

Improved Spectral Clustering using Adaptive Mahalanobis Distance

Xiping Fu

Dept. of Computer Science
University of Otago
Dunedin, NZ
xiping@cs.otago.ac.nz

Shawn Martin

Software Systems Research
Sandia National Laboratories
Albuquerque, NM 87185, USA
smartin@sandia.gov

Steven Mills

Dept. of Computer Science
University of Otago
Dunedin, NZ
steven@cs.otago.ac.nz

Brendan McCane

Dept. of Computer Science
University of Otago
Dunedin, NZ
mccane@cs.otago.ac.nz

Abstract—In this paper, we consider the manifold clustering problem. In manifold clustering, data are sampled from multiple manifolds and the goal is to partition the data accordingly. Spectral clustering algorithms have been developed to solve this problem, but they tend to fail when the underlying manifolds are very close to each other and/or they intersect. We propose an improvement to spectral clustering algorithms using adaptive neighborhoods computed using Mahalanobis distance. We show the effectiveness of this approach on some artificial data. We further incorporate the modification into recent related algorithms and compare the results on datasets in motion segmentation, handwritten digit recognition, and object rotation.

Keywords—spectral clustering; adaptive Mahalanobis distance; multiple manifolds clustering; motion segmentation;

I. INTRODUCTION

Clustering is widely used in fields such as computer vision, machine learning, and information science. The goal of clustering is to partition data into different groups according to inherent similarity. Clustering has been widely studied and there are many approaches available [2]. As an example, k -means is a simple algorithm which partitions a dataset into k groups by minimizing within cluster Euclidean distance.

Spectral clustering algorithms have been developed to overcome some of the shortcomings of traditional clustering methods [6], [9]. Traditional algorithms often assume normally distributed clusters, while spectral algorithms are capable of clustering non-Gaussian, non-linear data. Spectral clustering works by representing data similarity as a weighted graph. Each point in the dataset is a node in the graph, and the non-zero edges in the graph connect similar data points. This graph can be mapped into a new space by computing eigenvectors (spectra) of an associated matrix. Finally, k -means is used to cluster the data in the new space.

In the past few years, a lot of work has been done to extend clustering algorithms to even more difficult problems. Specifically, algorithms have been designed to cluster data sampled from multiple manifolds. These manifolds may be very close to each other and may even intersect. For this kind of clustering problem, we need to construct elaborate similarity matrices to group the data using spectral clustering. Chen *et al.* have constructed a similarity matrix based on the polar sine, which is a high dimensional generalization of the sine function [3]. This is a global algorithm which has good performance on linear manifolds. Wang *et al.* incorporate

tangent space information into the similarity matrix [14]. They use a mixture of probabilistic principal component analysis (MPPCA) [11] to obtain the tangent space information, followed by distance information to construct a similarity matrix. Gong *et al.* estimate the local tangent space using a weighted low-rank matrix factorization which penalizes neighborhood points according to their distance from the neighborhood center [4]. This tangent space information is used to construct a similarity matrix using both distance and angle information. Finally, Arias-Castro *et al.* have proposed three algorithms which address the manifold intersection problem [1]. The central idea behind these algorithms is to incorporate local covariance information.

In this paper, we propose an algorithm designed to improve the selection of neighborhoods in the case of data sampled from multiple neighborhoods. By improving neighborhood selection, we improve the similarity matrix used by spectral clustering algorithms. Like previous works [4], [14], [1], we are trying to construct a better similarity matrix for spectral clustering. Unlike previous work [4], [14], we do not explicitly estimate local tangent spaces, nor do we use a single covariance measurement to reject certain edges [1]. Instead, we select edges by employing an iterative Mahalanobis distance calculation. Since we are concerned only with neighborhood selection, the modified neighborhood selection method can be applied as a pre-processing step for various spectral clustering algorithms [4], [14], [9].

We organize our presentation as follows. In Section II, we introduce the necessary background for the modified neighborhood selection method, including details on spectral clustering and the Mahalanobis distance. In Section III, we describe how to select manifold respecting neighborhoods using an iterative computation of Mahalanobis distance. We provide examples and discuss computational complexity. In Section IV, we incorporate the modified neighborhood selection method into other algorithms and compare the resulting performance on some real data. In Section V, we conclude the paper and discuss directions for future work.

