

Mode Seeking with an Adaptive Distance Measure

Guodong Pan, Lifeng Shang, Dirk Schnieders, and Kwan-Yee K. Wong

Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
{gdpan, lfshang, sdirk, kykwong}@cs.hku.hk

Abstract. The mean shift algorithm is a widely used non-parametric clustering algorithm. It has been extended to cluster a mixture of linear subspaces for solving problems in computer vision such as multi-body motion segmentation, etc. Existing methods only work with a set of subspaces, which are computed from samples of observations. However, noises from observations can distort these subspace estimates very much and greatly influences clustering accuracy. We propose to use both subspaces and observations to improve performance. Furthermore, while these mean shift methods use fixed metrics for computing distances, we prefer an adaptive distance measure. The insight is, we can use temporary modes in a mode seeking process to improve this measure and obtain better performance. In this paper, an adaptive mode seeking algorithm is proposed for clustering linear subspaces. By experiments, the proposed algorithm is better than the state-of-the-art algorithm in terms of clustering accuracy.

Keywords: Mean Shift Algorithm, Metric Learning

1 Introduction

The mean shift (MS) algorithm is a non-parametric clustering method for finding centers of arbitrarily distributed points in a vector space [1]. Characterized as a gradient ascent algorithm, it has been successfully applied to tackle problems like image segmentation and tracking [1][2].

Recently the MS algorithm has been extended to Grassmann manifolds for clustering linear subspaces [3][4]. A Grassmann manifold is a collection of linear subspaces endowed with a distance measure. A Grassmann point is a linear subspace represented by an orthonormal matrix. If we concatenate columns of this matrix into a long vector, the Grassmann manifold is a surface in a high dimensional space. We will use the terms Grassmann point and linear subspace interchangeably. The basic idea of the MS algorithm over Grassmann manifolds is to compute a set of Grassmann points from samples of observations, find modes or local centers of these Grassmann points, and cluster the points using the modes.

Subbarao and Meer proposed the nonlinear mean shift algorithm [3], which started from a Grassmann point and iteratively moved it by logarithm and exponential mappings. One problem of their method is these mappings are computationally expensive. The intrinsic mean shift (Intrinsic MS) algorithm proposed by Cetingül and Vidal [4] used discrepancy measures and QR decompositions to simplify the computation. Unfortunately, clustering accuracy of both methods is greatly restricted by accuracy of estimation of Grassmann point sets. The accuracy of clustering depends on how well sampled Grassmann points can reflect the underlying density functions. Due to noise, these Grassmann points could deviate very much from the ground truth, and this heavily impairs performance. Actually the sampled points only retrieve part of observation information, and there is much information left behind. Reusing observation information will improve clustering accuracy.

In this paper, we propose a mode seeking algorithm (GOPS MS) to improve accuracy of clustering linear subspaces. It reuses observations as well as linear subspaces. Given a linear subspace, we find observations in this subspace, and encapsulate these observations and the subspace into an entity. This entity is viewed as a point in a Grassmann manifold and Observations Product Space, or simply a GOPS point. We will define a distance measure in GOPS such that, if two linear subspaces are similar, their corresponding GOPS points are close. Equipped with this measure, the GOPS MS performs mode seeking in the GOPS, and linear subspaces are clustered according to these GOPS modes.

In tradition, MS algorithms [5][1][4][3] have fixed distance measures. In a mode seeking process, they use a fixed distance to shift a temporary mode to the next. However, we find a temporary mode actually provides useful information to measure similarity between GOPS points. The proposed GOPS MS is equipped with a distance measure adapted to temporary modes. When this measure is updated with such a mode, distances between similar observations decrease and these observations become more dense (see Figure 1). This facilitates clustering tasks.

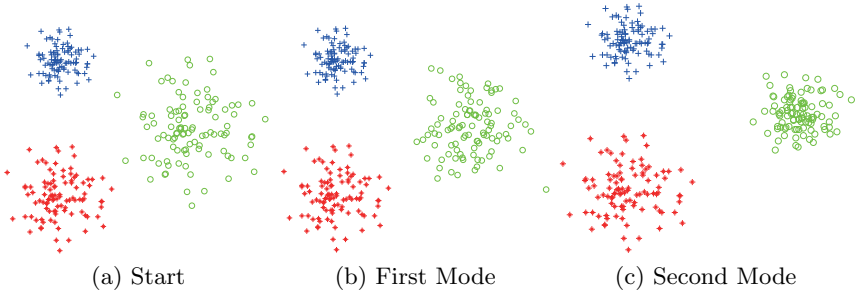


Fig. 1. Illustration of change of distances. (a) shows distances between points with an initial distance measure. The right green circles scatter in a large region. (b)(c) show distances between points with distance measures updated by modes. These green circles become more and more dense.

