

A Grassmannian Graph Approach to Affine Invariant Feature Matching

Supplemental Material

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

November 5, 2017

This supporting document presents additional experimental results to support the main Grassmannian Graph manuscript. Specifically we focus on experiments that evaluate the effects of the two free parameters in our GrassGraphs framework: the ϵ parameter for graph construction and the number of LBO eigenvectors used as an affine invariant coordinate representation. These parameters arise during three main phases of in our approach: (1) obtaining a Grassmannian representation (GR) of the pointset, (2) building an ϵ -graph on the GR, and (3) using the eigenvectors of its Laplace-Beltrami Operator (LBO) as affine invariant coordinates. The final shape correspondences are computed directly on these coordinates.

The remainder of this document is organized as follows. Section 1 looks at how different combinations of eigenvectors affect the registration performance in 2D. Section 2 investigates the same issue of eigenvector combination (as Section 1) but for 3D using different combinations of more than 2 eigenvectors. Section 3 gives examples of the affine invariant coordinate representations (GrassGraph coordinates) for different 2D and 3D affine transformed shapes. The works in the previous sections were all investigated using a single ϵ value. Finally in Section 4 we showcase the effect of the ϵ value on the registration and show how the corresponding eigenvalues change. These

additional experiments illustrate the effectiveness of our GrassGraph framework and we believe will cement it as the first-choice practitioners will turn to for affine invariant coordinate representations and registration.

1 Effect of Eigenvector Combination on 2D Registration

The first rows of Figures 1, 2, and 3 show three different 2D fish shapes. The original shape (source) is shown in red and the affine transformed version (target) is shown in blue: we randomly selected 50 points on the target shape shown as purple asterisks and shifted them to the green circles. This was done to show our method’s ability in recovering the true underlying affine transformation under some point perturbation. Each of the figures previously mentioned correspond to results obtained at a single ϵ value but the number and order of the eigenvectors used are different. Note that the constant eigenvector obtained from the eigendecomposition (corresponding to the smallest eigenvalue) is not used, so for example the eigenvector combination of 1, 2, 3 are actually the 2nd, 3rd and 4th eigenvectors (and please see the main manuscript for more details).

The recovered correspondences are used to estimate the true affine transformation \hat{A} that produced the shapes. This \hat{A} is applied to the source shape and the quality of the registration is determined by how well the affine transformed source shape lines up with the target shape. The reader should infer from Figures 1, 2, and 3 that the order and number of eigenvectors selected does have an effect on the registration. Specifically, the (1,2,3) combination showed the best overall registration performance. We refer the reader to Section 3 of the main manuscript where we detail how to address the sign ambiguity that is inherent to eigendecomposition.

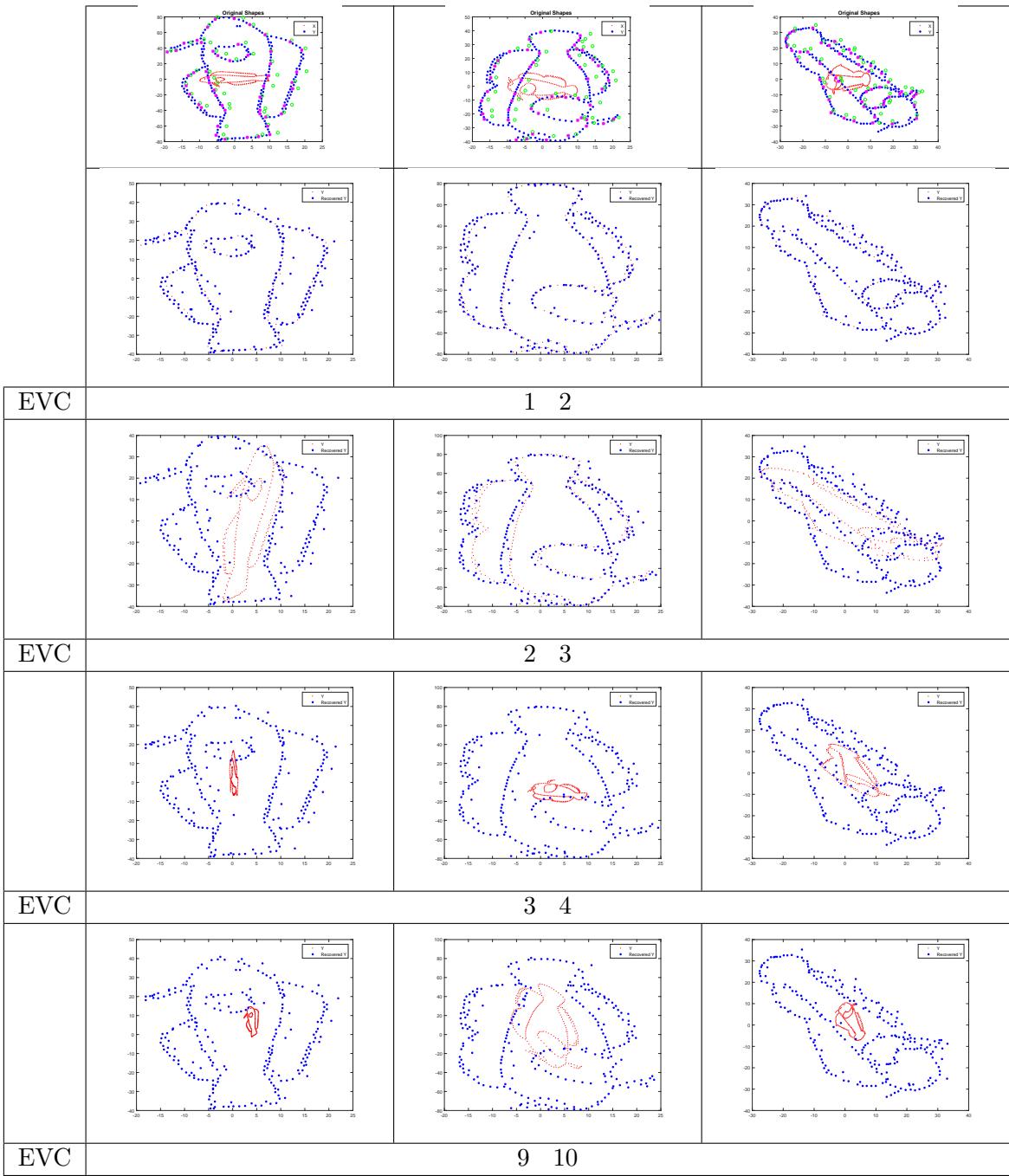


Figure 1: Effect on 2D registration from using only a pair of eigenvectors (EVC refers to eigenvector combination) as affine invariant coordinates. Three different source (red) and target (blue) shapes are shown in the first row. The purple colored points represent the points on the target shape that have been moved to the green circles: this was done to show GrassGraph’s ability to recover the affine transformation under point perturbation. In this scenario, a pair of low ordered eigenvectors are used as affine invariant coordinates—with the pairs used for the shapes given below the images. Notice that as the order of the eigenvector increases, the registration performance worsens. In many numerous empirical evaluations, the first two low ordered eigenvectors (1,2) consistently produced the best registration results.

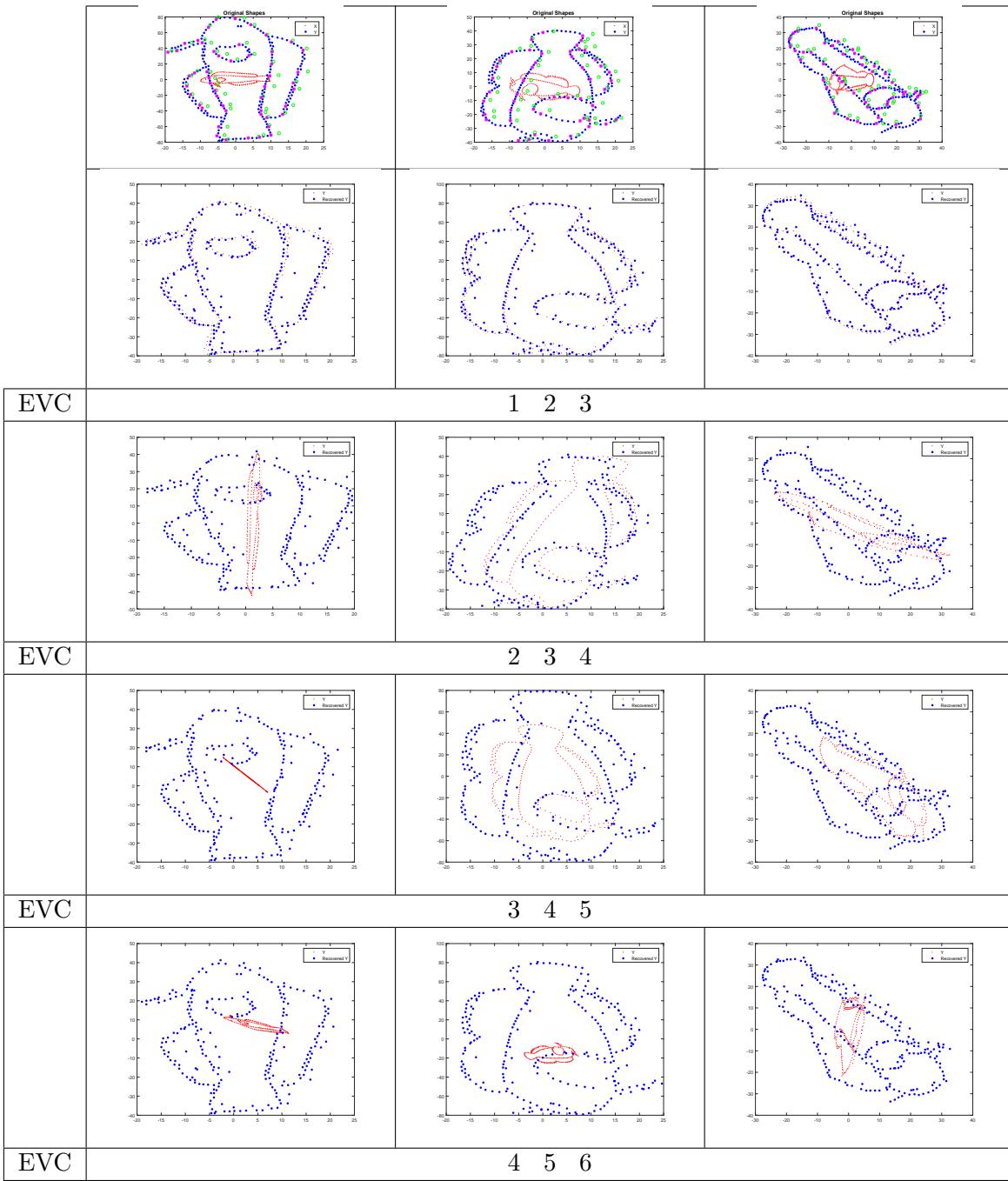


Figure 2: Effect on 2D registration from using a triplet of eigenvectors (EVC again refers to eigen-vector combination) as affine invariant coordinates. Three different source (red) and target (blue) shapes are shown in the first row. The purple colored points represent the points on the target shape that have been moved to the green circles: this was done to show GrassGraph’s ability to recover the affine transformation under point perturbation. In this scenario, a triplet of low ordered eigenvectors are used as affine invariant coordinates—with the triplet used for the shapes given below the images. Notice that as the order of the eigenvector increases, the registration performance worsens. As supported by over 2 million trials in our main manuscript, the first three low ordered eigenvectors (1,2,3) consistently produced the best results.