II. BACKGROUND

Suppose we have a dataset $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^D$. The first step in any spectral clustering algorithm is the construction of a weighted similarity graph. In this graph, vertices correspond to data points \mathbf{x}_i and edges give the

similarity between two points \mathbf{x}_i and \mathbf{x}_j . For example, we might form the weighted similarity graph using ϵ -balls to specify neighborhoods: for each data point \mathbf{x}_i , we connect it to point \mathbf{x}_j if the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j) \leq \epsilon$. Another common approach for generating weighted similarity graphs is to connect each data point to its K nearest neighbors.

After assigning an appropriate weight to each edge, typically the distance $d(\mathbf{x}_i, \mathbf{x}_j)$, we get a weighted adjacency matrix $W = (w_{ij})$ for the graph. Note that $w_{ij} = 0$ if an edge does not exist. Spectral clustering [10] is done by calculating the normalized Laplacian $L = I - D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with $D_{ii} = \sum_{j=1}^N w_{ij}$. Next, we compute the smallest k eigenvalues of the generalized eigenvalue problem $L\mathbf{u} = \lambda D\mathbf{u}$, where $\mathbf{u}_1, \dots, \mathbf{u}_k$ are the corresponding eigenvectors. If we form a matrix $U = (\mathbf{u}_1, \dots, \mathbf{u}_k) = (\mathbf{y}_1^T, \dots, \mathbf{y}_N^T)^T \in \mathbf{R}^{N \times k}$, then \mathbf{y}_i can be viewed as a representation of \mathbf{x}_i . Finally, we cluster $\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ into k clusters using k -means.

A. Mahalanobis Distance

The modified neighborhood selection method uses the Mahalanobis distance to select neighborhoods. To define the Mahalanobis distance, we suppose that $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{R}^D$ and that $\Sigma \in \mathbf{R}^{D \times D}$ is a symmetric positive semidefinite covariance matrix. The Mahalanobis distance is defined as $d_M = ((\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j))^{-\frac{1}{2}}$. Under the Mahalanobis distance, the space \mathbf{R}^D can be viewed as normalized by Σ . In Figure 1, we show a unit sphere in the Mahalanobis distance using two different covariance matrices.



Fig. 1. Unit balls under different Mahalanobis distances. On the left we use $\Sigma = I$, and on the right we use a diagonal matrix with entries $(3, 1, 1)$ for Σ .

III. ALGORITHM

We propose an algorithm for selecting manifold respecting neighborhoods, regardless of nearby manifolds or manifold intersections. The algorithm is based on iteratively recomputing the Mahalanobis distance for a given neighborhood center. To describe the algorithm suppose we have a point \mathbf{x}_i and we want to identify its K nearest neighbors. In the absence of prior information, we assume an isotropic Mahalanobis distance ($\Sigma = I$) to find the closest K neighbors. Using these neighbors, we compute the covariance and re-compute the Mahalanobis distance. We then select a new set of K neighbors based on the new Mahalanobis distance and repeat. In practice, we do not wait for convergence, but terminate after a fixed number of iterations. After the neighborhood selection process, we apply a spectral clustering algorithm. We denote the resulting algorithm as Modified-SC. The modified neighborhood selection method is demonstrated in Figure 2 and summarized below.

Algorithm 1 Identify neighborhood using Mahalanobis Distance

Input: Data set X with N elements, neighborhood size K .
Output: Neighborhood for each data point.
for $i = 1$ to N **do**
 1. Let $\Sigma = I$.
 2. Use distance $(\mathbf{x}_j - \mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x}_j - \mathbf{x}_i)$ to find K nearest neighbors of \mathbf{x}_i .
 3. Calculate the covariance Cov of $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$.
 4. If Cov = Σ , go to step 5, else, set $\Sigma = \text{Cov}$, and return to step 2.
 5. Record the points $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ as \mathbf{x}_i 's neighborhood.
end for