Our algorithm can be applied to many problems in computer vision [4][6][7][8]. In this paper, we apply it to the problems of multi-body motion segmentation, shape categorization and face clustering, and compare it with the Intrinsic MS, a fixed metric algorithm. The corresponding overall clustering accuracy of the proposed algorithm are 99.2%, 89.5% and 98.0%, and better than those of the Intrinsic MS as 82.1%, 64.5%, 76.5% respectively.

The rest of the paper is organized as follows. In Sect. 2, Euclidean mean shift is reviewed, from which we will motivate our GOPS MS. In Sect. 3, we construct the GOPS and explain the GOPS MS algorithm. Sect. 4 shows experiments and comparisons with the Intrinsic MS. Conclusion and future work are discussed in Sect. 5.

2 Euclidean Mean Shift

The Euclidean MS algorithm [9] finds local centers of a point set in a Euclidean space, and clusters all points in this set by these centers. If the point set is viewed as sampled points from a density function over the Euclidean space, the local centers are actually modes of this set, which are the points of local maximum density. So this algorithm is also considered as to seek modes and group points sharing the same mode together.

Formally, let a set of observations $x_n \in \mathbb{R}^d, n = 1, \dots, N$, be points sampled independently from a density function f . The kernel density estimate of f at x , denoted by $\hat{f}(x; h)$, is given by $\hat{f}(x; h) = c \sum_{n=1}^N \Phi(u_n^2; h)$, where $\Phi(\cdot)$ is a kernel function, $u_n = \|x - x_n\|$ is a distance measure, c is a normalization term, and $h > 0$ is the bandwidth. To locate a mode, the Euclidean MS algorithm seeks next temporary mode $y^{(k+1)}$ by minimizing a sum of weighted distance squares [10]

$$y^{(k+1)} = \arg \min_y \sum_{n=1}^N \|x_n - y\|^2 \psi(\|x_n - y^{(k)}\|^2; h). \quad (1)$$

where $\psi(u_n; h) = -\nabla \Phi(u_n; h)$. The distance $\|\cdot\|$ play an important role to measure similarity between two points. Although it is a metric, the MS algorithm does not require that. In the next section, we will define a distance measure that is not a metric to compute similarity between points in GOPS. In addition, it is shown in [10] that the convergence of the MS algorithm depends on the convexity of the kernel $\Phi(\cdot)$.

3 Mode Seeking in GOPS

In the following, we construct the GOPS to reuse observations for clustering and explain the GOPS MS algorithm.

3.1 GOPS

The GOPS is a set of GOPS points with a distance measure over it. The GOPS point is an entity including a linear subspace and observations in this subspace,

denoted by $G = \{\mathbf{X}, \mathbf{S}_{\mathbf{X}}\}$. Here \mathbf{X} is a $m \times p$ orthonormal matrix indicating a p -dimensional linear subspace in \mathbb{R}^m . Each observation is a vector in \mathbb{R}^m . And $\mathbf{S}_{\mathbf{X}}$ is an observation set in this subspace, denoted by a $m \times N$ matrix, where N is the number of observations.

Suppose we have two GOPS points $G_1 = \{\mathbf{X}_1, \mathbf{S}_{\mathbf{X}_1}\}$, $G_2 = \{\mathbf{X}_2, \mathbf{S}_{\mathbf{X}_2}\}$. The GOPS distance d_{GO} is defined as

$$d_{GO}^2(G_{\mathbf{X}_1}, G_{\mathbf{X}_2}) = c_1 d_G^2(\mathbf{X}_1, \mathbf{X}_2) + c_2 d_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_1}, \mathbf{S}_{\mathbf{X}_2}), \quad (2)$$

where c_1, c_2 are weights and $d_G, d_{\mathbf{X}_1}$ are defined as follows. The Grassmann distance [4] $d_G^2(\mathbf{X}_1, \mathbf{X}_2)$ computes distance between two linear subspaces as

$$d_G^2(\mathbf{X}_1, \mathbf{X}_2) = p - \text{tr}(\mathbf{X}_2^T \mathbf{X}_1 \mathbf{X}_1^T \mathbf{X}_2). \quad (3)$$

