Robust Subspace Clustering by Combined Use of kNND Metric and SVD Algorithm

Qifa Ke and Takeo Kanade
Computer Science Department, Carnegie Mellon University
{ke+,tk}@cs.cmu.edu

Abstract

Subspace clustering has many applications in computer vision, such as image/video segmentation and pattern classification. The major issue in subspace clustering is to obtain the most appropriate subspace from the given noisy data. Typical methods (e.g., SVD, PCA, and Eigen-decomposition) use least squares techniques, and are sensitive to outliers. In this paper, we present the k-th Nearest Neighbor Distance (kNND) metric, which, without *actually clustering the data, can exploit the intrinsic data cluster structure to detect and remove influential outliers as well as small data clusters. The remaining data provide a good initial inlier data set that resides in a linear subspace whose rank (dimension) is upper-bounded. Such linear subspace constraint can then be exploited by simple algorithms, such as iterative SVD algorithm, to (1) detect the remaining outliers that violate the correlation structure enforced by the low rank subspace, and (2) reliably compute the subspace. As an example, we apply our method to extracting layers from image sequences containing dynamically moving objects.*

1 Introduction

Subspace clustering has many applications in computer vision, such as image and video segmentation [18], layer segmentation [12], and pattern classification (see [7]). In these approaches, the input high-dimensional data are projected or mapped onto a low dimensional subspace where the clusters become tighter and easier to identify.

The major issue of subspace clustering is to obtain the most appropriate subspace from the given noisy data. Traditional approaches to subspace estimation, such as Singular Value Decomposition (SVD), Principal Component Analysis, and Eigen-Decomposition, are based on the least squares technique, and are therefore sensitive to outliers in the given data (see [5]).

In previous approaches, outliers are detected using subspace constraint itself [19, 5], i.e., inliers must lie in the subspace, while outliers violate the correlation structure enforced by subspace constraint. The subspace is computed using the following two iterative steps: (1) compute the subspace model using current set of weighted data; (2) reweight each data item based on its distance to the current subspace model. Such iterative approach is sensitive to initialization (the weight of each data item, or the subspace model). Good initialization is required for the algorithm to converge to a desirable optimal solution. When the rank (dimension) of the subspace is unknown, some heuristic methods are usually included in the iterative process to estimate the subspace dimension [5], which makes the behavior of the iterative algorithm more un-predictable.

This paper presents a robust subspace clustering technique that exploits another data constraint to address the above difficulty of *initialization* and *rank detection*. One should note that in clustering applications, besides residing in the subspace, the inliers must also reside in some clusters. In other words, isolated or sparse data items not in any clusters are considered to be outliers. For convenience we name such constraint the *cluster constraint*. We use the kNND metric (1) to exploit the cluster constraint to detect outliers, but *without* actually clustering the data; and (2) to bound the subspace dimension. By using kNND, we can obtain a good initial inlier data set that resides in a linear subspace whose rank (dimension) is upper-bounded. Such subspace constraint can then be exploited by some simple algorithm, such as iterative SVD algorithm, to (1) detect the remaining outliers that violate the correlation structure enforced by the subspace, and (2) reliably compute the subspace.

As an example, we apply our approach to extracting layers from image sequences using subspace clustering. Layer representation has many important applications, including video compression, motion and scene analysis, and 3D scene representation [9, 25, 13, 1, 21]. Various approaches have been proposed for layer extraction [8, 9, 25, 16, 26, 17, 15, 24, 12, 27]. The subspace approach [12] has the advantage of avoiding unstable grouping in high dimensional motion parameter space. As a grouping approach, it also avoids the dominant plane assumption and the difficulty of layer initialization that are typically encountered in layer motion segmentation. However, there are two unsolved problems in the subspace-based approach. First, it is not clear how to guarantee a low-dimensional subspace for a video of dynamic scene. Second, the subspace is estimated by SVD algorithm, which is sensitive to outliers that often exist in local motion measurements. We will apply our robust subspace analysis technique to address these two problems.

0-7695-2158-4/04 $20.00 (C) 2004 IEEE

2 Subspace and Clustering

In subspace clustering, the input high dimensional data are first projected or mapped onto some low dimensional subspace, where the clusters become tighter and easier to identify. The major issue in subspace clustering is to obtain the most appropriate subspace from the given noisy data for data mapping or projection. In this section we review existing approaches to obtaining the subspace. We will show that these approaches are sensitive to outliers.

There are two popular approaches to map the data onto a low-dimensional subspace: the linear data approximation approach (e.g., SVD or PCA), and the non-linear spectral mapping.

2.1 Linear Subspace as Data Approximation

Let $\mathtt{W}_{M\times N} = [\mathbf{m}_1, \cdots, \mathbf{m}_N]$ be the $M \times N$ matrix that is formed by the input data, where each column $\mathbf{m}_i$ is an input data item, M is the dimension of input data, and N is the number of input data items. We center $\mathtt{W}$ by subtracting its column mean.