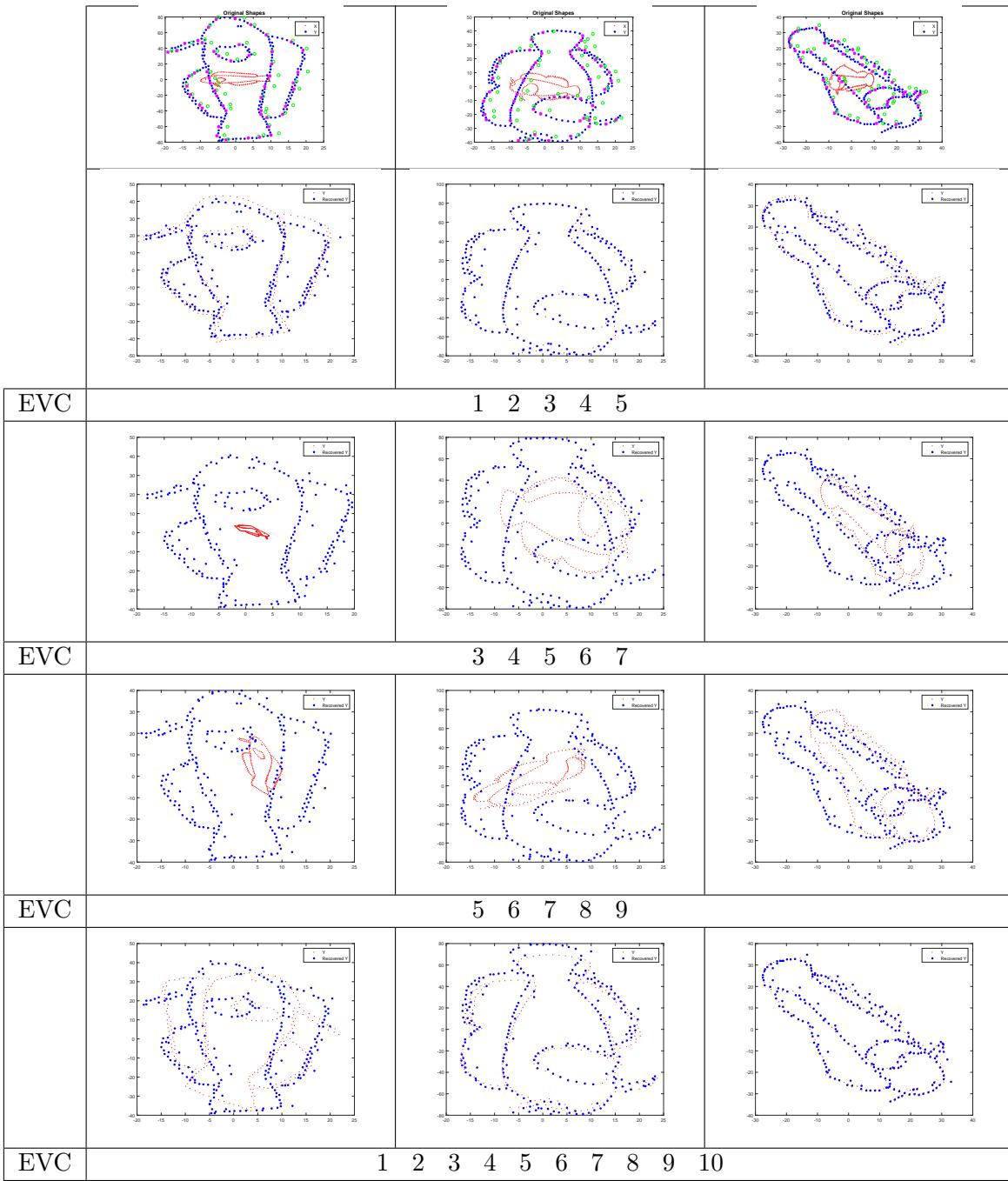


Figure 3: Effect on 2D registration from using 5+ eigenvectors (EVC refers to eigenvector combination) as affine invariant coordinates. Three different source (red) and target (blue) shapes are shown in the first row. The purple colored points represent the points on the target shape that have been moved to the green circles: this was done to show GrassGraph’s ability to recover the affine transformation under point perturbation. In this scenario, multiple low ordered eigenvectors are used as affine invariant coordinates—with the eigenvectors used for the shapes given below the images. Notice that as the order of the eigenvector increases, the registration performance worsens provided the 1st and 2nd eigenvectors are not included. There is no distinct advantage in including more eigenvectors into the invariant coordinate representation: it only adds unnecessary computational time. This justifies the use of at least the first three low ordered eigenvectors (1,2,3).

2 Effect of Eigenvector Combination on 3D Registration

In Figure 2, we performed a set of 3D registration experiments similar to the ones in Section 1 of the main manuscript. Here the purpose was to determine which eigenvector combination would perform the best for 3D registration. We can see that the 1, 2, 3 combination yields the best registration results. These experiments justify the use of the 1, 2, 3 EVC for both the 2D and 3D registration cases.

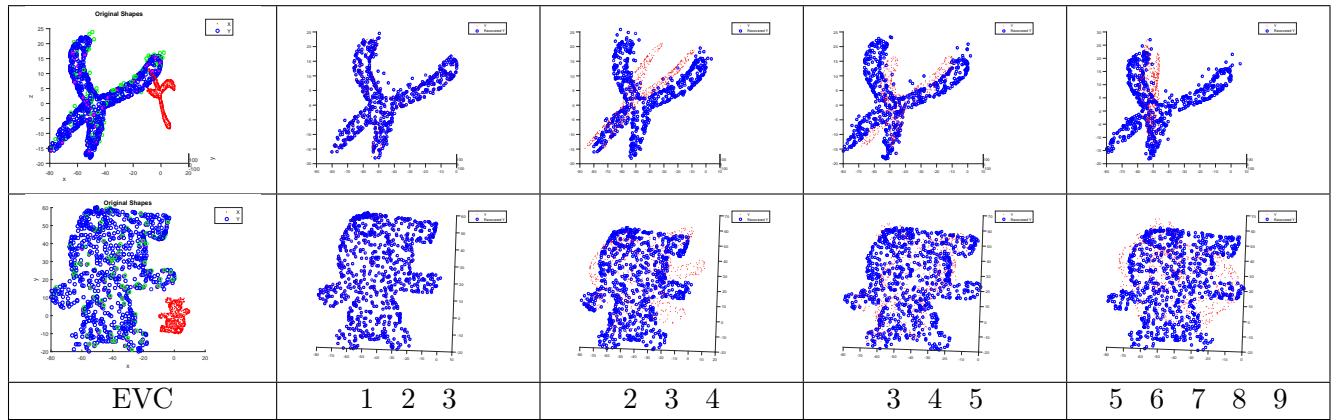


Figure 4: Effect on 3D registration from using 3+ eigenvectors (EVC refers to eigenvector combination) as affine invariant coordinates. Three different source (red) and target (blue) shapes are shown in the first row. The purple colored points represent the points on the target shape that have been moved to the green circles: this was done to show GrassGraph’s ability to recover the affine transformation under point perturbation. In this scenario, multiple low ordered eigenvectors are used as affine invariant coordinates. The eigenvector combinations used for the shapes are given below the images. Notice that as the order of the eigenvector increases, the registration performance worsens provided the 1st eigenvector is not included. It is clear that the 1, 2, 3 eigenvector combination yields the best representation for registration.

3 Eigenvector Representations

Since the GrassGraph coordinates (GGCs) are affine invariant, given two affinely related shapes one would expect to see the same GGC representation for each shape. This is demonstrated in Figure 5, where the GGC for two different 2D and 3D shapes under different affine transformations are shown. From the previous figures, it is clear that the first three eigenvectors are a suitable representation of affine invariant coordinates.

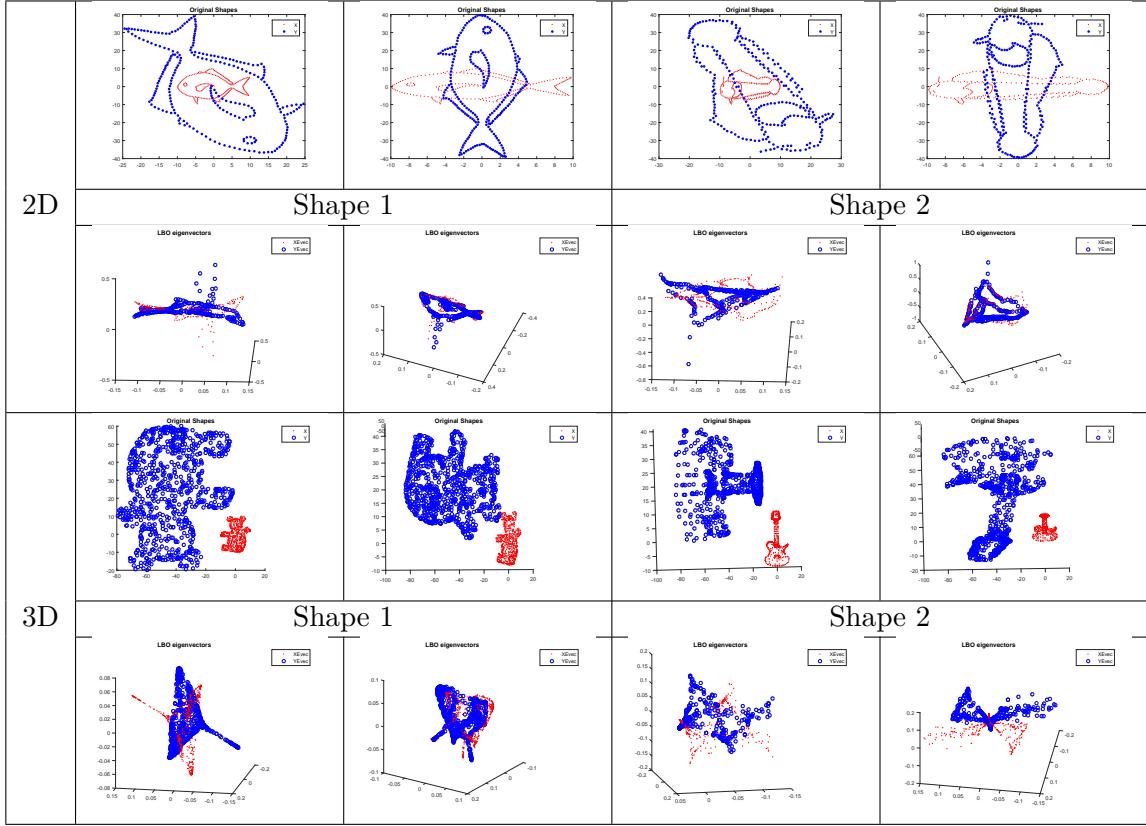


Figure 5: GrassGraph coordinate representations of 2D and 3D shapes. The source and target shape are shown in red and blue, respectively. There are two 2D and 3D shapes, each with two different affine transformations. Beneath each shape image, the GrassGraph coordinate (GGC) representation of the shape is displayed. It is evident that for the same shape, regardless of the affine transformation, the same GGC representation is obtained. Since these coordinates are obtained from an eigendecomposition, there exists a sign flip ambiguity that must be resolved. This explains why the GGCs look similar but off by some sign flip. Once the sign flip is resolved (Section 3 of the main manuscript) we get the same GGC for the source and target shape allowing for accurate correspondence recovery. All coordinates use the (1,2,3) EVC.

