Adjustment Learning and Relevant Component Analysis

Noam Shental[1], Tomer Hertz[1], Daphna Weinshall[1], and Misha Pavel[2]

[1] School of Computer Science and Engineering and the Center for Neural Computation, The Hebrew University of Jerusalem, Jerusalem, Israel 91904

[2] Department of Electrical and Computer Engineering, Oregon Health and Science University, Beaverton OR 97006

fenoam,tomboy,daphna@cs.huji.ac.il, pavel@ece.ogi.edu

Abstract. We propose a new learning approach for image retrieval, which we call *adjustment learning*, and demonstrate its use for face recognition and color matching. Our approach is motivated by a frequently encountered problem, namely, that variability in the original data representation which is not relevant to the task may interfere with retrieval and make it very difficult. Our key observation is that in real applications of image retrieval, data sometimes comes in small chunks - small subsets of images that come from the same (but unknown) class. This is the case, for example, when a query is presented via a short video clip. We call these groups *chunklets*, and we call the paradigm which uses chunklets for unsupervised learning *adjustment learning*. Within this paradigm we propose a linear scheme, which we call Relevant Component Analysis; this scheme uses the information in such chunklets to reduce irrelevant variability in the data while amplifying relevant variability. We provide results using our method on two problems: face recognition (using a database publicly available on the web), and visual surveillance (using our own data). In the latter application chunklets are obtained automatically from the data without the need of supervision.

1 Introduction

We focus in this paper on a fundamental problem in data organization and data retrieval, which is often ignored or swept under the carpet of "feature selection". More specifically, our domain is image retrieval and image organization based on between-image distance. It is assumed that the specifics of the image representation and between-image distance were pre-determined, typically based on some engineering design that took into account a very large scope of tasks and data. Within this domain, data organization essentially requires graph-based clustering. Retrieval is based on the distance from a query image to stored images, and is essentially nearest neighbor classification.

We identify the following problem: Assuming that the data is represented compactly, a typical distance (whether Euclidean in feature space or other) is affected by all the variability that is maintained in the data representation indiscriminantly. However, for a particular query some of this variability is irrelevant.

For example, if we submit as a query the image of Albert Einstein sticking his tongue out, the relevant features in the image may be the expression (if we want to retrieve other silly faces), or the relevant features may be the facial features (if we want to retrieve other pictures of Albert Einstein).

Irrelevant variability is also a problem in data organization, e.g., in clustering (which has been used in vision for image retrieval [7]). Typically we collect data and rely on the pre-determined representation and distance function to cluster the data. However, depending on the context of the data collection, some variability is always irrelevant. Unaccounted for, this will damage the results, since clustering only gives as good results as the underlying representation and distance function.

More precisely, we first define data variability to be *correlated with a specific body of data and a specific task*, if the removal of this variability from the data deteriorates (on average) the results of clustering or retrieval. We then define *irrelevant variability* as data variability which satisfied the following conditions:

- it is normally maintained in the data (i.e., it is used in the representation and/or distance function explicitly or implicitly);
- it is **not** *correlated with the specific task* according to the definition above and the task at hand.

Intuitively, in a task where irrelevant variability is large, performance using the general and indiscriminating pre-determined distance measure can be poor. For example, when a certain irrelevant environmental feature dominates all other features because of its large variability, nearest neighbor classification using the indiscriminating distance function could perform poorly. This is the problem addressed in the present paper.

We propose an unsupervised adjustment scheme, which will modify the distance function so as to enhance its discriminative capabilities on the given test data. For this purpose we identify and use relatively small sets of data points, in which the class label is constant, but unknown. We call these small sets of data points *Chunklets.*

We justify our use of *chunklets* for data clustering and retrieval by the following observations:

Data organization: we observe that very often data naturally arrives in chunks in which the message is held constant. For example, in speaker identification, short utterances of speech are likely to come from a single speaker. In surveillance, short video clips obtained by tracking a moving target are likely to come from a single target. In both examples, the magnitude of the variance due to the irrelevant variability is comparable if not larger than the relevant variability. In both examples, we can automatically detect chunks of data where the message is constant while the context changes.

