

# TECHNICAL REPORT

## Visual Exploration of High-Dimensional Data: Subspace Analysis through Dynamic Projections

*Shusen Liu*<sup>\*</sup>, *Bei Wang*<sup>\*</sup>, *Jayaraman J. Thiagarajan*<sup>†</sup>, *Peer-Timo Bremer*<sup>†</sup>, *Valerio Pascucci*<sup>\*</sup>

<sup>\*</sup>SCI Institute, University of Utah, USA

<sup>†</sup>Lawrence Livermore National Laboratory

UUSCI-2014-003

Scientific Computing and Imaging Institute  
University of Utah  
Salt Lake City, UT 84112 USA

August 22, 2014

### **Abstract:**

Understanding high-dimensional data is rapidly becoming a central challenge in many areas of science and engineering. Most current techniques either rely on manifold learning based techniques which typically create a single embedding of the data or on subspace selection to find subsets of the original attributes that highlight the structure. However, the former creates a single, difficult-to-interpret view and assumes the data to be drawn from a single manifold, while the latter is limited to axis-aligned projections with restrictive viewing angles. Instead, we introduce ideas based on subspace clustering that can faithfully represent more complex data than the axis-aligned projections, yet do not assume the data to lie on a single manifold. In particular, subspace clustering assumes that the data can be represented by a union of low-dimensional subspaces, which can subsequently be used for analysis and visualization. In this paper, we introduce new techniques to reliably estimate both the intrinsic dimension and the linear basis of a mixture of subspaces extracted through subspace clustering. We show that the resulting bases represent the high-dimensional structures more reliably than traditional approaches. Subsequently, we use the bases to define different viewpoints, i.e., different projections onto pairs of basis vectors, from which to visualize the data. While more intuitive than non-linear projections, interpreting linear subspaces in terms of the original dimensions can still be challenging. To address this problem, we present new, animated transitions between different views to help the user navigate and explore the high-dimensional space. More specifically, we introduce the view transition graph which contains nodes for each subspace viewpoint and edges for potential transition between views. The transition graph enables users to explore both the structure within a subspace and the relations between different subspaces, for better understanding of the data. Using a number of case studies on well-know reference datasets, we demonstrate that the interactive exploration through such dynamic projections provides additional insights not readily available from existing tools.

# Visual Exploration of High-Dimensional Data: Subspace Analysis through Dynamic Projections

Shusen Liu\*

SCI Institute, University of Utah

Bei Wang<sup>†</sup>

SCI Institute, University of Utah

Jayaraman J. Thiagarajan<sup>‡</sup>

Lawrence Livermore National Laboratory

Peer-Timo Bremer<sup>§</sup>

Lawrence Livermore National Laboratory

Valerio Pascucci<sup>¶</sup>

SCI Institute, University of Utah

**Abstract**— Understanding high-dimensional data is rapidly becoming a central challenge in many areas of science and engineering. Most current techniques either rely on manifold learning based techniques which typically create a single embedding of the data or on subspace selection to find subsets of the original attributes that highlight the structure. However, the former creates a single, difficult-to-interpret view and assumes the data to be drawn from a single manifold, while the latter is limited to axis-aligned projections with restrictive viewing angles. Instead, we introduce ideas based on subspace clustering that can faithfully represent more complex data than the axis-aligned projections, yet do not assume the data to lie on a single manifold. In particular, subspace clustering assumes that the data can be represented by a union of low-dimensional subspaces, which can subsequently be used for analysis and visualization. In this paper, we introduce new techniques to reliably estimate both the intrinsic dimension and the linear basis of a mixture of subspaces extracted through subspace clustering. We show that the resulting bases represent the high-dimensional structures more reliably than traditional approaches. Subsequently, we use the bases to define different “viewpoints”, i.e., different projections onto pairs of basis vectors, from which to visualize the data. While more intuitive than non-linear projections, interpreting linear subspaces in terms of the original dimensions can still be challenging. To address this problem, we present new, animated transitions between different views to help the user navigate and explore the high-dimensional space. More specifically, we introduce the view transition graph which contains nodes for each subspace viewpoint and edges for potential transition between views. The transition graph enables users to explore both the structure within a subspace and the relations between different subspaces, for better understanding of the data. Using a number of case studies on well-know reference datasets, we demonstrate that the interactive exploration through such dynamic projections provides additional insights not readily available from existing tools.

**Index Terms**—High-dimensional data, Subspace, Dynamic projection

## 1 INTRODUCTION

As our ability to collect a wide variety of large, complex datasets grows, techniques to understand and mine such data are becoming increasingly important. Typically, data is given as a large number of points in high dimension describing anything from physical experiments to collections of images or social network connections. Fortunately, the vast majority of these datasets appear to describe structures of low intrinsic dimension that are embedded in high dimensional space. This has led to the development of a large number approaches aimed at finding and exploiting the low dimensional structures. In particular, the two most common concepts are manifold learning [19, 27] and subspace finding [2, 9]. Manifold learning assumes the data can be described by a single (smooth) manifold and a number of different ideas have been proposed to find non-linear projections that best preserve the high dimensional structure. However, the resulting projection(s) are difficult to interpret with respect to the original dimensions, results can vary drastically between techniques, and the single manifold assumption is often violated in practice.

Subspace finding/search techniques, instead, assume that there exists a (small) subset of the original dimensions which adequately describes the data. Typically, these approaches will first cluster and then rank different subsets of dimensions according to how “interesting” the corresponding projection appears to be and then use axis-aligned 2D projections, i.e., scatterplots to explore these subspaces. However, even for structures of low intrinsic dimension the required number of original dimensions can be quite large making the subsequent analysis and visualization intractable. Furthermore, especially in the presence

of non-linear structures axis aligned projections may contain significant artifacts which can severely distort the ranking.

Our proposed analysis and visualization techniques are based on recent advances in subspace clustering. Subspace clustering attempts to strike a middle ground between the two traditional approaches by assuming the data to be well represented through a collection of linear (or affine) subspaces not necessarily of uniform dimensions. This removes the single manifold assumption yet restricts the subspace to be linear. However, in practice this restriction is often an advantage as linear subspaces are easier (though not necessarily easy) to interpret than non-linear ones and non-linear subspaces can always be approximated by a collection of linear ones. Furthermore, subspace clustering naturally produces a number of interesting projections, given by the subspace bases, without the potential artifacts of evaluating interestingness after a projection.

Nevertheless, a number of open challenges remain. First, existing techniques in subspace clustering are designed to cluster, not to find intrinsic dimensions or linear subspaces. Since the clustering, especially for intersecting subspaces, cannot be perfect, applying traditional dimension estimation and subspace finding techniques directly to the clusters produces suboptimal results (see Section 3.2). Second, the user is still left with a collection of non-intuitive subspaces and their bases vectors that must be explored and analyzed. This paper presents a new framework for the visual exploration of high-dimensional data that addresses both these problems.

In this work, we introduce a new dimension and bases estimation method directly tied into the subspace clustering and less susceptible to outliers or intersecting subspaces than traditional techniques. Furthermore, in order to better understand and analyze the subspace clustering and basis estimation result, we propose adopting animated view transitions to interactively explore individual subspaces as well as the relationship between subspaces. For each pair of basis vectors within or between subspaces we create a 2D *view* defined as the corresponding projection into the plane defined by the given vectors. We then generate the view layout for intuitive navigation among views between and within subspaces. One can analyze and infer the relation-

\*e-mail: shusenl@sci.utah.edu

<sup>†</sup>e-mail: beiwang@sci.utah.edu

<sup>‡</sup>e-mail: jayaramanthi1@llnl.gov

<sup>§</sup>e-mail: bremer5@llnl.gov

<sup>¶</sup>e-mail: pascucci@sci.utah.edu

ship between views by exploiting the smooth and continuous dynamic projection seamlessly connecting these two views. Using several well-known reference datasets, we demonstrate that our framework is able to recover complex subspaces and that the animated transitions reveal insights not easily accessible with traditional techniques.