4 Effect of ϵ on Eigenvalues and Registration

In this section, we illustrate some 2D/3D registration results for several values of ϵ used in the graph construction. The corresponding first ten eigenvalues of the GrassGraph coordinates of the source and target shape are shown in Figure 6. It is clear from the plots that the value of ϵ influences registration performance. The spectrum of the LBO is also plotted to illustrate that much of the “information” is captured in the eigenvectors corresponding to the first three eigenvalues. When

the appropriate ϵ value is selected, we get good registration performance and the LBO spectrum reflects importance of the low order eigenvectors—validating our use of the first three eigenvectors as invariant coordinates.

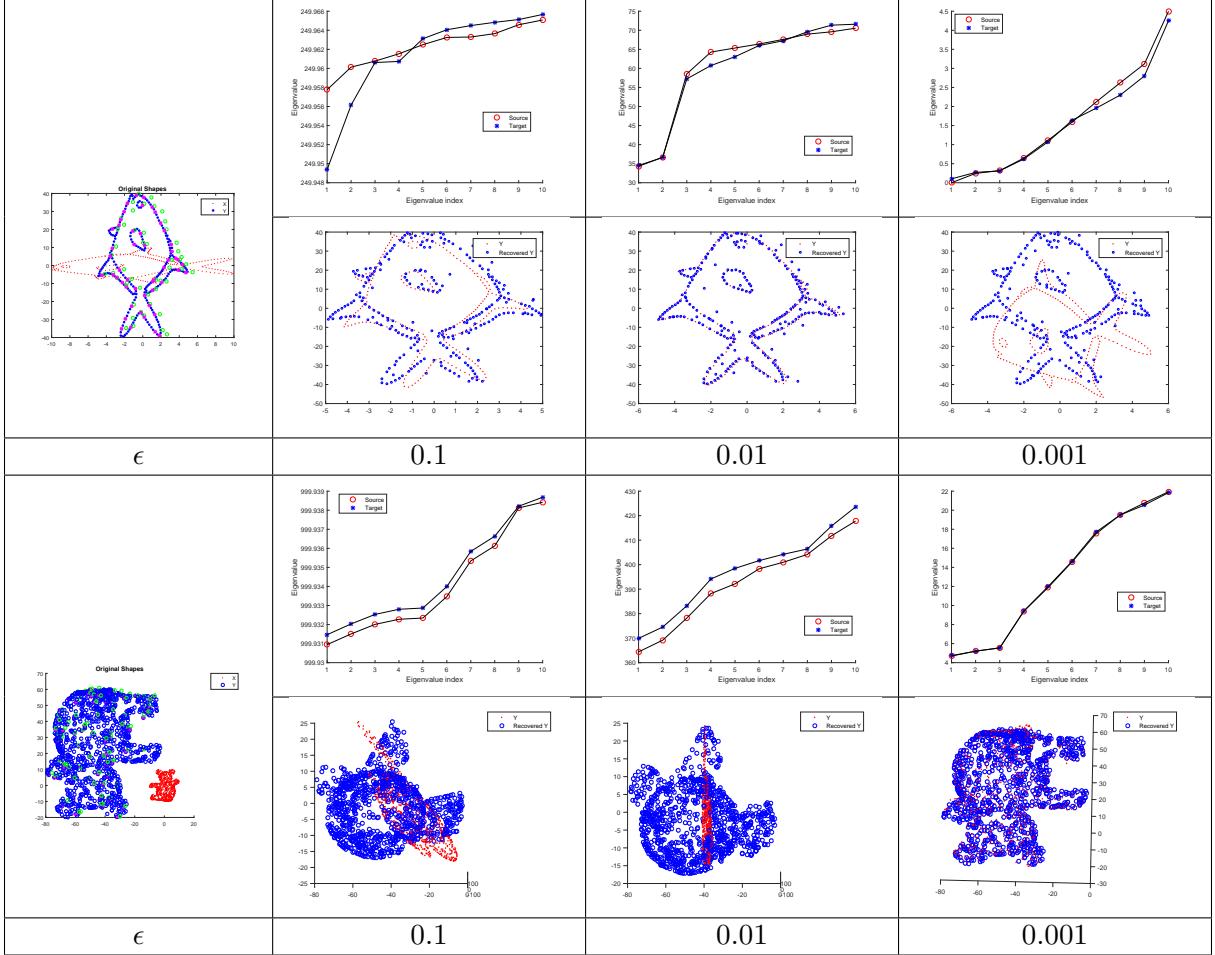


Figure 6: Effect of ϵ on eigenvalues and registration of 2D and 3D shapes. The source and target shape are shown in red and blue, respectively. For a given value of ϵ , the first 10 eigenvalues from the eigendecomposition of the LBO are shown; beneath each eigenvalue plot the corresponding registered source shape is shown. The (1,2,3) EVC was used to register all the shapes. For the 2D case, an ϵ value of 0.01, and for 3D an ϵ value of 0.001 produced the best registration results. Notice that their first three corresponding eigenvalues are essentially equal, which means the eigenvector GrassGraph coordinates, even under point perturbations, are roughly the same (as desired).

5 Related Work Continued

Space limitations in the main manuscript precluded us from including several other related works that have positively impacted the shape registration. In what follows, we detail these works and provide brief analyses with regards to our proposed GrassGraphs approach.

In [6], Ma *et al.* address non-rigid pointset registration. Their method requires an initial correspondence set from another registration algorithm and uses an EM based algorithm to remove outliers. Hence, their method would benefit greatly by running our proposed Grassmannian Graphs (GG) approach first to obtain a good, initial set of correspondences. Since our GG approach is able to produce affine invariant coordinates, we are able to estimate and remove large-scale deformations which are known to cause problems with non-rigid techniques. Also, their iterative, numerical optimization scheme for the non-rigid objective is quite a bit more complicated than ours. We are able to provide a very robust solution to the affine invariance problem without the need for an iterative optimization. We simply use an SVD, followed by an elementary nearest neighbor rule for correspondence recovery. We contend that it is the strong theoretical (Grassmannian) footing coupled with ease of implementation that makes our method appealing to most computer vision practitioners. Their work also explicitly address outlier rejection and noise stability. Even though our experiments demonstrate GG to be fairly robust to noise without an explicit robustness measure, we can readily incorporate this in future extensions.

Lovell *et al.* [4] use the Grassmannian manifold to solve a image classification problem. They use the Grassmannian manifold to represent and compare sets of images—the distance on the manifold is used to determine similarity between the sets of images. Our method does not utilize the Grassmannian distance. Instead, we use its subspace equivalence class properties to construct affine invariant coordinates for registration problems Both the authors and our work use Grassmannians in interesting ways, although in for unrelated applications.

The authors in [5] proposed a method to solve the correspondence problem by finding the most similar subgraphs within a graph constructed on the pointsets. Correspondence between two nodes is determined by solving for a scale parameter between graph edges using an EM based approach.

The main difference with our GG framework is that this method does not address affine invariance at all and instead leverages spectral characteristics of graphs. Also, they have an iterative algorithm unlike ours. However, their subgraph finding method could be incorporated in future extensions of our work to help overcome sensitivity to outliers.

Boscaini *et al.* [1] developed a shape feature descriptor based on anisotropic oriented learnable diffusion filters. The main similarity to this work is the use of the spectral properties of the Laplace-Beltrami Operator (LBO). In our method we use the eigenvectors of the LBO obtained from the graph Laplacian of the pointsets. Boscaini *et al.*'s work uses eigenvectors and eigenvalues of the LBO in a deep learning framework to learn direction invariant shape descriptors which are used to find correspondences on similar shapes. However, their approach automatically assumes there is no global affine transformation between a pair of shapes. For example, if a large shearing transform existed between two shapes, their approach would not produce good descriptors for matching. Our proposed GG technique directly addresses this problem.

Raviv and Kimmel's work [7] work creates a distance metric on the surface of the mesh that persists under affine transformations. The main difference is Raviv's work focuses on the surface geometry built on the triangulated mesh whereas we work directly with pointsets. Our entire representation is affine invariant, but his 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, and considerably more complicated than GG. Both approaches utilize the LBO, but are fundamentally different in their use and construction.

Cho *et al.* [2] focused on finding the best active graphs from maximal graphs. A maximal graph was built on all feature points, whereas the active graph was the best sub-graph in each feature set. By using a Bayesian approach and an initial set of correspondences, their graph matching procedure finds the best correspondences in the active graphs. The graph progression updates the active graphs and their similarity matrix to boost the matching score. This approach is a generalized graph matching approach and does not claim any specific invariance properties and is iterative, whereas we address affine invariance without an iterative scheme.

In [3], Rezatofighi *et al.* solve the matching problem by incorporating multiple hypotheses into their similarity measure. They propose a generic approach to efficiently approximate the marginal distributions by exploiting the m -best solutions of the original matching. The method is hindered by the growing number of m constraints as the number of iterations increases and also by not being generalizable to arbitrary problems. The work is definitely a step in the right direction of incorporating more constraints into the matching problem. They do not address the construction of affine invariant coordinates as our GG method and estimation of large-scale deformations.

6 Correspondence Recovery on Large Cardinality Pointsets

We provide further empirical evidence in support of the GrassGraph framework by examining large cardinality pointsets. Below, the first figure shows a series of images that demonstrate our method is able to register large cardinality pointsets (greater than 1500 points) and recover correct correspondences. The second gives a comparison to the experiments conducted in other state-of-the-art methods. Figures 7, 8 and 9 showcase our approach on two large cardinality pointsets under different affine transformations. From the results, it is evident that our method is able to recover correspondences and the true underlying affine transformation between the pointsets.