The SVD algorithm computes the d-dimensional linear subspace determined by $\mathtt{W}$:

$$\mathtt{W}_{M\times N} = \mathtt{U}_{M\times N}\Sigma_{N\times N}\mathtt{V}^{\top}_{N\times N} \tag{1}$$

The diagonal elements of Σ are the singular values λ_i of $\mathtt{W}$ in non-increasing order. The subspace dimension d (the actual rank of $\mathtt{W}$) is detected by [10]:

$$d_t = \arg\min_m \left(\frac{\sum_{i=0}^{m} \lambda_i^2}{\sum_{i=0}^{N} \lambda_i^2} > t \right) \tag{2}$$

Here $1-t$ determines the noise level we want to tolerate. The first d columns of $\mathtt{U}$ in equation (1) form the bases of the signal subspace. Under Gaussian noise assumption, this rank d linear subspace approximates $\mathtt{W}$ in an optimal way in the sense of least squares error.

Linear subspace approximation for clustering works best when the input high dimensional data reside in a low dimensional linear subspace. In clustering applications, assuming Gaussian noises the clusters by themselves enforce an $(L-1)$ dimensional linear subspace, where L is the number of clusters in the input data (see [7], pp.91). Linear subspace constraint can also be enforced by some intrinsic physical constraint underlying in the process of data formation, such as the rank-3 subspace in [23], or the homography subspace [29].

2.2 Nonlinear Spectral Mapping

While the linear subspace approximation looks for a subspace that best preserves the signal energy, the spectral mapping [18, 14] seeks a nonlinear subspace such that the data mapped onto it are optimal clustering.

Spectral mapping [18, 14] starts by first collecting the distance (similarity) between every two data points to construct the normalized affinity matrix $\mathtt{L} = \mathtt{D}^{-1/2}\mathtt{S}_{N\times N}\mathtt{D}^{-1/2}$. Here $\mathtt{S}$ is defined by:

$$S_{ij} = \begin{cases} \exp(-\frac{(\|\mathbf{m}_i - \mathbf{m}_j\|^2)}{\sigma^2}) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3}$$

where $\mathtt{D}$ is a diagonal matrix defined by $D_{ii} = \sum_j S_{ij}$.

The first K largest eigenvectors of $\mathtt{L}$ are stacked in columns to form the matrix $\mathtt{X}_{N\times K}$. Here K is the number of clusters. Each row of $\mathtt{X}$ is normalized to have unit length so that each row of $\mathtt{X}$ is a point on a K-dimensional sphere. The original data item $\mathbf{m}_i$ is mapped to the i-th row of $\mathtt{X}$, i.e., the i-th point on the K-dimensional sphere.

Spectral mapping therefore maps the original M dimensional data points onto the K-dimensional sphere, which is a $(K-1)$-dimensional non-linear space. Such non-linear low dimensional embedding can greatly improve the clustering structure of the input data, so that some simple clustering algorithm can easily identify the clusters. Note that spectral mapping requires the knowledge of K, the number of clusters.

2.3 Basic Subspace Clustering Algorithm

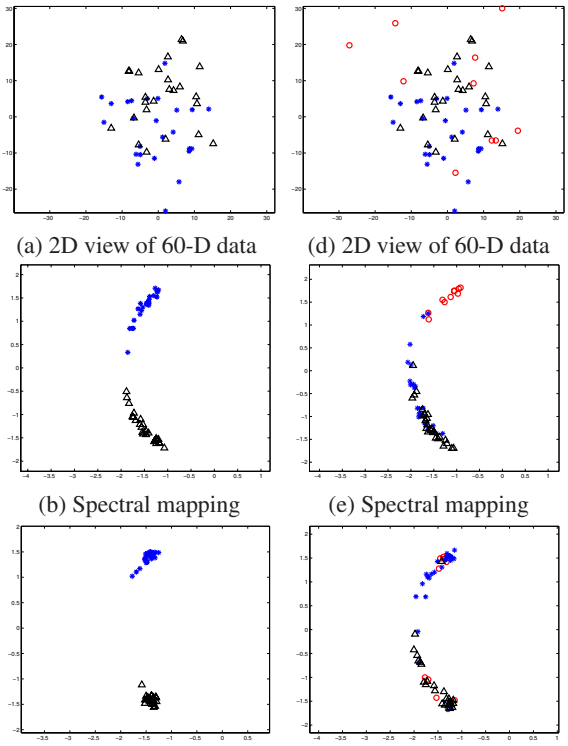
Assuming Gaussian noises, if the given high dimensional data reside in a low dimensional linear subspace, then projecting them onto the subspace will make the clustering structure more discriminative. In [11] we prove that such linear subspace projection can improve the cluster discriminability by a factor of $\frac{M}{d}$, where M is the original data dimension, and d is the subspace dimension. Better cluster discriminability will provide a better affinity matrix for spectral mapping. Based on above observations, we adopt the following basic algorithm for subspace clustering:

- Project the data onto the global linear subspace;
- If the number of clusters K is unknown, apply mean shift algorithm [4] to the projected data to detect the number of modes to initialize K;
- Use spectral mapping to map the projected low dimensional data onto a K-dimensional sphere.