Data retrieval: in image retrieval based on nearest-neighbor classification chunklets may also come by naturally. For example, often a query includes a number of images, especially when the user is allowed to use some of the retrieved images in the first stage to refine the answers of the system in a second stage. In the future we may expect to see queries in the form of short video clips, which under some circumstances can also provide *chunklets*.

We claim that, in data for which chunklets are known, irrelevant variability can be reduced without the use of a fully labeled training set and without building an explicit model of the sources that created it. The first contribution of the present paper is to note this observation and propose to use it for unsupervised learning. We call this learning paradigm *adjustment learning. Adjustment learning* lies between supervised and unsupervised learning (see Section 2); we place it closer to unsupervised learning, since it extracts the information from the test data and not from prior labeled data.

The second contribution of this paper is to develop a method to drive *adjustment learning* in simple cases where such linear methods as Fisher Linear Discriminant (FLD) are effective. We call the proposed method Relevant Component Analysis (RCA), and discuss it more formally in Section 3. In Section 4 we demonstrate the use of RCA in a facial image classification task, using a public database of faces which was obtained from the web. Section 5 demonstrates the use of RCA in a visual surveillance and color constancy problem.

Our approach is related to the work on multi-view representation, and the use of image manifolds for image classification and retrieval ([9, 3], see also [6]). However, whereas these approaches rely on prior training data to extract specific regularities in the overall data, our approach works to identify regularities in the test data which are *not* generally seen in the training data.

The problem of irrelevant variability has been addressed in the literature under different guises. One approach is based on adaptation. In signal processing, adaptation is a convenient way of tailoring the filter to the environment, if the external environment is not known in advance [2]. In learning, the recognizer is trained on a limited training set, but allowed to adapt to the specific values (range of values) of the irrelevant variables. In some situations this approach is viable, but it requires labeled training set for the "adaptation" process.

Another approach uses a "mixture of experts" [11], where multiple recognizers are trained in such a way as to encourage each "expert" to differ from the rest. The outputs of the individual experts are then gated by some estimate of that expert's confidence, and then combined in a principled manner. The mixture-of-experts solution has, in principle, the potential to yield the best possible solution, but the solution is critically dependent on the requirement that the training data captures the true distribution of the real-world observations. We note in passing recent statistical approaches such as boosting, and for that matter support vector machines [11], which greatly improve the training processes, but are plagued by the same dependencies on the veridicality of the training set.

2 What is *adjustment learning*?

Adjustment learning, as we pursue it here, is distinguished by the following features:

- We focus on classification - the generation of equivalence classes at the output that would be relevant to the task. This classification scheme accomplishes recognition by exemplars, where a query is associated with a set of examples rather than an abstract label.
- We attempt to diminish the effects of irrelevant variability, without modeling the exact sources of this variability and without reconstructing the exact mixture of sources that created it.
- We *do not* assume access to labeled data for training and adaptation. Thus our approach is cast into the domain of unsupervised learning. We only assume that chunklets can be identified in the data.

More specifically, we assume that our test data comes naturally divided into chunklets, which are small subsets of input equivalence classes. Each chunklet may include a small number of measurements (1 or more). In each chunklet the class label is known to be fixed, although the exact label is *unknown.* A chunklet must be defined naturally by the data – we cannot rely on some teacher to label the data for us.

The relationship among the data points, chunklets, and the desired classes is illustrated in Fig. 1, using for illustration two-dimensional input data. On the left side of Fig. 1 we illustrate the typical situation for unsupervised classification. The other extreme, when a complete labeled set is available for supervised learning, is illustrated on the right side of Fig. 1. The intermediate situation is illustrated in the middle of Fig. 1, where each ellipse denotes a chunklet - a set of test data where all data points have an identical but unknown class label.

