Combined Central and Subspace Clustering for Computer Vision Applications

No Author Given

No. 316

Abstract. *Central and subspace clustering methods are at the core of many segmentation problems in computer vision. However, both methods fail to give the correct segmentation in many practical scenarios, e.g., when data are close to the intersection of subspaces or when two cluster centers in different subspaces are spatially close. In this paper, we address these challenges by considering the problem of clustering a set of points lying in a union of n subspaces and distributed around m cluster centers inside each subspace. We obtain an initial estimate of the subspace bases using Generalized PCA (GPCA) and of the cluster centers using Kmeans. We then search for the optimal subspace bases and cluster centers by minimizing a suitable cost function that incorporates both central and subspace clustering constraints. The resulting method is a natural generalization of Kmeans and Ksubspaces to combined central and subspace clustering. Experiments on synthetic data compare our method favorably against four other methods in terms of both classification error and estimation accuracy. We also present applications of our method in vision problems such as face clustering with varying illumination and video shot segmentation of dynamic scenes. The ability of our approach to obtain a piecewise linear approximation of a curved manifold is also demonstrated.*

1 Introduction

Many problems in computer vision require the efficient and effective organization of huge-dimensional visual data for information retrieval purposes. Unsupervised learning, mostly clustering, provides a possible way to handle these challenges.

Central and subspace clustering are arguably the most studied clustering problems. In *central clustering*, data samples are assumed to be distributed around a collection of cluster centers, e.g., a mixture of Gaussians. This problem shows up in many vision tasks, e.g., image segmentation, and can be solved using techniques such as Kmeans [2], Expectation Maximization (EM) [1] and spectral clustering [9]. In *subspace clustering*, data samples are assumed to be distributed in a collection of subspaces. This problem shows up in various vision applications, such as motion segmentation [7, 11, 12], face clustering with varying illumination [5, 13], image and video segmentation [6, 13], etc. Subspace clustering can also be used to obtain a piecewise linear approximation of a manifold, as we will demonstrate in our real data experiments.[1] Existing subspace clustering methods include Ksubspaces [5] and Generalized Principal Component Analysis (GPCA) [14, 13], which make no assumption about the distribution of the data inside the subspaces. Methods such as Mixtures of Probabilistic PCA (MPPCA) [10] further assume that the distribution of the data inside each subspace is Gaussian and use EM to learn the parameters of the mixture model.

Unfortunately, there are many applications in which neither central nor subspace clustering individually are appropriate. In motion segmentation, for example, two objects in different locations (different cluster centers) may undergo the same rigid motion (same motion subspace). If one is interested in grouping based on motion only, one may argue that the problem can be solved by subspace clustering alone. Similarly, if one is interested in grouping the individual objects, one may argue that the problem can be solved using central clustering alone, because multiple cluster centers inside multiple subspaces can also be considered as multiple cluster centers. However, there are many cases in which neither central nor subspace clustering can solve the combined problem. For instance, subspace clustering fails when the data set contains points close to the intersection of two subspaces, as shown in Figure 1. Similarly, central clustering fails when two clusters in different subspaces are spatially close, as shown in Figure 2.

Paper contributions. In this paper, we propose a new clustering approach that combines both central and subspace clustering. We obtain an initial solution by grouping the data into multiple subspaces using GPCA and grouping the data inside each subspace using Kmeans. This initial solution is refined by minimizing an objective function composed of both central and subspace distances. This combined optimization leads to improved performance of our method over four different clustering approaches in terms of both clustering error and estimation accuracy. Real examples on illumination-invariant face clustering and video shot detection are also performed. Through the visualization of real face or video data in their feature space, we find that data samples in $\mathbb{R}^3$ are usually distributed on complex shaped curves or manifolds. With encouraging clustering results, our experiments also validate that subspace clustering can be effectively used to obtain a piecewise linear approximation of complex manifolds.

[1] While techniques for learning manifolds from data already exist, e.g., [15], manifold parsing is a very difficult machine learning problem and has not been so well studied.

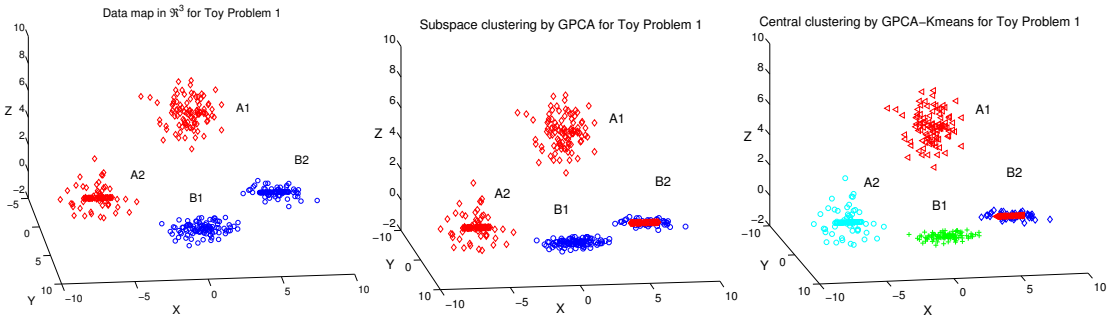

Fig. 1. Left: A set of points drawn from the x-y plane (blue) and the y-z plane (red), and distributed around two cluster centers inside each plane. Note that some data points are drawn from the intersection of the two planes (y-axis). **Center:** Subspace clustering by GPCA incorrectly assigns all the points on the y-axis to the y-z plane (red). **Right:** Central clustering by applying Kmeans to the two groups given by GPCA incorrectly assigns points in B_2 to A_1.