2.4 Effect of Outliers

The SVD in the linear subspace approximation and the Eigen-decomposition in the spectral mapping are sensitive to outliers. One may notice that Equation (3) in the spectral mapping is one kind of robust distance function. However, it reduces the effect of one outlier *only if* such outlier is far away from all other data points (including other outliers). In real applications, if two or more outliers are close to each other, they will have unwanted significant contribution to the affinity matrix and therefore to the eigen-decomposition in spectral mapping. Another difficulty in Equation (3) is the selection of appropriate σ. Without a good σ, the eigensystem will be either unstable, or effected by outliers.

We use a synthetic example to illustrate the effect of outliers. We generate two Gaussian clusters (each has 25



(a) 2D view of 60-D data (d) 2D view of 60-D data

(b) Spectral mapping (e) Spectral mapping

(c) Basic Subspace clustering (f) Basic Subspace clustering

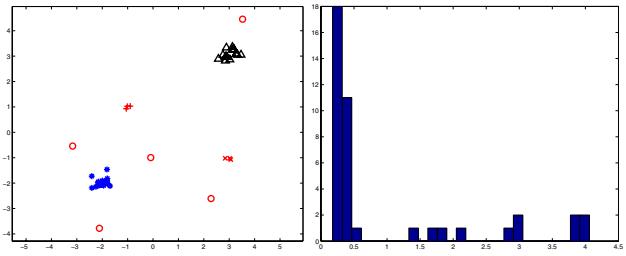
Figure 1. The effect of outliers on subspace analysis. (a–c): Spectral mapping of data without outliers. (d–f): Spectral mapping of data with outliers. See text for details.

points) in 60-D space. The first two dimensions of the 60-D data are shown in Figure 1(a). These 50 inliers in the two clusters are shown by $\triangle$ and $*$. The spectral mapping of these 60-D inliers is shown in Figure 1(b). The mapped data look separable, but it is still difficult for simple clustering algorithm to identify the two clusters. Since the two Gaussian clusters reside in a 1D subspace, we first project the data onto this subspace, and then apply spectral mapping to the projected 1D data. Figure 1(c) shows the spectral mapping of the projected 1D data. As we can see, two clusters are clearly distinguishable.

Now let us add 10 outliers to the original data set, shown as o in Figure 1(d). The spectral mapping of the original 60-D data is shown in Figure 1(e). Two clusters and the outliers are not separable in the mapped data. Figure 1(f) shows the spectral mapping of the projected 1D data. Due to outliers the subspace estimated by SVD is incorrect, and the two clusters are not separable.

3 Robust Subspace Clustering

In subspace clustering applications, there are two constraints available to detect the outliers: (1) the global linear subspace constraint, i.e., the inliers must reside in a low dimensional linear subspace; (2) the cluster constraint, i.e., the inliers must reside in some clusters instead of being isolated or sparsely distributed.



(a): Input data; (b): 1D histogram of kNNDs.

Figure 2. Example to illustrate the kNND procedure. There 41 data points in (a), including: (1) two large clusters each containing 15 data points shown by $\triangle$ and $*$; (2) two small clusters each containing 3 data points shown by $\times$ and $+$; and (3) five sparse outlier points shown by o.

The subspace constraint has been used for robust subspace estimation [19, 5], where the following two steps are iterated until converge: 1) compute the subspace model using weighted data; 2) re-weight the data using the distance between each data item and the current subspace model. Such iterative method, however, requires good initialization (the subspace dimension and the set of inliers).

We resolve the above difficulty of initialization by enforcing the cluster constraint using kNND metric. Specifically, we use kNND metric to provide a good initialization of inlier-outlier classification, and an upper-bound on the subspace dimension, which are then used by simple iterative algorithms (1) to detect the remaining outliers that violate the subspace constraint and (2) to reliably compute the subspace.

3.1 kNND for Outlier Detection

We use kNND metric to utilize the intrinsic cluster structure to detect and remove outliers and small clusters (size less than k), *without actually performing data clustering*. The remaining data reside in a linear subspace with its dimension upper-bounded.

The kNND metric for outlier detection is based on the fact that a data point has small kNND if it is in a cluster with size larger than k; otherwise, its kNND will be large. Denote x_i as the kNND of the i-th data point. The data points in *all* clusters with size larger than k will form the first dominant peak in the *one-dimensional* histogram of $\{x_i \,|\, i = 1, \cdots, N\}$. Points not in this peak have large kNNDs, and are removed as outliers. The size of any remaining cluster must be larger than k.

We use a simple example in Figure 2 to illustrate the above kNND procedure. In this example, if we set $k = 5$, the two clusters whose sizes are larger than 5 will form the first peak in the 1D histogram, as shown in Figure 2(b). The sparse points and the two small clusters will be removed as outliers since they are not in the first peak of the one-dimensional histogram.