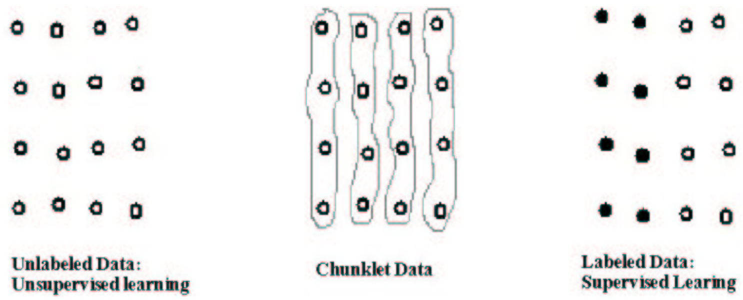

Fig. 1. In the supervised learning paradigm (right), each data point $x \in X$ is given with an associated label $y \in Y$; the system learns the association between data and label. In the unsupervised learning paradigm (left) each data point $x \in X$ is given without an associated label; the system can only learn the characteristics of the distribution of data points. In our paradigm (middle), the data is given in chunklets of points. For each chunklet it is known that the class label has been kept fixed for all the data points that belong to it, but the label itself is unknown.

3 Relevant Component Analysis (RCA)

We assume that the data is represented as vectors of features, and that the Euclidean distance in the feature space is used for classification and retrieval. Our method modifies the feature representation of the data space by a linear transformation W, such that the Euclidean distance in the transformed space is less affected by irrelevant variability. Effectively, the Euclidean distance in the transformed space is equivalent to the Mahalanobis distance in the original data space with weight matrix $W^T W$ (thus W may be viewed as accomplishing feature selection). We call W the *rescaling transformation.*

Our method is related to principal component analysis (PCA). PCA is a linear method for redundancy reduction which compresses the description of the data along the dimensions of smallest variability. It is guaranteed to achieve the least expected mean-square error for a given size of representation. In a similar way, we define RCA (Relevant Component Analysis) as the compression of the data description along the dimensions of highest *irrelevant* variability. Specifically, in RCA we transform the data space by a transformation matrix W which assigns large weights to "relevant dimensions" and low weights to "irrelevant dimensions"; thus we maintain concurrently as much relevant variability as possible and as little irrelevant variability as possible within the given data set.

We now present our method for computing the *rescaling transformation* W. We start in Section 3.1 with a text-book approach to the computation of W for normal class distribution using labeled data. In Section 3.2 we describe how to estimate W from unlabeled data, where chunklets are available. Finally we summarize the proposed algorithm in Section 3.3.

The assumptions made in Section 3.1 are needed to prove that our method may give optimal classification (i.e., it is equivalent to the Bayes classifier). Regardless of this, we propose RCA as a general method that may be used under rather general circumstances, just like such classifiers as the *Distance Classifier* and the *Matched Filter* [1].

3.1 Computation of the *rescaling* transformation using labeled data

In this section we show how to compute the desired *rescaling transformation* W from labeled data. We define the assumptions on data distribution which allow us to prove that nearest neighbor classification based on the Euclidean distance in the transformed space is statistically optimal.

The Whitening transformation Let $\boldsymbol{x} \in \Omega$ denote the sampled data, and $|\Omega|$ denote the size of the sample. Similarly, let $|\Omega_m|$ denote the size of the sample from class m where $\bigcup \Omega_m = \Omega$. Let the random variable $\mathbf{C}_m$ denote the distribution of the m-th class, and M denote the number of classes. For simplicity we also assume that $\mathbf{C}_m$ is distributed normally, i.e. $\mathbf{C}_m \sim N(\mu_m, \Sigma_m)$, where μ_m denotes the mean of the class and Σ_m denotes its covariance matrix. Let S_W

denote the within-class scatter of the data set, defined by

$$S_W = \frac{1}{|\Omega|}\sum_{m=1}^{M} |\Omega_m|\hat{\Sigma}_m \tag{1}$$

where

$$\hat{\Sigma}_m = \frac{1}{|\Omega_m|}\sum_{\boldsymbol{x}\in\mathbf{C}_m} (\boldsymbol{x}-\mu_m)(\boldsymbol{x}-\mu_m)^T$$

Assume first that the covariance matrix of all the classes is identical, i.e. $\Sigma_m = \Sigma \;\; \forall m$, and therefore $S_W = \Sigma$ for large enough sample. In this case the optimal *rescaling transformation* [1] is the whitening transformation W defined as

$$W = V\Lambda^{-\frac{1}{2}}$$

where V is the orthogonal matrix of eigenvectors of S_W, and Λ is its corresponding matrix of singular values. V and Λ may be obtained from the singular value decomposition of $S_W = V\Lambda V^T$.