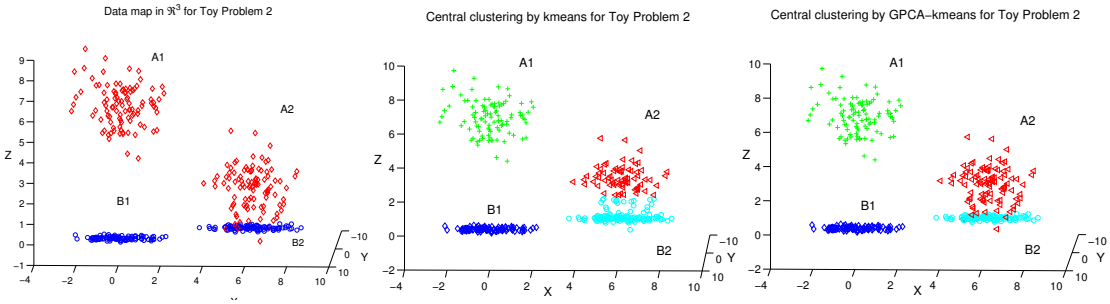

Fig. 2. Left: A set of points drawn from the x-y plane (blue) and the y-z plane (red), and distributed around two cluster centers inside each plane. Note that one of the blue clusters (B_2) is spatially close to one of the red clusters (A_2). **Center:** Central clustering by Kmeans assigns some points in A_2 to B_2. **Right:** Subspace clustering using GPCA followed by central clustering inside each subspace using Kmeans gives the correct clustering into four groups.

2 Combined Central and Subspace Clustering

In this paper, we consider the following problem.

Problem 1. Let $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^P$ be a collection of points lying approximately in n subspaces $S_j = \{\boldsymbol{x} : B_j^\top \boldsymbol{x} = 0\}$ with normal bases $\{B_j \in \mathbb{R}^{(D-d_j)\times D}\}_{j=1}^n$. Assume that within each subspace S_j the data points are distributed around m cluster centers $\{\mu_{jk} \in \mathbb{R}^D\}_{j=1\ldots n}^{k=1\ldots m}$. Given $\{\boldsymbol{x}_i\}_{i=1}^P$, estimate $\{B_j\}_{j=1}^n$ and $\{\mu_{jk}\}_{j=1\ldots n}^{k=1\ldots m}$.

Note that when $n = 1$ Problem 1 reduces to the standard central clustering problem. A popular method for solving this problem is the Kmeans algorithm, which solves for the cluster centers μ_k and the membership of the i-th point to the k-th cluster center, $w_{ik} \in \{0, 1\}$, by minimizing the within class variance

$$J_{KM} \doteq \sum_{i=1}^{P} \sum_{k=1}^{m} w_{ik} \|\boldsymbol{x}_i - \mu_k\|^2. \tag{1}$$

Given the cluster centers, the optimal solution for the memberships is to assign each point to the closest center. Given the memberships, the optimal solution for the cluster centers is given by the means of the points within each group. The Kmeans algorithm proceeds by alternating between these two steps until convergence to a local minimum. Initialization is usually done at random, or by choosing a random subset of the data points as initial cluster centers.

Note also that when $m = 1$ Problem 1 reduces to the classical subspace clustering problem. As shown in [5], this problem can be solved with an extension of Kmeans, called Ksubspaces, which solves for the subspace normal bases B_j and the membership of the i-th point to the j-th subspace, $w_{ij} \in \{0, 1\}$, by minimizing the cost function

$$J_{KS} \doteq \sum_{i=1}^{P} \sum_{j=1}^{n} w_{ij} \|B_j^\top \boldsymbol{x}_i\|^2 \tag{2}$$

subject to the constraints $B_j^\top B_j = \mathcal{I}$, for $j = 1, \ldots, n$, where $\mathcal{I}$ denotes the identity matrix. Given the normal bases, the optimal solution for the memberships it to assign each point to the closest subspace. Given the memberships, the optimal solution for the normal bases can be obtained from the null space of the data within each group. The Ksubspaces algorithm proceeds by alternating between these two steps until convergence to a local minimum.

In this section, we are interested in the more general problem of $n > 1$ subspaces and $m > 1$ centers per subspace. In principle, we could also solve this problem using Kmeans by interpreting Problem 1 as a central clustering problem with mn cluster centers. However, Kmeans does not fully employ the data's structural information and can cause undesirable clustering results, as shown in Figure 2. Thus, we propose a new algorithm which combines the objective functions (1) and (2) into a single objective, and is a natural generalization of both Kmeans and Ksubspaces to simultaneous central and subspace clustering.

For the sake of simplicity, we further assume that the subspaces are of co-dimension one, i.e. hyperplanes, so that we can represent them with a single normal vector $\boldsymbol{b}_j \in \mathbb{R}^D$. We discuss the extension to subspaces of any dimensions in Remark 2. Our method computes the cluster centers and the subspace normals by solving the following optimization problem

$$\min \sum_{i=1}^{P} \sum_{j=1}^{n} \sum_{k=1}^{m} w_{ijk} \left((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2 \right) \tag{3}$$

$$\text{subject to } \boldsymbol{b}_j^\top \boldsymbol{b}_j = 1, \quad j = 1, \ldots, n \tag{4}$$

$$\boldsymbol{b}_j^\top \mu_{jk} = 0, \quad j = 1, \ldots, n, k = 1, \ldots, m \tag{5}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} w_{ijk} = 1, \quad i = 1, \ldots, P, \qquad w_{ijk} \in \{0, 1\} \tag{6}$$

where w_{ijk} denotes the membership of the i-th point to the jk-th cluster center. Equation (3) ensures that for each point $\boldsymbol{x}_i$, there is a (j, k) such that both $|\boldsymbol{b}_j^\top \boldsymbol{x}_i|$ and

$\|\boldsymbol{x}_i - \mu_{ijk}\|$ are small[2], equation (4) ensures that the normal vectors are of unit norm, equation (5) ensures that each cluster center lies in its corresponding hyperplane, equation (6) ensures that each point is assigned to only one of the mn cluster centers.

