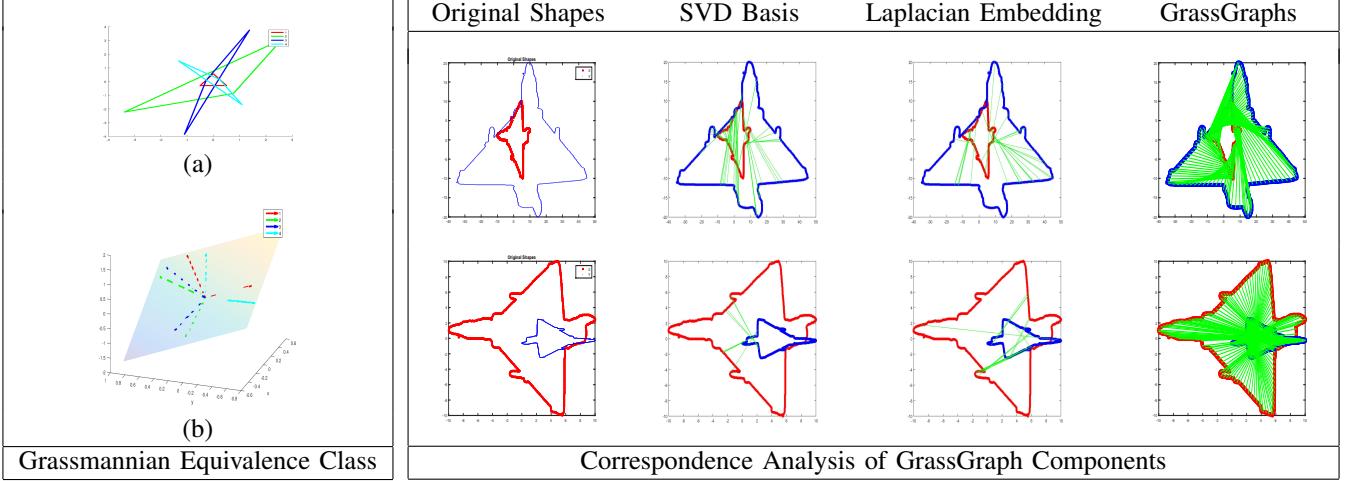


Figure 6. Grassmannian equivalence class and correspondence analysis. On the left, “Grassmannian Equivalence Class”, (a) illustrates a 2D equilateral triangle (in red) and three other triangles obtained by the action of three different affine transformations. Each of these shapes can be represented by a 3×2 matrix since we only need 3 points to define a triangle. In this matrix representation, all four triangles live in the same 2D subspace of the larger 3D space. The resulting U matrix [eq. (7)] of each triangle is a basis for the 2D subspace (the plane shown in the figure), differing only by a possible rotation. The basis vectors and the plane are illustrated in Figure 6(b); the color of the basis vectors corresponds to the color of their originating triangles. The figures on the right, “Correspondence Analysis of GrassGraph Components”, highlight the value of combining the Grassmannian with the Laplacian embedding. The original shapes are shown where Y (blue) is an affine transformed version of X (red). The “SVD Basis” and “Laplacian Embedding” columns show the correspondences retrieved from using only that component coordinate representations. When both components are fused, we recover dense correspondences and are truly affine invariant.

from each other only by affine transformations A_j , then the columns of all the point sets \tilde{X}_j are in $W \subset \mathbb{R}^N$. For a fixed X , the subspace W derived from X can be used to build an *affine invariant equivalence class* of point-sets. Since Grassmannians are the set of all subspaces of \mathbb{R}^N , W picks out *one* element of the Grassmannian.

In the homogeneous representation, $X \in \mathbb{R}^{N \times (D+1)}$. If the columns of X are linearly independent, they constitute a basis for W with dimension $d = D + 1$. Consequently, W is an element of the Grassmannian—a $d = (D + 1)$ dimensional linear subspace of \mathbb{R}^N .

With this understanding of Grassmannians, we make the final link from Grassmannians to their representation as orthogonal projection operators. Since the columns of X are a basis for the subspace W (assuming linear independence of the columns), an orthogonal projection onto this subspace is given by projection matrix $P_X = X(X^T X)^{-1} X^T$ (from standard linear algebra). Hence, there is an immediate identification between the subspace W spanned by the columns of X (a point g_X on the Grassmann manifold) and the projection matrix P_X that projects onto the same subspace. The identification makes the set of all d -dimensional subspaces of \mathbb{R}^N , $\text{Gr}(d, \mathbb{R}^N)$, isomorphic to the set of all $N \times N$ projection matrices \mathcal{P}_N^d that project onto these d -dimensional subspaces, i.e. $\text{Gr}(d, \mathbb{R}^N) \simeq \mathcal{P}_N^d$. Hence, for any $g_X \in \text{Gr}(d, \mathbb{R}^N)$ we can choose to work with the corresponding projection matrix $P_X \in \mathcal{P}_N^d$. Next, the columns of X may not, in general, be an orthogonal basis for the subspace. We can readily obtain an orthogonal basis through the SVD of X , i.e. $X = U_X S_X V_X^T$. Under the U_X orthogonal basis, the projection matrix becomes $Q_X = U_X U_X^T$. In fact, the projection matrices constructed from $X(X^T X)^{-1} X^T$ and $U_X U_X^T$ are identical, i.e. $P_X = Q_X$. This establishes the

rigorous identification of the orthogonal projector Q_X and its interpretation as representation of a Grassmannian point.

We now state the following well known theorem which allows us to move from Grassmannians to orthogonal projection operators.

Theorem 2. [4] Let $\text{Gr}(d, \mathbb{R}^N)$ denote the Grassmannian of d -dimensional subspaces of \mathbb{R}^N . Let $M(N, \mathbb{R})$ denote the space of real $N \times N$ matrices. Consider the set of matrices $G(d, N) \subset M(N, \mathbb{R})$ defined by $Q \in G(d, N)$ if and only if the three conditions are satisfied: (i) Q is a projection operator with $Q^2 = Q$. (ii) Q is symmetric with $Q^T = Q$. (iii) Q has a trace with $\text{trace}(Q) = d$. Then $G(d, N)$ and $\text{Gr}(d, \mathbb{R}^N)$ are homeomorphic, with a correspondence established (since each Q is unique) between each element of the Grassmannian and a corresponding Q .

Theorem 2 establishes the equivalence between each element of $\text{Gr}(d, \mathbb{R}^N)$ and a corresponding orthogonal projection matrix Q . Given a point set X , the theorem implies that we construct an orthogonal projector Q_X which projects vectors into the $d = D + 1$ dimensional subspace spanned by the columns of X , namely W . This can be readily constructed via $Q_X = X(X^T X)^{-1} X^T$ for X and likewise for a point set Y . Provided the relevant matrix inverses exist (and this is guaranteed if the columns of X are independent), two point sets X and Y with both in $\mathbb{R}^{N \times (D+1)}$ project to the same element $G \in \text{Gr}(d, \mathbb{R}^N)$ (of the Grassmannian) if and only if $Q_X = Q_Y$, see Figure 6 for a graphical illustration of these subspace concepts.