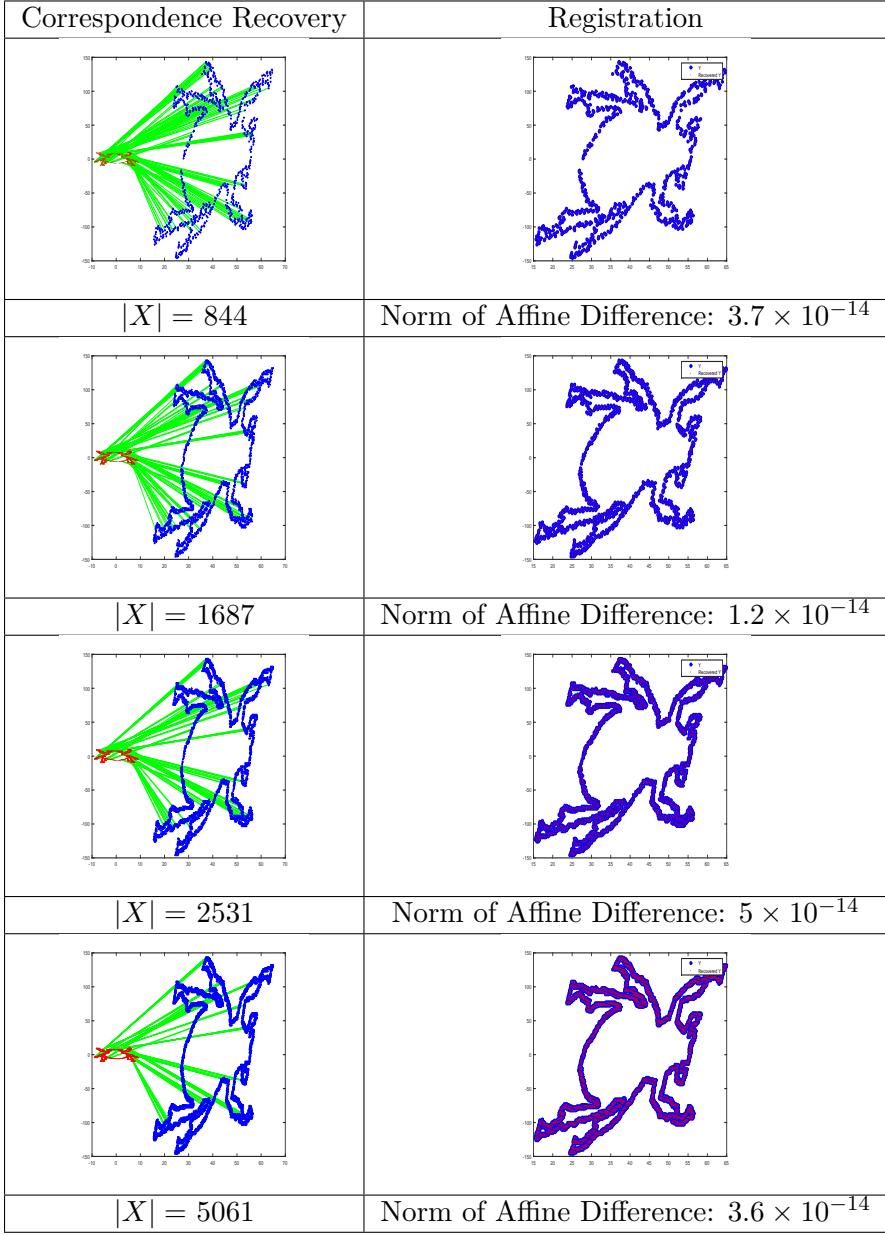


Figure 7: Correspondence recovery and registration performance for pointsets of large cardinality. The left column ('Correspondence Recovery') shows a subset of correspondences; as the rows increase so does the cardinality. Note that we are able to recover correspondences regardless of the cardinality. The 'Registration' column shows the pointset with the recovered affine applied to the source shape. The number below each plot is the Frobenius norm of the difference between the true affine matrix and the recovered one. It is clear that our method is able to recover the true underlying affine transformation regardless of the cardinality.

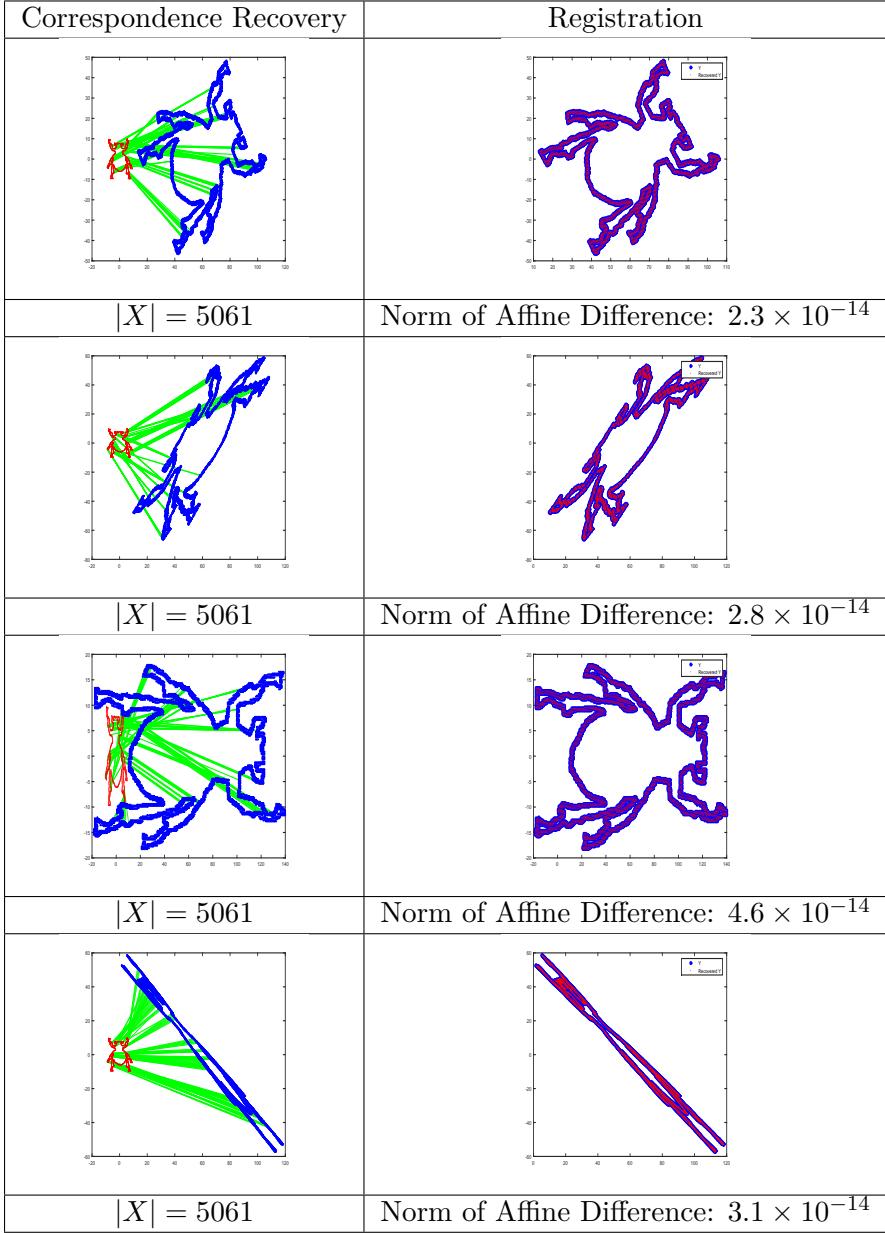


Figure 8: Correspondence recovery and registration performance for pointsets of large cardinality under different affines. The left column ('Correspondence Recovery') shows a subset of correspondences; each row contains a unique affine transformation. Note that we are able to recover correspondences regardless of the cardinality and affine transformation, showcasing the utility of our invariant framework. The 'Registration' column shows the pointset with the recovered affine applied to the source shape. The number below each plot is the Frobenius norm of the difference between the true affine matrix and the recovered one. It is clear that our method is able to recover the true underlying affine transformation with large cardinality pointsets under a wide range of affine transformations.

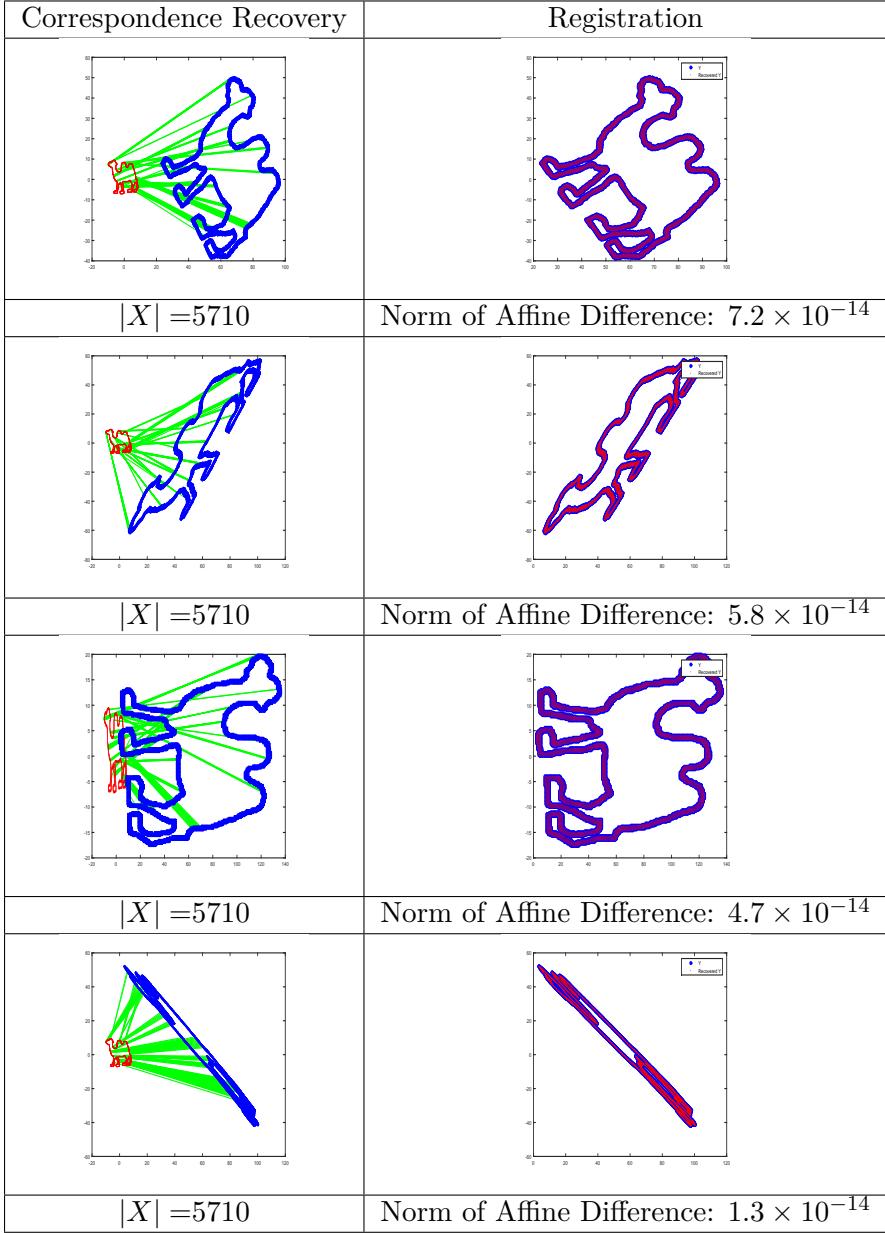


Figure 9: Correspondence recovery and registration performance for different pointsets of large cardinality under different affines. The left column ('Correspondence Recovery') shows a subset of correspondences; each row contains a unique affine transformation but a different shape to Figure 8. Note that we are able to recover correspondences regardless of the cardinality, affine transformation and pointset, showcasing the utility of our invariant framework. The 'Registration' column shows the pointset with the recovered affine applied to the source shape. The number below each plot is the Frobenius norm of the difference between the true affine matrix and the recovered one. It is clear that our method is able to recover the true underlying affine transformation with large cardinality pointsets under a wide range of affine transformations and pointsets.

