Clustering Appearances of Objects Under Varying Illumination Conditions

Jeffrey Ho[†] Ming-Hsuan Yang[⋆] Jongwoo Lim[‡] Kuang-Chih Lee[‡] David Kriegman[†]

jho@cs.ucsd.edu myang@honda-ri.com jlim1@uiuc.edu klee10@uiuc.edu kriegman@cs.ucsd.edu

[†] Computer Science & Engineering
University of California at San Diego
La Jolla, CA 92093

[⋆] Honda Research Institute
800 California Street
Mountain View, CA 94041

[‡] Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

We introduce two appearance-based methods for clustering a set of images of 3-D objects, acquired under varying illumination conditions, into disjoint subsets corresponding to individual objects. The first algorithm is based on the concept of illumination cones. According to the theory, the clustering problem is equivalent to finding convex polyhedral cones in the high-dimensional image space. To efficiently determine the conic structures hidden in the image data, we introduce the concept of conic affinity which measures the likelihood of a pair of images belonging to the same underlying polyhedral cone. For the second method, we introduce another affinity measure based on image gradient comparisons. The algorithm operates directly on the image gradients by comparing the magnitudes and orientations of the image gradient at each pixel. Both methods have clear geometric motivations, and they operate directly on the images without the need for feature extraction or computation of pixel statistics. We demonstrate experimentally that both algorithms are surprisingly effective in clustering images acquired under varying illumination conditions with two large, well-known image data sets.

1 Introduction

Clustering images of 3-D objects has long been an active field of research in computer vision (See literature review in [3, 7, 10]). The problem is difficult because images of the same object under different viewing conditions can be drastically different. Conversely, images with similar appearance may originate from two very different objects. In computer vision, viewing conditions typically refer to the relative orientation between the camera and the object (i.e., pose), and the external illumination under which the images are acquired. In this paper we tackle the clustering problem for images taken under varying illumination conditions with the object in fixed pose. Recent studies on illumination have shown that images of the same object may look drastically different under different lighting conditions [1] while different objects may appear similar under different illumination conditions [12].

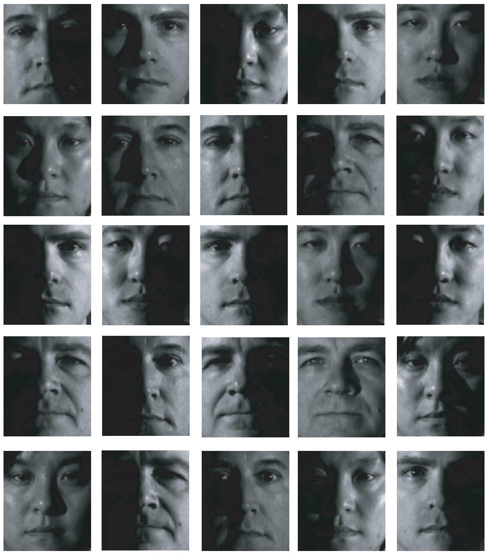

Figure 1: Images under varying illumination conditions: Is it possible to cluster these images according to their identities?

Consider the images shown in Figure 1. These are images of five persons taken under various illumination conditions. For this collection of images, there are two natural clustering problems to be considered: we can cluster them by external illumination condition or by identity. Since the shapes of human faces are very similar, the shadow formations in images taken under the same lighting condition are more or less the same for different individuals. This can be exploited directly by computing some statistical correlations among pixels. Numerous algorithms for estimating lighting direction have been proposed in the literature, e.g., [18, 25, 27], and undoubtedly many of these algorithm can be applied with few modifications to clustering according to lighting. On the other hand, clustering by identity is considerably more challenging. In face recognition for instance, the appearance variation of the same person under different lighting condition is almost always larger than the appearance variation of different people under the same lighting condition [1].

A first glance at the images from the CMU PIE database [23] (See sample images in Figure 1) or the Yale Face Database B [11] (See sample images in Figure 6) may suggest that it is a daunting task to develop an *unsupervised*

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

clustering algorithm to group these images based on *identity*. However, the main contribution of this paper is to show that such pessimism is unwarranted. We propose two simple algorithms for clustering unlabeled images of 3-D objects acquired at fixed pose under varying illumination conditions.

Given a collection of unlabeled images, our clustering algorithms proceed by first evaluating a measure of similarity or affinity between every pair of images in the collection; this is similar to many previous clustering and segmentation algorithms e.g., [13, 22]. The affinity measures between all pairs of images form the entries in an affinity matrix, and spectral clustering techniques [6, 17, 26] can be applied to yield clusters. The novelty of this paper is in the two different affinity measures that form the basis of two different algorithms.