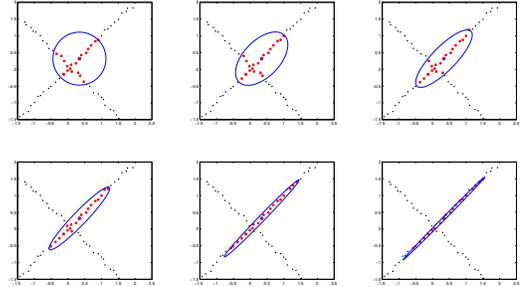


Fig. 2. Neighborhood selection. Here we consider a dataset with two intersecting lines and a neighborhood center near the intersection point. From the top left to the bottom right, we show snapshots of Algorithm 1 as it identifies a suitable neighborhood. The neighborhood center is shown enclosed within a circle surrounding the neighborhood. Non-neighbors are shown in grey. As the algorithm converges, the neighborhood improves so that the intersecting line is ignored.

A. Examples

Next, we examine some artificial examples which cannot be partitioned using traditional spectral clustering algorithms. The first example is two intersecting lines. For this example, we generated 400 points uniformly sampled from two lines. We used $K = 10$ nearest neighbors and identified $k = 2$ clusters. The results of a traditional spectral clustering algorithm [10] compared with the same spectral clustering algorithm modified using our neighborhood selection is shown in Figure 3.

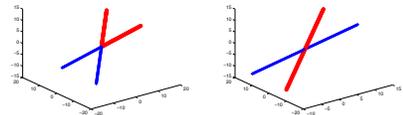


Fig. 3. Clustering for two intersecting lines. On the left we show the result of a traditional spectral clustering algorithm, and on the right the result of the same algorithm modified by first applying Algorithm 1.

In our next example, we use two intersecting planes. For this example, we generated 200 points sampled from a

Gaussian distribution from each plane. We again used $K = 10$ nearest neighbors and identified $k = 2$ clusters. Clustering by our method yielded only two misclassified points. A comparison with the traditional method is shown in Figure 4.

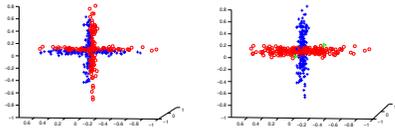


Fig. 4. Clustering for two intersecting planes. On the left we show the results of a traditional spectral clustering algorithm, and on the right the result of the same algorithm modified by our neighborhood selection scheme. There are two misclassified using our approach, shown in green.

The previous two examples show that for manifold intersections, the performance of spectral clustering can be improved using our iterative algorithm for learning the Mahalanobis distance. We note that when the data lies on a manifold without error, the method will work for a large range of K . However, if there is noise, the value of K should be large enough to eliminate the noise. For example, in Figure 5, we show that for two intersecting lines with noise, the data can still be partitioned successfully if we choose $K = 20$.

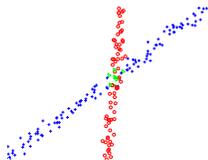


Fig. 5. Clustering for two intersecting lines with noise. The Modified-SC algorithm is able to cluster noisy data using larger neighborhoods. Errors are shown in green.

B. Computational complexity

Spectral clustering involves finding the nearest neighborhood points $O(N^2)$ and solving a generalized eigenvalue problem $O(N^3)$. Our modification involves the neighborhood calculation. Although we have to find the nearest neighbors iteratively, we can record a larger number of neighbors in the first step of the algorithm, and use only those neighbors from that list for further calculations. Thus the modified algorithm has roughly the same complexity as the traditional spectral clustering algorithm.

IV. RESULTS

A. Motion segmentation by trajectories

The Hopkins 155 motion segmentation database (<http://www.vision.jhu.edu/data/>) [12] is a benchmark for testing motion segmentation and subspace learning algorithms. There are 155 data sets, including 13 articulated sequences which relate to people’s motion, 38 traffic sequences, and 104 checkerboard sequences. The checkerboard sequences display different combinations of movements of both checkerboard and camera, such as rotation and translation. Some of the sample images are shown in Figure 6.

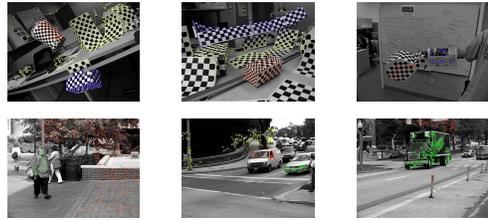


Fig. 6. Sample images from Hopkins 155 motion segmentation database [12].