We have shown that orthogonal projectors can be used to construct an equivalence class of feature sets with the geometric implication being that different feature sets in the same affine invariant equivalence class project to the same $D + 1$ dimensional subspace of \mathbb{R}^N . Furthermore, orthogonal

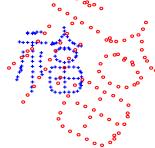
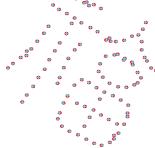
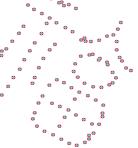
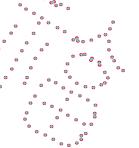
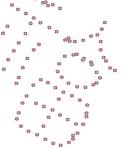
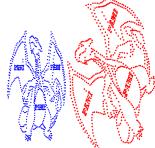
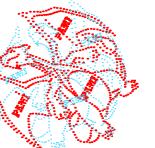
Original Shapes	RPM-L2E	KC	VBPSM	GG
				
Chinese Character (150)	7.18 sec	7.16 sec	12.02 sec	0.07 sec
				
PAMI Dragon (800)	1273.35 sec	43.03 sec	140.97 sec	1.91 sec
				
Octo-robo-fish (800)	586.78 sec	25.70 sec	140.52 sec	1.97 sec
				
Rastafish (800)	2596.87 sec	32.95 sec	136.26 sec	1.76 sec

Figure 7. 2D registration results for state-of-the-art non-rigid methods: RPM-L2E [29], KC [49], VBPSM [37]. The original pointset is shown in blue and its affine transformed version shown in red. The number below each pair of registered image shows the time taken to perform the registration. Since our method (GG) is truly affine invariant we have a relatively constant registration time for pointsets with the same cardinality. Notice in the first row, for transformations with small angles all the methods perform well. However, as the degree of the transformation increases, the methods fail to register the pointsets, especially for larger angles; whereas our method consistently recovers the true transformation.

Algorithm 1 The GrassGraph Algorithm

Input: $X, Y \in \mathbb{R}^{N \times (D+1)}$, as in (5)

Output: Estimated correspondences \hat{P} and affine transformation \hat{A}

1. **SVD:** $X = U_X S_X V_X^T$ and $Y = U_Y S_Y V_Y^T$

2. **Graph Laplacian (GL):**

Retain the top $D + 1$ columns of U_X and $U_Y \rightarrow \hat{U}_X$ and \hat{U}_Y , respectively.*

Build weighted graph L_X from the rows of \hat{U}_X

Build weighted graph L_Y from the rows of \hat{U}_Y

3. **GL Eigenvectors:**

Take top 3 eigenvectors* of L_X (E_X) and L_Y (E_Y)

4. **Estimate P and A :**

\hat{P} : Correspondence using rows of E_X and E_Y^\dagger

\hat{A} : Apply P to X , then $\hat{A} = (X^T X)^{-1} X^T Y$

*Treated as points in \mathbb{R}^{D+1} .

*Can use any k combinations where $2 \leq k \leq N$.

†Simple mutual nearest neighbor assignment works well.

B. Grassmannian Graphs

Orthogonal projectors can be represented using the singular value decomposition (SVD) of X . If $X = U_X S_X V_X^T$, with U_X being $N \times (D+1)$ and S_X, V_X being $(D+1) \times (D+1)$, then $Q_X = U_X U_X^T$. If feature sets X and Y project to identical elements of the Grassmannian, then

$$Q_X = U_X U_X^T = U_Y U_Y^T = Q_Y \quad (6)$$

where U_Y is from the SVD of Y . This suggests that we look for new affine invariant coordinates of X via its SVD decomposition matrix U_X . Unfortunately, this is not straightforward since

$$Q_X = Q_Y \implies U_X U_X^T = U_Y U_Y^T \implies U_X R = U_Y \quad (7)$$

where R is an *unknown* orthonormal matrix in $\mathbb{O}(D+1)$. Or in intuitive terms, orthonormal matrices U_X and U_Y differ by an arbitrary rotation (and reflection). If we seek affine invariant *coordinate* representations, as opposed to affine invariant Grassmannian elements, then we must overcome this rotation problem. In the literature on Grassmannians, the emphasis is usually placed on extensions of Plücker coordinates (in low dimensions) to general Grassmann coordinates [16] with the

projectors are unique. Despite this attractive property, they do have a drawback which we now address.

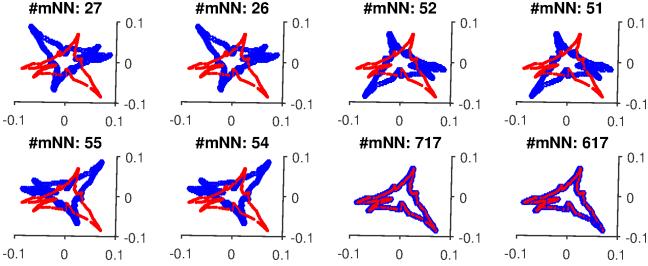


Figure 8. Resolving eigenvector sign-flip discrepancies for the graph Laplacian eigendecomposition phase of the GrassGraph framework (steps 3 & 4 in Algorithm 1). The images show the eight combinations of eigenvector sign flips, the text above each pointset indicates the number of mutual nearest neighbor (mNN) correspondences resulting from the respective flip. The largest mNN score indicates the correct flip combination—this is verified by overlapping eigenvector representations, leading to dense correspondences.

latter constructed from all $(D+1) \times (D+1)$ minors of the rows of X . We do not pursue this route, however, but turn to an alternative approach wherein the arbitrary orthonormal matrix R mentioned above is quotiented out. It is worth pointing out that affine invariant hypergraphs can be constructed from the set of minors followed by higher order eigendecomposition to obtain affine invariant coordinates. We favor a simpler approach as detailed below.

The unknown rotation problem can be overcome by introducing the *Grassmannian graph representation*. In a nutshell, we (after computing the SVD of X) build a rotation invariant weighted graph from the rows of U (treated as points in \mathbb{R}^{D+1}). The Euclidean distances between rows of U are invariant under the action of an arbitrary orthonormal matrix R . Consequently, weighted graphs constructed from U with each entry depending on the Euclidean distance between rows is an affine invariant of X . This Grassmannian graph representation, which we now introduce via the popular Laplace-Beltrami operator approach, is therefore central to the goals of this paper.