For a Lambertian object, it has been proven that the set of all images taken under all lighting conditions forms a convex polyhedral cone in the image space [4], and this polyhedral cone can be approximated well by a low-dimensional linear subspace [2, 8, 20]. Recall that a polyhedral cone in $\mathbb{R}^s$ is defined by a finite set of generators (or extreme rays) $\{x_1, \cdots, x_n\}$ such that any point x in the cone can be written as a linear combination of $\{x_1, \cdots, x_n\}$ with *non-negative* coefficients. With these observations, the k-class clustering problem for a collection of images $\{I_1, \cdots, I_n\}$ can be cast as finding k polyhedral cones that best fit the data.

For each pair of images I_i, I_j, we define a non-negative number a_{ij}, their conic affinity. Intuitively, a_{ij} measures how likely I_i and I_j come from the same polyhedral cone. The major difference between the conic affinity we introduce here and other affinity measures commonly defined in other clustering and segmentation problems, e.g., [13, 22], is that the conic affinity has a global characteristic while other affinity measures are purely local (e.g., affinity between neighboring pixels). The algorithm operates directly on the underlying geometric structures, i.e., the illumination cones. Therefore, potentially complicated and unreliable procedures such as image features extraction or computation of pixel statistics can be completely avoided.

While the algorithm outlined above exploits the hidden geometric structures (the convex polyhedral cones) in the image space, the second algorithm exploits the effect of the 3-D geometric structure of a Lambertian object on its appearances under varying illumination. In [5], it has been shown that there is no such notion of illumination invariants that can be extracted from an image. However, [5] demonstrated that image gradients can be utilized in a probabilistic framework to determine the likelihood of two images originating from the same object. The important conclusion of [5] is that while the lighting direction can be random, the direction of the image gradient is not. The second algorithm utilizes directly this illumination insensitive property of the image gradient vector. For a pair of images, we define another affinity measure, the gradient affinity. The image gradient vectors at each pixel are first computed. The magnitude and orientation of the image gradient vectors at the corresponding pixels are compared, and the results are aggregated over the entire image to form the gradient affinity.

The first algorithm computes the affinity measures globally in the sense that the affinity between any pair of images is actually determined by the entire collection. The second algorithm, more akin to the usual approach in the literature, computes the affinity between a pair of images using just two images. Both methods are straightforward to implement. We will demonstrate experimentally that these two simple algorithms are surprisingly effective when applied to cluster large collections of unlabeled images. Unlike some clustering problems [13] studied earlier, the clustering problem studied in this paper benefits greatly from many structural results concerning illumination effects that emerged in the past few years, e.g., [2, 4, 20]. It is clear from Figure 1 that a direct approach using the usual L^2-distance metric coupled with standard clustering techniques will not yield promising results. However, this paper shows that it is precisely the use of these subtle structural results which is the gist of the problem; simple and effective solutions can be designed by appealing directly to these structural results.

This paper is organized as follows. In Section 2, we present the two clustering algorithms. Two large image data sets developed for studying illumination variation effects, the CMU PIE database and the Yale database B, are used for the experiments. The results and comparisons with other algorithms are reported in Section 3. We conclude this paper in Section 4 with remarks on this work and future research plans.

2 Clustering Algorithms

In this section, we detail the two proposed clustering algorithms. Schematically, they are similar to other clustering algorithms previously proposed, e.g., [3, 22]. That is, we define similarity measures between all pairs of images. These similarity or affinity measures are represented in a symmetric N by N matrix $A = (a_{ij})$, i.e., the affinity matrix. The second step is a straightforward application of any standard spectral clustering method [17, 26]. The theoretical foundation of these methods has been studied quite intensively in combinatorial graph theory [6]. The novelty of our clustering algorithms lay in the definition of the two affinity measures described below.

This section is organized as follows. In the first subsection, we give the definition of conic affinity and the motivation behind the definition. In the second subsection, we describe an affinity measure based on image gradients. For completeness, we include a brief description of the spectral clustering method used in this paper. The final subsection presents the K-subspaces clustering algorithm, which is a generalization of the usual K-means algorithm. According to [2, 20], we know that images from each cluster should be well approximated by some low dimensional linear subspace. The K-subspace algorithm is designed specifically to ensure that the resulting clusters have this property.