7 Unequal Cardinality Analysis for GrassGraph Algorithm

As mentioned above, at this current phase in our research, due to our use of the Grassmannian manifold as a building block for affine invariance we inherit an equal pointset cardinality requirement on our matching. We support the AE’s suggestion that altering this requirement is not a trivial process. The following exposition focuses on the open research problem of using the Grassmannian representation (GR) for unequal cardinality affine invariant pointset registration. We will detail the differences, challenges and opportunities for solving this unequal cardinality requirement. The remainder of this section will be in the following order: recap of current affine invariance technique; analysis in the Grassmannian space with equal pointset cardinality under affine transformations; analysis in the Grassmannian space with evenly subsampled pointsets (unequal pointset cardinality) without affine transformations; analysis in the Grassmannian space with evenly subsampled pointsets under affine transformations, analysis in the Grassmannian space with uneven sampling without affine transformations and finally analysis in the Grassmannian space with uneven sampling under affine transformations.

Recap of Current Affine Invariance Framework

In the current phase of our research, we leverage two main components—the Grassmannian representation and the Laplace-Beltrami Operator’s eigenvectors—to obtain our affine invariant coordinate representation. First, the pointsets to be matched are mapped to the Grassmannian coordinate representation (GR). If the GR coordinate representations were truly affine invariant on their own, any affine related pointsets would be represented as the same pointset in this space. However, in the GR coordinate representation, any arbitrary affine transformation applied on the pointsets is reduced to an arbitrary rotation. To overcome this rotation we turn to a graph representation. More specifically, an epsilon graph is constructed on the pointsets in this space. Using this graph we compute its Graph Laplacian, find its associated Laplacian embedding and use this as an affine invariant coordinate representation. Although we inherited some of the well-known artifacts from Laplacian embeddings such as eigenvector sign-flips, we have not found it to be detrimental to our

current approach. In Figure 10, we see the main components of our affine invariance approach. The top row shows an aircraft pointset under four different affine transformations. The second row shows the top row’s associated pointsets under the GR. The third row shows the second row’s associated Laplacian embedding, where sign-flip resolution is used to align pointsets.

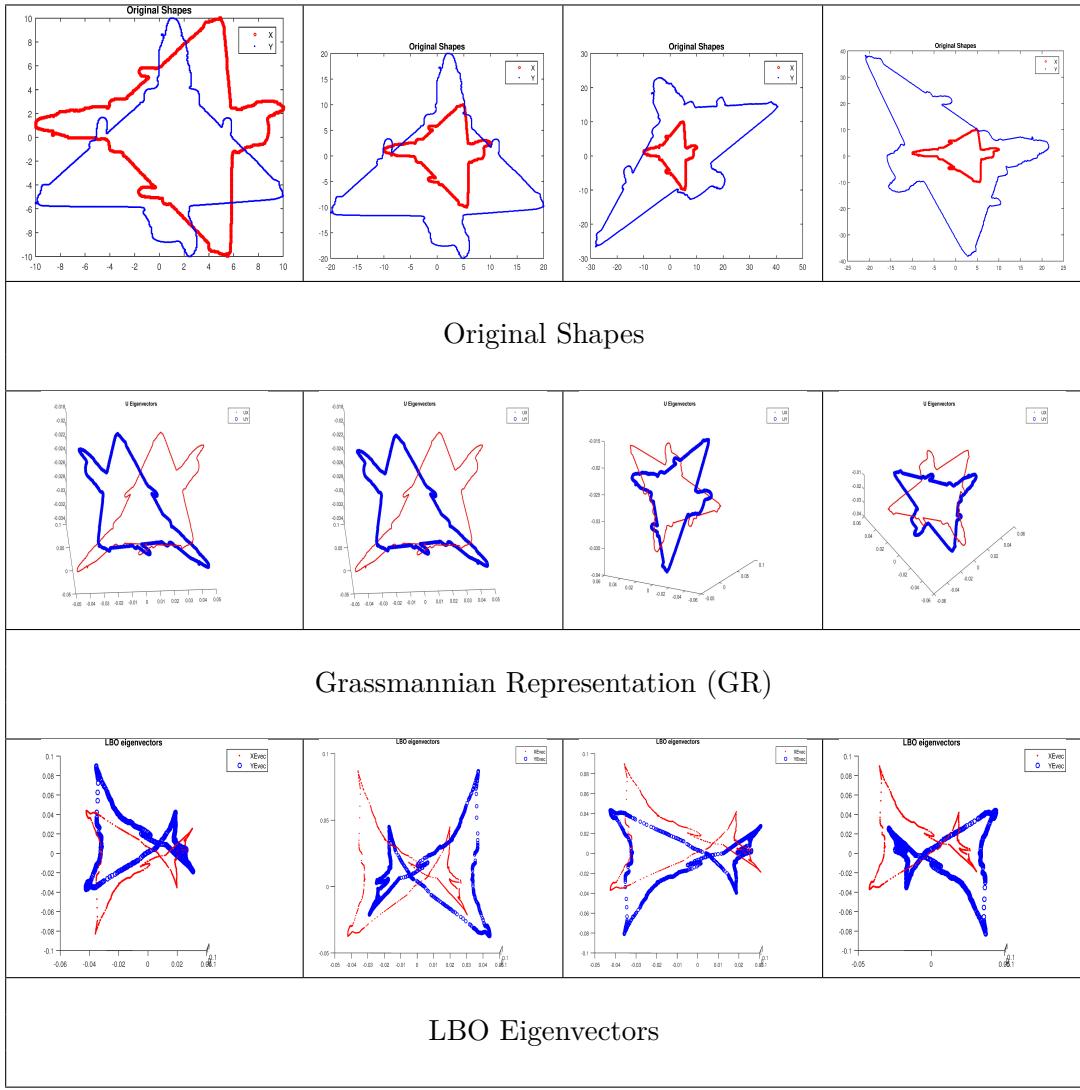


Figure 10: Equal pointset cardinality analysis of the GrassGraph framework. The first row shows four affine transformed versions of a 2D aircraft shape. The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). Note the large degrees of rotation and scales shown (translations are not shown due to space constraints). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main conclusion to draw is that any arbitrary affine transformation between pointsets in the original space is reduced to an arbitrary rotation between the pointsets in the Grassmannian space. One aspect to pay attention to is the scale of the pointsets under this GR—both pointsets have equal scale and no translation. The third row shows the Laplacian embedding of the shape above it. The displayed eigenvectors are off by a sign-flip which we are able to overcome to recover dense correspondences on the original pointsets. Observe that there are no scale differences between the Laplacian embeddings of the pointsets (this will be different for the case with unequal cardinality pointsets).

Analysis of Pointsets in the GR with Equal Cardinality Under Affine Transformations

Our current technique fully addresses this scenario. The purpose of this section is to re-highlight the aspects of the method that change as we move to addressing the unequal cardinality pointset matching case. Our method currently has two main parameters: the epsilon value used to create the graphs in the GR, and the number of LBO eigenvectors used to perform correspondence matching. Currently, we have found that using a triplet of low-ordered LBO eigenvectors yields the best correspondence recovery. This may be considered a fixed parameter, hence our method can be reduced to having one free parameter: epsilon (ϵ). In the unequal cardinality scenarios below, we see very distinct characteristics of the pointsets under the GR. Any effect in the pointsets in the GR has a direct effect on the LBO eigenvectors, hence an effect on the overall matching. Resolving these issues is not trivial but we have made progress here by identifying the exact problem that needs to be addressed for the method to be extended.

Analysis of Evenly Subsampled Pointsets (Unequal Cardinality) in the GR without Affine Transformations

In this scenario, we show the response of the main components of our framework to unequal cardinality (even subsampling) pointsets without affine transformations. The pointsets shown in the first row of Figure 11 and 12 are ordered. The transformed shape Y was created by downsampling X evenly across the entire shape. This ensured that the overall shape of the pointset was preserved with a smaller number of points. The goal was to investigate the effect of this downsampling on the GR and LBO eigenvector representations. Recall, from Figure 10 that affine transformations in the original space were mapped to arbitrary rotations in the GR. In this scenario, we have a pointset and its evenly subsampled version without any affine transformations. Note the second row of Figure 11. In the previous section, in the GR we saw pointsets having the same scale but different rotations. In the GR of this case we see a distinct scale and translation difference between the pointsets. Keep in mind that the trivial counter case is to have the same pointset with no affine transformation, which would result in the exact same pointset in the GR (not shown in images).

The pointsets in the GR still retain the overall shape of the original pointsets but their scalings are different. As mentioned before, this increase in scale in the GR has a direct effect on the LBO eigenvectors as shown in the third rows of Figures 11 and 12. Recall, that our approach (see Section 7) essentially has a single free parameter, the choice of epsilon of the GR graph. In order to mimic our current framework, the scale of the pointsets in the LBO space need to be equal for our correspondence algorithm to find the best matches. In this case therefore, a scale factor must be introduced in the LBO eigenvector space to address this scaling ambiguity due to cardinality differences in the original pointsets. The effect of adding such a scaling factor is shown in the fourth row of Figures 11 and 12. Note that the sign-flips persist (which we account for in our framework) but the scale of the pointsets is almost the same which gets us closer to our original formulation.