Suppose there are F frames in one video sequence. A specific feature can be characterized by its position (x_i, y_i) in the i -th frame. The trajectory of this feature is defined as $(x_1, y_1, x_2, y_2, \dots, x_F, y_F) \in \mathbb{R}^{2F}$. Typically trajectories belonging to a specific rigid movement lie in the same low dimension manifold. Therefore, we can use multi-manifold learning algorithms to separate them.

We applied our neighborhood connection method to the standard spectral clustering algorithm (SC), spectral multi-manifold clustering (SMMC) [14], and robust multi-manifold structure learning (RMMSL) [4]. We name the modified algorithms Modified-SC, Modified-SMMC and Modified-RMMSL. We compare these modified algorithms with SC, SMMC, RMMSL, generalized principal component analysis (GPCA) [13], and spectral curvature clustering (SCC) [3] on the Hopkins 155 motion segmentation database. The code for GPCA [13], SCC [3], and SMMC [14] was downloaded from corresponding authors’ homepages. The results of the comparison are shown in Table I.

Note that some of these algorithms have already been tested on this data set, and the results may be different from those shown in Table I. This is probably due to differences in parameter selection and data pre-processing used in each algorithm. For example, Wang *et al.* [14] tested the SCC and SMMC algorithms without data pre-processing [14], i.e. on the $2F$ dimensional data directly, while Chen and Lerman [3] pre-processed the data using PCA. Further Gong *et al.* [4], performed their experiments by first tuning the parameters, then choosing the best parameter for the trials, whereas we do not tune the parameters. Hence, the result shown in Table I can only be compared in the context of our trials. In other words, we tested the methods using the same parameters and pre-processing. We only varied the method of edge connection (neighborhood selection).

The results in Table I compare a large number of algorithms across many datasets. In most cases, our modification improves the results of the standard method, and in certain cases the improvement is significant. Due to the large number of methods considered, however, there are some subtleties in our analysis, which we now discuss.

First, the variety of algorithms requires a standard method of pre-processing and parameter selection which will work for all methods. For pre-processing, we first use PCA to project the trajectories into 5 dimensional space. We chose $D = 5$ dimensions because the original trajectories lie on at most 3 dimensional multi-manifold spaces. Thus after projection the trajectories are still on different low dimensional manifolds

	1R2RCT_A	2RT3RCT_B	1R2TCR	1R2TCRT	ARM	people2	two_cranes_g23	cars5_g13	kanatani1	Average
GPCA	5.94%	35.67%	24.28%	25.65%	1.30%	0.21%	19.48%	1.17%	0%	10.08%
SC	3.80%	4.25%	5.58%	0%	9.09%	28.14%	0%	17.32%	0.74%	9.44%
Modified-SC	6.65%	1.27%	4.86%	0%	2.60%	27.51%	0%	10.23%	0%	8.68%
SMMC	12.17%	1.02%	8.63%	0%	37.66%	0.26%	3.03%	9.45%	1.93%	10.30%
Modified-SMMC	12.30%	1.02%	8.63%	0%	37.66%	0.26%	3.03%	9.71%	0.18%	9.67%
RMMSL	34.44%	40.42%	0%	0%	49.35%	0.21%	32.47%	2.62%	0%	12.36%
Modified-RMMSL	4.51%	38.64%	0%	0%	1.30%	0.21%	1.56%	6.03%	0%	10.15%
SCC	1.20%	1.59%	2.70%	1.01%	0.52%	27.97%	9.22%	0%	0%	3.38%

Tab. I. Performance comparison. Here we compare a host of spectral clustering algorithms using the Hopkins 155 motion segmentation database [12]. Mis-classification rates are reported, and bold-face is used when our modification is as good or better than the result of the standard algorithm. In some cases there are significant improvements.

in \mathbb{R}^5 . Another reason we chose $D = 5$ is that GPCA is designed to address the subspace intersection problem, and a 3 dimension affine space can be viewed as a subset in 4 dimensional space. Thus $D = 5$ is the minimum dimension for GPCA.