Let $\{I_1, \cdots, I_n\}$ be a collection of unlabeled images. We assume:

1. The images were taken from N different objects with *Lambertian* reflectance. That is, there is an assignment function $\rho : \{I_1, \cdots, I_n\} \rightarrow \{1, \cdots, N\}$.
2. For each cluster of images, $\{I_i | \rho(I_i) = z, 1 \leq z \leq N\}$, all images were taken with the same viewing conditions (i.e., relative position and orientation between the object and the camera). However, the external illumination conditions under which the images were taken may vary widely.
3. All images have the same number of pixels, s.

In the subsequent discussion, n and N will always denote the number of sample images and the number of clusters, respectively.

2.1 Conic Affinity

Let $C = \{x_1, \cdots, x_n\}$ be points in the image space (i.e., the non-negative orthant of $\mathbb{R}^s$) obtained by raster scanning the images. We assume that there is no non-trivial linear dependency among elements of C. This condition is usual satisfied when 1) the dimension of the image space s is greater than the number of samples n and 2) there are no duplicate images in C. As mentioned in Section 1, the clustering problem is equivalent to determining a set of k polyhedral cones that best fit the input data based on the theory in [4]. However, it is rather ineffective and inefficient to search for such a set of k polyhedral cones directly in the high-dimensional image space.

The first step of our algorithm is to define a good metric of the likelihood that a pair of points come from the same cone. In other words, we want a numerical measure that can detect the conic structure underlying in the high-dimensional image space. Recall that at a fixed pose, the set of images of any object under all possible illumination conditions forms a polyhedral cone, and any image in the cone can be represented as a non-negative linear combination of the cone's generators (extreme rays). For each point x_i, we seek a non-negative linear combination of all the other input samples that approximates x_i. In other words, we find non-negative coefficients $\{b_{i1}, \cdots, b_{i(i-1)}, b_{i(i+1)}, \cdots, b_{in}\}$ such that

$$x_i = \sum_{j, j \neq i}^{n} b_{ij} x_j \tag{1}$$

in the least square sense, and $b_{ii} = 0$ for all i.

Let $\{y_1, \cdots, y_k\}$ be a subset of the collection C, i.e., for each j, $y_j = x_k$ for some k. If x_i actually belongs to the cone generated by this subset, this will imply that $b_{ij} = 0$ for any x_j not in the subset. If x_i does not belong to the cone yet lies close to it, x_i can be decomposed as the sum of two vectors $x_i = x_i^c + r_i$ with x_i^c the projection of x_i on the cone and r_i, the residue of the projection. Clearly, x_i^c can be written as a linear combination of $\{y_1, \cdots, y_k\}$ with non-negative coefficients. For r_i, because of the *non-negative* constraint, the non-negative coefficients in the expansion

$$r_i = \sum_{j, j \neq i}^{n} b_{ij}^r x_j. \tag{2}$$

will be dominated by the magnitude of r_i. This follows from the following simple proposition. The proof of the proposition is omitted since it is straightforward. Note that this proposition is false without non-negative constraint on the coefficients. In addition, the proposition holds only for image vectors, vectors in image space $\mathbb{R}^s$ with non-negative components.

Proposition 2.1 *Let I and $\{I_1, \cdots, I_n\}$ be a collection of images. Considered as vectors in the image space $\mathbb{R}^s$, their components are all non-negative. If I can be written as a linear combinations of $\{I_1, \cdots, I_n\}$ with non-negative coefficients:*

$$I = \alpha_1 I_1 + \cdots + \alpha_k I_n \tag{3}$$

where $\alpha_i \geq 0$ for $1 \leq i \leq n$, then $\alpha_i \leq I \cdot I_i$ and $\alpha_i \leq \|I\| / \|I_i\|$.

Therefore, we expect the coefficients in the expansion of x_i to reflect the fact that if x_i were well-approximated by a cone generated by $\{y_1, \cdots, y_k\}$, then the corresponding coefficients b_{ij} would be large (relatively) while others would be small or zero. That is, the coefficients in the expansion should serve as good indicators of the hidden conic structures.

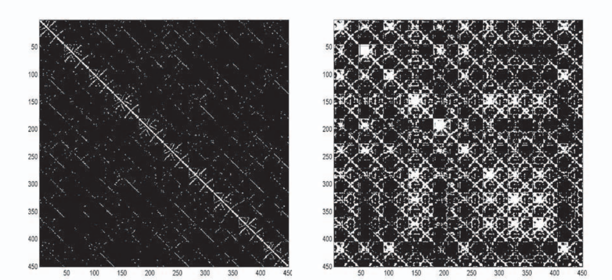

Figure 2: Non-zero matrix entries for **Left**: A matrix using non-negative linear least square approximations. **Right**: A matrix using the usual linear least square approximation without non-negativity constraints.