The *Laplace-Beltrami operator* (LBO) generalizes the Laplacian of Euclidean spaces to Riemannian manifolds. For computational applications, one has to discretize the LBO which results in a finite dimensional operator. Though several discretization schemes exist, probably the most widely used is the graph Laplacian [12]. The Laplacian matrix of a graph is a symmetric positive semidefinite matrix given as $L = D - K$, where K is the adjacency matrix and D is the diagonal matrix of vertex degrees. The spectral decomposition of the graph Laplacian is given as $Lv = \lambda v$ where λ is an eigenvalue of L with a corresponding eigenvector v . The eigenvalues of the graph Laplacian are non-negative and constitute a discrete set. The spectral properties of L are used to embed the feature points into a lower dimensional space, and gain insight into the geometry of the point configurations [18], [34]. (Note: For the LBO, $\lambda = 0$ is always an eigenvalue for which its corresponding eigenvector is constant and hence discarded in most applications, including ours). Note that in this framework, Gaussian weighted ϵ -neighborhood graphs are constructed and used for the discretization of the graph Laplacian.

We can utilize the LBO to realize the Grassmannian graph's

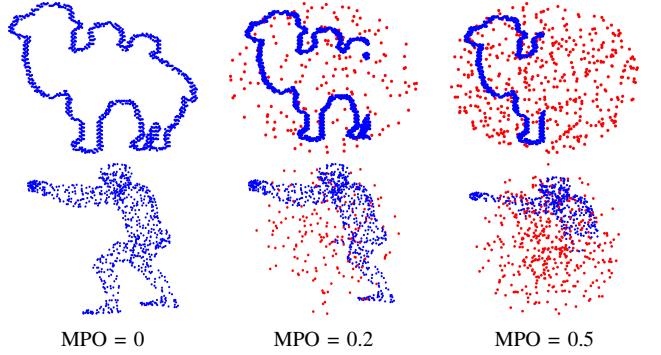


Figure 9. Examples of 2D and 3D shapes with increasing values of missing point/outliers (MPO). The MPO values used in the experiment ranged from 0 to 0.5. (Shapes are randomly subsampled in experiments, ordered sampling is shown for illustration only).

goal to eliminate the arbitrary orthonormal matrix R present in the relationship between U_X and U_Y . To achieve this, we leverage the graph Laplacian approximation described above to construct new coordinates from the Grassmannian graph's $N \times N$ Laplacian matrix by taking its top $(D+1)$ eigenvectors (with the rows of the eigenvectors serving as coordinates). Since the Grassmannian graph is affine invariant, so are its eigenvectors. We can now conduct feature comparisons in this eigenspace to obtain correspondences, clusters and the like. For our present application of affine invariant matching, we recover the correspondences between point configurations X and Y by representing each in the LBO eigenspace, and then use nearest neighbor (kNN) selection to recover the permutation matrix P . The ability to simply use kNN arises from the fact that the affine transformation has been rendered moot in LBO coordinates. Algorithm 1 details the steps in our GrassGraph approach.

More specifically, the correspondence algorithm used is a simple mutual nearest neighbor search. Consider a point set X and its target Y in \mathbb{R}^D . First, X is held fixed and the nearest neighbors in Y are found through the minimum Euclidean distance. Next, Y is held fixed and the nearest neighbors in X are found using the same distance measure. For a pair of points to be in correspondence, they must both be each other's nearest neighbors. This reduces the chances of assigning a single point in Y to many points in X . Though it is possible to incorporate a more sophisticated correspondence techniques like the Hungarian algorithm [23] or its variants [22], we purposefully employ NN to illustrate the power of our affine invariant coordinates. Although we are afforded a simple nearest neighbor search, we pay a computational price due to sign ambiguities of the eigenvectors resulting from the eigendecomposition of the graph Laplacian matrix.

C. Eigenanalysis Sign Ambiguities

As formulated, the GrassGraph algorithm requires two eigendecompositions—one for the SVD to obtain the orthogonal projector and the other to get the eigenvectors of the graph Laplacian. It is well known that numerical procedures for eigenanalysis can introduce arbitrary sign flips on the eigenvectors. Though there have been previous attempts at

addressing the sign ambiguity issue [5], they are commonly considered as application specific or highly unreliable. Hence, the only solution remains to evaluate all possible sign flips, i.e. for k eigenvectors we have 2^k possibilities. In GrassGraph, we have two such decompositions, so one may construe that we require evaluation of $2^{k_1+k_2}$ sign flips, where k_1 is the number of eigenvectors selected ($k_1 = 3$ for 2D point sets and $k_1 = 4$ for 3D point sets), and k_2 is the number of graph Laplacian eigenvectors (typically $k_2 = 3$ for 2D and 3D). It turns out, however, that the graph Laplacian eigendecomposition is invariant to any sign flips induced by the initial SVD. This is due to the fact that in forming the graph we use nearest-neighbor relationships which are determined using the standard Euclidean distance. The argument below details how the distance metric nullifies the sign ambiguity.

Suppose $u = (u_1, u_2, \dots, u_{k_1})$ are coordinates obtained via a numerical eigendecomposition procedure. This introduces the possibility that any coordinate u_i may be sign flipped, i.e. $\pm u_i$. Now consider the calculation of pairwise distances between any two different points $u^{(1)}$ and $u^{(2)}$ in the same coordinate space under the presence of a sign ambiguity:

$$\begin{aligned} d(u^{(1)}, u^{(2)}) &= \sqrt{\sum_{i=1}^{k_1} (\pm u_i^{(1)} - \pm u_i^{(2)})^2} \\ &= \sqrt{\sum_{i=1}^{k_1} (u_i^{(1)} - u_i^{(2)})^2} \quad (8) \end{aligned}$$

Hence, when we are forming the graph using the GR coordinates, we are invariant to sign flips introduced by the SVD and subsequently only have to resolve the sign ambiguity in the eigenvectors of the graph Laplacian. Since GrassGraph only uses three eigenvectors for the spectral coordinates, this is a low order search space that allows us to easily determine the best eigenvector orientation from the set of eight possibilities, shown in Figure 8.

It is important to note that using the number of correspondences from a simple NN search as a flip selection score, would not result in the correct sign flip combination being chosen. This occurs because there exist cases where the embedded pointsets would not align, but still have the most correspondences resulting in flawed affine transformation recovery. We addressed this issue by using a mutual nearest neighbor (NN) approach: a high score only results when maximum overlap occurs, leading to accurate eigenvector sign flip selections. Having addressed the eigenvector sign flipping problem, we now provide comprehensive experimental validation of the proposed GrassGraph method.