The fit measure $d_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_1}, \mathbf{S}_{\mathbf{X}_2})$ measures how well a linear subspace \mathbf{X}_1 fit observations $\mathbf{S}_{\mathbf{X}_1}, \mathbf{S}_{\mathbf{X}_2}$. Let \mathbf{I}_m be a $m \times m$ identity matrix. The fitness of \mathbf{X}_1 to $\mathbf{S}_{\mathbf{X}_2}$ is

$$V_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_2}) = \text{tr}(\mathbf{S}_{\mathbf{X}_2}^T (\mathbf{I}_m - \mathbf{X}_1 \mathbf{X}_1^T) \mathbf{S}_{\mathbf{X}_2}), \quad (4)$$

The fit measure between two GOPS points is defined as

$$d_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_1}, \mathbf{S}_{\mathbf{X}_2}) = \frac{V_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_1}) + V_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_2})}{(|\mathbf{S}_{\mathbf{X}_1}| + |\mathbf{S}_{\mathbf{X}_2}|)^2}, \quad (5)$$

where $|\mathbf{S}|$ provides the cardinality of \mathbf{S} . Since $\mathbf{S}_{\mathbf{X}_1}$ lies in \mathbf{X}_1 , $V_{\mathbf{X}_1}^2(\mathbf{S}_{\mathbf{X}_1}) \approx 0$ and the fit measure actually measures the residual of $\mathbf{S}_{\mathbf{X}_2}$ to \mathbf{X}_1 . And \mathbf{X}_1 is called the reference subspace of this fit measure.

3.2 Adaptive Mode Seeking

Algorithm 1 Adaptive Mode Seeking

Input: A set of linear subspaces $\mathbf{X}_n, n = 1, 2, \dots, N$, an observation set \mathbf{S} ;

Output: A clustering of subspaces;

Algorithm:

for Each \mathbf{X}_n **do**

 Compute observations $\mathbf{S}_{\mathbf{X}_n}$ in \mathbf{X}_n by subset selection;

$\mathbf{Y} = \mathbf{X}_n$ and $\mathbf{S}_{\mathbf{Y}} = \mathbf{S}_{\mathbf{X}_n}$;

repeat

 Update \mathbf{Y} by fixing $\mathbf{S}_{\mathbf{Y}}$ and solving Equation (7);

 Update $\mathbf{S}_{\mathbf{Y}}$ using subset selection with the updated \mathbf{Y} ;

until $\mathbf{Y}, \mathbf{S}_{\mathbf{Y}}$ changes little;

end for

Group together GOPS points sharing their modes.

The GOPS MS algorithm is summarized in Algorithm 1. It starts with a GOPS point $G_{\mathbf{Y}^{(1)}} = \{\mathbf{Y}^{(1)}, \mathbf{S}_{\mathbf{Y}^{(1)}}\}$, sets $\mathbf{Y}^{(1)}$ as the reference subspace of the fit

measure (5) and computes its distances with other points $d_{GO}(G_{\mathbf{Y}^{(k)}}, G_{\mathbf{X}_n}), n = 1, \dots, N$. We follow (1) and locate the next temporary mode $G_{\mathbf{Y}^{(k+1)}}$ by

$$G_{\mathbf{Y}^{(k+1)}} = \arg \min_{G_{\mathbf{Y}}} \sum_{n=1}^N d_{GO}^2(G_{\mathbf{Y}}, G_{\mathbf{X}_n}) w_n(G_{\mathbf{Y}^{(k)}}), \quad (6)$$

where

$$w_n(G_{\mathbf{Y}^{(k)}}) = \psi(d_{GO}^2(G_{\mathbf{Y}^{(k)}}, G_{\mathbf{X}_n}); h).$$

The equation (6) is simplified by substituting (2) into (6) (see supplementary material), and yields

$$G_{\mathbf{Y}^{(k+1)}} = \arg \min_{G_{\mathbf{Y}}} \|\mathbf{Y}\mathbf{Y}^T - \mathbf{C}^{(k)}\|_F^2, \quad (7)$$

where

$$\mathbf{C}^{(k)} = \sum_{n=1}^N w_n(G_{\mathbf{Y}^{(k)}}) (c_1 \mathbf{X}_n \mathbf{X}_n^T + c_2 [\mathbf{S}_{\mathbf{Y}} \mathbf{S}_{\mathbf{X}_n}] [\mathbf{S}_{\mathbf{Y}} \mathbf{S}_{\mathbf{X}_n}]^T). \quad (8)$$