Another important characteristic of the non-negative combinations is that there are only a few coefficients having significant magnitude. Typically there are only a few nonzero b_{ij} in Equation 3. This is indeed what has been observed in our experiments as well as in prior work on non-negative matrix factorization [14]. Figure 2 shows coefficients of the affinity matrix A (defined below) computed with and without non-negative constraints using a set of 450 images of the Yale B database.

We form a matrix B by taking the coefficients in the expansion in Equation 1 as the entries of $B = (b_{ij})$. We normalize each column of B to so that the sum is 1. This step ensures that the overall contribution of each input image is the same. By construction, $b_{ij} \neq b_{ji}$ in general, i.e., the B matrix is not symmetric. So we symmetrize B to obtain the affinity matrix $A = (B + B^T)/2$.

The time complexity of the algorithm is dominated by the computation of the non-negative least square approximation for each point in the collection. For a collection with a large number of images, solving the least square approximation for every single image is time-consuming. Therefore, we introduce a parameter m which gives the maximum number of images used in non-negative linear least squares estimation. That is, we only consider the m closest neighbors of x_i in computing Equation 1. Here, the distance involved in defining neighbors can be taken to be any similarity measure. We have found that the usual L^2-distance metric is sufficient for the clustering task considered in this paper.

The proposed algorithm, summarized in Figure 3, is very easy to implement and the clustering portion of the algorithm takes less than twenty lines of code in Matlab. The last step involves an optional K-subspace clustering algorithm which will be discussed in Section 2.4.

One previous clustering algorithm that shares some similarities with ours is the work by Basri et. al. [3]. Both methods exploit the underlying geometric structures in the image space: appearance manifolds [16] vs. illumination cones [4]. However, our approach differs fundamentally from theirs in one crucial aspect: their method is based on *local* geometry while ours is based on *global* characteristics. This is because the geometric structures on which the two algorithms operate are different. The algorithm proposed in [3] deals mainly with clustering problems with pose variation but under fixed illumination conditions. The affinity is computed based on local linear structures represented by the tangent planes of the appearance manifold. The non-linear nature of the appearance manifold is reflected by the local affinity measures in the absence of a global linear structure. However, it is clear that one is likely to obtain clustering results based on lighting directions instead of identity by applying such method to the images shown in Figure 1. Note that under similar lighting conditions, the shadow formations on different faces are roughly the same. This implies that the tangential estimation in [3] would produce tangent planes with tangent vectors equal to zeros in the shadowed region. This is, put in the terminology of [3], images taken under the same lighting conditions are more likely to have tangent planes that are nearly parallel.

1. **Non-negative Least Square Approximation**
   Let $\{x_1, \cdots, x_N\}$ be the collection of input samples. For each input sample x_i, compute a non-negative linear least square approximation of x_i by all the samples in the collection except x_i

   $$x_i \approx \sum_{j, j \neq i} b_{ij} x_j$$

   with $b_{ij} \geq 0 \quad \forall j \neq i$, and set $b_{ii} = 0$. Normalize $\{b_{i1}, \cdots, b_{ik}\}$: $$b_{ij} = \frac{b_{ij}}{\sum_l b_{il}}.$$

   (If N is too large, use only m closest neighbors of x_i for the approximation.)
2. **Compute the Affinity Matrix**
   (a) Form the B matrix $B = (b_{ij})$.
   (b) Let $A = (B + B^T)/2$
3. **Spectral Clustering**
   Using A as the affinity matrix, apply any standard spectral method for clustering.
4. **(Optional) K-subspace Clustering**
   Apply K-subspace clustering to further exploit the linear geometric structures hidden among the images.

Figure 3: Clustering algorithm based on conic affinity.

To cluster these images according to identity, the underlying linear structure is actually a global one, and the problem becomes finding polyhedral cones for each person in which an image of that person can be reconstructed by a non-negative linear combination of basis images (generators of the cone). Given an image I, our algorithm considers all the other images in order to find the set of images (i.e., the ones in the same illumination cone) that best reconstruct I. However, this cannot be realized by the approach in [3] which operates only on a pairwise basis.