IV. EXPERIMENTAL RESULTS

In this section we detail the 2.1 million experiments (2D and 3D combined) used to benchmark the GrassGraph algorithm. The goal was to understand the capabilities of the GrassGraph (GG) approach against other well known registration methods: Coherent Point Drift (CPD) [35]; Registration using Mixtures of Gaussians (GMM) [20]; Algebraic Affine (AA) [17]; Robust L_2E (RPM-L2E) [29], Kernel Correlation (KC) [49], and Variational Bayesian (VBPSM) [37]. Due to longer computational

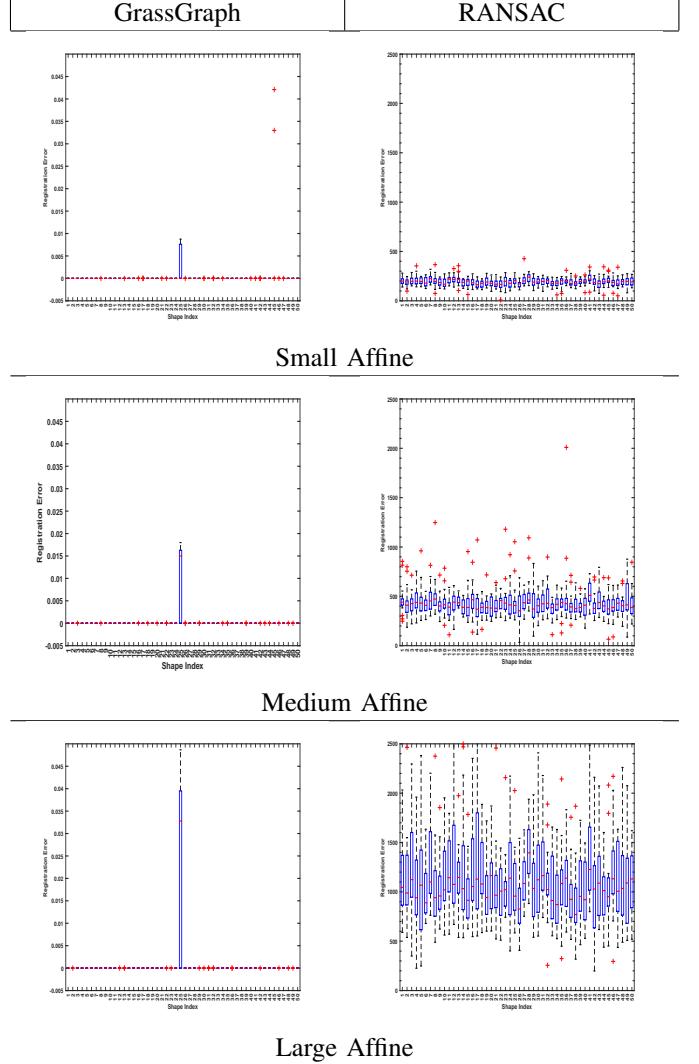


Figure 10. **Affine transformation recovery comparison to RANSAC.** The left column of images shows the results for GrassGraph and the right, RANSAC [48]. The x-axis shows the index of the shapes used to perform the experiments. The y-axis shows “Registration Error” which is the Frobenius norm between the registered target shape and the original target shape. Three levels of affine transformations (small, medium, large; see supplemental material for parameters) were used test the performance of the algorithms. **Note:** The max error of the GrassGraph plots is 0.05 vs RANSAC’s of 2500. No box bars are shown for GrassGraph plots because the registration error was practically zero. However, RANSAC performs progressively worse the larger the affine transformation to be recovered. This empirically shows the GrassGraph is a strong alternative to RANSAC for affine registration.

runtimes of RPM-L2E, KC and VBPSM, we chose to use CPD, GMM and AA in the large-scale experimental trials, and demonstrate the remaining methods visually in Figure 7. The performance of a method was determined by testing the accuracy of the affine transformation matrix recovered in the presence of simulated artifacts: noise and missing points/outliers (MPO). For 2D experiments, CPD, GMM and AA were used but only CPD and GMM were used for 3D. The freely available codes for the competing methods were used to run the experiments. For CPD and GMM, the affine versions of their methods were used to ensure fair evaluations.

In each trial, the target shape was created by applying an affine transformation (see the supplemental material for the

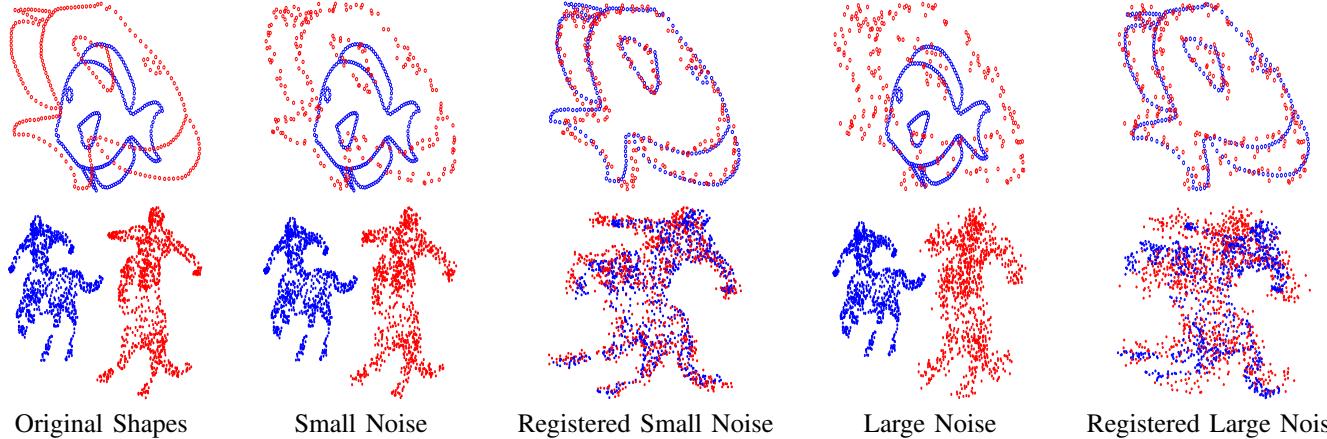


Figure 11. Registration results for 2D and 3D affine related shapes in the presence of small and large levels of noise. The original shape and its affine version are shown in the “Original Shapes” column. The “Small Noise” column on the top row shows the shape and its affine version with a small amount of noise. The corresponding registered shape is shown in the 3rd column. The “Large Noise” column and resulting registration are arranged similarly. It is clear that under large amounts of noise the GrassGraphs method is still able to recover the transformation accurately with just a simple mutual nearest neighbor correspondence algorithm.

affine transformation parameters) to the source shape (referred to as an “affine shape”) with additional artifacts being added depending on the experiment. **Each experiment measured the ability of the various methods to recover the true affine transformation that generated the target shape, with the error metric being the Frobenius norm between the true and recovered affine matrices.** Figure 12 shows more successful examples of the GrassGraph’s correspondence recovery on SIFT features extracted from real image pairs related by a variety of affine transformations. It is clear that our algorithm performs well on unstructured pointsets and can handle large affine transformations effectively.