Using the technique of Lagrange multipliers to minimize the cost function in (3) subject to the constraints (4)–(6) leads to the new objective function

$$\mathcal{L}=\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2+\|\boldsymbol{x}_i-\mu_{jk}\|^2)+\sum_{j=1}^{n}\sum_{k=1}^{m}\lambda_{jk}(\boldsymbol{b}_j^\top\mu_{jk})+\sum_{j=1}^{n}\delta_j(\boldsymbol{b}_j^\top\boldsymbol{b}_j-1). \quad (7)$$

Similarly to the Kmeans and Ksubspaces algorithms, we minimize $\mathcal{L}$ using the following coordinate descent minimization technique.

Algorithm 1 (Combined Central and Subspace Clustering)

1. *Initialization*: Obtain an initial estimate of the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and cluster centers $\{\mu_{jk}\}_{j=1\ldots n}^{k=1\ldots m}$ using GPCA followed by Kmeans in each subspace.
2. *Computing the memberships* Given the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and the cluster centers $\{\mu_{jk}\}_{j=1\ldots n}^{k=1\ldots m}$, compute the memberships $\{w_{ijk}\}$.
3. *Computing the cluster centers*: Given the memberships $\{w_{ijk}\}$ and the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$, compute the cluster centers $\{\mu_{jk}\}_{j=1\ldots n}^{k=1\ldots m}$.
4. *Computing the normal vectors*: Given the memberships $\{w_{ijk}\}$ and the cluster centers $\{\mu_{jk}\}_{j=1\ldots n}^{k=1\ldots m}$, compute the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$.
5. *Iterate*: Repeat steps 2,3,4 until convergence of the memberships.

The following subsections describe each step of the algorithm in detail.

Initialization: Since the data lies in a collection of hyperplanes, we can apply GPCA to obtain an estimate of the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and segment the data into n groups. Let $\boldsymbol{X}_j \in \mathbb{R}^{D\times P_j}$ be the set of points in the j-th hyperplane. If we use the SVD of $\boldsymbol{X}_j$ to compute a rank $D-1$ approximation of $\boldsymbol{X}_j \approx U_j S_j V_j$, where $U_j \in \mathbb{R}^{D\times(D-1)}$, $S_j \in \mathbb{R}^{(D-1)\times(D-1)}$ and $V_j \in \mathbb{R}^{(D-1)\times P_j}$, then the columns of $\boldsymbol{X}'_j = S_j V_j \in \mathbb{R}^{(D-1)\times P_j}$ are a set of vectors in $\mathbb{R}^{D-1}$ distributed around m cluster centers. We can apply Kmeans to segment the columns of $\boldsymbol{X}'_j$ into m groups and obtain the projected cluster centers $\{\mu'_{jk} \in \mathbb{R}^{D-1}\}_{k=1}^m$. The original cluster centers are then given by $\mu_{jk} = U_j\mu'_{jk} \in \mathbb{R}^D$.

Computing the memberships: Since the cost function is always positive and linear in w_{ijk}, the minimum is attained by choosing $w_{ijk} = 0$. However, since $\sum_{jk} w_{ijk} = 1$, the w_{ijk} multiplying the smallest $\left((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2\right)$ must be 1, hence

$$w_{ijk} = \begin{cases} 1 & \text{if } (i,j) = \arg\min \left((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2\right) \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

[2] The other reason to employ the sum of two distances here is to be consistent with the log-likelihood formulation in remark 1.

Computing the cluster centers: From the first order condition for a minimum we have

$$\frac{\partial \mathcal{L}}{\partial \mu_{jk}} = -2\sum_{i=1}^{P} w_{ijk}(\boldsymbol{x}_i - \mu_{jk}) + \lambda_{jk}\boldsymbol{b}_j = 0. \tag{9}$$

After left-multiplying (9) by $\boldsymbol{b}_j^\top$ and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$ and $\boldsymbol{b}_j^\top \boldsymbol{b}_j = 1$, we obtain

$$\lambda_{jk} = 2\sum_{i=1}^{P} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i). \tag{10}$$

After substituting (10) into equation (9) and dividing by two, we obtain

$$-\sum_{i=1}^{P} w_{ijk}(\boldsymbol{x}_i - \mu_{jk}) + \sum_{i=1}^{P} w_{ijk}\boldsymbol{b}_j\boldsymbol{b}_j^\top \boldsymbol{x}_i = 0 \implies \boxed{\mu_{jk} = (\mathcal{I} - \boldsymbol{b}_j\boldsymbol{b}_j^\top)\frac{\sum_{i=1}^{P} w_{ijk}\boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}}}$$

where $\mathcal{I}$ is the identity matrix in $\mathbb{R}^D$. Note that μ_{jk} has a simple geometric interpretation: it is the mean of the points in the jk-th cluster, projected onto the j-th hyperplane.

Computing the normal vectors: From the first order condition for a minimum we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_j} = 2\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)\boldsymbol{x}_i + \sum_{k=1}^{m} \lambda_{jk}\mu_{jk} + 2\delta_j \boldsymbol{b}_j = 0. \tag{11}$$

After left-multiplying (11) by $\boldsymbol{b}_j^\top$ to eliminate λ_{jk} and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$, we obtain

$$\delta_j = -\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2. \tag{12}$$

After substituting (10) into equation (11) and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$, we obtain

$$\boxed{(\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})\boldsymbol{b}_j = 0.} \tag{13}$$

Then $\boldsymbol{b}_j$ is the eigenvector of $(\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})$ associated with its smallest eigenvalue, which can be computed using the SVD.

Remark 1 (Maximum Likelihood Solution). Notice that in the combined objective function (7) the term $|\boldsymbol{b}_j^\top \boldsymbol{x}_i|$ is the distance to the j-th hyperplane, while $\|\boldsymbol{x}_i - \mu_{jk}\|$ is the distance to the jk-th cluster center. Since the former is mostly related to the variance of the noise in the orthogonal direction to the hyperplane, σ_b^2, while the latter is mostly related to the within class variance, σ_μ^2, the magnitudes of these two distances need to be taken into account. One way of doing so is to assume that the data is generated by a mixture of mn Gaussians with means μ_{jk} and covariances $\Sigma_{jk} = \sigma_b^2\boldsymbol{b}_j\boldsymbol{b}_j^\top + \sigma_u^2(\mathcal{I} - \boldsymbol{b}_j\boldsymbol{b}_j^\top)$.