2.2 Gradient Affinity

In the previous two subsections, we have explored the possibilities of defining affinities by exploiting the hidden conic structures in the image space. In this section, we explore the possibilities of defining affinities using the object's 3-D geometry. The effect of the object's geometry on its images taken under different illumination conditions has been analyzed in great detail in [5]. There, it has been shown that the important quantity to compute for studying illumination variation is the image gradient. For a Lambertian surface, the image gradient ∇I depends on the object geometry (surface normal $\mathbf{n}$) and the albedo α as:

$$\nabla I = (\hat{u}\kappa_u S_u + \hat{v}\kappa_v S_v) + (\nabla \alpha) S \cdot \mathbf{n}. \qquad (4)$$

Here, the $\hat{u}$ and $\hat{v}$ are local tangential directions defined by the principal directions, κ_u and κ_v are the two principal curvatures, and S is the lighting direction. Further analysis based on this equation has shown that the magnitudes and orientations of the image gradient vectors form a joint distribution which can be utilized to compute the likelihood of two images originating from the same object.

We take a simpler approach using direct comparison between image gradients. That is, we sum over the image plane the differences in the magnitude of the image gradient and the relative orientation (i.e., angular difference between the two corresponding image gradients). Given a pair of images I_i and I_j. Let ∇I_i and ∇I_j denote their image gradients. First, we define M_{ij} as the sum over all pixels of the squared-differences between the magnitudes of ∇I_i and ∇I_j.

$$M_{ij} = \sum_{w=1}^{s} (\|\nabla I_i(w)\| - \|\nabla I_j(w)\|)^2 \qquad (5)$$

Next, we calculate the difference in orientation. O_{ij} is defined as the sum over all pixels of the squared-angular differences

$$O_{ij} = \sum_{w=1}^{s} (\angle(\nabla I_i(w), \nabla I_j(w))^2 \qquad (6)$$

Prior to computing the gradients, the image intensities are normalized to $\{0, 1\}$, and the angular difference between the two image gradients are also normalized from the range of $\{-\pi, \pi\}$ to $\{0, 1\}$. The algorithm, summarized in Figure 4, is again very easy to implement.

2.3 Spectral Clustering

For completeness, we briefly summarize the spectral method [17] that we use in this paper though other spectral clustering methods could have been incorporated. Let A be the affinity matrix, and D be a diagonal matrix where D_{ii} is the sum of i-th row of A. First, we normalize A by computing $M^{'} = D^{-1/2} A D^{-1/2}$. Second, we compute the N largest eigenvectors $w_1, \ldots, w_k$ of $M^{'}$ and form a matrix $W = [w_1 w_2 \ldots w_N] \in \mathbb{R}^{n \times N}$ by stacking the column eigenvectors. We then form the matrix Y from W by re-normalizing each row of W to have unit length, i.e., $Y_{ij} = W_{ij} / (\sum_j W_{ij}^2)^{1/2}$. Each row of Y can now be viewed as a point on a unit sphere in $\mathbb{R}^N$. The main point of [17] is that after this transformation, the projected points on the unit sphere should form N tight clusters. These clusters on the unit sphere can then be detected easily by an application of the usual K-means clustering algorithm. We let $\rho(x_i) = z$ (i.e., cluster z) if and only if row i of the matrix Y is assigned to cluster z.

1. **Compute Image Gradients**
   Let $\{I_1, \cdots, I_N\}$ be the collection of input images with s pixels. Let ∇I_i denote the image gradient of I_i. For $1 \leq i, j, \leq N$, define
   $$M_{ij} = \sum_{w=1}^{s} (\|\nabla I_i(w)\| - \|\nabla I_j(w)\|)^2$$
   and
   $$O_{ij} = \sum_{w=1}^{s} (\angle(\nabla I_i(w), \nabla I_j(w))^2$$
2. **Compute Affinity Matrix**
   Set the entries of the affinity matrix A as
   $$A_{ij} = \exp(-\frac{1}{2\sigma^2}(M_{ij} + O_{ij}))$$
   for some real number σ.
3. **Spectral Clustering**
   Using A as the affinity matrix and apply any standard spectral method for clustering.
4. **(Optional) K-subspace Clustering**
   Apply K-subspace clustering to further exploit the linear geometric structures hidden among the images.

Figure 4: Clustering algorithm based on gradient affinity.

2.4 K-Subspace Clustering

A typical spectral clustering method analyzes the eigenvectors of an affinity matrix of data points where the last step often involves thresholding, grouping or normalized cuts [26]. For the clustering problem considered in this paper, we know that the data points come from a collection of convex cones which can be approximated well by low dimensional linear subspaces. Therefore, each cluster should also be well-approximated by some low-dimensional subspace. We therefore exploit this particular aspect of the problem and supplement with one more clustering step on top of the results obtained from spectral analysis. The algorithm we are using is a variant of the usual K-means clustering algorithm. While the K-means algorithm basically finds K cluster centers using point to point distance metric, the task here is to find k linear subspaces using point to plane distance metric.