A. Comparison to RANSAC

In this section we compare the GrassGraph algorithm to the RANSAC algorithm [48] for recovering the underlying affine transformation between two pointsets. The trials were conducted by applying 1000 random affine transformations of varying degrees (small, medium, large) to 50 shapes (with 250 points each) and submitting both the source X and target Y shapes to the algorithms for performance comparison. An increase in affine transformation degree is composed of larger degrees of rotation, scale and translation. The goal was for each algorithm to recover the underlying affine transformation between the pointsets. The recovered affine transformation of each algorithm is applied to the source shape X to obtain a recovered target shape \hat{Y} . Note that the ordering of all target shapes are shuffled, so the algorithms must find the correct correspondences to recover the true underlying affine transformation. The registration error was the frobenius norm between the registered target shape \hat{Y} and the true target shape Y . If the correct affine transformation is recovered, this norm should be close to zero. Figure 10 shows the performance of the two algorithms in recovering the correct affine transformations for the various shapes. RANSAC is one of premiere algorithms for transformation estimation, but for affine transformations without any noise or outliers, it fails to recover the correct affine transformation. The error

increases as the size of the affine transformations increase. Since our algorithm is affine invariant, we have almost no error in recovering the range of affine transformations. See Figure 4 for an example of shapes registered by the GrassGraph algorithm. The RANSAC algorithm was not included in the following large scale evaluations due to its poor performance in recovering the underlying affine transformations.

B. Runtime Complexity

In the GrassGraph framework, our overall time complexity is $O(n^3)$ which comes from the two main computationally intensive components related to steps 1 and 3 in Algorithm 1. Step 1 computes the singular value decomposition (SVD) for an $m \times n$ matrix which has a time complexity of $O(\min\{mn^2, m^2n\})$; the complexity of eigenvalue decomposition (EVD) in step 3 for a $n \times n$ matrix is $O(n^3)$.

In Figure 7, we present the runtime comparison for the shapes shown in Figure 7. The method was implemented in Matlab 2015b with an Intel Xeon E5607 (2.27Ghz) processor, 24.0 GB RAM and Windows 8.1. It is evident that methods based on nonlinear optimization take longer to solve for the transformation on shapes with larger affines. Our GrassGraph algorithm is significantly faster and scales fairly well. Due to affine invariance, the runtimes are constant for a particular number of shape points regardless of the degree of the affine transformation.

C. Data and Artifact Generation

To generate data for the large number of conducted trials, 50 2D and 3D shapes were taken from the established GatorBait [36] and SHREC’12 [27] datasets, respectively. The GatorBait 100 fish database consisted of 100 images of individual fishes from 27 different fish families. The images contain unordered contours of the fish including the body, fins, eyes and other interior parts. The SHREC’12 Track: Sketch-Based 3D Shape Retrieval Contest used a dataset that contained 13 categories with 10 shapes per category in its basic version. Only the 3D

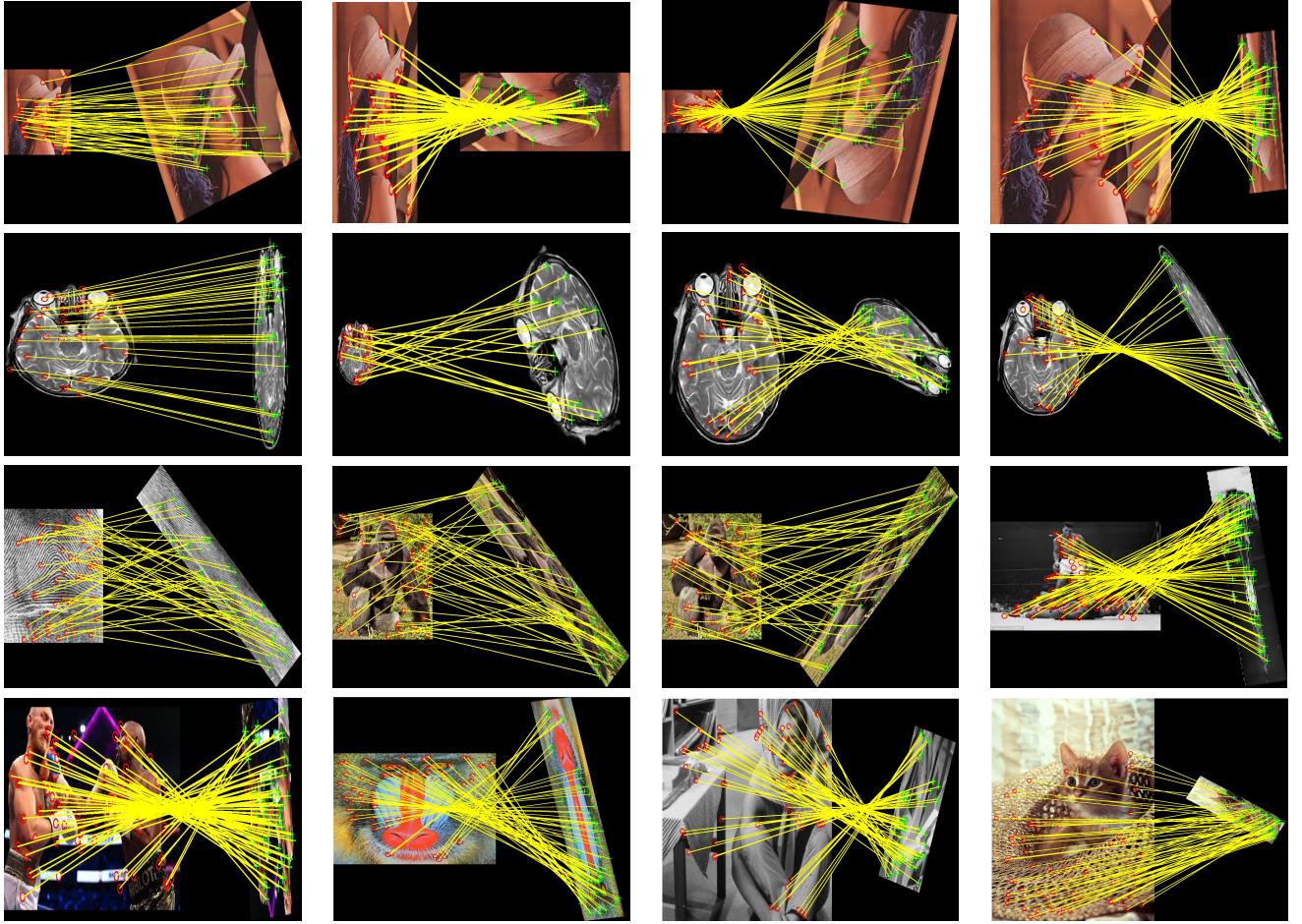


Figure 12. Retrieved correspondences from GrassGraph algorithm on SIFT features of images under various affine deformations. It is evident that GrassGraph is able to find dense correspondences on a variety of unstructured pointsets, despite the degree of affine transformation. (Zoom in to see detailed results)

models were selected as they provided non-similar categories such as ants, planes, and sunglasses.

The number of points in the shapes collected varied between 250 to 10,000 points. The current version of GrassGraph approach requires equal point-set cardinalities between the source and target shapes. Future extensions will incorporate unequal point-set cardinalities and more dataset evaluations. To obtain equal cardinalities, all of the shapes were clustered using the k-means algorithm (using 250 clusters) where the closest point from the shape to a cluster center was used as the new point. The choice to use 250 points was due to the large scale of the experiments conducted; using a higher number of points would have resulted in longer trial times. Given the base shape, we now explain how to generate the various artifacts on the shapes.