This automatically allows the variances inside and orthogonal to the hyperplanes to be different. Application of the EM algorithm to this mixture model leads to the minimization of the following normalized objective function

$$\mathcal{L} = \sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}\Big(\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{2\sigma^2} + \frac{\|\boldsymbol{x}_i - \mu_{jk}\|^2}{2\sigma_\mu^2} + \log(\sigma_b) + (D-1)\log(\sigma_u)\Big) + \sum_{j=1}^{n}\sum_{k=1}^{m}\lambda_{jk}(\boldsymbol{b}_j^\top \mu_{jk}) + \sum_{j=1}^{n}\delta_j(\boldsymbol{b}_j^\top \boldsymbol{b}_j - 1)$$

where $w_{ijk} \propto \exp(-\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{2\sigma^2} - \frac{\|\boldsymbol{x}_i - \mu_{jk}\|^2}{2\sigma_\mu^2})$ is now the probability that the i-th point belongs to the jk-th cluster center, and $\sigma^{-2} = \sigma_b^{-2} - \sigma_\mu^{-2}$. The optimal solution can be obtained using coordinate descent, similarly to Algorithm 1, as follows

$$\lambda_{jk} = 2\sum_{i=1}^{P} w_{ijk}\frac{\boldsymbol{b}_j^\top \boldsymbol{x}_i}{\sigma_\mu^2} \qquad \delta_j = -\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{\sigma^2}$$

$$\mu_{jk} = (\mathcal{I} - \boldsymbol{b}_j\boldsymbol{b}_j^\top)\frac{\sum_{i=1}^{P} w_{ijk}\boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}} \qquad 0 = (\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\frac{\boldsymbol{x}_i}{\sigma^2} + \frac{\mu_{jk}}{\sigma_\mu^2})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})\boldsymbol{b}_j$$

$$\sigma_b^2 = \frac{\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}} \qquad \sigma_u^2 = \frac{\sum_{ijk} w_{ijk}\big(\|\boldsymbol{x}_i - \mu_{jk}\|^2 - (\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2\big)}{(D-1)\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}}$$

Remark 2 (Extension from hyperplanes to subspaces). In the case of subspaces of co-dimension larger than one, each normal vector $\boldsymbol{b}_j$ should be replaced by a matrix of normal vectors $B_j \in \mathbb{R}^{D\times(D-d_j)}$, where d_j is the dimension of the j-th subspace. Since the normal bases and the means must satisfy $B_j^\top \mu_{jk} = 0$ and $B_j^\top B_j = \mathcal{I}$, the objective function (3) should be changed to

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m} w_{ijk}(\|B_j^\top \boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2) + \\ &\sum_{j=1}^{n}\sum_{k=1}^{m}\lambda_{jk}^\top(B_j^\top \mu_{jk}) + \sum_{j=1}^{n}\text{trace}\big(\Delta_j(B_j^\top B_j - \mathcal{I})\big).\end{aligned} \tag{14}$$

where $\lambda_{jk} \in \mathbb{R}^{(D-d_j)}$ and $\Delta_j \in \mathbb{R}^{(D-d_j)\times(D-d_j)}$ are, respectively, vectors and matrices of Lagrange multipliers. Given the normal basis B_j, the optimal solution for the means is given by

$$\mu_{jk} = (I - B_jB_j^\top)\frac{\sum_{i=1}^{P} w_{ijk}\boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}}.$$

One can show that the optimal solution for Δ_j is a scaled identity matrix whose diagonal entry is $\delta_j = -\sum_{i=1}^{P}\sum_{j=1}^{m} w_{ijk}\|B_j^\top \boldsymbol{x}_i\|^2$. Given δ_j and μ_{jk}, B_j can still be solved from the null space of $(\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})$ which can be proved to have dimension $D - d_j$.

3 Experiments

3.1 Comparison of clustering performance with simulated data

We randomly generate 600 data points in $\mathbb{R}^3$ lying in 2 intersecting planes $\{S_j\}_{j=1}^{2}$ with 3 clusters in each plane $\{\mu_{jk}\}_{j=1,2}^{k=1,2,3}$, as shown in Figure 3. 100 points are drawn around each one of the six cluster centers according to a Gaussian distribution with $\sigma_\mu = 1.5$. The angle between the two planes is randomly chosen from $20^o \sim 90^o$, and the distance among the three cluster centers is randomly selected in the range $2.5\sigma_\mu \sim 5\sigma_\mu$. Gaussian noise with standard deviation σ_b is added in the direction orthogonal to each plane. Based on the simulated data, we compare 5 different clustering methods:

- Kmeans clustering in $\mathbb{R}^3$ with 6 cluster centers, then merging into 2 planes (**KM**),
- MPPCA clustering using 6 cluster centers[3], then merging them into 2 planes (**MP**),
- Ksubspaces clustering in $\mathbb{R}^3$, then Kmeans within each plane $S_j; j = 1, 2$ (**KK**),
- GPCA clustering in $\mathbb{R}^3$, then Kmeans within each plane $S_j; j = 1, 2$ (**GK**),
- GPCA-Kmeans clustering for initialization followed by combined central and subspace clustering (**JC**) as described in Section 2 (Algorithm 1).