The K-subspace clustering algorithm, summarized in Figure 5, iteratively assigns points to a nearest subspace (cluster assignment) and, for a given cluster, it computes a subspace that minimizes the sum of the squares of distance to all points of that cluster (cluster update). Similar to the K-means algorithm, the K-subspace clustering method terminates after a finite number of iterations. This is the consequence of the following two simple observations:

1. There are only finitely many ways that the input data points can be assigned to k clusters.
2. Define an objective function (of a cluster assignment)

1. **Initialization**
   Starting with a collection $\{S_1, \cdots, S_K\}$ of K subspaces of dimension d, where $S_i \subset \mathbb{R}^s$. Each subspace S_i is represented by one of its orthonormal bases, U_i (represented as a s-by-d matrix).
2. **Cluster Assignment**
   We define an operator $P_i = \mathrm{I}_{\mathrm{s}\times\mathrm{s}} - U_i U_i^T$ for each subspace S_i. Each sample x_i is assigned a new label $\rho(x_i)$ such that
   $$\rho(x_i) = \arg\min_q \|P_q(x_i)\| \tag{7}$$
3. **Cluster Update**
   Let Σ_i be the scatter matrix of the sampled labeled as i. We take the eigenvectors corresponding to the top d eigenvalues of Σ_i to form an orthonormal basis U_i of $S_i^{'}$. Stop when $S_i^{'} = S_i$ for all i. Otherwise, go to Step 2.

Figure 5: K-subspace clustering algorithm.

as the sum of the square of the distance between all points in a cluster and the cluster subspace. It is obvious that the objective function decreases during each iteration.

The result of the K-subspace clustering algorithm depends very much on the initial collection of k subspaces. Typically, as for the case with K-means clustering, the algorithm only converges to some local minimum which may be far from optimal. However, after applying the clustering algorithm using either the conic or gradient affinity, we have a new assignment function ρ', which is expected to be close to the true assignment function ρ. We will use ρ' to initiate the K-subspace algorithm by replacing the assignment function ρ in the Cluster Assignment (see Figure 5) with ρ'.

3 Experiments and Results

We performed numerous experiments using the Yale Face Database and the CMU PIE database, and compared the results with those obtained by other clustering algorithms. From the Yale Face Database B, we drew two subsets; in one subset all images are in frontal pose while in the other (nonfrontal), the viewing direction is 22 degrees from frontal. Each of these two subsets consists of 450 images with 45 images of each person acquired under varying light source directions ranging from frontal illumination to 70 degrees from frontal (See [11] for more details). Figure 6 shows sample images of two persons from these subsets. Each image is then manually cropped and downsampled to 21 × 24 pixels for computational efficiency.

From the CMU PIE database, we used a subset (PIE 66) of 21 frontal images of 66 individuals which were taken under different illumination conditions but without an ambient light. See Figure 1 for sample images from the PIE 66 subset. Note that this is a more difficult subset than the subset of the PIE database containing images taken with ambient background light. Similar to pre-processing with the Yale dataset, each image is manually cropped and downsampled to 21 × 24 pixels. Clearly the large appearance variation of the same person in these data sets makes the face recognition problem rather difficult [11, 24], and thus the clustering problem extremely difficult. Nevertheless we will show that our methods achieve very good clustering results, and outperform numerous alternative algorithms.

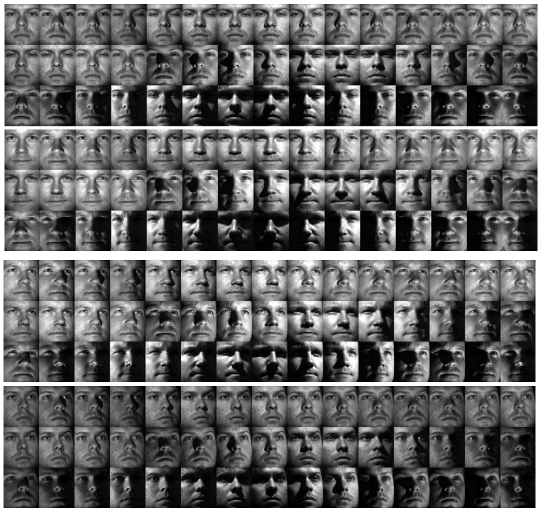

Figure 6: Sample images acquired at frontal view (Top) and a non-frontal view (Bottom) in the Yale database B.