**Conventions and notations.** To eliminate any possible confusion from the naming convention, here we define some of the fundamental concepts and names used in this paper. In our work, we consider any linear projection as a “view” for peeking into the high-dimensional space, therefore, a view means a linear projection. Also, our proposed method uses subspace clustering to identify a subset of data that shares a similar bases. A cluster corresponds to a subspace, therefore, cluster and subspace may be used interchangeably. Subspace search/finding most commonly refers to the subspace discovery method proposed in database community. However, they also use subspace clustering to describe the searching operation for a subset of dimensions. To avoid the confusion we will not refer to them as subspace clustering algorithms, but as subspace searching/finding algorithms.

## 2 BACKGROUND

In this section, we reviewed several topics closely related to our proposed method.

### 2.1 Subspace Clustering

Several of the advances in analyzing high-dimensional data have relied on the observation that, even though these samples were drawn from a high-dimensional space, their intrinsic dimension is much smaller than the ambient dimensions. For example, the number of pixels in an image may be large. However, we typically use only a few parameters such as the geometry or the dynamics to describe the appearance. Data models inferred with such assumptions are often simple, in the number of parameters, and interpretable. This has motivated the design of techniques that can provide meaningful low-dimensional representations for the high-dimensional data. A wide variety of low-dimensional models have been considered in the machine learning and data analysis literature, and they have found widespread applications in pattern recognition, data compression, bio-informatics etc. Conventional approaches such as the Principal Component Analysis (PCA) [13] assume the data lies along a single, linear (or affine) low-dimensional subspace of the ambient dimensions. However, in practice, this assumption could be restrictive, and hence we often use a more general assumption that the data samples are drawn from a union of subspaces. The memberships of the samples to the subspaces are unknown, and each of the subspaces can be of different dimensions. This is more challenging as there is a need to simultaneously cluster the data into multiple subspaces and find a low-dimensional subspace fitting each group of samples.

Existing subspace clustering methods can be broadly classified into three classes: (a) algebraic, (b) iterative, and (c) spectral clustering-based. All these methods are typically developed for noise-free data and their behavior in presence of noise and outliers vary significantly. Algebraic methods for subspace clustering include performing low-rank matrix factorization of the data matrix [29] and generalized PCA [30] which is based on the idea that one can fit a union of  $n$  subspaces with a set of polynomials of degree  $n$ , whose derivatives at a sample provide a vector orthogonal to the subspace containing that sample. Iterative methods follow the construction of traditional clustering algorithms such as K-means, which given an initial segmentation, fit a model (example: PCA) to each subspace and in the next step assign the samples to the nearest subspace. Some examples for this class of methods include K-planes [6] and K-subspace clustering [1]. The last class of methods attempt to construct graph affinities, that capture the subspace structure, for use in spectral clustering [22]. This approach has been shown to be provably correct when the data is noise-free and can be effective in handling noise and outliers. Some of the recently successful methods based on this idea include local subspace affinity [32], sparse subspace clustering [10], low-rank representation [20] and spectral curvature clustering [8].

### 2.2 Visual Exploration through Subset of Dimension

Due to the structural complexity and high-dimensionality of many datasets, some of the recent advances in high-dimension data visualization rely on selecting a related subset of dimensions for analysis. Approaches such as representative factor generation [28] and dimension projection matrix/tree [33] can allow interactive exploration in the dimension space, in addition to enabling analysis in the data space. This allows the user to visually explore the relationship among dimensions and filter out unrelated dimensions. While other methods, such as the TripAdvisorND [21], adopt the ENCLUS[9] algorithm to identify related subset of dimensions (or subspaces). In [26], the authors proposed an approach for summarizing the large number of “subspaces” (subset of dimensions) generated from subspace search algorithms. These methods introduce some very interesting exploration strategies for high-dimensional datasets, and they can be particularly effective when the dimensions are not closely coupled. One issue with such approaches is that only axis-aligned features are easily discoverable. In addition, the subset of dimensions also means we are only using partial information in the data, so it could be difficult to determine whether the discovered features are truthful structures in high-dimensional space or artifacts due to the use of incomplete data. In our proposed framework, we utilize subspace clustering techniques, which group points that share common low dimensional linear spaces, therefore reliably capturing the intrinsic structure in high-dimensional space.

### 2.3 Informative View vs. Interpretable Axis

Informative view refers to the low dimensional embedding that reveals the hidden innate structure of the high-dimensional data (class separation for example). Fully understanding these structures often proves to be difficult due to the lack of interpretability of the axes. Trade-offs exist between the “informative view” and “interpretable axis”. To begin with, let us consider the simplest projection: the axis parallel projection used in a scatterplot / scatterplot matrix. Based on the worst case lemma[3], the axis-parallel projections will tend to be non-informative for high-dimensional problems. This implies that even if we select the most “interesting” view from the scatterplot matrix, as carried out by approaches such as Scagnostics work [31], we will still lack the information to reliably capture the intrinsic structure. However, the benefit of axis-aligned plots is obvious: one can easily interpret them. On the other hand, even with a linear projection such as PCA, we have no idea about what the principle component represents, and hence there is no meaningful label for the axes in the plot. As discussed in [14], even a simple linear basis could be difficult to interpret. In [14], Gleicher attempts to associate a linear basis with a certain meaningful concept. These concepts are defined based on user defined examples. Machine learning techniques can then be employed to find a set of simple linear bases that could achieve accurate projection according to the example prior. This allows for an interesting and meaningful view to be generated based on user intention. However, there is no guarantee such a projection will reveal any interesting structure that was masked in the high-dimensional space. For non-linear dimension reduction techniques, interpreting of the axis would be almost impossible. In order to combine the advantages of both worlds, our method tries to find an informative view through subspace clustering and basis animation, and with the help of transitional animation, we could better interpret and understand the subspace bases.

### 2.4 Animation Augmented Exploration

Besides identifying suitable/informative views, navigation and animated transition between different scatterplots has been introduced to enhance perception, and gain intuition about the high-dimensional data. Our brain has been trained to track the correspondence in animated transitions as well as recognize high-dimensional objects (3D mostly) by viewing a continuous 2D projection of them. This observation could be exploited for designing high-dimensional visualization tools. The Rolling the Dice approach for navigating a scatterplot matrix [11] provides a navigation interface for transition between pairs of scatter plots, as well as a smooth 3D transformation animation to

help visualize the point correspondence and structural changes. In addition, users can iteratively build queries using bounding volumes in the dataset, sculpting the query from different viewpoints. In NavGraph [18], an interesting subset of scatterplots in scatterplot matrix is selected based on the measures proposed in Graph-Theoretic Scagnostics [31]. Subsequently, a graph is constructed using the pair of views with common variables. Navigating along each edge of this graph will result in a smooth transition animation that mimics the rigid body transformation in 3D. Compare to NavGraph, our work relies on a very different view selection scheme. Instead of attempting to find interesting views among all scatterplots (that may originate from artifacts), we use subspace clustering to capture the data’s intrinsic low dimensional structure. Sanftmann et.al [24] further extended the 2D scatterplot transition by introducing similar operation for 3D scatterplots. The GGobi system[25] introduced the guided tour concept, it combines grand tour[4] with projection pursuit[12], allowing transition to be guided to the more “interesting” views (based on projection pursuit index). Instead of relying on a fully animated transition, some methods like the TripAdvisorND[21] use a limited rotation to create an illustration that gives the user an impression of the high-dimensional structure. This system provides a Focus+Context approach, where a number of “tourist sites” corresponding to a best view of each subspace (the subset of dimensions) is provided as an overview and context. The user can delve into each of these “tourist sites” for a more focused study, including slightly tilting the projection plane around their neighborhood. In comparison to the TripAdvisorND system, our proposed approach adopts a similar navigation structure to guide the exploration, with an important difference that our method allows transiting between different subspaces, which is often crucial for understanding complex data.