The most interesting aspect of this investigation is the induced scaling of the pointsets in the GR—the smaller the number of points, the larger the representation of the pointset. To the best of our knowledge, this is not a known phenomenon which poses an interesting research question: **what is the relationship between the cardinality of pointsets and their corresponding Grassmannian representations?**

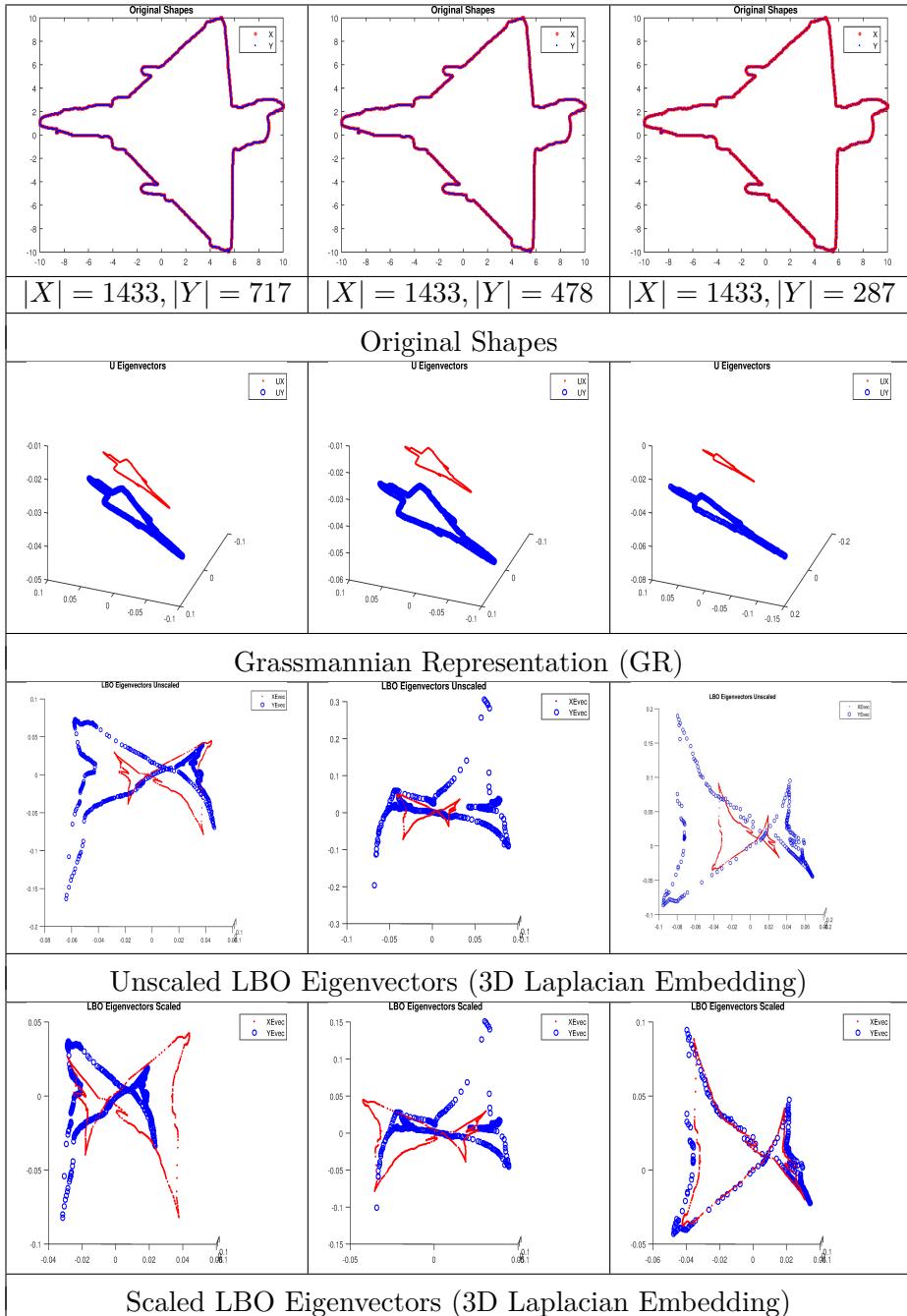


Figure 11: Analysis of the GrassGraph framework for unequal pointset cardinality (for even subsampling, see Section 7) without affine transformations. (Zoom in to the first row to see the different number of points between the red and blue pointsets). The first row shows three affine transformed versions of a 2D aircraft shape (the cardinality of the pointsets are listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different, a translation now exists and no arbitrary rotation exists between the pointsets. The absence of the arbitrary rotation is due to the lack of an affine transformation between the original pointsets, which is expected in this analysis. However, this induced scale in the GR, carries over into the LBO eigenvectors as shown in the third row where the LBO embeddings are now off by a sign-flip and scale. Hence, a new scaling parameter must be incorporated to bring the eigenvectors to the same scale so our original method can be leveraged.

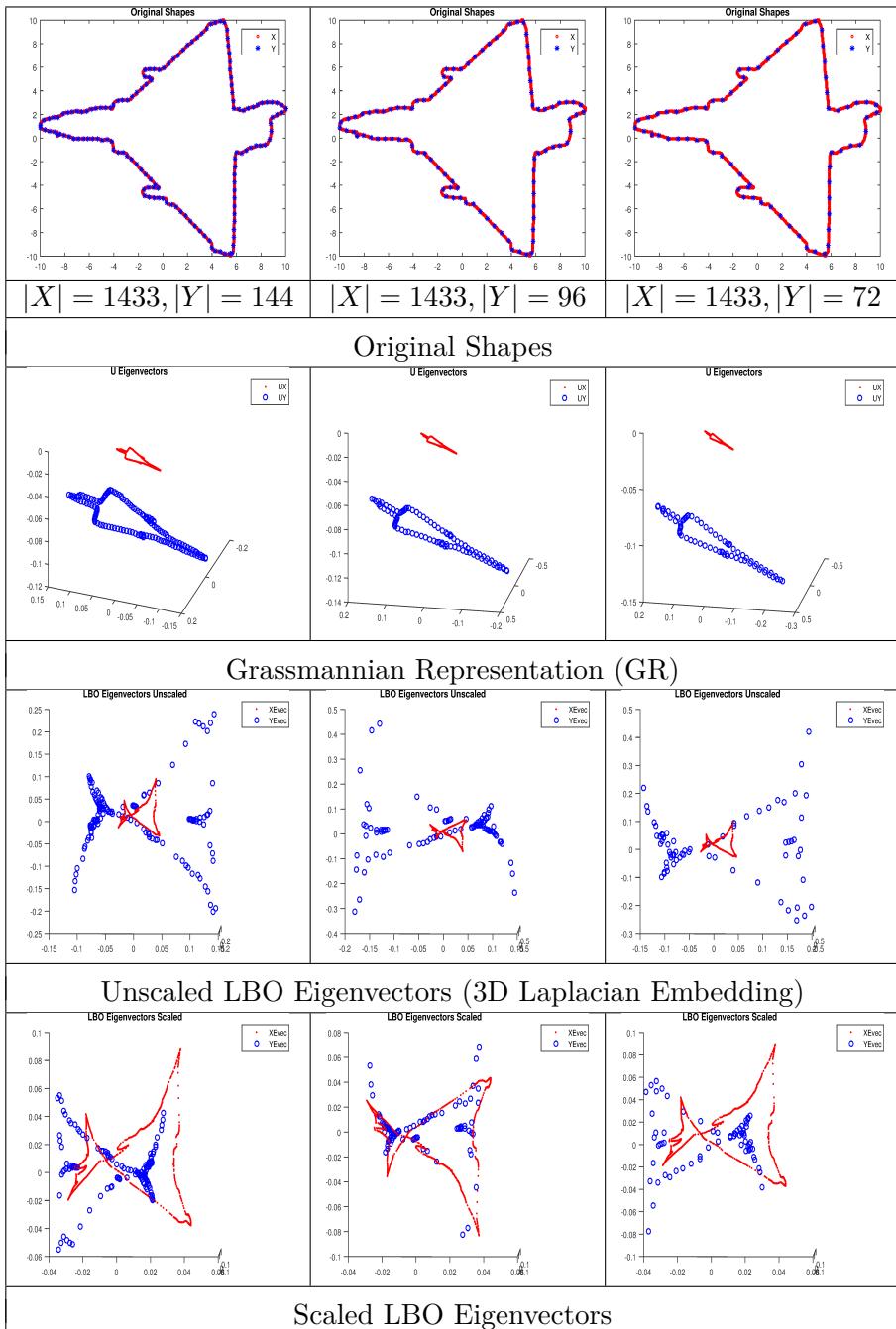


Figure 12: Analysis of the GrassGraph framework for unequal pointset cardinality (even subsampling, see Section 7) without affine transformations continued from Figure 11. (Zoom in to the first row to see the different number of points between the red and blue pointsets). The first row shows three affine transformed versions of a 2D aircraft shape (the cardinality of the pointsets is listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different, a translation now exists and no arbitrary rotation exists between the pointsets. Note that as the cardinality of the original pointset decreases the scale of the GR pointsets (in this case Y) increases. The absence of the arbitrary rotation is due to the lack of an affine transformation²¹ between the original pointsets, which is expected in this analysis. However, this induced scale in the GR carries over into the LBO eigenvectors as shown in the third row where the LBO embeddings are now off by a sign-flip and scale.

In Section 7, we investigated the response of our method to evenly subsampled pointsets without affine transformations. There we saw that in the GR, the pointsets were off by a scale and a translation but not an arbitrary rotation. As a result of this, the LBO eigenvectors were seen to be off by a scale as well, which prompted us to incorporate an additional scaling parameter to equalize the scale between the pointsets. This section now investigates this same subsampling but with affine transformations included. Figures 13, 14, and 15 show the same shape subsampled at different rates under different affine transformations. Within each figure, the affine transformation is the same and three subsampled versions of the shape are shown. Therefore, the effect of subsampling on the components of the GrassGraphs across a particular affine transformation can be inferred from each figure.

From the three figures it is clear that the change in scale in the GR is due to the subsampling and not the affine transformation itself. However, the arbitrary rotation is caused by the affine transformation. Fortunately, our LBO eigenvector representation solves this rotation problem. The main change that affects our framework is the scale factor in the LBO space resulting from the induced scale in the GR. The fourth row of Figures 13, 14, and 15 show that incorporating a scale parameter brings us closer to our original framework. To be effective, this scaling parameter must first be solved in order for our sign-flip resolution and correspondence algorithm to be able to recover the correspondences. Figure 15, shows the same pointsets under the same affine transformation as Figure 14 but subsampled at a lower rate. It confirms the fact that the scale in the GR and LBO is a direct result of the subsampling or difference in cardinality of the pointsets.