Some algorithms, such as SMMC, Modified-SMMC, RMMSL, Modified-RMMSL, and SCC need intrinsic dimension information, which we always set as $d = 3$. For SC, Modified-SC, SMMC, Modified-SMMC, RMMSL, Modified-RMMSL the neighborhood size K is set to $2\lceil\log(N)\rceil$, as suggested when using SMMC [14]. For SMMC, the number of components of MPPCA M is set to 9 and the adjustable parameter ρ is set to 8.

When comparing SC and Modified-SC, the only difference is the parameter for the number of iterations to use in the Mahalanobis distance computation. For SC, we use a single iteration, and for Modified-SC we use 10 iterations. For the comparison of SMMC and Modified-SMMC, our results depend heavily on the outcome of the MPPCA process. Thus for comparison, we fix the MPPCA procedure. In other words, we use the same result from MPPCA, then run the different algorithms. For the RMMSL and Modified-RMMSL, we note that the outliers have been previously removed in the Hopkins 155 dataset. Thus we can skip the outlier detection process in RMMSL, so we set $\sigma_c = 0.01$, $\sigma_n = 1$, and $K_{th} = 5$ for the self-tuning step. After estimating the weighted tangent space for each data point, we apply the different edge connection methods to continue the experiment.

The results in Table I are produced using a set of fixed parameters. For real world data sets, this is not necessarily the best way to choose parameters. In the cars_g13 dataset, for example, it seems that several algorithms have poor performance. When we analyzed this data set in detail, we found that the poor performance was related to the distribution of the points. The cars_g13 data consists of 36 points for car one, and 307 points for the background. By visualizing the data, we noticed that the background points are less localized than the car points, as shown in Figure 7 (left). If we choose $K = 12$ there are in fact three clusters, as shown in Figure 7 (right). Even though two of the three clusters are in the same subspace, they are not connected, thus explaining the poor performance of the algorithms. To address the problem we tried other parameters. For $K = 20$ we obtained better performance (13.11% and 0% for SC and Modified-SC respectively).

Although our modification generally improves the results over the standard algorithm, we noticed that in some cases the

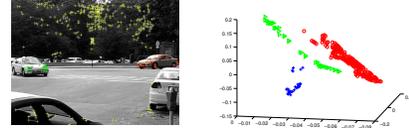


Fig. 7. The cars5_g13 dataset. On the left we show the features provided with the data. On the right we show how the data clusters into three groups in its projection into \mathbb{R}^3 .

results are worse. For example, in Table I, the Modified-SC methods has a higher mis-classification rate than the original SC algorithm for 1R2RCT_A data set. After a detailed analysis, we noticed that for certain data points, the iterative Mahalanobis distance learning may introduce incorrect connections. This situation is shown in Figure 8. The data shown in Figure 8 has a complicated distribution, and this is probably the main reason that the modified neighborhood selection method has failed. To some extent, this problem can be mitigated by allowing larger neighborhoods. For example with $K = 20$ the mis-classification rate is 3.56% for the Modified-SC algorithm and 4.04% for the SC algorithm. However, this is a minor improvement, and we tend to prefer a fixed choice of parameter for fair comparisons.

B. MNIST and COIL20 data set

MNIST [5] and COIL20 [8] are two other commonly used benchmark datasets for object recognition and clustering algorithms. MNIST is a dataset of handwritten digits, with each digit represented by a 28×28 image matrix. The COIL20 data set has 20 objects, and each object has been captured by 72 images taken as the object is rotated by 5° increments. Some of the sample images are shown in Figure 9.

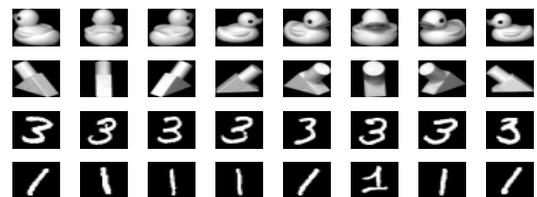


Fig. 9. The top two rows are example images taken from COIL20. The other images are taken from MNIST.