The notion $[\mathbf{S}_{\mathbf{Y}} \mathbf{S}_{\mathbf{X}_n}]$ stands for making a matrix by putting columns of $\mathbf{S}_{\mathbf{X}_n}$ after those of $\mathbf{S}_{\mathbf{Y}}$.

The two components of $G_{\mathbf{Y}^{(k+1)}}$ are computed by fixing one and updating the other respectively. First we fix $\mathbf{S}_{\mathbf{Y}} = \mathbf{S}_{\mathbf{Y}^{(k)}}$ in (8). Since $\mathbf{Y}\mathbf{Y}^T$ is the closest rank- p matrix to $\mathbf{C}^{(k)}$ under Frobenius norm, following [11], we decompose $\mathbf{C}^{(k)} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ with singular value decomposition (SVD) and set $\mathbf{Y}^{(k+1)} = \mathbf{U}(:, 1:p)$ (i.e., the columns of \mathbf{U} associated with the p -largest eigenvalues). The $\mathbf{S}_{\mathbf{Y}^{(k+1)}}$ is updated by performing subset selection introduced later.

The subspace $\mathbf{Y}^{(k+1)}$ is a low dimensional representation of the column space of $\mathbf{C}^{(k)}$. Since those GOPS points $G_{\mathbf{X}_n}$ close to $G_{\mathbf{Y}^{(k)}}$ have large weights in Equation (8), $\mathbf{Y}^{(k+1)}$ essentially fits to observation sets $\mathbf{S}_{\mathbf{X}_n}$ close to $\mathbf{Y}^{(k)}$. Intuitively, we use more observations roughly in $\mathbf{Y}^{(k)}$ to refine estimation of linear subspace. Therefore, $\mathbf{Y}^{(k+1)}$ is more accurate than $\mathbf{Y}^{(k)}$.

To compute the next mode $\mathbf{Y}^{(k+2)}$, we reset the fit measure's reference subspace to the temporary mode $\mathbf{Y}^{(k+1)}$ and repeat above. Since the fit measure (5) is adapted to $\mathbf{Y}^{(k+1)}$, the GOPS distance (2) is updated accordingly. To see why this adaptive distance measure improves clustering accuracy, notice that $\mathbf{Y}^{(k+1)}$ fits better than $\mathbf{Y}^{(k)}$ to observation components of GOPS points near $\mathbf{Y}^{(k)}$. The fit measure between these points becomes smaller. Therefore, those GOPS points are more close and dense as illustrated in Figure 1. This makes clustering tasks easier. We will keep on updating $G_{\mathbf{Y}^{(k+1)}}$ until there is no more change. A mode is then obtained.

3.3 Subset Selection

The observation set $\mathbf{S}_{\mathbf{Y}^{(k+1)}}$ is updated by performing subset selection with $\mathbf{Y}^{(k+1)}$. Suppose observations are generated from a model and have Gaussian noises. According to [12], a normalization of their residual to the model, called

the studentized residual, follows a student's t distribution. We will apply t -tests to these residuals to find a subset of observations lying in $\mathbf{Y}^{(k+1)}$. This process is called subset selection. Before this, let us see how to compute the scalar studentized residual of a linear regression model. Given N scalar observations $\mathbf{S} \in \mathbb{R}^{1 \times N}$, N explanatory d -dimensional vectors $\mathbf{Q} \in \mathbb{R}^{p \times N}$, a Gaussian noise $\epsilon \sim N(0, \sigma^2 I)$ and a regression coefficient vector $\mathbf{X} \in \mathbb{R}^{1 \times p}$ such that $\mathbf{S} = \mathbf{X}\mathbf{Q} + \epsilon$, the studentized residual for an observation $\mathbf{S}_{(n)}$ is defined as

$$t_{(n)} = \frac{\mathbf{S}_{(n)} - \mathbf{X}\mathbf{Q}_{(n)}}{\hat{\sigma}_n \sqrt{(1 - \mathbf{V}_{(n,n)})}}, \quad (9)$$

where $\mathbf{S}_{(n)}$ is the n -th element of \mathbf{S} , $\mathbf{Q}_{(n)}$ is the n -th column of \mathbf{Q} , $\hat{\sigma}_n$ is the estimated variance, $\mathbf{V} = \mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q}$ and $\mathbf{V}_{(n,n)}$ is the n -th diagonal element of \mathbf{V} .