The major advantage of using kNND is that it transforms the high dimensional data to a *one-dimensional* histogram, where the outliers are clearly distinguishable from inliers that form the first dominant peak. Detecting the first peak in such 1D histogram is easy. We simply smooth the 1D histogram and then detect the first peak by looking for the first and largest local maximum. A good initialization of inlier-outlier classification can therefore be obtained by labeling the data items not in the first peak as initial outliers.

We use EM algorithm [6] to refine the final inlier-outlier classification result by maximizing the likelihoods of the observed kNNDs, i.e., the normalized 1D histogram of $\{x_i \,|\, i = 1, ..., N\}$. We model the likelihoods by the mixture distributions of inlier's and outlier's kNNDs:

$$Pr(x) = \gamma f(x, \lambda) + (1 - \gamma)\frac{1}{v} \tag{4}$$

Here γ is the mixing parameter. $f(x, \lambda)$ is the distribution of inlier's kNND, which is modelled by Poisson distribution (see [3]), with λ its only parameter representing the dense of the data points (the rate of point process). $\frac{1}{v}$ is the uniform distribution of outlier's kNND, where v is determined by the range of x (i.e., $v = \max(x_i) - \min(x_i)$). By experiments we have observed that the distribution of outlier's kNND is well approximated by uniform distribution (see also [20]).

The missing component in the EM algorithm is the indicator for each data item, denoted by $\eta_i \in \{0, 1\}$, where $\eta_i = 1$ if the i-th data item is an inlier, and $\eta_i = 0$ if the i-th data item is an outlier. The EM algorithm estimates the parameters $(\eta_i, \gamma, \lambda)$ by maximizing the likelihood in Equation (4). It consists of the following iterative E-step and M-step:

E-step: Estimate the indicator variables:

$$Pr(\eta_i = 1 \,|\, \gamma, \lambda) = \frac{f(x_i, \lambda)}{\gamma f(x_i, \lambda) + (1 - \gamma)\frac{1}{v}}$$

M-step: Compute the parameters based on the indicators given in E-step:

$$\gamma = \frac{\sum_{i=1}^{N} \eta_i}{N}, \qquad \lambda = \frac{k \sum_{i=1}^{N} \eta_i}{\alpha_M \sum_{i=1}^{N} (x_i)^M \eta_i}$$

Here $\alpha_M = \frac{2\pi^{M/2}}{d\Gamma(M/2)}$ is a constant in the Poisson distribution; $\Gamma(\cdot)$ is the Gamma function; and M is the data dimension.

Since the initial inliers from the first dominant peak of the 1D histogram provide a good initialization of the missing component $\eta_i \in \{0, 1\}$, the above EM algorithm converges quickly. The final inlier-outlier classification is based on the ratio ρ:

$$\rho_i = \frac{Pr(\eta_i = 1 \,|\, \gamma, \lambda)}{Pr(\eta_i = 0 \,|\, \gamma, \lambda)} \tag{5}$$

Point i is marked as an inlier if $\rho_i > 1$, or outlier if $\rho_i \leq 1$.

kNND was used to model point process in clutter backgrounds [3], where the goal was to separate points into two Poisson processes with different rates λ. It was also used to determine the window size in kernel density estimation [7], which can also be used to detect sparse points. We directly use kNND for outlier detection in subspace clustering application, because of its following advantages:

- kNND transforms the high dimensional data into 1D data, where outliers are clearly distinguishable from inliers. Therefore kNND implicitly uses the cluster constraint to detect outliers, but does ***not*** actually cluster the data.
- The parameter k is directly related to the expected cluster size, and therefore can be used to guarantee a low dimensional subspace (Section 3.2).
- kNND is adaptive to cluster shapes. This is different from sparse point detection using kernel-based density estimation, where the window shape is predetermined.

3.2 kNND for Bounding the Dimension of Subspace

Suppose the given data set contains L clusters. Assuming Gaussian noises, the centroids of these L clusters form an $(L-1)$-dimensional linear subspace[1] (see [7]). We can therefore reduce the subspace dimension by reducing the number of clusters. This is done by applying kNND metric to remove small clusters with size less than k.

Suppose the number of input data is N. After the kNND procedure, the size of a survived cluster is expected to be larger than k. The number of survived clusters, denoted as L, must be bound by:

$$L \leq \frac{N}{k}$$

The subspace dimension d is then bound by:

$$d \leq L - 1 \leq \frac{N}{k} - 1$$

We can assure the subspace dimension to be lower than a pre-defined value d_0 by setting the parameter k as:

$$k \geq \frac{N}{(d_0 + 1)} \tag{6}$$

3.3 Robust Subspace Estimation

Our algorithm for robust subspace estimation consists of the following four steps:

1. Use kNND to initialize the set of inliers with weights initialized to 1, and the upper-bound of subspace dimension d_u.
2. Apply SVD and rank detection (Equation (7)) to the weighted inliers to compute the subspace model.
3. Use subspace constraint to detect the remaining outliers and re-weight the inliers (details in Appendix).