Figure 4 shows a comparison of the performance of the above methods in terms of clustering error ratios[4] and the error in the estimation of the normal directions in degrees. The results are the mean of the errors over 100 trials. It can be seen in Figure 4 that the errors in clustering and normal vectors of all four algorithms increase as a function of noise. **MP** achieves better performance than **KM** and **KK** for large levels of noise, probably via its probabilistic formulation. The two stage algorithms, **KK**, **GK** and **JC**, in general perform better than **KM** and **MP** in terms of clustering error. The random initialization based methods, **KM**, **MP** and **KK**, have non-zero clustering error even with noise-free data. Within the two stage algorithms, **KK** begins to experience subspace clustering failures more frequently with more severe noises, due to its random initialization, while GPCA in **GK** and **JC** employ an algebraic solution of one-shot subspace clustering, thus avoiding the initialization problem. The incorrect subspace clustering of **KK** can cause the estimate of the normals to be very inaccurate, which explains why **KK** has worse errors in the normal vectors than **KM** and **MP**[5]. In summary, **GK** and **JC** have smaller average errors in clustering and normal vectors than **KM**, **MP** and **KK** . The combined optimization procedure of **JC** converges within $2 \sim 5$ iterations according to our experiments, which further advocates **JC**'s clustering performance.

[3] Software available from www.ncrg.aston.ac.uk/netlab/

[4] By considering both central and subspace clustering, each data point has 2 labels after computation. Therefore the final error percentage is the result of the total error counts divided by $1200 = 2 \times 600$.

[5] In order to estimate the plane normals, we group 6 clusters returned by **KM** or **MP** into 2 planes. The idea is that 3 clusters which lie in the same plane have the dimensionality of 2 instead of 3. A brute-force search with $\binom{6}{3}/2$ selections is employed to find the 2 best fitting planes, by considering the minimal strength of the data distributed in the third dimension via Singular Value Decomposition [4].

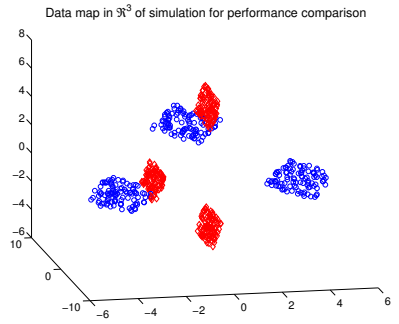

Fig. 3. 600 simulated data points in $\mathbb{R}^3$ lying in 2 intersected planes with 3 clusters in each plane.

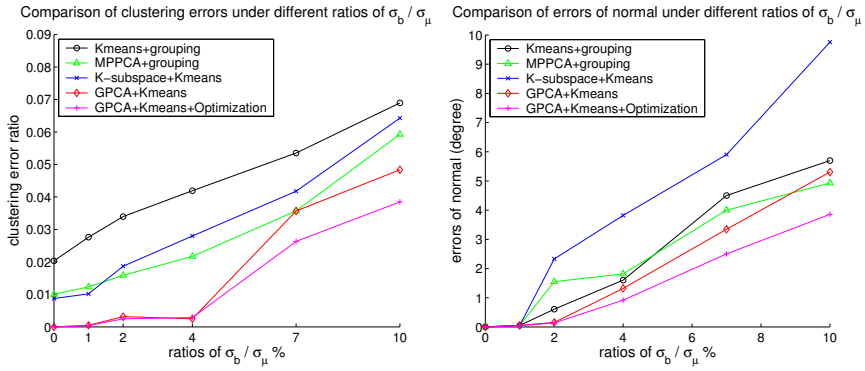

Fig. 4. Left: Clustering error as a function of noise in the data. **Left:** Error in the estimation of the normal vectors (degrees) as a function of the level of noise in the data.

3.2 Applications with real data

Face clustering with varying illumination The Yale face database B [3, 8] contains a collection of face images $I_j \in \mathbb{R}^K$ of 10 subjects taken under 576 viewing conditions (9 poses $\times$ 64 illumination conditions). Here we only consider the illumination variation for face clustering in the case of frontal face images. Thus our task is to sort the images taken for the same person by using our combined central-subspace clustering algorithm. From [3, 5, 8], we know that the set of all images of an assumed Lambertian object (such as a human face with a fixed pose) taken under all lighting conditions forms a cone in the image space which can be well approximated by a low dimensional subspace. Thus image sets of different subjects consist of different subspaces. Since the number of pixels K of each image is in general much larger than the dimension of the underlying subspace, PCA [2] is first employed for dimensionality reduction. Successful GPCA clustering results have been reported by [13] for a subset of 3x64 images of subjects 5,8,10. The image resolution used in [13] is 30x40 pixels and 3 PCA components are considered as image features in homogeneous coordinates. In this paper, we further explore the performance of combined central-subspace face clustering under more complex conditions. We consider 4x64 images of subjects 5,6,7,8, as shown in Figure 5, with a resolution of 240x320 pixels. This gives more background details and makes PCA features capture less information with 3 components. Figures 6 (a,b)

show the imperfect clustering result of GPCA due to the intersection of the subspace of subject 5 with the subspaces of subjects 6 and 7. GPCA classifies all the images on the intersections of subspaces into subject 5. Mixtures of PPCA is implemented in Netlab as a probabilistic variation of subspace clustering with one spatial center per subspace. It can be initialized with Kmeans (originally in Netlab) or GPCA, which all result in imperfect clustering outputs. We show one example of the subspaces of subjects 6 and 7 mixed (Kmeans initialization) in Figure 6 (c,d). Our combined subspace-central optimization process successfully corrects the clustering labels of portions of images for subjects 6 and 7, as demonstrated in Figure 6 (e,f). In the optimization, the local clusters in the subspace of subject 6,7 contribute smaller central distances to their misclassified images, which re-classifies them to the correct subspaces using our combined subspace-central clustering algorithm. In this experiment, 4 subspaces with 2 clusters per subspace are used. Compared with the result in [13], we obtain the perfect illumination-invariant face clustering where data distribution is more complex.