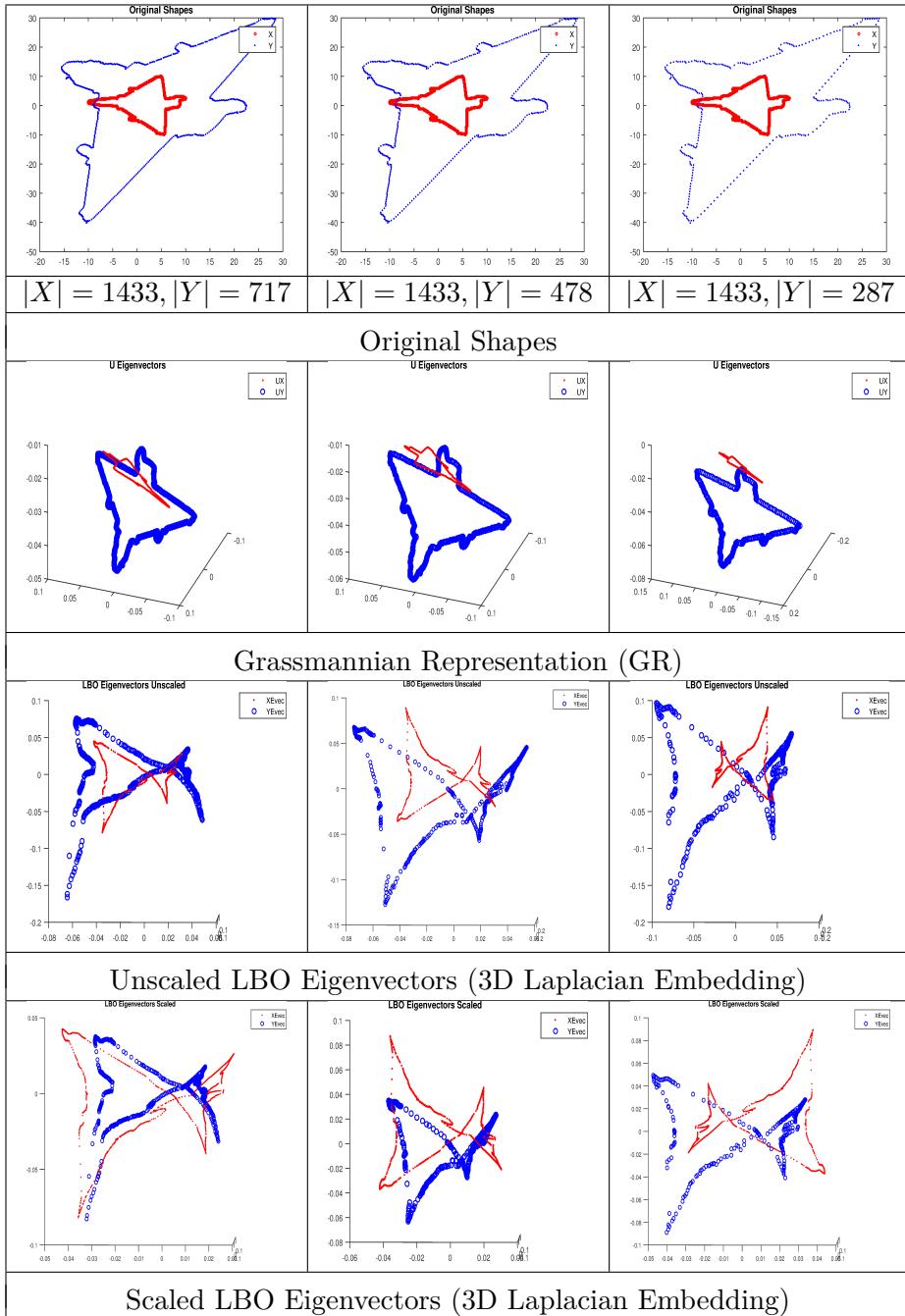


Figure 13: Analysis of the GrassGraph framework for unequal pointset cardinality (even subsampling, see Section 7) with affine transformations. The first row shows three affine transformed versions of a 2D aircraft shape with different affine transformations to Figure 14 and 15 (the cardinality of the pointsets are listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different, a translation now exists and an arbitrary rotation exists between the pointsets. Note that as the cardinality of the original pointset decreases, the scale of the GR pointsets (in this case Y) increases. However, this induced scale in the GR carries over into the LBO eigenvectors as shown in the third row where the LBO embeddings are now off by a sign-flip and scale. Hence, a new scaling parameter must be incorporated to bring the eigenvectors to the same scale so our original method can work.

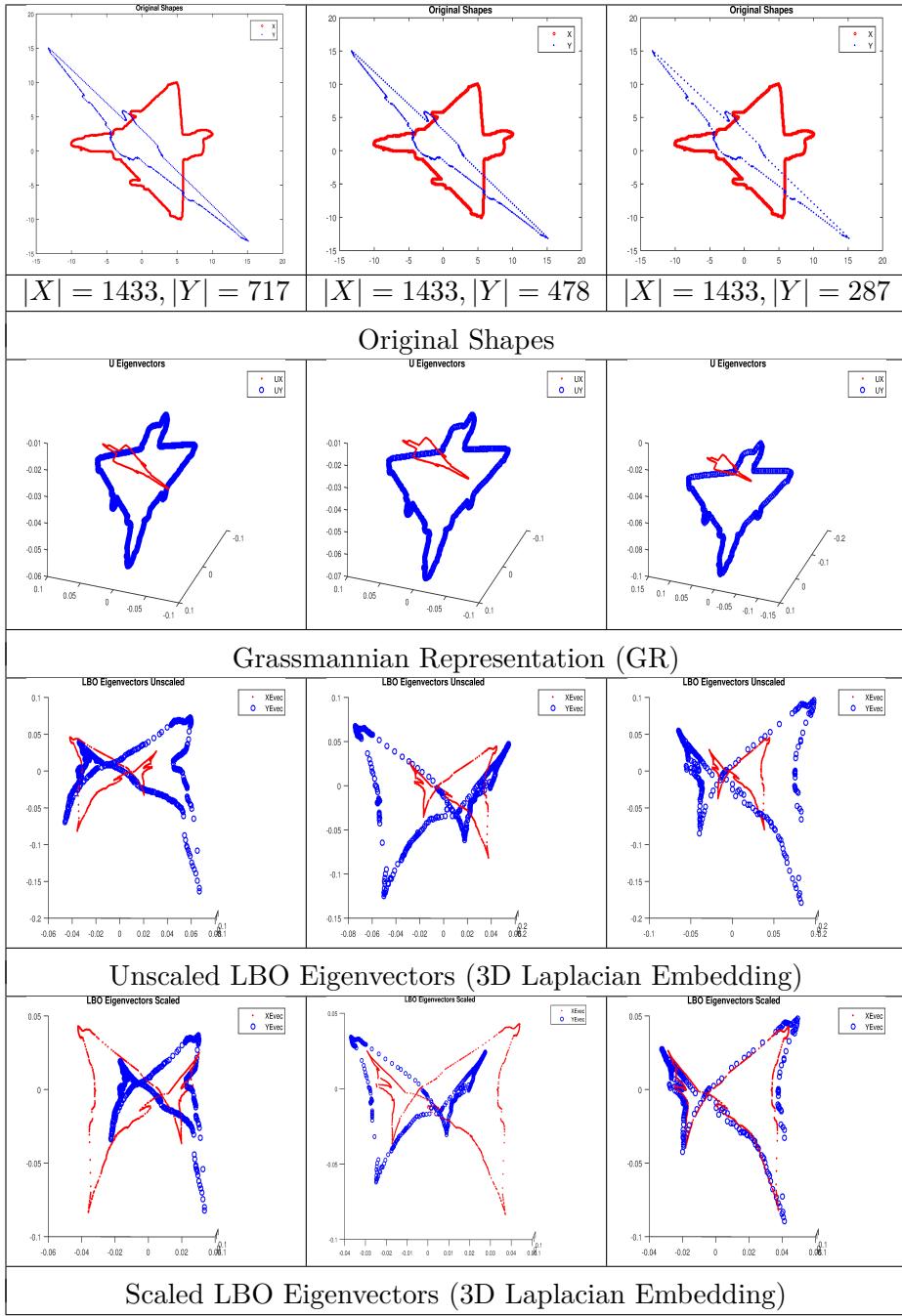


Figure 14: Analysis of the GrassGraph framework for unequal pointset cardinality (even subsampling, see Section 7) without affine transformations. The first row shows three affine transformed versions of a 2D aircraft shape with different affine transformations to Figure 13 and 15 (the cardinality of the pointsets are listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different, a translation now exists and an arbitrary rotation exists between the pointsets. Note that as the cardinality decreased the scale of the original pointsets (in this case Y) increases. However, this induced scale in the GR, carries over into the LBO eigenvectors as shown in the third row where the LBO embeddings are now off by a sign flip and scale. Hence, a new scaling parameter must be incorporated to bring the eigenvectors to the same scale so our original method can work.

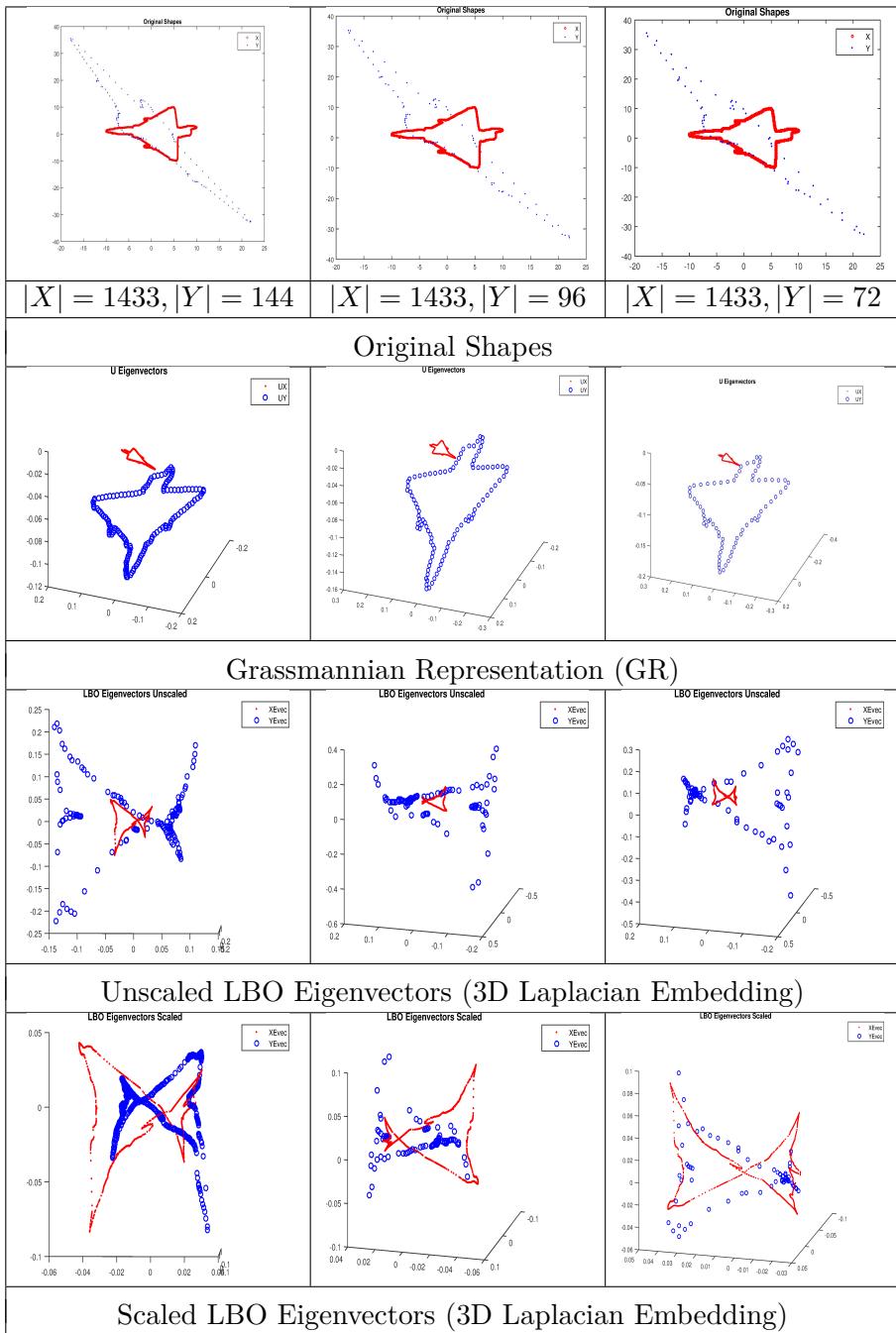


Figure 15: Analysis of the GrassGraph framework for unequal pointset cardinality (even subsampling, see Section 7) without affine transformations. The first row shows three affine transformed versions of a 2D aircraft shape (the cardinality of the pointsets is listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different, a translation now exists and an arbitrary rotation exists between the pointsets. Note that as the cardinality decreased the scale of the original pointsets (in this case Y) increases. However, this induced scale in the GR, carries over into the LBO eigenvectors as shown in the third row where the LBO embeddings are now off by a sign-flip and scale. Hence, a new scaling parameter must be incorporated to bring the eigenvectors to the same scale so our original method can work.