We tested several clustering algorithms with different setups and parameters, where we further assume the number of clusters, i.e., k, is known. Recent results on spectral clustering algorithms show that it is feasible to select an appropriate k value by analyzing the eigenvalues [6, 17, 26, 19]. The distance metric for experiments with the K-means and K-subspace algorithms are the L^2-distance in the image space, and the parameters were empirically selected. We repeated experiments several times to get average results since they are sensitive to initialization and parameter selections, especially in the high-dimensional space.

Table 1 summarizes the experimental results achieved by each method: the proposed conic affinity method with K-subspace method (conic+non-neg+spec+K-sub), variant of our method where K-means algorithm is used after spectral clustering (conic+non-neg+spec+K-means), method using conic affinity and spectral clustering with K-subspace method but without non-negative constraints (conic+no-constraint+spec+K-sub), the proposed gradient affinity method (gradient aff.), straightforward application of spectral clustering where K-means algorithm is utilized as the last step (spectral clust.), straightforward application

COMPARISON OF CLUSTERING METHODS

Method	Error Rate (%) vs. Data Set		
	Yale B (Frontal)	Yale B (Non-frontal)	PIE 66 (Frontal)
Conic+non-neg +spec+K-sub	0.44	4.22	4.18
Conic+non-neg +spec+K-means	0.89	6.67	4.04
Conic+no-constraint +spec+K-sub	62.44	58.00	69.19
Gradient aff.	1.78	2.22	3.97
Spectral clust.	65.33	47.78	32.03
K-subspace	61.13	59.00	72.42
K-means	83.33	78.44	86.44

Table 1: Clustering results using various methods.

of K-subspace clustering method (K-subspace), and the K-means algorithm. The error rate is computed based on the number of images that are assigned to the wrong cluster as we have the ground truth of each image in these data sets.

Our experimental results suggest a number of conclusions. First, the results clearly show that our methods using conic or gradient affinity outperform other alternatives by a large margin. Comparing the results on rows 1 and 3, they show that the non-negative constraints play an important role in achieving good clustering results. Second, the proposed conic and gradient affinity metric facilitates spectral clustering method in achieving very good results. The use of K-subspace further improves the clustering results after applying conic affinity with spectral methods (See also Figure 7). Finally, a straightforward application of the K-subspace or K-means algorithm fails miserably.

COMPARISON OF CLUSTERING METHODS

Method	Error Rate (%) vs. Data Set	
	Yale B (Non-frontal) Subjects 1-5	Yale B (Non-frontal) Subjects 6-10
Conic+non-neg +spec+K-sub	0	0
Gradient+spec K-sub	8.90	6.67

Table 2: Clustering results with high-resolution images.

To further analyze the strength of the conic and gradient affinities, we applied the proposed metrics to cluster high-resolution images (i.e, the original 168×184 cropped images). Table 2 shows the experimental results using the non-frontal images of the Yale database B. For computational efficiency, we further divided the Yale database B into two sets. The results demonstrate that the method using conic affinity metric and spectral clustering renders perfect results. The experiments also show that applying the gradient affinity metric to low-resolution images gives better clustering results than that in high-resolution images. This suggests that computation of gradient metric is more reliable in low-resolution images, and surprisingly such information is sufficient for the clustering task considered in this paper.

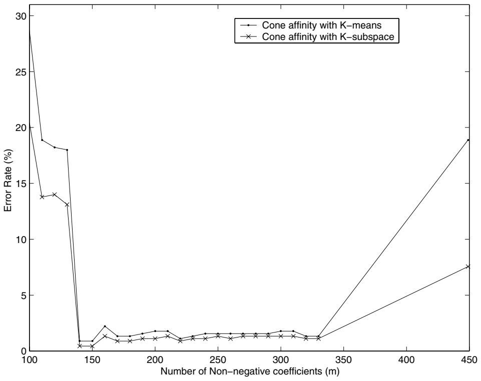

Figure 7: Effects of parameter selection on clustering results with the Yale database B.

For the conic affinity, the main computational load lies in the non-negative least square approximation. When the number of sample images is large, it is not efficient to use all the other images in the data set for approximation. Instead, the non-negative least square are only computed for m nearest neighbors of each image. Figures 7 and 8 show the effects of m on the clustering results for the proposed method with or without K-subspace clustering using the Yale database B and the PIE database. The results show that our method with conic affinity is robust within a wide range of parameter selection (i.e., number of non-negative coefficients in linear approximation).