Video shot segmentation Unlike face images under different illumination conditions, video data provides continuous visual signals. Video structure parsing and analysis applications need to segment the whole video sequence into several video shots. Each video shot may contain hundreds of image frames which are either captured with a similar background or have a similar semantical meaning. Figure 7 shows 2 sample videos, *mountain.avi* and *drama.avi*, containing 4 shots each. Archives are publicly available from *http://www.open-video.org*. For the mountain sequence, 4 shots are captured. The shots display different backgrounds and show either multiple dynamic objects and/or severe camera motions. In this video, the frames between each pair of successive shots are gradually blended from one to another[6]. In order to explore how the video frames are distributed in feature space, we plot the first 3 PCA components for each frame in Figure 9 (b, d, f). Note that a manifold structure can be observed which starts from red dots to green, black dots and ends in blue as shown in Figure 9 (f). The video shot segmentation results of the mountain sequence by Kmeans,GPCA and GPCA-Kmeans followed by combined optimization are shown in Figure 9 (a,b), (c,d) and (e,f), respectively. Because Kmeans is based on the central distances among data, it segments the data into spatially close blobs. There is no guarantee that these spatial blobs will correspond to correct video shots. Comparing Figure 9 (b) with the correct segmentation in (f), the Kmeans algorithm splits shot 2 into clusters 2 and 3, while it groups shots 1 and 4 into cluster 1. By considering the data's manifold nature, GPCA provides a more effective approximation with multiple planes to the manifold in $\mathbb{R}^3$ than the spatial blobs given by central clustering. The essential problem for GPCA is that it only deploys the co-planar condition in $\mathbb{R}^3$, without any constraint relying on their spatial locations. In the structural approximation of the data's manifold, there are many intersecting data points among 4 planes. These data points represent video frames with the clustering ambiguity solely based on the subspace constraint. Fortunately this limitation can be well tackled by GPCA-Kmeans with combined optimization. Combining central and subspace distances provides correct video shot clustering results for the mountain sequence, as demonstrated in Figure 9 (e,f).

[6] Due to this reason, the correct video shot segmentation is considered to split every two successive shots at their blending frames.

Fig. 5. Sample images of subjects 5,6,7,8 shown in different colors.

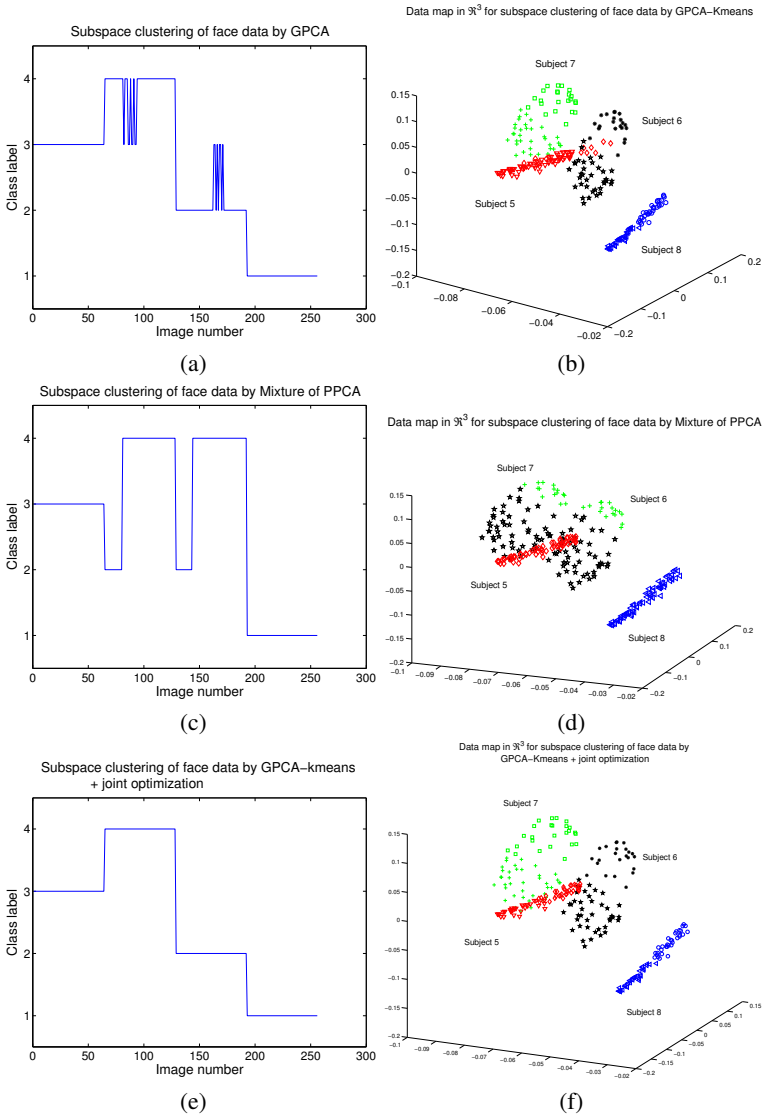

Fig. 6. According to the correct clustering in (e-f), subjects 5, 6, 7, 8 are grouped as clusters 3, 4, 2, 1 on the left, and shown in red, black, green and blue on the right respectively. (a-b) Subspace based face clustering by GPCA. (c-d) Face clustering by MPPCA. (e-f) Combined central-subspace clustering by alternating minimization initialized with GPCA-Kmeans. The colors of points in this figure also match the colors of subjects 5,6,7,8 in Figure 5.