### 3 METHOD

As shown in Figure 1, our system contains two major components: the preprocessing step and the interactive exploration step. The preprocessing is responsible for subspace identification and subspace model estimation. First, we applied subspace clustering on the input high-dimensional dataset. Then our proposed model estimation algorithm is utilized to identify each subspaces’ intrinsic dimension and the corresponding basis. After that we can feed all the preprocessed information into the interactive exploration tool, where we use animated transitions between subspace view to understand the structure of the high-dimensional space.

#### 3.1 Subspace Clustering

Let us assume that the set of data samples  $\{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^T$  are drawn from an unknown union of  $n \geq 1$  linear subspaces  $\{S_j\}_{j=1}^n$ . The dimensions of the subspaces,  $0 < d_j < D$  ( $j = 1, \dots, n$ ), are unknown and each subspace is described as

$$S_j = \{\mathbf{x} \in \mathbb{R}^D : \mathbf{x} = \mathbf{U}_j \mathbf{y}\}, \quad (3.1.1)$$

where  $\mathbf{U}_j \in \mathbb{R}^{D \times d_j}$  is a basis for the subspace  $S_j$  and  $\mathbf{y} \in \mathbb{R}^{d_j}$  is the low-dimensional representation of the sample  $\mathbf{x}$ . When  $n = 1$ , this problem reduces to Principal Component Analysis (PCA). In this case, the basis  $\mathbf{U}$  and the low-dimensional representation  $\mathbf{Y}$  can be easily obtained using the rank- $d$  singular value decomposition of the mean-removed data matrix. The dimension  $d$  can be fixed as the rank of  $\mathbf{X}$  when the data is noise-free, and a suitable model selection technique can be adopted for noisy data [29]. Ideally, when the clusters are known, PCA can be directly used to estimate the basis. However, when  $n > 1$ , the problems of identifying subspaces and model estimation are coupled. One primary challenge with this general setting is that in extreme cases we can either fit  $T$  one-dimensional subspaces to the data, i.e., one subspace for each sample, or a single  $D$ -dimensional subspace, and neither of them are appropriate. As a result, a model selection strategy that can balance the number of subspaces and the number of dimensions becomes crucial.

A wide variety of subspace clustering algorithms have been proposed in the machine learning literature [29], and in particular methods based on spectral clustering have been very effective. Spectral

clustering belongs to the family of subspace clustering which aims at finding a low-dimensional subspace for each group of points [22]. These methods rely on the assumption that the high-dimensional data could be better grouped in a projected space. Thus, spectral clustering mainly contains two tasks, projecting the dataset into an latent space and calculating the cluster membership of the dataset in that space. It works by constructing an affinity matrix  $\mathbf{A} \in \mathbb{R}^{T \times T}$ , where  $A_{ij}$  measures the similarity between samples  $i$  and  $j$ . Following this, the cluster membership of the data points are estimated using the spectrum of the affinity matrix. Given the affinity matrix, we apply K-means clustering to the eigenvectors of the graph Laplacian  $\mathbf{L}$ . The Laplacian can be constructed as  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is a diagonal matrix with each diagonal element is set to the sum of the corresponding row (or column) of the symmetric matrix  $\mathbf{A}$ . Alternatively, a normalized Laplacian can be defined as  $\mathbf{L} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ . Assuming that the number of clusters  $K$  is known, we extract  $K$  smallest eigenvectors,  $\mathbf{V} \in \mathbb{R}^{T \times K}$ , of  $\mathbf{L}$  and apply K-means clustering to the rows of  $\mathbf{V}$ .

The construction of the affinity matrix lies at the heart of the spectral clustering algorithm and there are two general kinds of metrics to build a similarity graph, i.e., pairwise distances and reconstruction coefficients. The first approach uses pairwise distances to determine the neighborhood for each sample and appropriately computes similarities for the edges in the graph. One popular similarity measure is based on the heat kernel,  $A_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 / \sigma)$ . The primary challenge in using distance-based affinities for subspace clustering is that two samples that are close to each other, may actually lie on different subspaces, e.g. near the intersection of two subspaces. On the contrary, two samples that are far from each other may belong to the same subspace. Furthermore, pairwise distances are very sensitive to noise and outliers. An alternative class of methods is based on the idea of representing each sample as a linear combination of other data samples. Since samples belonging to the same subspace share a common basis, this linear combination can reveal the underlying subspace structure, if one exists.

The general model considered for reconstructing each sample using the other data samples can be written as

$$\mathbf{X} \approx \mathbf{XW}, \quad \text{s.t.} \quad W_{ii} = 0 (i = 1 \cdots T), \quad (3.1.2)$$

where the matrix  $\mathbf{W} = [\mathbf{w}_i]_{i=1}^T$  describes the relationships between the different samples and the condition  $W_{ii} = 0$  ensures that a sample is not used for its own reconstruction. In principle, this leads to a highly *ill-posed* problem with several possible solutions. This calls for using appropriate *regularization* to this optimization problem. In this paper, we consider three regularization penalties on  $\mathbf{W}$  for the problem in (3.1.2) : (i) sparsity [10], (ii) low-rank [20], and (iii) L2 [23]. The details on these three approaches have been included in the supplementary material. Given the coefficient matrix  $\mathbf{W}$ , the adjacency matrix for spectral clustering is computed as  $\mathbf{A} = |\mathbf{W}| + |\mathbf{W}^T|$ . Given the subspace segmentation, the next step is to fit a linear model for each subspace. Though PCA can be directly used to infer the basis and dimension for each subspace, we propose an alternative approach for model estimation based on graph embedding. The proposed approach is more resilient to outliers, and can provide an improved data fitting when data does not completely adhere to the subspace assumption, which is often the case in practice.

#### 3.2 Proposed Model Estimation

The primary advantages of the different affinity matrix construction approaches described in the previous section is that the neighborhood for each sample is chosen automatically and by design, they search for low-dimensional subspaces to fit the data. In this paper, we propose to use dynamic projections to analyze a subspace and its connection to other subspaces. To achieve this, we need to reliably estimate the dimension of each subspace and its corresponding basis. The problem of understanding relations between subspaces can then be efficiently solved using the models estimated for each subspace.

A natural way to estimate the dimension and basis of a linear subspace is to use PCA. However, using PCA for model estimation with

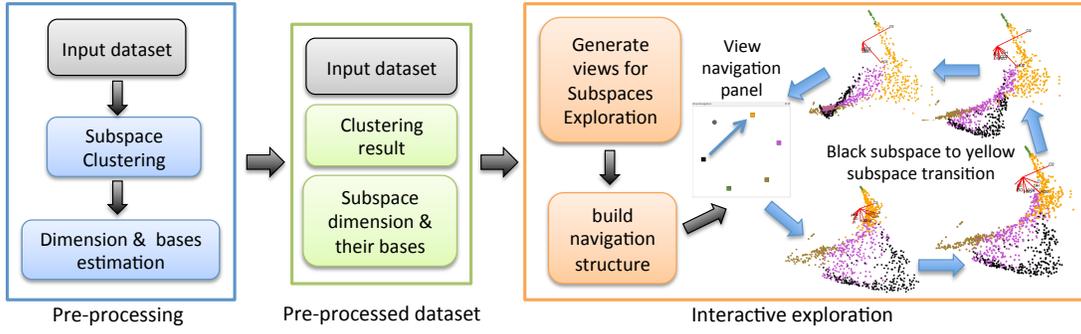


Fig. 1: An overview of our work-flow. Firstly, subspace clustering is applied on the input dataset. Then a model estimation algorithm is applied to identify each subspace's intrinsic dimension and the corresponding basis, eventually all the preprocessed information is fed into the interactive exploration tool, where we can use animated projection transition between subspace views to understand the structure of the high-dimensional data.