[1] In many computer vision applications $L - 1$ is usually much smaller than the input data dimension.

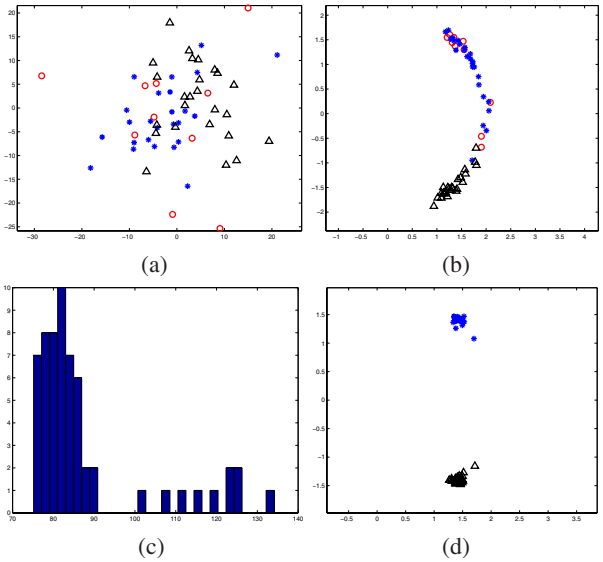


Figure 3. A synthetic example for comparing subspace clustering with and without robust analysis using kNND. (a): First two dimensions of the 60-D synthetic data (with 10 outliers shown by o); (b): Spectral mapping without robust subspace analysis; (c): The 1-dimensional histogram of kNNDs ($k = 8$); (d): Spectral mapping with robust subspace analysis.

4. Repeat Step 2 and 3 until converge.

In Step 2, the subspace dimension d (rank of $\mathtt{W}$) can be computed by:

$$d = \min(d_u, d_t) \tag{7}$$

Here d_u is the upper-bound of the subspace dimension guaranteed by kNND according to Equation (6), and d_t is the expected-noise-bound subspace dimension computed by Equation (2). By enforcing the upper-bound d_u in the rank detection, we are guaranteed a global linear subspace constraint. Since kNND provides a good initialization of inliers, the above iterative procedure will converge quickly to a desirable subspace model.

Finally, to cluster the spectral-mapped data, we can use simple clustering algorithms, such as mean shift, k-means, or a more advance approach in [28]

Figure 3 shows an example of applying our subspace analysis using synthetic data. We generate two Gaussian clusters in 60-D space, with 10 outliers. The first two dimensions are shown in Figure 3(a). Without robust analysis, the two clusters and outliers are mixed together in the spectral mapping, as shown in Figure 3(b). Figure 3(c) shows the histogram of kNNDs. Our algorithm detects all of the 10 outliers not in the first peak. The inliers are projected onto the resulted 1D subspace. Figure 3(d) shows the spectral mapping of the projected 1D data. Two clusters are clearly distinguishable.

4 Application: Layer Extraction from Video

We apply our robust subspace analysis technique to layer extraction. The goal of layer extraction is to segment a video image into some number of 2D regions (sub-images), in each of which pixels share the same apparent motion model. The robust subspace clustering technique will solve two problems not addressed in previous work [11]: (1) dealing with outliers in subspace estimation; and (2) bounding the subspace dimension for dynamic scene.

4.1 Subspace Approach to Layer Extraction

We are given $F + 1$ images $\{I^f \mid f = 0, 1, \cdots, F\}$ of a scene. Select I^0 as the reference view. We divide I^0 into small $n \times n$ image patches (we use 48×48 overlapped blocks[2]). For the i-th patch in I^0, we can estimate a 2×3 affine transformation (homography) between I^0 and any other view I^f. The affine transformation[3] is reshaped into a 6-D column vector denoted by $\mathbf{m}_i^f$. The measurement matrix $\mathtt{W}$ is constructed by stacking all such column vectors together in the following way:

$$\mathtt{W}_{6F\times N} = \begin{bmatrix} \mathbf{m}_1^1 & \cdots & \mathbf{m}_N^1 \\ \mathbf{m}_1^2 & \cdots & \mathbf{m}_N^2 \\ & \vdots & \\ \mathbf{m}_1^F & \cdots & \mathbf{m}_N^F \end{bmatrix}_{6F\times N} \tag{8}$$

where N is the total number of image patches in I^0. Each column in $\mathtt{W}$ contains the motions of one $n \times n$ image patch across all images. We center the data in $\mathtt{W}$ by subtracting their column mean.

The subspace-based layer extraction approach [12] is based on the fact that the rank of $\mathtt{W}$ is no more than *three*, given a static scene (or equivalently a scene containing a single moving body). The SVD algorithm is used in [12] to compute the low rank subspace. The high dimensional data points (each column in $\mathtt{W}$ is a 6F-dimensional data point) are projected onto this subspace, in which they are clustered into layers.