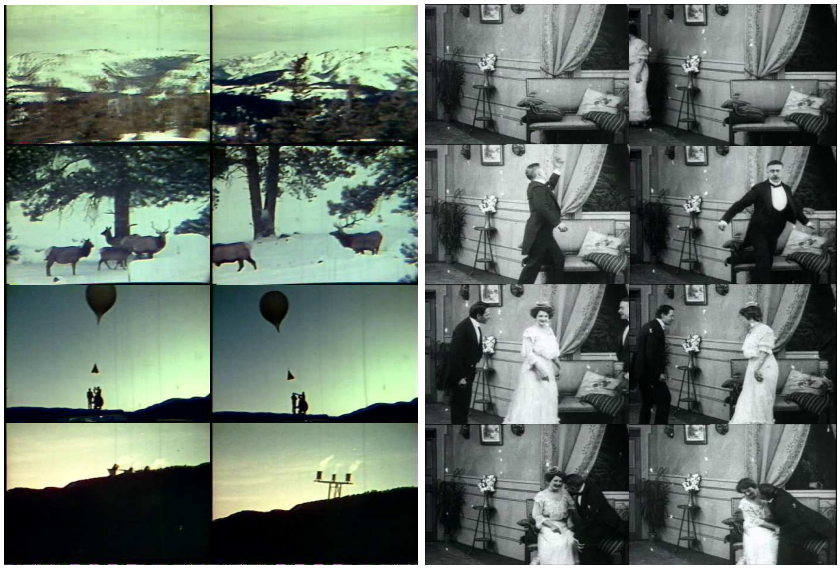

Fig. 7. Video shot segmentation. **Left:** Sample images from the mountain sequence. There are 4 shots in the video. Each row shows two images from each shot. All 4 shots are dynamic scenes, including large camera panning in shot 1, multiple animals moving in shot 2, a balloon rolling left and right in shot 3 and a rocket firing with the camera moving in shot 4. **Right:** Sample images from the drama sequence. There are 4 shots in the video. Each row shows 2 images from each shot. Shot 1 mainly shows the background only with no or little appearance of the actor or actress; shot 2 shows the actor's motion; shot 3 shows a scene of the actor and actress talking while standing; shot 4 shows the actor and actress kissing each other and sitting.

The second video sequence shows a drama scenario which is captured with the same background. The video shots should be segmented by the semantic meaning of the performance of the actor and actress. In Figure 7 (Right), we show 2 sample images for each shot. This drama video sequence contains very complex actor and actress' motions in front of a common background, which results in a more complex manifold data structure[7] than that of the mountain video. For better visualization, the normal vectors of data samples recovered by GPCA or the combined central-subspace optimization, are drawn originating from each data point in $\mathbb{R}^3$ with different colors for each cluster. For this video, the combined optimization process shows a smoother clustering result in Figure 8 (c,d), compared with (a,b). In summary, GPCA can be considered as an effective way to group data in a manifold into multiple subspaces or planes in $\mathbb{R}^3$ which normally better present video shots than central clustering. GPCA-Kmeans with combined optimization can then associate the data at the intersection of planes into the correct clusters by optimizing combined distances. Subspace clustering seems to be a better method to group the data on a manifold by somehow preserving their geometric structure. Central clustering, such as Kmeans, provides a piecewise constant approximation; while subspace clustering shows a piecewise linear approximation. On the other hand, subspace

[7] Because there are image frames of transiting subject motions from one shot to another, the correct video shot segmentation is considered to split successive shots at their transiting frames.

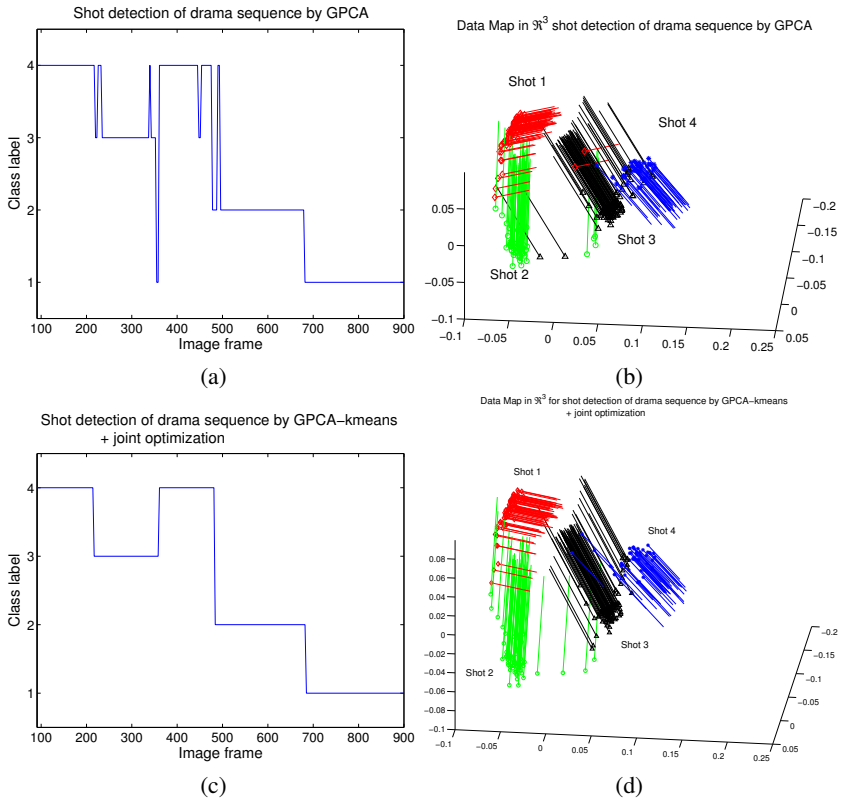

Fig. 8. According to the correct clustering in (c-d), video shots 1, 2, 3, 4 are grouped as clusters 4, 3, 2, 1 on the left, and shown in red, green, black and blue on the right respectively. (a-b) Video shot segmentation of drama sequence by GPCA. (c-d) Video shot segmentation by GPCA-Kmeans with optimization.

clustering can meet severe clustering ambiguity problems when the shape of the manifold is complex, as shown in Figure 8 (b,d). In this case, there are many intersections of subspaces so that subspace clustering results can be very sparse, without considering the spatial coherence. Combined optimization of central and subspace distances demonstrates superior clustering performance with real video sequences.