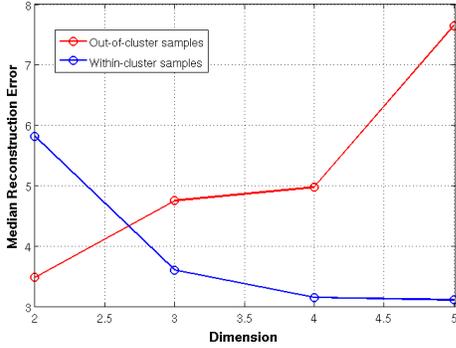


Fig. 2: Estimating the intrinsic dimension of linear subspaces - Median misfit for the within-cluster ( $E_{in}^d$ ) and out-of-cluster samples ( $E_{out}^d$ ) using the basis estimated for a cluster. The dimension is set as the lowest value beyond which  $E_{in}^d < E_{out}^d$ , and  $E_{in}^d \leq \tau E_{in}^{D-1}$ .

each subspace independently does not exploit the relationships between the different subspaces. As a result, the model estimates might not be very useful when two subspaces intersect, for example. Furthermore, since PCA attempts to determine directions of maximal variance, outliers that might arise due to the subspace clustering process can significantly affect both the dimension estimation and the basis computation. Another inherent challenge in using subspace clustering methods for exploratory data analysis is the need to validate the union of subspaces assumption on the data as part of the analysis. When the subspaces are not independent and disjoint, the affinity matrix is not block-diagonal and hence accurately estimating the number of subspaces is not possible. In practice, similar to other clustering methods, it is assumed that the user can input the number of subspaces. When the input number of subspaces is not accurate, the basis vectors provided by PCA might over-fit the data, and it might not be apparent from our analysis that the model assumption needs to be re-evaluated.

To address these challenges, we propose to use ideas from graph embedding to perform model estimation for each subspace. The graphs constructed using all three methods can contain edges, with non-zero weights, for a sample with other samples from both within its subspace, and other subspaces. For example, samples near the intersection of two subspaces might have contributions from samples in both subspaces. In order to infer a subspace model, while taking into account its relation to other subspaces, we extract a block-diagonal matrix from the coefficient matrix  $\mathbf{W}$ , corresponding to only the samples in that subspace to compute the basis vectors. For a subspace  $S_j$ , we denote the set of indices of samples belonging to that cluster by  $\Lambda_j$ . Since we are interested in computing a linear embedding, we pose the following problem similar to the one in [17]:

$$\mathbf{U}_j = \underset{\mathbf{U}}{\operatorname{argmin}} \sum_{i \in \Lambda_j} \left\| \mathbf{U}^T \mathbf{x}_i - \sum_{k \neq i, k \in \Lambda_j} W_{ik} \mathbf{U}^T \mathbf{x}_k \right\|_2^2 \quad \text{s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}. \quad (3.2.1)$$

Here the matrix  $\mathbf{U} \in \mathbb{R}^{D \times d_j}$  contains the set of basis functions, and  $d_j$

is the dimension of the subspace. We can rewrite this minimization problem as

$$\min_{\mathbf{U}} \mathbf{U}^T \mathbf{X}_j (\mathbf{I} - \mathbf{W}_j)^T (\mathbf{I} - \mathbf{W}_j) \mathbf{X}_j^T \mathbf{U} \quad \text{s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}, \quad (3.2.2)$$

where  $\mathbf{X}_j$  is the subset of samples belonging to cluster  $j$ , and  $\mathbf{W}_j \in \mathbb{R}^{T_j \times T_j}$  is the block-diagonal matrix with edge weights corresponding to only samples in  $\Lambda_j$ . The solution to this problem can be obtained using the generalized eigenvalue decomposition of  $\mathbf{X}_j (\mathbf{I} - \mathbf{W}_j)^T (\mathbf{I} - \mathbf{W}_j) \mathbf{X}_j^T$  and the basis  $\mathbf{U}$  is fixed as the eigenvectors corresponding to the  $d_j$  smallest eigenvalues.

Using the proposed embedding strategy, we develop a new technique for reliably estimating the subspace dimension  $d_j$ . Unless the number of subspaces is highly overestimated, the basis set of a particular subspace will be a poor fit to the data from other classes. However, subspace clustering algorithms might not provide a perfect subspace segmentation, and as a result some samples can be wrongly clustered into a subspace. PCA does not take this into account, and provides a model that fits the outlier samples also, which is not desirable for understanding relations between subspaces. In contrast, our embedding in (3.2.2) penalizes samples that have non-zero coefficients with samples from other subspaces and hence the resulting model is more meaningful. For a subspace  $S_j$ , we vary the number of dimensions  $d$  between 2 and  $D - 1$ , and for each case repeat the following steps:

- Compute the basis  $\mathbf{U}_j^d$  for the samples  $\mathbf{X}_j$  using (3.2.2).
- In the low dimensional space, construct a  $L2$  graph for samples in the low dimensional space and measure the reconstruction error for each sample.
- Compute  $E_{in}^d$  as the median reconstruction error for all within-cluster samples.
- Project samples from other subspaces,  $\{\mathbf{x}_i : i \notin \Lambda_j, i = 1 \dots T\}$ , onto the basis  $\mathbf{U}_j^d$ .
- Construct  $L2$  graph for the out-of-cluster samples and measure the reconstruction error for each sample.
- Compute  $E_{out}^d$  as the median reconstruction error for all out-of-cluster samples.

Here median error is more resilient to the outliers compared to the mean error. We fix  $d_j$  as the lowest dimension beyond which  $E_{in}^d < E_{out}^d$ , and  $E_{in}^d \leq \tau E_{in}^{D-1}$ . The parameter  $\tau$  balances between the number of dimensions and the desired reconstruction error. Figure 2 plots the errors  $E_{in}$  and  $E_{out}$  for varying values of the dimension  $d$ , for an example case. For this case, the dimension is estimated to be 3.

### 3.3 Dynamic View Transition for Subspace Exploration

In this section, we will first describe the underlying principle for dynamic transition between a pair of projections. After that a dynamic

view transition based subspace exploration scheme is proposed. The method we adopt for transition between a pair of bases was originally proposed by Buja [7] and implemented in their *GGobi* software. However, we use that general idea for the transition under a very different context. *GGobi* defines a series of transition targets, by either randomly generating the target (the grand tour scheme) or utilizing the projection pursuit, and then cycling through them one by one. This process is referred to as a “tour”. In contrast, the source and target projections in our proposed setup are obtained from low-dimensional subspaces describing the inherent structure of the data. And, we develop a navigation infrastructure that allows for intuitive exploration among the different views. In addition, the primary goal of our animation transition is to study the structure for each subspace and their relationships to other subspaces.

### 3.3.1 Transition Between Projection Planes

In the proposed framework, a linear projection (a view) can be defined using a projection matrix. Let  $\mathbb{D}$  denote the original data dimension, and  $d$  denote the desired dimension for the projection. We define the matrix  $\mathbf{F} \in \mathbb{D} \times d$  to denote a projection, where each column of the matrix is an orthonormal basis vector. Calculating the dynamic transition between projections is equivalent to generating a set of projection matrices  $\mathbf{F}$ , interpolating between  $\mathbf{F}_a$  and  $\mathbf{F}_z$ . Note that, for any useful visual analysis, we need to fix  $d = 2$ , and this implies we essentially attempt to smoothly interpolate between a pair of projection matrices defined using the basis vectors  $(f_{a1}, f_{a2})$  and  $(f_{z1}, f_{z2})$  respectively. Transitioning between a set of bases comprises of two components: in-plane rotation and rotation of the projection plane. The within plane rotation will only alter the orientation of the projection but the plane rotation will change the projection’s shape and visual pattern. From visualization standpoint the difference in the orientation will not impact shape and pattern recognition. Therefore, we could remove the in-plane rotation part in the transition calculation to achieve a more direct transition animation. However, this means we are no longer transitioning from  $\mathbf{F}_a$  and  $\mathbf{F}_z$ , instead, we rotate the plane  $\text{span}(f_{a1}, f_{a2})$  to the plane  $\text{span}(f_{z1}, f_{z2})$  as illustrated by the red and blue circles in Figure 3.