There are two problems that remained unsolved in [12]. First, the SVD algorithm is sensitive to outliers, which are regions of motion outliers. It is widely known that local motion measurements often contain outliers due to various reasons, such as lighting change, motion or depth discontinuity, and/or non-rigid motions. Second, it is not clear in [12] how to bound the subspace dimension for dynamic scene. For dynamic scene, the rank of $\mathtt{W}$ is scene dependent. We can show that [11]:

$$d = \mathrm{rank}(\mathtt{W}) \leq \min(4B - 1, L - 1) \tag{9}$$

Here B is the number of independently moving objects, and L is the number of different planes in the scene. In general

[2] Overlapped blocks can effectively deal with occlusions.

[3] We scaled the elements in the parameters of affine transformation so that the motion parameter space is approximately isotropic [25]. Such scaling does not change the rank of $\mathtt{W}$ [12].

we do not know B and L.

4.2 Layer Extraction Algorithm

Our robust subspace clustering technique in Section 3.3 can be directly applied to address these two problems. Here we just point out how to bound the subspace dimension using Equation (6). Suppose the image size (pixel number) is w. Since each image patch contains n^2 pixels, according to Equation (6), to bound the subspace dimension below d_0 we set the k to be:

$$k \geq \frac{w}{(d_0+1)n^2} \tag{10}$$

Our overall subspace-based layer extraction algorithm consists of the following major steps: (1) robustly compute the low dimensional subspace from W, and project the measurement data onto the subspace; (2) cluster the projected inliers into initial layers, which are large layers; (3) progressively extract the previously excluded small layers, and refine layers using layer competition.

The layer competition in the Step 3 simply assigns an image region r to the initial layer (cluster) that best describes the motions of r. To utilize the spatial coherence existing in a single image, we assign homogeneous color regions instead of individual pixels to layers. The assumption we use here is that each homogeneous color region corresponds to a planar patch in the scene. Such assumption is generally valid for images of natural scenes, and has been used in motion analysis and stereo [2, 26, 22]. We use color oversegmentation [4] to assure the validity of the assumption.

Note that small layers are removed by kNND procedure at the very beginning. It is widely known that motion measurements of small layers are not stable, since they tend to have non-rigid or other complex motions, as well as depth discontinuity at layer boundaries. It is therefore desirable for any layer extraction algorithm to exclude them at the beginning for the extraction of larger layers. Once the large initial layers are available, the well-established progressive techniques [8, 16] can be applied to extract these previously excluded small layers.

4.3 Experimental Results

We show the experimental results on several real video sequences of dynamic and static scenes. See the complementary movie files for more results. In all experiments, we set $k = 8$ in kNND procedure to bound the subspace dimension to be no more than *three*, and set $t = 95\%$ in Equation (2) and (7) rank detection.

4.3.1 Mobile & Calendar Sequence

In this standard MPEG sequence, the calendar is moving upward, the train is pushing the ball, and the camera is zooming and tracking the train. The input to our algorithm is a 5-frame image sequence, with the middle frame chosen as the reference image, which is shown in Figure 4(a).

Due to outliers in local motion estimations, the direct spectral mapping, without detecting outliers, results in incorrect layer segmentation, as shown in Figure 4(b) and (c). Our robust subspace analysis technique has detected those patches of motion outliers, as shown in Figure 4(e). As we can see, these patches contain either multiple motions or little texture, and their local motion estimations are unstable and become motion outliers. For example, in Figure 4(e), Patch A contains little texture and Patch B contains two different motions. By removing these detected motion outliers, our algorithm obtains a 2D linear subspace of $\mathbb{R}^{24}$, as shown in Figure 4(f). As a result, the spectral mapping of the low dimensional data has tight cluster structure that are easily identifiable, as shown in Figure 4(g). The final layer extraction result is shown in Figure 4(h). Four layers (background, calendar, train, and ball) are correctly extracted.

4.3.2 Walking-Person Sequence

This sequence is taken by a hand-held video camera, where three persons are walking in front of a building. Figure 5(a) show the middle frame in this 5-frame sequence.

Our algorithm correctly extracts five layers from this sequence, including the ground, the building wall, and the three walking persons. See the caption in Figure 5 for more details.