As mentioned previously, the two standard artifacts that we applied to the clustered shapes were noise and missing points/outliers (MPO). The process of adding noise and MPO artifacts will be referred to as “noise protocol” and “MPO protocol”, respectively. To generate noise in 2D, we uniformly sampled a new point from a circle of radius r around each point. The uniformly sampled point replaced the original point (center of circle) in the shape—the larger the radius of the circle the more noisy the shape. The same principle is applied

to 3D shapes. Here, we uniformly sampled from a sphere of radius r . To generate a noisy shape for experiments, an affine shape was first created, the points are shuffled to remove the correspondence and then the noise protocol is applied.

In the GrassGraph framework we combine the missing points and outliers into a single artifact. The number of points removed corresponding to the missing point (MP) percentage ($0 \leq mp \leq 0.5$) became the same number of points added as outliers. To create outliers, we uniformly draw samples from a circle in 2D and a sphere in 3D of radius s . To obtain s , the max spread of the points across all coordinate directions was divided by two and multiplied by 1.2. This ensured that the circle or sphere fully encompassed the shape volume, leading to outliers as shown in Figure 9. To generate shapes with MPO for experiments, the artifact was applied to the source shape and then an affine transformation was applied to the artifact shape. The experiments conducted required various affine transformations with the parametrization of these transformations discussed, next.

D. Experimental Trials

The GrassGraph framework evaluation was composed of two main cases: increasing levels of noise and MPO, each with a 2D and 3D component. It is important to point out that

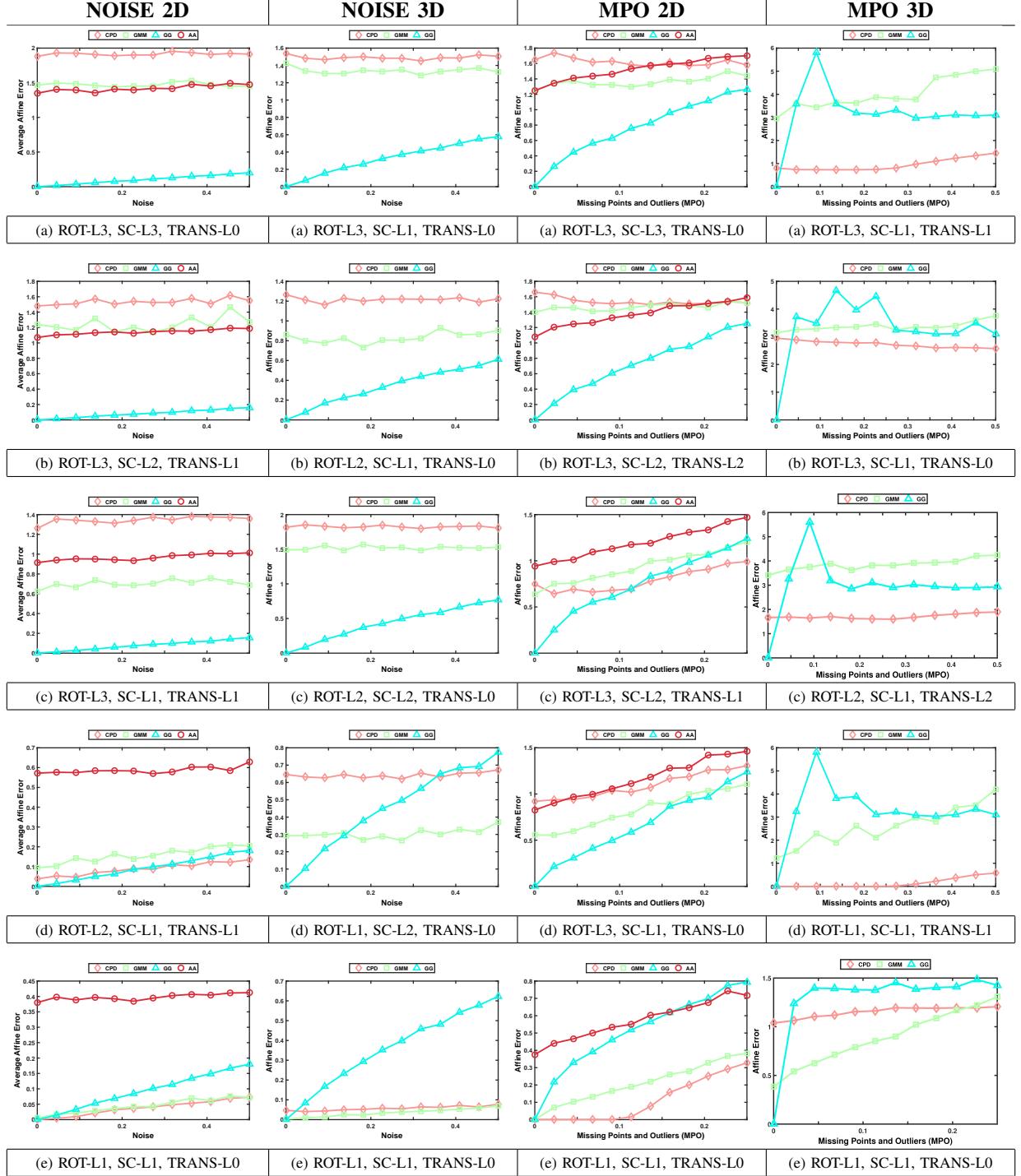


Figure 13. Registration results for 2.1 million experiments in the presence of increasing noise and missing points/outliers (MPO). The text under each image represents the level of Rotation (ROT), Scale (SC) and Translation (TRANS) for levels (L) Large (3), Medium (2), Small (1) and None (0). See Supplemental Material for affine parameter ranges corresponding to the levels. (Zoom in to see detailed results)

our method does not explicitly incorporate any strategies to handle noise or MPO, and only uses NN for correspondence matching. This is in contrast to many of the competing methods we test against, which employ sophisticated optimization schemes, noise/outlier rejection, and richer correspondence matching. Our goal is to illustrate a novel utilitarian, non-iterative GG framework, which, in future extensions, can

readily be modified with similar features.

1) Case 1: Noise: The goal was to investigate the performance of the methods in recovering the true affine transformations under the influence of noise. The noise experiments used 50 base shapes for both 2D and 3D. To perform the experiments, each base shape was transformed with 50 different affine transformations (available in supplemental material) to

form affine shapes. The noise protocol was applied to each of these affine shapes for the twelve values of noise ranging from 0:1. Figure 11 provides examples of noisy shapes in 2D and 3D. All of the affine shapes are evaluated at a single noise level for a particular method: the errors across all the affine shapes are averaged and this error value is assigned to the noise level for that method. For example in Figure 13 top row (a), each marker along a curve represents the average error of the 2500 (50 base shapes \times 50 affines) affine shapes evaluated at that noise level for the method corresponding to the curve’s color. The results of the noise experiments in 2D and 3D are shown in the first two columns of Figure 13. For 2D, across the twelve noise levels with the four methods (GrassGraph and three competitors) we get a total of 600,000 experimental trials and in 3D for the three methods we ran a total of 450,000 trials. Visual registration results are shown in Figure 4.