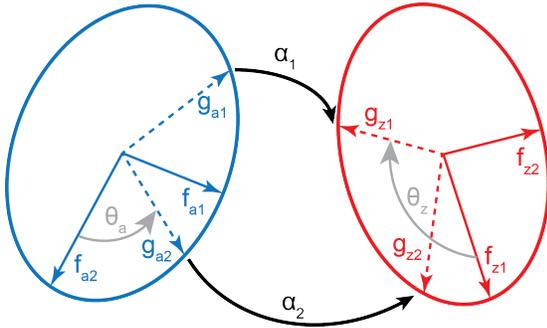


Fig. 3: Illustration of the relative positions of given basis  $(f_{a1}, f_{a2}), (f_{z1}, f_{z2})$  and the pairs of principal directions,  $(g_{a1}, g_{a2}), (g_{z1}, g_{z2})$ . The  $\theta_a$  and  $\theta_z$  are the in-plane rotation the  $\mathbf{F}_a, \mathbf{F}_z$  require to match  $\mathbf{G}_a, \mathbf{G}_z$ .

As shown in Figure 3, performing an in-plane rotation of  $\mathbf{F}_a$  by  $\theta_a$  results in the set  $\mathbf{G}_a = (g_{a1}, g_{a2})$ . Similarly we can form  $\mathbf{G}_z$  by rotating  $\mathbf{F}_z$  by  $\theta_z$ . We know that for two planes of dimension  $d$ , orthonormal  $d$  dimensional frames,  $\mathbf{G}_a$  and  $\mathbf{G}_z$ , that span the planes can exist, when  $\mathbf{G}_a^T \mathbf{G}_z$  is a diagonal matrix. Therefore the bases in  $\mathbf{G}_a$  and  $\mathbf{G}_z$  are orthogonal to each other. The columns of the  $\mathbf{G}_a$  and  $\mathbf{G}_z$  are the “principal directions” of the pair of planes, and the diagonal entry  $\lambda_j$  of the matrix  $\mathbf{G}_a^T \mathbf{G}_z$  corresponds to the cosine of the principle angles. Therefore, the geodesically shortest path between the planes can be obtained by rotating the columns of  $\mathbf{G}_a$  into the corresponding columns of  $\mathbf{G}_z$  based on the principle angle. In Figure 3,  $g_{a1}$  are orthogonal to  $g_{z1}$  and  $g_{a2}$  are orthogonal to  $g_{z2}$ ,  $\alpha_1$  and  $\alpha_2$  are the principal angles between these two planes.

The diagonal matrix can be easily computed using the singular value decomposition (SVD) of  $\mathbf{F}_a^T \mathbf{F}_z$ .

$$\mathbf{F}_a^T \mathbf{F}_z = \mathbf{V}_a \Lambda \mathbf{V}_z^T, \quad \Lambda = \text{diag}(\lambda_1 \geq \dots \geq \lambda_d), \quad (3.3.1)$$

$\mathbf{V}_a$  and  $\mathbf{V}_z$  are the in-plane rotation matrices, we now have

$$\mathbf{G}_a = \mathbf{F}_a \mathbf{V}_a \quad \text{and} \quad \mathbf{G}_z = \mathbf{F}_z \mathbf{V}_z, \quad \text{where} \quad \mathbf{G}_a^T \mathbf{G}_z = \Lambda \quad (3.3.2)$$

Following this, we can construct the rotation matrix which moves  $\mathbf{G}_a$  towards  $\mathbf{G}_z$ . As mentioned earlier, the rotation is not a direct mapping from  $\mathbf{F}_a$  to  $\mathbf{F}_z$ , instead we rotate  $\mathbf{F}_a$  to another basis  $\mathbf{F}'_z$  that shares the same span of  $\mathbf{F}_z$  to achieve a less complicated animation. Such a setup will cause the end rotation to have a different orientation compared to the target. In order to resolve the discrepancy, we update the target to  $\mathbf{F}'_z$  to achieve a smooth and continuous transition. Additional details about the formulation of this transition can be found in [7].

### 3.3.2 Subspace Exploration Pipeline

The dynamic transition between projection planes is the essential building block for our visualization system. When compared to existing works that utilize dynamic view transitions (grand tour, guided tour), the novelty of our approach lies in using the animated transition to understand the (union of) subspaces and their complex interrelations. The pipeline of the proposed interactive exploration tool is illustrated in Figure 1. The tool loads the original high-dimensional data along with the clusters and their corresponding models (dimension, bases) estimated during the pre-processing step. The subspace views, defined by a pair of bases vectors, are automatically generated for each subspace. Therefore, without duplications, we will have one view per 2D subspace, three views per 3D subspace and six views per 4D subspace, etc. Our subspace views are designed in a similar fashion as a scatter plot matrix. However, using linear subspaces instead of all possible axis-aligned subspaces, greatly simplifies the analysis. Since the estimated subspaces are typically low dimensional, the number of views is significantly lower than that of a scatter plot matrix. However, it is still challenging to organize the reduced number of views in a logical and meaningful way for effective navigation and visualization.

Once all the views are created, we construct a k-nearest neighbors graph, where the distance between a pair of views is determined by the maximum value of dot products among all pair of bases between the two selected views. A force-based graph layout algorithm is then applied to generate the spatialization of the view in the canvas. Our proposed view distance measure not only captures the similarity between pairs of projections, but also allows the views belonging to one subspace to be nicely clustered together (see the view navigation panel in Figure 5) to facilitate easy subspace navigation.

With the view navigation panel, we could navigate among views in the same subspace (intra-subspace transition), or between views belonging to different subspaces (inter-subspace transition). Due to the fact that our subspace bases can be described as some linear combination of the original dimensions, the axes are difficult to interpret, especially in a static plot. We propose to resolve this limitation by connecting these scatter plots with animated transitions, which can reveal important correspondence information between the views for understanding their structures. From the computational perspective, there is no difference between transitioning between subspace views, and transitioning between any arbitrary pairs of views. During the dynamic projection, we project all points indiscriminately based on the current interpolated bases, even though they belong to different subspaces. To understand the importance of the dynamic transition between a pair of subspace views, we need to revisit the subspace finding techniques. Though a high-dimensional dataset may contain highly complex structure, the idea of subspace learning assumes that certain parts of the complex structure can be well approximated by some low dimensional linear space. Subspace clustering techniques try to partition the data into clusters where within each cluster the points share the same set of bases. From a visualization standpoint, such decomposition can provide enough information for understanding the complex high-dimensional space through a divide and conquer paradigm. Since each subspace is much simpler than the overall data, we could first focus on each of the subspaces. Then based on the understanding of

the individual subspaces, we could then explore the relationship between subspaces through inter-subspace transitions. During such transitions, we could infer the structural relationship between subspaces through the motion pattern exhibited during the transition. In Section 5 we demonstrate example cases where we are able to discover patterns corresponding to distinct structure in the data that were not apparent otherwise.

## 4 IMPLEMENTATION

This section will discuss our user interface and available interactions as well as some implementation details and performance numbers.

### 4.1 User Interface & Interaction Design

Fig. 5 shows an overview of the system that the three main windows marked as (A), (B), and (C) respectively. We consider (A) the main canvas showing the dynamic projections as well as a number of informative insets. (B) is the view navigation panel showing the view transition graph from which the user drives the exploration. Finally, (C) contains the file menu, lets the user select which dimension reduction technique to apply and provides a raw data display for detailed queries.

**Dynamic projection panel.** This is the main canvas that displays the current, dynamically changing projection (A-1). This panel also include the controls for choosing colormaps, the type of transitions, and allows the user to turn on/off different visual components in the view. The points can be colored based on the subspace label, the original dimensional value or the ground truth class label. To prevent the view clutter when focusing on a subspace, we allow the user to fade out the non-active subspace. During a transition, if the source and target projections are from different subspaces we will fade-in and out the corresponding points in a continuous manner. In addition, we augmented the projection with basis axis or the original dimension axis, to provide a spatial reference during the transition. Alongside the projection view we display small insets (A-2) showing both the source and the target projections. Note that the target projection may exhibit some discrepancies from the final projection shown in the dynamic view, due to differences of the in-plane rotations (see Section 3.3). Finally, we provide an additional inset (A-3) to display the data in its natural domain, e.g., images as images, if applicable. Especially for the face and digit dataset (see Section 5) the ability to refer back to the original data has proven helpful.