4.3.3 Static Scenes

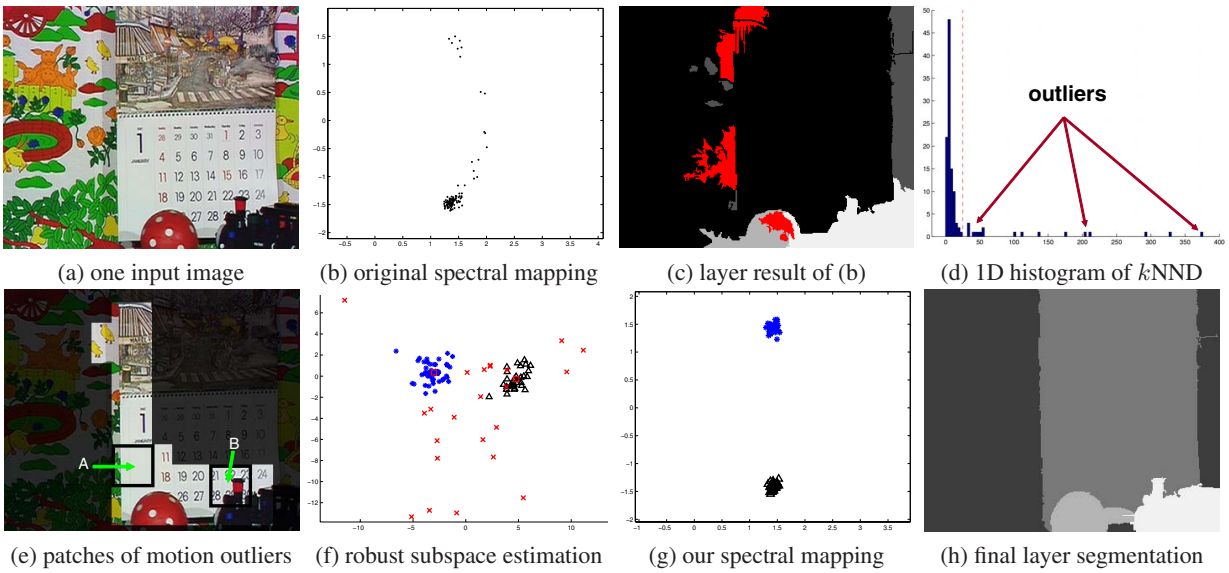
We apply our algorithm to two sequences of static scenes, but without using the knowledge that the scenes are static.

In the standard *MPEG flower garden* sequence, the camera is translating to the right. Figure 6(a) shows one image of the input sequence. Our algorithm extracts four layers: the ground, the house in the background, the tree trunk, and the tree branch.

The stop-sign sequence is taken by a hand-held video camera. It has a cluttered background. Figure 6(c) shows one frame. Our algorithm outputs four layers: the ground, the stop sign, the building on the left, and the tree on the right. Note that the building wall behind the tree does not contain enough textures, and is assigned to the tree layer.

5 Conclusion

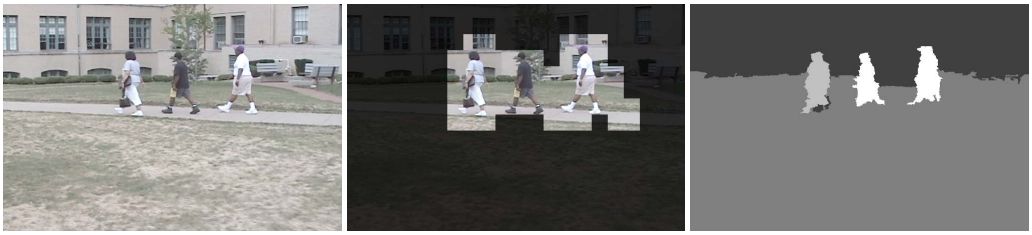
We exploit both subspace constraint and cluster constraint to detect outliers in high dimensional data for robust subspace estimation. The kNND metric is simple and effective for exploiting the cluster constraint *without* actually clustering the data. This is important since it is a chick-and-egg problem if one wants to detect outliers by clustering high dimensional noisy data. The parameter k in kNND can be used to bound the subspace dimension to assure global linear subspace constraint on the data. Such subspace constraint is then used by simple iterative algorithms to detect the remaining outliers and to estimate the final subspace. Since kNND provides a good initial inlier set, the itera-



(a) one input image (b) original spectral mapping (c) layer result of (b) (d) 1D histogram of kNND

(e) patches of motion outliers (f) robust subspace estimation (g) our spectral mapping (h) final layer segmentation

Figure 4. Layer extraction result of *mobile* sequence. Our algorithm derives a good 2D subspace (of $\mathbb{R}^{24}$) by excluding the outliers in the local motion estimations and small layers. The remaining initial large layers (the background and the calendar) become tight clusters in the spectral mapping of the projected low dimensional inlier data. Given such large initial layers, the well-established progressive technique [8, 16] is applied to extract the small layers (train and ball), which are previously excluded by robust subspace clustering. (b): spectral mapping without robust subspace analysis; (c): layer extraction result of clustering the spectral-mapped data in (b); (d): 1D histogram of kNND and the result of inlier-outlier classification; (e) excluded outliers, which are patches containing either discontinuous motions or small layers (ball and train); (f): 2D subspace by our algorithm, where $*$ and $\triangle$ are inliers, and $\times$ are detected outliers projected onto this subspace; (g): spectral mapping after projecting inliers onto the 2D subspace, where 2 clusters (corresponding to the two initial layers of background and calendar) are easily identifiable; (h): final layer extraction result, where four layers (background, calendar, train, and ball) are correctly extracted.



(a) one input image (b) patches of motion outliers (c) final layer segmentation

Figure 5. Layer extraction result of *Walking-person* sequence. The walking persons have non-rigid motions and their motion estimations are not reliable. Five layers are extracted, including three walking persons, ground, and building wall. The ground and build wall are extracted as large initial layers. The walking persons are progressively extracted against the large initial layers.