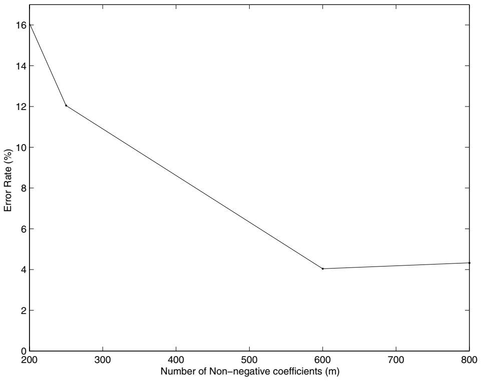

Figure 8: Effects of parameter selection on clustering results with the PIE 66 (frontal) database.

4 Conclusion and Future Work

We have proposed two appearance-based algorithms for clustering images of 3-D objects under varying illumination conditions. Unlike previous image clustering problems, the clustering problem studied in this paper is highly structured. We have demonstrated experimentally that the algorithms are very effective with two large data sets. The most striking aspect of the algorithms is that the usual computer vision techniques such as the image feature extraction and computation of pixel statistics are completely unnecessary.

Our clustering algorithms and experimental results complement the earlier results on face recognition [11, 24, 15]. Invariably, these algorithms aim to determine the underlying linear structures using only a few training images. The difficulty is how to effectively use the limited training resource so that the computed linear structures is close to the real one. In our case, the linear structures are hidden among the input images, and the task is to detect them for clustering.

The holy grail in image clustering is an efficient and robust algorithm that can group images according to their identity with both pose and illumination variation. While illumination variation produces a global linear structure, only local linear structures are meaningful for pose variation [3, 9]. Clustering with local linear structures has been proposed in [19] based on the work of [21]. A clustering method based on these algorithms and our work may therefore be able to handle both pose and illumination variation. On the other hand, the algorithm we proposed can be applied to other problem domains where the data points are known to originate from some linear or conic structures. We will address these issues from combinatorial and computational geometry perspectives in our future work.

Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. This work was carried out at Honda Research Institute, and was partially supported under grants from the National Science Foundation, NSF EIA 00-04056, NSF CCR 00-86094 and NSF IIS 00-85980.

References

[1] Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):721–732, 1997.

[2] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. In *Proc. Int'l Conf. on Computer Vision*, volume 2, pages 383–390, 2001.

[3] R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 414–420, 1998.

[4] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. In *Int'l Journal of Computer Vision*, volume 28, pages 245–260, 1998.

[5] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 254–261, 2000.

[6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[7] S. Edelman. *Representation and Recognition in Vision*. MIT Press, 1999.

[8] R. Epstein, P. Hallinan, and A. Yuille. 5+/-2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE Workshop on Physics-Based Modeling in Computer Vision*, 1995.

[9] A. W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *Proc. European Conf. on Computer Vision*, pages 304–320, 2002.

[10] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12):1312–1328, 1999.

[11] A. Georghiades, D. Kriegman, and P. Belhumeur. From few to many: Generative models for recognition under variable pose and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(6):643–660, 2001.

[12] D. W. Jacobs, P. N. Belhumeur, and R. Basri. Comparing images under varying illumination. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 610–617, 1998.

[13] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[14] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:781–791, 1999.

[15] K.-C. Lee, J. Ho, and D. Kriegman. Nine points of lights: Acquiring subspaces for face recognition under variable lighting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 519–526, 2001.

[16] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int'l Journal of Computer Vision*, 14(1):5–24, 1995.

[17] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 15*, pages 849–856, 2002.

[18] A. P. Pentland. Finding the illuminant direction. *J. of Optical Society America A*, 72(4):448–455, 1982.

[19] P. Perona and M. Polito. Grouping and dimensionality reduction by locally linear embedding. In *Advances in Neural Information Processing Systems 15*, pages 1255–1262, 2002.

[20] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *Proc. SIGGRAPH*, pages 117–228, 2001.

[21] S. T. Roweis and l. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

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

[23] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, pages 53–58, 2002.

[24] T. Sim and T. Kanade. Combining models and exemplars for face recognition: An illuminating example. In *CVPR workshop on Models versus Exemplars in Computer Vision*, 2001.

[25] S. Ullman. On visual detection of light sources. *Biological Cybernetics*, 21:205–212, 1976.

[26] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. Int'l Conf. on Computer Vision*, volume 2, pages 975–982, 1999.

[27] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(7):680–702, 1991.