**View navigation panel.** This panel provides the navigation interface for our dynamic view transitions. As illustrated in Figure 5(B), each view is represented by a node, where: a square node represents a scatterplot view using a subspaces basis; a circular node represents the global PCA view; and a triangular node is an axis-aligned projection. Node colors indicate the different clusters. Transitions are allowed between any pair of nodes.

**Data operation panel.** The final panel contains various operations such as loading data and manually applying projections. The panel is part of an interlinked system, so changes made to the dataset are instantly reflected through other views. The panel consists of three sub-panels. The meta-information panel provides a direct table view of the data, in terms of its dimensions and statistics. The dimensionality reduction panel enables the user to choose different dimensionality reduction techniques and specify their parameters. In our current application, only the PCA and dimension aligned projection are allowed (have meaningful linear basis). Once a projection is applied the corresponding view node will appear in the view navigation panel.

### 4.2 Software Implementation

Our software is divided into two major parts. The pre-processing code responsible for subspace clustering as well as dimension and basis estimation and the visualization tool provides an interactive exploration of the different views. For the preprocessing we have implemented three different subspace clustering techniques and the proposed basis estimation technique in MATLAB. The dynamic projection tool is an integral part of our high-dimensional data exploration framework. Our visualization system is implemented in C++, using Qt for the GUI

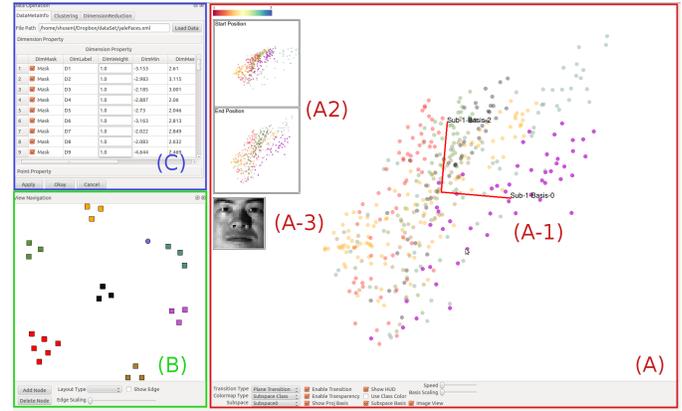


Fig. 5: The user interface of our interactive tool consists of three major parts: (A) the dynamic projection panel; (B) the subspace navigation panel; and (C) the data operation panel. In the dynamic projection panel, the dynamic projection, (A-1), augmented with an axis plot to provide a reference to the basis coordinate. The top inset (A-2) displays the source and target projection of the current transition. The bottom inset (A-3) provides a display of the original data if applicable.

components and point graphics drawing. The framework is designed to be easily extendable to include new visualization techniques, and provide an environment to combine different tools to complement their capabilities.

Here we gives a brief summary of our performance and timing information. For an interactive application smooth visual feedback and a responsive interface is important for a user to fully utilize the system. Our dynamic transition requires a linear projection re-calculation of the whole dataset in every frame, therefore an efficient implementation is important. The projection code is implementation in C++, using the *Eigen* linear algebra library [15] for faster matrix multiplication. For all of the datasets we tested, we achieve between 10-35 frame per second depending on the workload. For the pre-processing step, the full MNIST data took around 15 minutes; the Yale faces around 7 minutes; and the combustion data around 3 minutes.

## 5 RESULTS

In this section we demonstrate our system using a mix of well-known reference datasets, some application specific data, and a synthetic dataset for illustration.

### 5.1 Synthetic Dataset

To provide some intuition on how to interpret the animations between different projections and to setup the examples in the remainder of the section we use a simple synthetic dataset containing two intersecting 2D planes embedded in 3D. Figure 4 shows the transition left-to-right from one subspace (colored purple) to the other (colored black). The intersecting relationship becomes obvious in the transition, providing important insight not easily obtained from traditional axis-aligned views. Furthermore, note how the black subspace is projected onto a near perfect line in the purple view (left) yet the reverse is not true for the purple subspace in the black view (right). This mismatch indicates different spatial extents of both planes. In this case, both planes cover the entire  $xy$ -plane of the unit box yet the purple one is tilted in space and thus larger leading to the spread in the black view. Instead, in a perfectly symmetrical setup the projections also would have been symmetrical.

### 5.2 Combustion Simulation Dataset

The combustion simulation dataset contains a collection of 2.8K samples from a large scale turbulent simulation [16]. Each sample has the concentration of ten chemical components, such as  $H_2$ ,  $O_2$ , etc., as attributes and scientists are interested in understanding the shape of the underlying “chemical manifold” and especially how it might related to phenomena such as extinction or re-ignition.



Fig. 4: Two intersecting planes. Animated transitions between the purple and the black subspace.

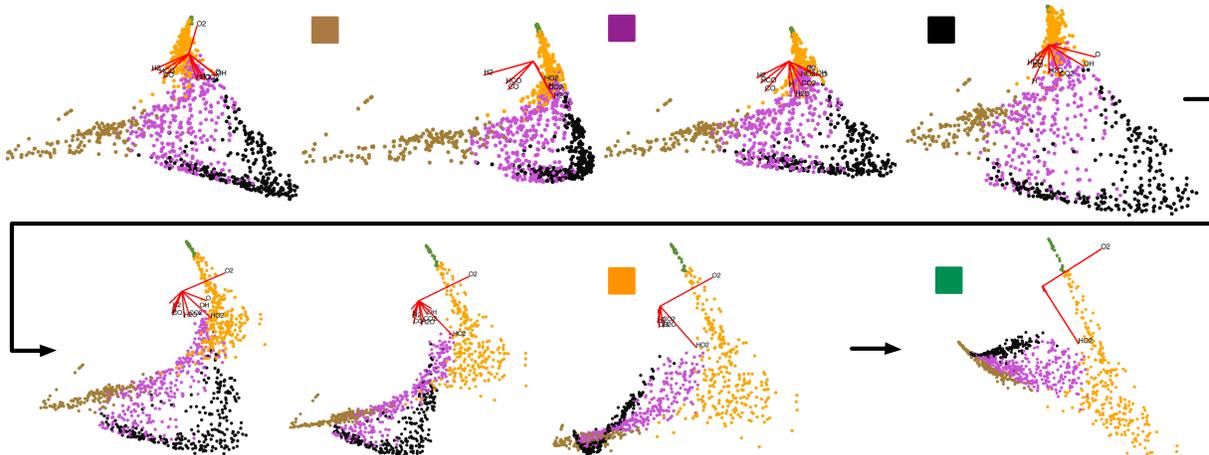


Fig. 6: Combustion. From left to right, from top to bottom: transitions from the PCA View, to the brown, purple and black Subspace Views; Animated transitions from the (top row) black subspace to the yellow subspace; and to the green subspace.



Fig. 7: Face. From left to right: transitions from the PCA View, to three of the yellow Subspace Views.

Subspace clustering results in five 2D subspaces. In Figure 6(top), we start our exploration from the PCA View. We then dynamically transition to the brown subspace View, then to the purple and eventually to the black subspace View. By rotating the bases associated with each view, we observe a small amount of tilting during the transitions. This observation helps us to develop an intuitive understanding that these 2D subspaces are glued together with small rotational angles and likely describe a gently curved, non-linear structure to some extent also visible in the PCA View.

However, transitioning from the black subspace to the orange one, causes a drastic expansion of the orange subspace and a compression of the brown, purple and black subspaces (see Figure 6(bottom left)). The animation indicates that the orientation of the orange subspace is very different from the other three subspaces we have just visited. Such an effect could also be observed in the axis plot, where the  $O_2$  and  $HO_2$  concentrations are the only dominating dimensions for the orange subspace, where other subspaces span among more dimensions. Finally, we transition from orange to the green subspace, where the animation demonstrates their similarities in terms of the small rotational angle (see Figure 6(bottom right)).