For the 2D noise experiments, the GrassGraph (blue) method performed the best across (a)-(c). In case (d) we outperform GMM and AA and are on par with CPD. As the size of the angle is decreased in (e) we are only marginally outperformed by CPD and GMM. Once the affine parameter values increased however, the true utility of our invariant method was highlighted as we outperformed the competing methods. CPD seems to be more susceptible to larger angles and scale whereas GMM is affected more by scale. In case (a)-(c), for large affines we see that the competing methods have almost flat curves: this suggests that the correspondences retrieved across all the noise levels were erroneous indicating failing methods. The Algebraic Affine (AA) method seems to perform the worst in comparison to the other competing methods for smaller angles. Our curve increases nicely with noise because we eliminate the affine transformation and deal with finding correspondences on the the noisy points directly. Our simple mutual nearest neighbor scheme applied to the affine invariant coordinates shows good promise for 2D compared to the competing methods that contain optimization schemes and motion constraints.

In the second column of Figure 13 we see that our method outperforms the competing methods for bigger angles in (a)-(c). As the angle becomes small with medium scale, the competing methods have smaller errors; the flat nature of their curves suggest that the methods are not recovering the transformations accurately. As the noise increases, the recovered affine transformations are nearly the same due to inability of the competing methods to separate the noise from the affine transformation. Our curves steadily increase because we only work on the noisy points due to our affine invariance. GMM performs better than CPD in all cases. As we moved up one dimension from 2D to 3D, our method still performs well despite using the simplest of correspondence finding algorithms. This shows that the GrassGraph approach should indeed be the first method considered when recovering correspondences under noisy conditions and for large transformations. Now we look at how our method performs in the presence of missing points and outliers.

2) *Case 2: Missing points/Outliers:* The MPO protocol established above was followed here with the same affine shapes used in the noise experiments, yielding 600,000 trials

for 2D and 450,000 for 3D with the results shown in the last two columns of Figure 13. In 2D, we outperform the competing methods in (a)-(d) and are outperformed by CPD and GMM in (e). Note that the case in (e) has a small angle and translation which means the affine transformation is not hard to recover, for the non-rigid methods CPD and GMM. In cases (a) and (b) the competing methods perform very similarly; however, as the scale, translation and angle are decreased, CPD and GMM begin to outperform AA. So for 2D, our method proves to be a viable approach for affine registration.

Our performance on 3D MPO is slightly different to 2D with the outlier rejection schemes built into the competing methods outperforming us in some cases. In cases (a)-(d) we are on par with GMM at low MPO levels and surpass them at high MPO levels, but not with CPD. The motion coherence constraint in CPD is able to withstand the increasing outliers and occlusion. As the amount of occlusions and outliers increase, our method has a distinct spike at lower levels and tapers off at higher levels. This is due to the mutual nearest neighbor correspondence finding scheme and increase in dimensionality. The extra dimension increases the likelihood that an outlier can be chosen as a mutual nearest neighbor. As the number of outliers begin to increase at small levels the chances of an outlier being classified as a mutual nearest neighbor is high (because a majority of the original points still exists between the point sets with the outliers being in close proximity) which results in bad correspondences and poor transformation recovery. (Recall we are removing points from the shape and replacing them with outliers). When more outliers are present, the chances of two points being mutual nearest neighbors decreases and better correspondences are chosen—resulting in more accurate affine transformation recovery. After a certain amount of outliers, the probability of choosing wrong correspondences decreases at first and then remains the same, hence our error curves flatten out. Although we do not completely outperform the competing methods in 3D, for a simple two stage algorithm without optimization and outlier rejection, we perform quite well compared to the state-of-the-art in non-rigid registration algorithms.

V. CONCLUSIONS

Feature matching is at the heart of many applications in computer vision. For example, image and point registration, object recognition, and shape analysis all rely heavily on robust methods for recovery of correspondences and estimation of geometric transformations between domains. As a core need, a myriad of pioneering efforts and mathematically sophisticated formulations have resulted in a multitude of approaches. However, very few offer the triadic balance of sound theoretical development, and ease of implementation as the proposed GrassGraph framework.

GrassGraph develops true affine invariant coordinates through a two-stage process. First, a Grassmannian representation (GR) is achieved through the use of standard SVD. Second, we approximate the Laplace-Beltrami operator (LBO) on the GR domain, whose eigenspace coordinates then free us from an inherent ambiguity present in the coordinates

extracted from the GR. Within this true affine invariant setting, establishing correspondences reduces to a simple mutual nearest-neighbor selection (though more complex linear assignment solvers can be used in the future). Correspondences in this new space are bijectively related to the original pair of feature points; hence, we are able to directly recover the affine transformation between them. Our method was evaluated on a broad spectrum of test cases, parameter settings, noise corruption levels, occlusions, and included comparisons to numerous state-of-the-art methods. GrassGraph has demonstrated state-of-the-art performance under large affine transformations providing credence to the efficacy of the approach.

Our aim in developing this method (from the theory) was to maintain its simplicity to reinforce its fundamental nature. With such a simple algorithm, we have clearly shown that true affine invariance is possible without the need to stepwise construct invariance by removing individual transformations. This work has hopefully provided a significant, yet extremely simple contribution to the affine invariance literature and should serve as a foundation for research into developing non-rigid invariant methods. One possible future extension to aid correspondence recovery in the LBO space is to combine shape contexts [3] (and SIFT etc.) features with the GrassGraph coordinates, thereby improving the nearest neighbor search in a manner similar to [30].