Our model has two major differences compared with the linear regression model above. First, $\mathbf{S}_{(n)}$ is a m -dimensional vector instead of a scalar. Second, our model is a mixture model rather than a single one. Unfortunately, it is unknown if a vector residual follows a t distribution or others, and, if multi-variable tests are proper. We adapt the testing as follows. Firstly, we perform the t test for each dimension of $\mathbf{S}_{(n)}$. If any dimension of $\mathbf{S}_{(n)}$ cannot pass the test, $\mathbf{S}_{(n)}$ is rejected. Secondly, a portion of observations is utilized to estimate variances. The detail is described in Algorithm 2. To guarantee the convergence of GOPS MS, we select the largest size subset $\mathbf{S}_{\mathbf{Y}^{(k+1)}}$ from subset selection such that the cost function (7) decreases.

Algorithm 2 Subset Selection Algorithm

Input: A linear subspace $\mathbf{X} \in \mathbb{R}^{m \times p}$, observations $\mathbf{S}_{\mathbf{X}} \in \mathbb{R}^{m \times N}$ in \mathbf{X} , and a single observation $S_{(h)} \in \mathbb{R}^m$;

Output: if $S_{(h)}$ lies in \mathbf{X} ;

Algorithm:

Let $\mathbf{Q} = \mathbf{X}^T \mathbf{S}_{\mathbf{X}}$ and $Q_{(h)} = \mathbf{X}^T S_{(h)}$;

Compute a residual vector $\mathbf{E} = \mathbf{S}_{\mathbf{X}} - \mathbf{X}\mathbf{Q}$ and sort the norm of each column of \mathbf{E} increasingly;

Select W columns of \mathbf{E} with the smallest norms;

Estimate j -th dimension's variance using these columns by $\hat{\sigma}_{(j,i)} = \frac{\sum_{i=1}^W \mathbf{E}_{(j,i)}}{M-p}$;

Compute a score $t_{(j,h)}$ for $S_{(h)}$'s j -th dimension by (9);

Perform the t test of $t_{(j,h)}$ with the degree of freedom $W - p - 1$ at the significance level of 5%;

If any dimension of $S_{(h)}$ fails this test, $S_{(h)}$ does not lies in \mathbf{X} .

4 Experiments

In this section, we describe experimental setup and apply the GOPS MS to problems of multi-body motion segmentation, affine shape categorization, and face clustering.

4.1 Experiments Setting

In all experiments, we construct GOPS points as follows. For each observation, we find its p nearest neighbors $\mathbf{S}_{\mathbf{X}} \in \mathbb{R}^{m \times p}$ in Euclidean space. Here p is the rank of a linear subspace. Perform SVD on $\mathbf{S}_{\mathbf{X}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and take $\mathbf{U}(:, 1:p)$ as a Grassmann point \mathbf{X} . The set $\mathbf{S}_{\mathbf{X}}$ is taken as observations in \mathbf{X} . The weight c_1, c_2 in d_{GO} is set by $c_1 = 1 - \lambda$ and $c_2 = \lambda$. For all applications, we set $\lambda = 0.9$. The Epanechnikov kernel [5] is selected to compute $w_n(\cdot)$. In subset selection, we use $W = 0.1N$, where N is the number of observations.

Given the number of clusters, the proposed algorithm is compared with the Intrinsic MS [4] in terms of clustering accuracy. The clustering accuracy is measured by the missing rate, which is the ratio of the number of points incorrectly clustered to the total.