Overall, one can infer that the brown, purple, and black subspaces share structural similarities; the green and orange spaces are closely related; yet both sets of subspaces are structurally very different. In addition, based on domain knowledge and previous studies we know that in the PCA View, parts of the orange and green subspaces exhibit large projection distortion compared to the rest of the points. Our visualization partially validates the previous discoveries and provides a likely explanation. PCA is known to maximize variance across all dimensions, which is visible through the axis plot of the PCA View. However for the green and orange subspaces, only two out of the ten

dimensions exhibit a large variance. As a result, the bases of the green and orange subspaces do not align well with the globally computed basis obtained from the PCA, resulting in large projection distortion.

### 5.3 Yale Face Dataset

The Yale Face dataset is a subsample from the original database [5]. It consists of 439 face images from seven people we roughly label as (in no particular order): one Asian female, two Asian males, one Caucasian male, one Indian male, one African female and one middle eastern male. The original images have a resolution of  $32 \times 32$ . For visualization purpose, we use random projection to reduce their resolution to  $10 \times 10$ , therefore the point cloud is embedded in 100D space. Our subspace clustering returns six 3D subspaces and one 4D subspace.

We start our exploration from the global PCA View where the yellow points appear to form a ring-like structure displayed in Figure 7. To verify such a structure, we transition to a yellow Subspace View for further intra-cluster investigation. By enabling subspace-based transparency, we highlight the selected subspace (yellow) by making other points more transparent. Via intra-cluster exploration with animated transitions, we can clearly identify the ring-like structure for the yellow subspace, and infer its structural separation from the remaining point cloud (see Figure 7). Referring back to the face images, for the yellow subspace, we can verify that the mis-classified points (which are highlighted by colors other than yellow in Figure 9a) exist in the intersection between the yellow subspace and the rest of the point cloud.

Similar intra-cluster transitions could be apply for exploring other subspace individually. The brown and dark green subspaces contain structures that are visibly separated from the rest of the of the dataset (see Figure 9b). Referring back to the face images, the brown and dark



Fig. 8: Face. From left to right: animated transitions from dark green to purple Subspace View.

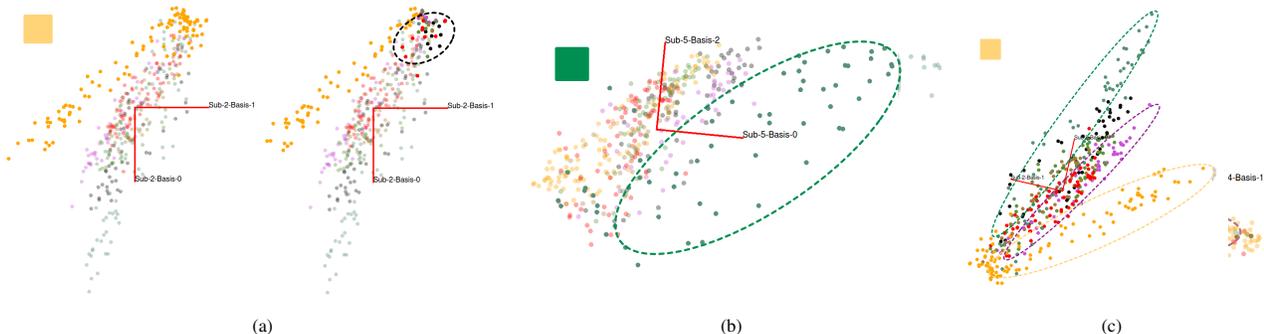


Fig. 9: (a) Face. Yellow subspace: mis-classification (circled by black dotted line) exists where the yellow subspace intersects with the rest of the point cloud. (b) Face. Dark green and brown Subspace Views with good separation from the rest of the point cloud. (c) Yellow Subspace View contains three stratified sets. Which corresponding to two asian male and one asian female.

green points correspond to images from an Indian male and an Asian male respectively. Which indicate the possible distinct facial features they possess.

To study the relationships between multiple subspaces we apply a number of inter-cluster transitions. As illustrated in Figure 9b, we start the transitioning between a dark green subspace to a purple subspace. The purple points rotate towards the dark green points and cross one another, while remaining separated from the rest of the point cloud. Visually, they appear to be partially overlapping (Figure 9b). Referring back to the face images, it turns out that these two subspaces correspond to two Asian males. The similarities between their facial features result in proximities between their corresponding subspaces, indicated by small rotational angles during the transitions. In addition, when we transition from the dark green subspace to the previously explored yellow subspace, we see that the dark green, purple, and yellow points form three stratified sets in the visualization (Figure 9c). The yellow subspace corresponds to the Asian female and the three stratified sets to the other faces from people with Asian origin.

#### 5.4 MNIST Dataset

The MNIST dataset is a sampled subset from the MNIST handwritten digits database. The original images have a resolution of  $28 \times 28$ . In our example, we downsample the images into  $12 \times 12$  and use 500 samples for the visualization for better interactive performance (in terms of generating smooth animations), which is essential for effective data exploration. We obtain three 3D subspaces and seven 2D subspaces from the subspace clustering.

From the PCA View in Figure 10 (first view from left), we notice the black points seem to reside in an area that is separable from the rest of the points. To investigate its connections with the other subspaces, we transition between the global PCA view to one of the black Subspace Views (Figure 10). As we perform the basis rotation towards the black subspace View, black points start to expand while other points begin to compress. Such compression is especially visible for the dark green and the purple points.

To understand the relationships among the black, the dark green, and the purple subspaces better, we perform inter-cluster transitions from the purple to the black subspace. As illustrated in Figure 11, the black cluster expands while purple and dark green clusters compress and become visibly overlapping. This indicates that the black points and the purple/dark green points reside in subspaces that are structurally distinct. Referring back to the original handwritten images, it

turns out that both the purple and the dark green subspace contain the handwritten digit 1. The only difference is that digits that are part of the purple subspace are slightly tilted, while the digits in the dark green cluster are mostly written vertically. This difference implies that the digit 1 images could not be well-approximated by a single 2D subspace due to these variations. Meanwhile, the black subspace contains mostly the handwritten digit 0, which is probably the digit that is least likely to be confused with digit 1. Therefore, there is no surprise the subspaces lie on the opposite ends of the high-dimensional space.

Similarly, as illustrated in Figure 12, while transitioning between a red Subspace View and a blue Subspace View, we observe the compression of the red cluster and expansion of the blue cluster. This implies that these two subspaces are also likely very different from each other. Referring back to the original images, the red cluster corresponds to the digit 6 while the blue cluster corresponds to the digit 7. The differences between their shapes likely contribute to the very different subspaces these two digits reside in. Since similar movements have been observed for the combustion dataset, we suspect that this type of compression/expansion patterns likely indicates big structural differences between source and target subspaces.

We now focus on the cyan and blue subspaces. In the PCA View, the cyan and blue cluster appear to partially overlap. To validate that such an observation is not just a projection artifact, we perform inter-cluster transitions between the cyan and the blue subspace, as well as intra-cluster transitions among views of the blue subspace. Figure 13 demonstrates that blue and cyan clusters remain intermingled during such transitions. Referring back to the original images, the cyan cluster corresponds to digit 9 and the blue cluster corresponds to digit 7. Their similarities in handwritten digits contribute to the observed difficulty in separating these two clusters.