REFERENCES

- [1] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 13(2):111–122, 1981.
- [2] E. Begelfor and M. Werman. Affine invariance revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2087–2094, 2006.
- [3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 24(4):509–522, 2002.
- [4] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2002.
- [5] T. Caelli and S. Kosinov. An eigenspace projection clustering method for inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:515–519, 2004.
- [6] M. Carcassoni and E. R. Hancock. Correspondence matching with modal clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1609–1615, 2003.
- [7] M. Carcassoni and E. R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193–204, 2003.
- [8] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *European conference on Computer vision*, pages 492–505, 2010.
- [9] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 398–405, 2012.
- [10] H. Chui and A. Rangarajan. A new feature registration framework using mixture models. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000.
- [11] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89:114–141, 2003.
- [12] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [13] A. Egozi, Y. Keller, and H. Guterman. A probabilistic approach to spectral graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):18–27, 2013.
- [14] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.
- [15] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.
- [16] P. Griffiths and J. Harris. *Principles of Algebraic Geometry*. Wiley-Interscience, 1st edition, 1994.
- [17] J. Ho, M. Yang, A. Rangarajan, and B. C. Vemuri. A new affine registration algorithm for matching 2D point sets. In *Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision*, page 25, 2007.
- [18] J. C. Isaacs and R. G. Roberts. Metrics of the Laplace-Beltrami eigenfunctions for 2D shape matching. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3347–3352, 2011.
- [19] V. Jain and H. Zhang. Robust 3D shape correspondence in the spectral domain. In *IEEE International Conference on Shape Modeling and Applications*, pages 19–31, 2006.
- [20] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Transactions on Pattern Analysis Machine Intelligence*, pages 1633–1645, 2011.
- [21] P. W. Jones, M. Maggioni, and R. Schul. Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. *Proceedings of the National Academy of Sciences*, 105(6):1803–1808, 2008.
- [22] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.
- [23] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics*, 2:83–97, 1955.
- [24] Y. Lamdan, J. Schwartz, and H. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions on Robotics and Automation*, 6(5):578–589, 1990.
- [25] S. Lang. *Linear Algebra*. Springer, New York, 3rd edition, 1987.
- [26] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision*, pages 1482–1489, 2005.
- [27] B. Li, T. Schreck, and A. Godil et al. SHREC’12 track: Sketch-based 3D shape retrieval. *Eurographics Workshop on 3D Object Retrieval*, pages 109–118, 2012.
- [28] B. Luo and E. R. Hancock. Alignment and correspondence using singular value decomposition. In *Advances in Pattern Recognition*, volume 1876, pages 226–235, 2000.
- [29] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu. Robust L_2E estimation of transformation for non-rigid registration. *IEEE Transactions on Signal Processing*, 63:1115–1129, 2015.
- [30] J. Ma, J. Zhao, and A. L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, 25:53–64, 2016.
- [31] X. Ma and R. J. Cripps. Shape preserving data reduction for 3D surface points. *Computer-Aided Design*, 43:902–909, 2011.
- [32] D. Mateus, F. Cuzzolin, R. Horaud, and E. Boyer. Articulated shape matching using locally linear embedding and orthogonal alignment. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.
- [33] D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [34] M. Moyou, K. E. Ihou, and A. M. Peter. LBO-Shape densities: A unified framework for 2D and 3D shape classification on the hypersphere of wavelet densities. *Computer Vision and Image Understanding*, 152:142–154, 2016.
- [35] A. Myronenko, X. Song, and M. Á. Carreira-Perpiñán. Non-rigid point set registration: Coherent Point Drift (CPD). In *Advances in Neural Information Processing Systems*, pages 1009–1016, 2006.
- [36] A. M. Peter and A. Rangarajan. Gatorbait 100, 2007. https://www.cise.ufl.edu/~anand/GatorBait_100.tgz.
- [37] H. B. Qu, J. Q. Wang, B. Li, and M. Yu. Probabilistic model for robust affine and non-rigid point set matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–14, 2016.
- [38] J. Qu, L. Gong, and L. Yang. A 3D point matching algorithm for affine registration. *International Journal of Computer Assisted Radiology and Surgery*, 6(2):229–236, 2011.
- [39] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. S. Duncan. A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 4(1):379–398, 1997.
- [40] D. Raviv, A. M. Bronstein, M. M. Bronstein, D. Waisman, N. Sochen, and R. Kimmel. Equi-affine invariant geometry for shape analysis. *Journal of Mathematical Imaging and Vision*, 50:144–163, 2014.

- [41] D. Raviv and R. Kimmel. Affine invariant geometry for non-rigid shapes. *International Journal of Computer Vision*, 111(1):1–11, 2015.
- [42] M. Reuter, F. Wolter, and N. Peinecke. Laplace-Beltrami spectra as ‘Shape-DNA’ of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
- [43] R. M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Eurographics Symposium on Geometry Processing*, pages 225–233, 2007.
- [44] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London Biological Sciences*, 244:21–26, 1991.
- [45] L. S. Shapiro and J. M. Brady. Feature-based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5):283 – 288, 1992.
- [46] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.
- [47] G. K. L. Tam, Z. Cheng, Y. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to non-rigid. *IEEE Transactions Vision Computer Graphics*, 19(7):1199–1217, 2013.
- [48] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78:138–156, 2000.
- [49] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision*, pages 558–569, 2004.
- [50] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 10(5):695–703, 1988.
- [51] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *EuroGraphics: State-of-the-Art Report*, pages 61–82, 2010.
- [52] Z. Wang and H. Xiao. Dimension-free affine shape matching through subspace invariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2487, 2009.
- [53] Z. Wang and Y. Yu. Matching and perturbation theories for affine-invariant shapes using QR factorization with column pivoting. *Journal of Mathematical Imaging and Vision*, 49(3):633–651, 2014.
- [54] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal Computer Vision*, 13:119–152, 1994.
- [55] F. Zhou and F. de la Torre. Factorized graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–134, 2012.
- [56] F. Zhou and F. de la Torre. Deformable graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2922–2929, 2013.



Anand Rangarajan is at the University of Florida. His research interests are machine learning, computer vision and the scientific study of consciousness.



John Corring is a data scientist at Microsoft. He received his Ph.D. in Computer Engineering at the University of Florida in 2017. He has a Master’s of Science degree in Mathematics from the University of Florida. His research focus is on representation, Mathematical analysis, and processing of point clouds, meshes, and images with an emphasis on invariant representations, linear embeddings, and fast and robust registration techniques. John has also worked on filtering and recognition of radar and ultrasound data.



Adrian M. Peter is an Associate Professor of Engineering Systems and Data Science at the Florida Institute of Technology. He has over 10+ years of algorithm development experience, with professional and academic work in areas such as statistical data analysis, mathematical optimization, and probabilistic modeling. He has previously held several positions in industry: Intel Corporation (technology initiatives program manager), Harris Corporation (research engineer), and Northrop Grumman (Chief Scientist of the Advanced Analytics Group in the Information Systems sector). Dr. Peter is the co-inventor on several patents which cover exploitation across a wide variety of sensors (e.g. LiDAR, SAR, and Hyper/Multi-spectral). He is a member of the IEEE and an active reviewer on several journal publications and leading conferences in computer vision and computational reasoning. Dr. Peter’s research interests are in applied machine learning with a focus on geometric information exploitation in a variety of domains, such as signal processing, image analysis and text mining.



Mark Moyou is a data scientist at Alstom USA. He received his Ph.D. and master’s degree in Systems Engineering at Florida Institute of Technology in 2017 and 2012, respectively. Mark served as a research assistant at the Information Characterization and Exploitation (ICE) Lab where he worked on information geometry based algorithms for shape matching, classification and retrieval. He has conducted industry-related research in anomaly detection, concrete crack detection and registration of LiDAR data for remote sensing applications.