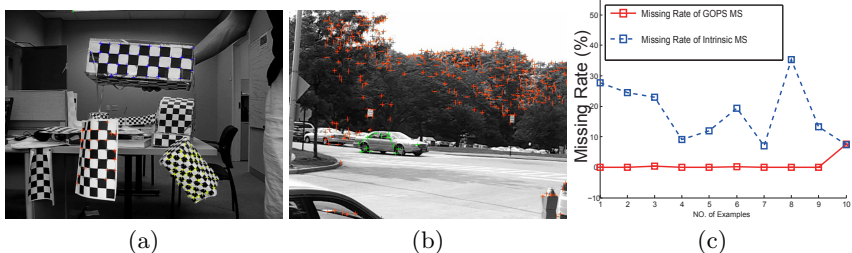


Fig. 2. (a)(b) Results for multi-body motion segmentation. Features of the same color belong to the same group. The segmentation for these examples are 100% correct. The results are better viewed on a screen. **(c) Missing rates of multi-body motion segmentation for ten examples in [4].**

4.2 Multi-Body Motion Segmentation

Multi-body motion segmentation is an essential problem for reconstructing or understanding a dynamic scene with multiple rigidly moving objects. In this task, trajectories of image features are segmented using their motion similarity without knowing the moving object number. Observations are trajectories represented by vectors in \mathbb{R}^{2F} , where F denotes number of frames. Under affine camera models, each motion is a 4 dimensional linear subspace in \mathbb{R}^{2F} [4].

The GOPS corresponds to the Motion-Trajectory space. To compare with the Intrinsic MS, we apply the GOPS MS to the ten examples of the Hopkins155

database selected by the Intrinsic MS. Some results are shown in Figure 2 (a)(b). The comparison of the GOPS MS and the Intrinsic MS is shown in Figure 2 (c). The average missing rate of GOPS MS is 0.8% and better than that of Intrinsic MS, which is 17.9%.

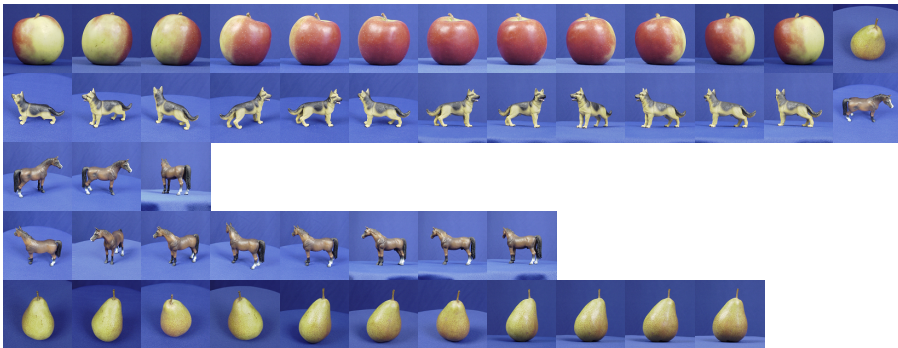


Fig. 3. Shape categorization results of GOPS MS. The images in the second and third rows belong to the dog category. Images in each other row are clustered into one category.

4.3 Affine Shape Categorization

Shape representation and analysis play an important role in object recognition, gait recognition and image registration. Landmark based shape analysis, which represents a shape by landmark points on its contour, is a popular approach for shape representation. The affine shape space method [8] models shapes as affine transformations of a base shape. Given a base shape represented by a $m \times 2$ matrix \mathbf{B} , the shape space is the set $\mathbb{P} = \{\mathbf{P} | \mathbf{P} = \mathbf{B}\mathbf{A}\}$, where \mathbf{A} is any 2×2 full rank matrix. Each shape space is mapped to a linear subspace spanned by \mathbf{B} . The study of affine shape spaces boils down to a study of the points on the Grassmann manifold.

Affine shape categorization attempts to group shapes generated from the same base shape \mathbf{B} together. To perform the GOPS MS, we reformulate the affine shape space as

$$\mathbf{P} = \begin{bmatrix} \mathbf{x} & \mathbf{y} & 0 & 0 \\ 0 & 0 & \mathbf{x} & \mathbf{y} \end{bmatrix} \begin{bmatrix} A_{11}^1 & \cdots & A_{11}^N \\ A_{21}^1 & \cdots & A_{21}^N \\ A_{12}^1 & \cdots & A_{12}^N \\ A_{22}^1 & \cdots & A_{22}^N \end{bmatrix}. \quad (10)$$

Here \mathbf{P} is a $2m \times N$ matrix, which includes N shapes and m landmarks for each shape. x, y are the X-coordinates and Y-coordinates of a base shape $\mathbf{B} = [\mathbf{x} \ \mathbf{y}]$. $A_{i,j}^n$ is the element of the i -th row j -th column of an affine transformation \mathbf{A}^n . In

this application, we use the database ETH-80 [13] as a benchmark. The database ETH-80 includes eight categories of objects. We select four categories and twelve images for each category for categorization.