4 Conclusions

In summary, this paper makes three contributions. First, we propose a combined central and subspace clustering algorithm via optimization of a suitable objective function incorporating both types of constraints. It is an intuitive, easy to implement, yet useful extension of the classical Kmeans and Ksubspaces/GPCA algorithms. Second, the advantage of our algorithm of handling more clustering conditions, compared with traditional Kmeans and Ksubspaces/GPCA methods is illustrated by two toy problems. The comprehensive comparison of four clustering solutions based on simulated data demonstrates our algorithm's priority in accuracy. Third, our method provides an effective and efficient piecewise linear approximation of curved manifolds of complex visual

data in video. The video data can be grouped through subspace clustering, while central distance constraints eliminate the clustering ambiguity of subspace intersections.

As a future work, we expect to extend the proposed combined central and subspace clustering formulation to recognize multiple complex curved manifolds. An example problem is to find which movie a given image appears in. Each manifold will be composed of multiple subspaces where each subspace is spatially constrained by central distances among data samples. Once the models are learned from movie data (using similar techniques for shot detection), the likelihood evaluation for a new data sample is simply to compute its combined central and subspace distances to the given models.

References

1. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.
2. R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2000.
3. A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.
4. H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2nd edition, 1996.
5. J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering apperances of objects under varying illumination conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
6. K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.
7. K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision*, volume 2, pages 586–591, 2001.
8. K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.
9. A. Ng, Y. Weiss, and M. Jordan. On spectral clustering: analysis and an algorithm. In *Neural Information Processing Systems*, 2001.
10. M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2), 1999.
11. R. Vidal and R. Hartley. Motion segmentation with missing data by PowerFactorization and Generalized PCA. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 310–316, 2004.
12. R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *European Conference on Computer Vision*, pages 1–15, 2004.
13. R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 510–517, 2004.
14. R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 621–628, 2003.
15. K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 988–995, 2004.

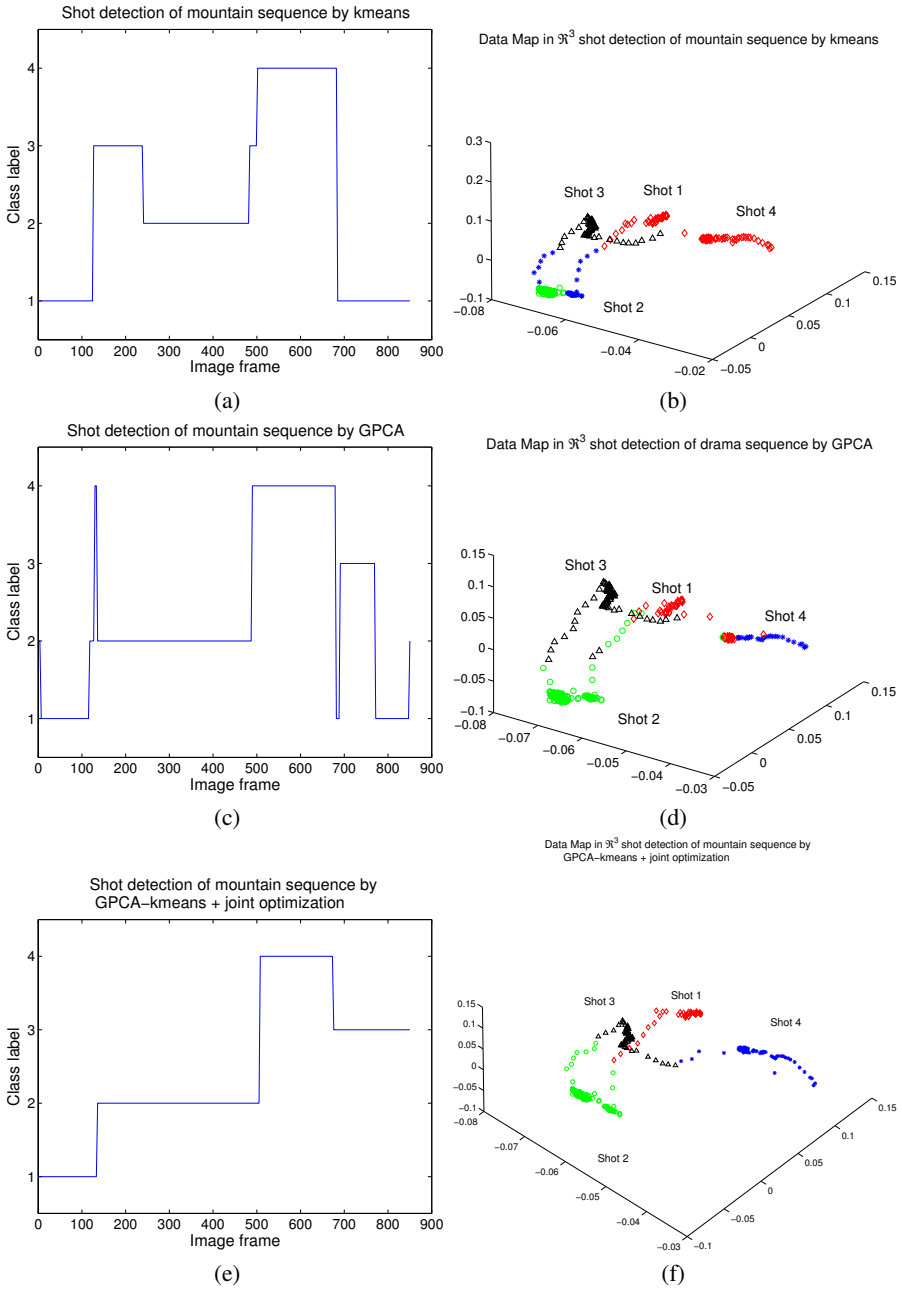

Fig. 9. According to the correct clustering in (e-f), video shots 1, 2, 3, 4 are grouped as clusters 1, 2, 4, 3 on the left, and shown in red, green, black and blue on the right respectively. (a-b) Video shot segmentation of mountain sequence by Kmeans. (c-d) Video shot segmentation of mountain sequence by GPCA. (e-f) Video shot segmentation of mountain sequence by GPCA-Kmeans with optimization.