## 6 DISCUSSION AND FUTURE WORKS

This paper presents an interactive framework for exploring linear low dimensional subspaces in high-dimension data, through animation transitions between pairs of subspace views. Through extensive case study, we demonstrate the effectiveness of our approaches for finding hidden and interesting low dimensional structure. Compared to previous techniques that rely on subset of dimension or visual interaction to identify the subspaces, the subspace clustering combined with robust dimension and bases estimation we proposed, capture more accurate and meaningful subspace structures. In addition, these subspace structural information is exploited in our interactive tool to provide

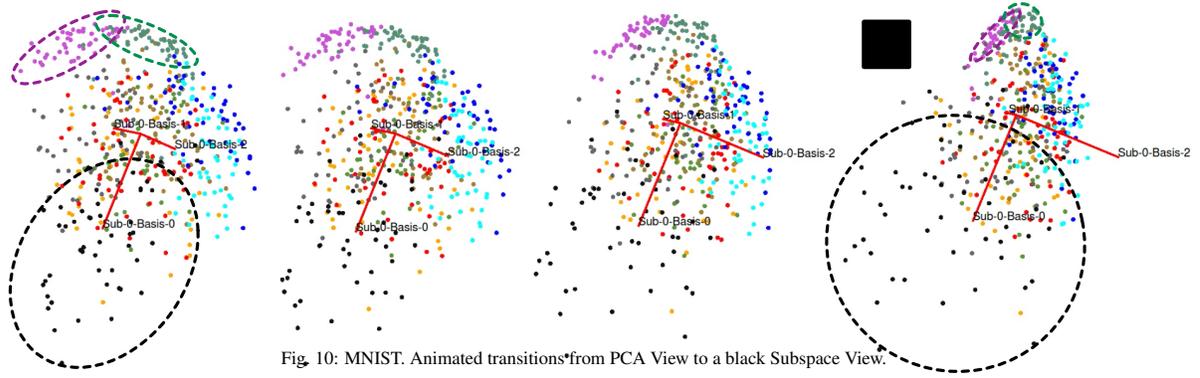


Fig. 10: MNIST. Animated transitions\*from PCA View to a black Subspace View.



Fig. 11: MNIST. Animated transitions from a purple to a black Subspace View where purple and dark green cluster appear to merge.

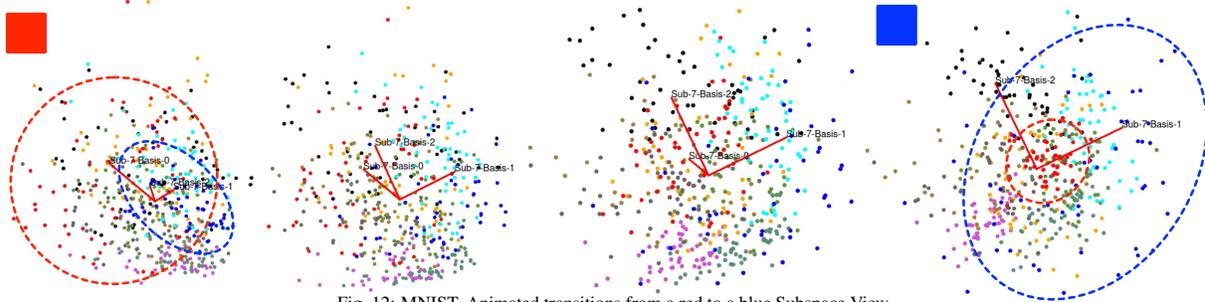


Fig. 12: MNIST. Animated transitions from a red to a blue Subspace View.

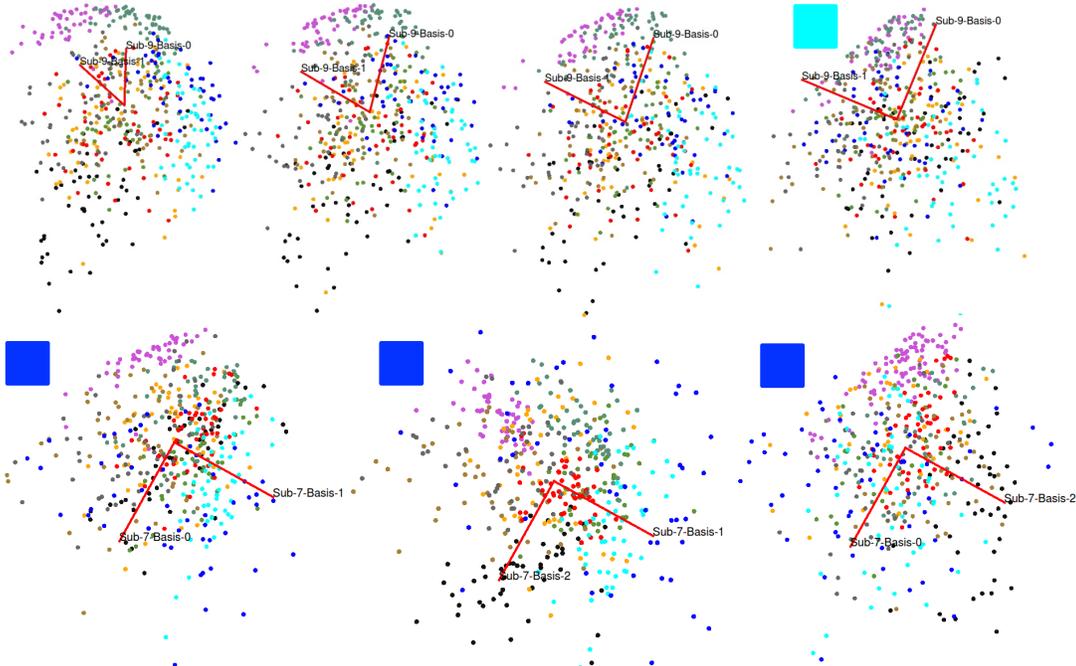


Fig. 13: MNIST. Transitions from a PCA View, to cyan, then to three of the blue Subspace Views.

new visual cues. One downside of the system is the heavy processing loads especially for high dimensions that prompted us to down-sample some data. One of the goals in future works will be to improve this performance through parallel implementations or the use of GPUs. In addition, we plan to further explore the various repeating patterns found during the animated transition, such as the compression/expansion connection, to better understand their connection with the underlying high-dimensional structure.

## REFERENCES

- [1] K-subspace clustering. volume 5782 of *Lecture Notes in Computer Science*. Springer, 2009.
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.
- [3] A. Anand, L. Wilkinson, and T. N. Dang. Visual pattern discovery using random projections. In *VAST*, pages 43–52. IEEE, 2012.
- [4] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.
- [5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projections. *IEEE Pattern Analysis and Machine Intelligence*, 19(7), 1997.
- [6] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1), Jan. 2000.
- [7] A. Buja, D. Cook, D. Asimov, and C. Hurley. Computational methods for high-dimensional rotations in data visualization. *Handbook of statistics: Data mining and data visualization*, 24:391–413, 2005.
- [8] G. Chen and G. Lerman. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330, 2009.
- [9] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93. ACM, 1999.
- [10] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE CVPR*, 2009.
- [11] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE TVCG*, 14(6):1539–1148, 2008.
- [12] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. 1973.
- [13] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, 1990.
- [14] M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE TVCG*, 19(12):2042–2051, 2013.
- [15] G. Guennebaud, B. Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [16] E. R. Hawkes, R. Sankaran, P. P. Pébay, and J. H. Chen. Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: II. Parametric study. *Combustion and Flame*, 145:145–159, 2006.
- [17] X. He, D. Cai, S. Yan, and H.-J. Zhang. Neighborhood preserving embedding. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, 2005.
- [18] C. Hurley and R. Oldford. Graphs as navigational infrastructure for high dimensional data spaces. *Computational Statistics*, 26(4):585–612, 2011.
- [19] A. J. Izenman. Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(5):439–446, 2012.
- [20] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 2013.
- [21] J. E. Nam and K. Mueller. Tripadvisor nd: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE TVCG*, 19(2):291–305, 2013.
- [22] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 2001.
- [23] X. Peng, L. Zhang, and Y. Zhang. Constructing l2-graph for subspace learning and segmentation. *preprint arXiv:1209.0841v4*, 2014.
- [24] H. Sanftmann and D. Weiskopf. 3d scatterplot navigation. *IEEE TVCG*, 2012.
- [25] D. F. Swayne, D. Temple Lang, A. Buja, and D. Cook. GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43:423–444, 2003.
- [26] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *VAST*, pages 63–72. IEEE, 2012.
- [27] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [28] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE TVCG*, 18(12):2621–2630, 2012.
- [29] R. Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magazine*, 2011.
- [30] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *CoRR*, 2012.
- [31] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *INFOVIS*, volume 5, page 21, 2005.
- [32] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proceedings of ECCV*, 2006.
- [33] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE TVCG*, 19(12):2625–2633, 2013.