(a) flower garden sequence (b) four layers extracted from (a) (c) stop-sign sequence (d) four layers extracted from (c)

Figure 6. Layer extraction from static scenes. We do not need the assumption of static scene.

tive algorithm will converge quickly to a desirable solution. Given cleaned data and a good linear subspace, the spectral mapping can be applied to further improve the cluster structure. We apply our robust subspace clustering technique to extracting layers from image sequences, and show promising results.

In our clustering applications, we treat all components in each data point equally. Dealing with *intra-sample* outliers is future work (Our technique provides a good initialization to apply the techniques in [19, 5]).

Appendix: Subspace Constraint for Data Weighiting

Assuming Gaussian distribution for measurement noises, the d dimensional signal space is optimally approximated by the subspace defined by the first d columns of $\mathtt{U}$ in equation (1), which form the bases of the signal subspace $\mathtt{U}_S$. The data can then be decomposed into the signal component and the noise component:

$$\begin{aligned}\mathtt{W}_{M\times N} &= (\mathtt{U}_S \,|\, \mathtt{U}_\perp) \begin{pmatrix} \Sigma_S & \\ & \Sigma_\perp \end{pmatrix} (\mathtt{V}_S \,|\, \mathtt{V}_\perp)^\top \\ &= \mathtt{U}_S\Sigma_S\mathtt{V}_S^\top + \mathtt{U}_\perp\Sigma_\perp\mathtt{V}_\perp^\top = \mathtt{S} + \mathtt{N}\end{aligned}$$

The noise component $\mathtt{N}$ is assumed to follow zero-mean Gaussian distribution with sampled covariance matrix $\mathtt{C}$:

$$\mathtt{C} = \frac{1}{N-1}\mathtt{U}_\perp\Sigma_\perp^2\mathtt{U}_\perp^\top \tag{11}$$

The noise component in the i-th data point is $\mathbf{n}_i = \mathtt{U}_\perp\Sigma_\perp\mathbf{v}_{\perp i}$, where $\mathbf{v}_{\perp i}^\top$ is the i-th row of $\mathtt{V}_\perp$ (a $(M-d)$-vector). The Mahalanobis distance z_i^2 of $\mathbf{n}_i$ is:

$$z_i^2 = \mathbf{n}_i^\top \mathtt{C}^{-1}\mathbf{n}_i = (N-1)\sum_{p=1}^{M-d} v_{\perp i,p}^2 \tag{12}$$

z_i^2 follows χ^2 distribution [10], with $(M-d)$ degrees of freedom[4]. A data point is marked as an outlier if its z_i^2 is outside of the p-th confidence interval of the corresponding χ^2 distribution. In this paper we set $p = 95\%$. The inliers are weighted based on the Mahalanobis distance z_i^2 (e.g., $w_i = \exp(-z_i^2)$).

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR98*.

[2] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, 1996.

[3] S. Byers and A. E. Raftery. Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619.

[5] F. de la Torre and M. Black. A framework for robust subspace learning. *IJCV*, 54(1):117–142.

[6] A. Dempster, N. Laird, and D. Rubin. Maximal likelihood form incomplete data via the em algorithm. *RoyalStat*, B 39:1–38, 1977.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[4] In reality, the elements in $\mathbf{n}_i$ may not be independent. We perform another rank detection based on equation (2) to detect the effective degree of freedom.

[8] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV92*.

[9] A. Jepson and M. Black. Mixture models for optical flow computation. In *CVPR 1993*, pages 760–761.

[10] I. T. Jolliffe. *Principal Components Analysis*. Springer, 1986.

[11] Q. Ke. A robust subspace approach to extracting layers from image sequences. Technical Report CMU-CS-03-173, Carnegie Mellon University, 2003.

[12] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *CVPR 2001*, pages I:255–262.

[13] M. Lee, W. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *CirSysVideo*, 7(1), 1997.

[14] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and algorithm. In *Proc. of NIPS*, 2002.

[15] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66:143–155, 1998.

[16] H. Sawhney and S. Ayer. Compact representations of videos through dominantand mulitple motion estimation. *PAMI*, 18:814–831, 1996.

[17] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV'98*.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000.

[19] H.-Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Trans. on PAMI*, 17(8):854–867.

[20] C. Stewart. Minpran: A new robust estimator for computer vision. *PAMI*, 17(10):925–938.

[21] H. Tao, H. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *PAMI*, 24(1):75–89, January 2002.

[22] H. Tao and H. S. Sawhney. Global matching criterion and color segmentation based seereo. In *WACV2000*.

[23] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 1992.

[24] P. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. In *ICCV99*.

[25] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5), 1994.

[26] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR96*.

[27] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cut. In *CVPR 2004*.

[28] S. Yu and J. Shi. Multiclass spectral clustering. In *Proc. of ICCV*, 2003.

[29] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *PAMI*, 22(10), 2000.