Analysis of Unevenly Subsampled Pointsets (Unequal Cardinality) in the GR without Affine Transformations

In this section we investigate the effect of a different type of subsampling on the component of the GR. Instead of sampling evenly over the entire pointset, we localize it to remove parts of the shape. The effect of removing up to 15% of the points is shown in Figure 16. When we look at the second row, there is a distinct rotation present. Even though we established above that a smaller cardinality pointset results in an increase in scale in the GR, it does not seem to be the case here (since this is a missing points rather than subsampling situation). However, it is important to note that this scale may not be noticeable for only a 15% reduction in cardinality. Figure 17 confirms that a reduction in cardinality does indeed increase the scale of the pointset in the GR. Looking at the third row of Figure 16, it is evident that a loss in information of the original pointset manifests itself in the LBO eigenvector representation (as well as the GR).

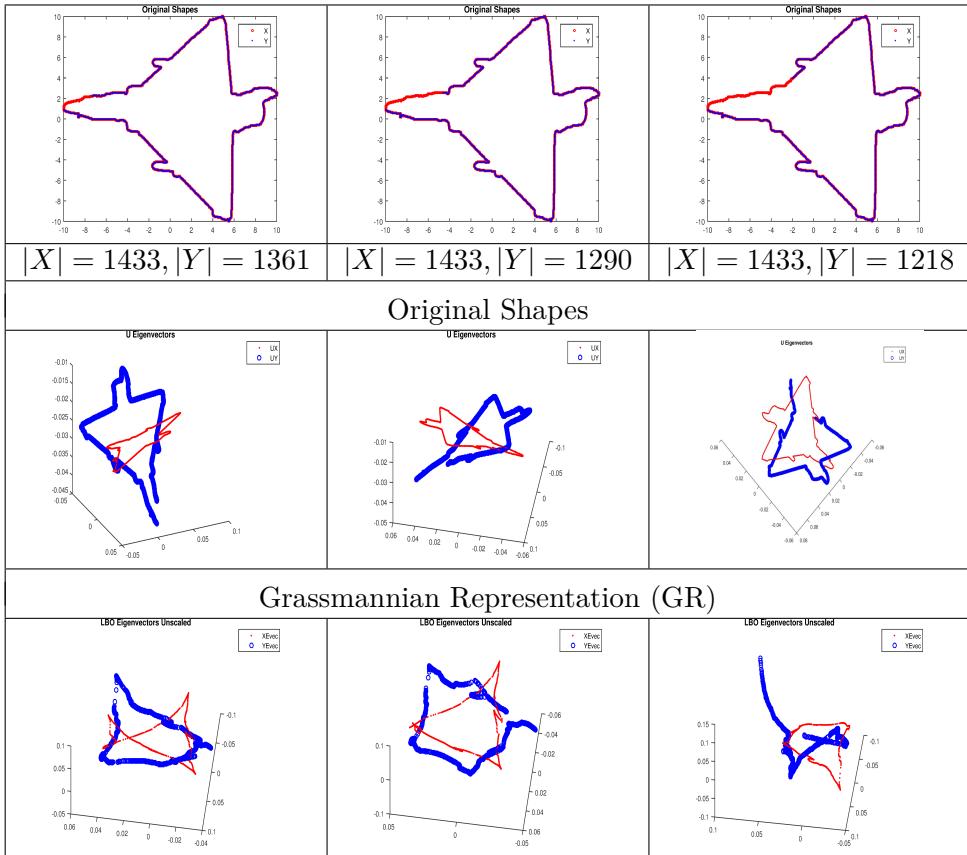


Figure 16: Analysis of the GrassGraph framework for unequal pointset cardinality without affine transformations. The first row shows three affine transformed versions of a 2D aircraft shape (the cardinality of the pointsets is listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The main difference in this second row unlike the second row of Figure 10, is that the scale of pointsets are different and rotation exists between the pointsets. In the third row, our LBO eigenvector representation reflects the removed parts of the original shape.

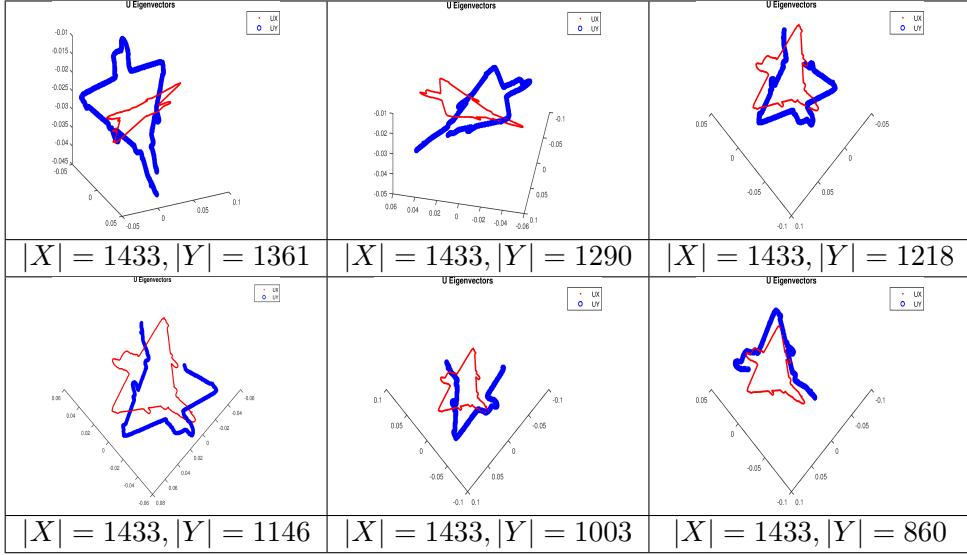


Figure 17: Grassmannian representation of pointsets at different uneven subsamplings.

Analysis of Unevenly Subsampled Pointsets (Unequal Cardinality) in the GR with Affine Transformations

In this section we investigate the response of our method to uneven subsampling of pointsets in the presence of affine transformations. Figure 18 shows the results of this investigation. An important thing to notice is that the removing 40% of the points of the shape only results in a marginal increase in scale in the GR. This is unlike the pointsets in Figure 15, where a small cardinality produces a significantly noticeable increase in scale in the GR. In order to solve the unequal points problem, our future formulation needs to account for the change in scale in the LBO eigenvector representation coming from the difference in cardinality of the pointset.

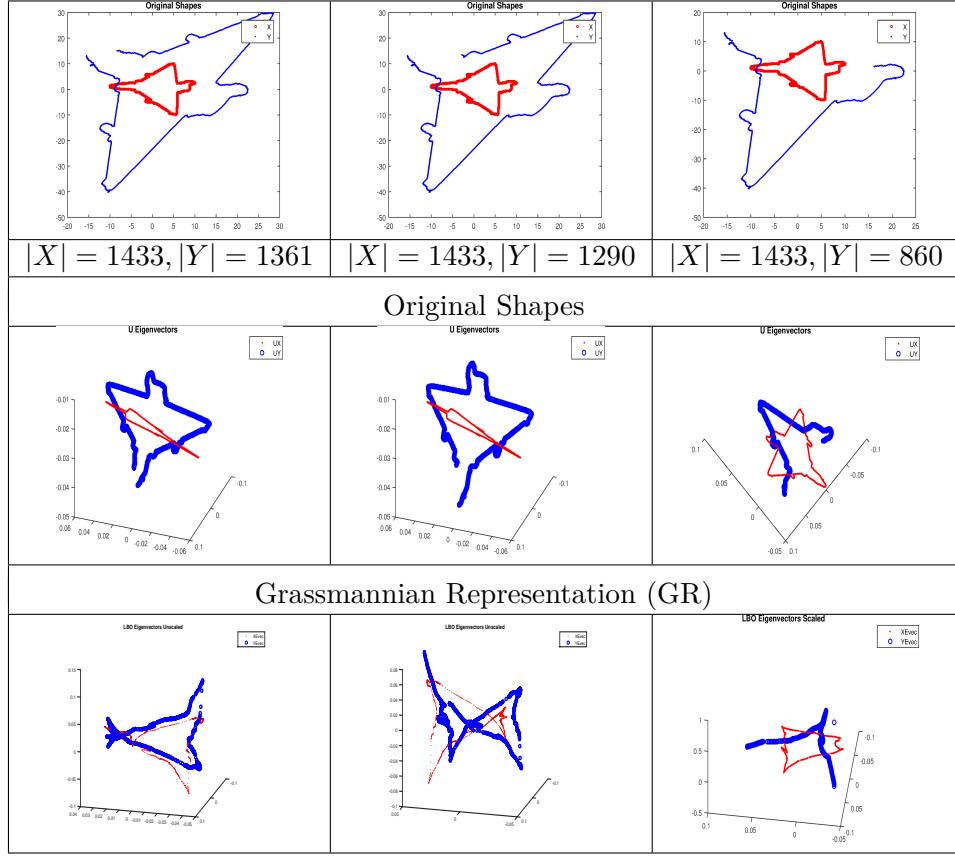


Figure 18: Analysis of the GrassGraph framework for unequal pointset cardinality with affine transformations. The first row shows three affine transformed versions of a 2D aircraft shape (the cardinality of the pointsets is listed below each figure). The original shape is shown in red and the transformed shape is shown in blue (this ordering persists through all the associated images). The second row shows the Grassmannian representation (GR) of the pointsets above it. The second row shows that the affine transformation is introducing the expected arbitrary rotation. In the third row, our LBO eigenvector representation reflects the removed parts of the original shape.

References

- [1] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441, 2016.
- [2] 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.
- [3] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid. Joint probabilis-

- tic matching using m-best solutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 136–145, 2016.
- [4] M. T. Harandi, C. Sanderson, S. A. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2705–2712. IEEE Computer Society, 2011.
- [5] H. Liu and S. Yan. Robust graph mode seeking by graph shift. In *International Conference on Machine Learning*, pages 671–678, 2010.
- [6] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014.
- [7] D. Raviv and R. Kimmel. Affine invariant geometry for non-rigid shapes. *International Journal of Computer Vision*, 111(1):1–11, 2015.