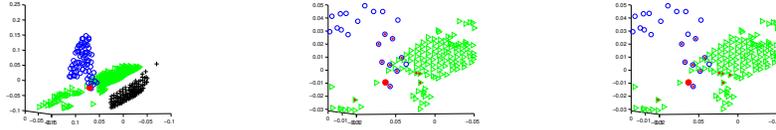


Fig. 8. Mis-connections caused by Mahalanobis learning. On the left, we visualize the features in \mathbb{R}^3 , with points belonging to different clusters distinguished by blue, green, and black. We want to find the neighbors of the red point which belongs to the blue cluster. In the middle, we show its neighbors, which have four mis-connections, in red after one iteration. On the right, we show in red the neighborhood learned by the iterative Mahalanobis algorithm. Notice it has introduced one more mis-connection.

For these data sets, we compared the performance of the SC and Modified-SC algorithms since these two algorithms need the smallest number of parameters. The performance comparison in this experiment is different from the previous one. For each algorithm, we used a range of parameters, then selected the best result for each algorithm. Finally, we compared the best performance of each algorithm.

For the COIL20 data set, we choose several objects for each test. We notice that the results depend on the objects we have chosen. This is because the similarity of different objects varies considerably. As an example, object 1 and 2 appear similar and are difficult to distinguish, while other objects appear very different, which means that it is easy to separate them. Here we show some similar examples and corresponding performance of different algorithms. For each test, the parameters D and K are chosen from 6 to 10 and from 10 to 30 respectively.

For MNIST, we first randomly choose 150 data points from different digits. We project the data points to \mathbb{R}^D , where D ranges from 10 to 20, and the number of nearest neighbors K ranges from 10 to 30. After testing on all of these parameters, we choose the best result for each algorithm. We run the test 10 times, and the average result is shown in Table II.

	COIL20 (1, 2, 3)	COIL20 (1, 2, 3, 4)	MNIST (1, 3)
SC	14.81%	22.57%	21.27%
Modified-SC	6.02%	18.40%	18.13%

Tab. II. Performance comparison. We show the results on the COIL20 and MNIST data sets. Numbers in brackets indicate the object digit number used in the experiment.

V. CONCLUSION

We have proposed an algorithm for using local information to learn an adaptive Mahalanobis distance, and thus provide high quality neighborhoods for various algorithms. We have applied a modified neighborhood selection method to spectral clustering, seeking improvement when data points belonging to different clusters are very near or intersect with each other. However, this type of neighborhood selection could be used in any situation where higher quality edge connections are required.

We incorporated our proposed modification into some recent spectral clustering algorithms [14], [4] and compared the performance of the modification versus the original algorithms. In most cases we obtained improvements with real-world benchmark image datasets.

During our experiments, we noticed that in some cases the modification produces worse performance. We have analyzed

this phenomenon, noting that a change in parameters can often improve the situation. In the future, we hope to find a better way to select the parameters. We also hope to find a convergence proof for the correctness of the proposed algorithm.

ACKNOWLEDGMENT

Sandia is a multipurpose laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. Figures 1-5, 8 and the right image of Figure 7 are licensed by the authors under the Creative Commons Attribution-ShareAlike 3.0 Unported License (CC-BY-SA, <http://creativecommons.org/licenses/by-sa/3.0/>). If reusing these figures please make reference to this article.

REFERENCES

- [1] E. Arias-Castro, G. Lerman, and T. Zhang. Spectral Clustering Based on Local PCA. *arXiv preprint arXiv:1301.2007*, 2013.
- [2] P. Berkhin. A survey of clustering data mining techniques *Grouping multidimensional data*, pp. 1-56, 2006.
- [3] G. Chen and G. Lerman. Spectral Curvature Clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, December 2008.
- [4] D. Gong, X. Zhao, and G. Medioni. Robust multiple manifolds structure learning. *Proc. of the 29th International Conference on Machine Learning*, (1), 2012.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278C2324, November 1998.
- [6] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, August 2007.
- [7] P.C. Mahalanobis. On the generalised distance in statistics *In Proceedings National Institute of Science, India*, 2(1): 49-55, April 1936.
- [8] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report CUCS-005-96, February 1996.
- [9] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information*, 2002.
- [10] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [11] M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 1999.
- [12] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2007.
- [13] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–59, December 2005.
- [14] Y. Wang, Y. Jiang, Y. Wu, and Z.H. Zhou. Spectral clustering on multiple manifolds. *IEEE transactions on neural networks*, 22(7):1149–61, July 2011.