We follow [14] to extract landmarks. After locating the outmost contour of every image, we take the pixel with the smallest x and y coordinate as a starting point, order the pixels on contour anti-clockwisely and sample 200 landmarks uniformly. Principal component analysis (PCA) is applied to project these shapes onto a subspace of 25 dimensions for computational efficiency. Then the GOPS MS is performed. Our results are shown in Figure 3. The missing rate of GOPS MS is 10.4%, which is better than 35.4% of the Intrinsic MS.

4.4 Face Clustering under Varying Illumination

Given a set of face images taken under varying illumination conditions, face clustering attempts to cluster these images corresponding to the same face. By assuming human faces are Lambertian objects, these images are approximately lying in a low-dimensional subspace [7]. Hence this problem can be solved by the GOPS MS.

We carry on experiments with the Yale Faces B database [15]. This database consists of photos of 10 faces taken under 9 different poses and 64 different illumination conditions. We follow [6] and select nine subsets of the ten faces as examples for clustering. For computational efficiency, these images are down sampled to 120×160 pixels and principal component analysis (PCA) is applied to further reduce each image into a 25-dimensional vector. Table 1 summarizes the missing rates for both algorithms. Compared with the Intrinsic MS algorithm, our algorithm outperforms it for all the examples.

Example Id	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Mean
GOPS	2.5	0	1.1	0	0.7	0.4	4.5	1.0	7.2	1.95
Intrinsic	28.4	0	18.2	24.0	38.4	20.3	19.5	30.4	31.8	23.4

Table 1. Missing rates for face clustering. Numbers in this table refer to the missing rate (%). Each example is a combination of photos of attendees. For simplicity, we use (1,5,8) to stand for a collection of photos of NO.1, NO.5 and NO.8 attendees. These selected combinations are Id1: (5,8), Id2: (1,5,8), Id3: (1,5,8,10), Id4: (1,4,5,8,10), Id5: (1,2,4,5,8,10), Id6: (1,2,4,5,7,8,10), Id7: (1,2,4,5,7,8,9,10), Id8: (1,2,3,4,5,7,8,9,10), Id9: (1,2,3,4,5,6,7,8,9,10).

5 Conclusion

In this paper, we propose a mode seeking algorithm using observations and linear subspaces to improve accuracy of clustering linear subspaces. As a mode seeking

process goes on, the distance measure in use is updated by temporary modes. Distances between similar linear subspaces are decreasing with the updated measure. This facilitates our clustering task. The proposed algorithm can be applied to many computer vision problems, and we demonstrate its effectiveness for multi-body motion segmentation, shape categorization and face clustering. The experimental results show this algorithm improves the clustering accuracy significantly compared with the state-of-the-art Intrinsic MS algorithm.

References

1. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (May 2002) 603–619
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** (2003) 564–577
3. Subbarao, R., Meer, P.: Nonlinear mean shift over riemannian manifolds. *International Journal of Computer Vision* **84**(1) (2009) 1–20
4. Çetingül, H., Vidal, R.: Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (June 2009) 1896–1902
5. Cheng, Y.: Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(10) (August 1995) 790–799
6. Vidal, R.: A tutorial on subspace clustering. In: *IEEE Signal Processing Magazine*. (to appear)
7. Ho, J., Yang, M.H., Lim, J., Leem, K.C., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (June 2003) 11–18
8. Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (June 2008) 1–8
9. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* **21** (1975) 32–40
10. Sheikh, Y., Khan, E., Kanade, T.: Mode-seeking by medoidshifts. In: *IEEE International Conference on Computer Vision*. (October 2007) 1–8
11. Absil, P.A., Mahony, R., Sepulchre, R. In: *Optimization algorithms on matrix manifolds*. Princeton University Press (2007)
12. Cook, R.D., S., W. In: *Residuals and Influence in Regression*. Chapman and Hall in New York (1982)
13. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Volume 2. (2003) 409–415
14. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(2) (February 2007) 286–299
15. Georgiades, A., Belhumeur, P., Kriegman, D.: From few to many: Generative models for recognition under variable pose and illumination. In: *IEEE Computer Society Conference on Automatic Face and Gesture Recognition*. (2000) 277–284