In the whitened space, nearest neighbor retrieval based on the Euclidean distance is equivalent to maximum likelihood estimation. This is because in the transformed whitened space, the covariance matrices of all the classes are spherical, which is equivalent to the assumption of additive white noise. It follows that in the whitened space the Bayes optimal classifier is equivalent to the distance classifier and the correlation classifier [1].

Irrelevant variability and the Whitening transformation Getting back to our model as stated in the introduction, we assume that the observable data depends not only on the class label, but also on the environmental conditions and sensor characteristics. Formally stated, we assume that for class m the distribution of the observed measurements is:

$$\mathbf{X}_{Observable} = \mathbf{C}_m + \mathbf{G} \tag{2}$$

In general the random variable $\mathbf{G}$ denotes the additive contribution of features due to the global environment and sensor characteristics. We assume that:

- $\mathbf{G}$ is additive with 0 mean and non-isotropic covariance.
- $\mathbf{G}$ is independent of C_m $\forall m$, and has an identical effect on all classes.
- The variability in $\mathbf{G}$ is relatively large compared to $\mathbf{C}_m$, i.e $|\Sigma_G| > |\Sigma_m|$.

More specifically, assume that $\mathbf{G}$ is distributed normally as $\mathbf{G} \sim N(0, \Sigma_G)$ where $\Sigma_G \neq I$. The distribution of the m-th class is therefore normal, with mean μ_m and covariance $\Sigma_m + \Sigma_G$. Under the assumption that $|\Sigma_G| > |\Sigma_m| \;\; \forall m$, the class distributions are dominated by $|\Sigma_G|$. This effectively brings us back to the previous case where $\Sigma_m \approx \Sigma_G \;\; \forall m$, and therefore

$$S_W \approx \hat{\Sigma}_G$$

3.2 RCA: chunklet approximation of the *rescaling* transformation

Without access to labeled data, we cannot compute S_W directly. In this section we discuss how we can approximate S_W using chunklets. Intuitively, this is done as follows: Each chunklet is first shifted so that its mean (centroid) coincides with the origin. The shifted chunklets are then superimposed, defining a distribution of points around the origin whose covariance matrix approximates S_W.

Specifically, let $\mathbf{H}_n$ denote the sample of the nth chunklet where $\bigcup \mathbf{H}_n = \Omega$. We assume that the number of chunklets N is much larger than the number of classes M. We further assume that $\forall n \; \mathbf{H}_n \subseteq \mathbf{C}_m$ for some unknown label m. Let $\hat{\mu}_n$ denote the mean of chunklet H_n, $\hat{\mu}_n = \frac{1}{|H_n|}\sum_{\boldsymbol{x}\in H_n} \boldsymbol{x}$, where $|H_n|$ is the size of the n'th chunklet. We define the chunklet scatter matrix S_{ch} as:

$$S_{ch} = \frac{1}{|\Omega|}\sum_{n=1}^{N} |H_n| Cov(H_n) = \frac{1}{|\Omega|}\sum_{n=1}^{N}\sum_{j=1}^{|H_n|} \left(\boldsymbol{x}_n^j - \hat{\mu}_n\right)\left(\boldsymbol{x}_n^j - \hat{\mu}_n\right)^T \qquad (3)$$

Under the ideal conditions where each chunklet is chosen randomly from the appropriate class and is large enough, it can be shown that $Cov(H_n)$ approaches $\Sigma_{m_n} + \Sigma_G$, where m_n is the class label of chunklet H_n. In this case $S_{ch} = S_W$. In reality, however, data points in the same chunklet tend to have stochastic dependency, and therefore cannot be assumed to be independently sampled from the data with distribution identical to the corresponding class distribution. Thus in general $S_{ch} \neq S_W$. In Section 3.4 we will show that even when this is the case, S_{ch} can still provide a good estimate of S_W, regardless of the size of the chunklets used.

3.3 The RCA Algorithm

Our RCA algorithm is defined as follows:

1. Compute $S_{ch} = \frac{1}{|\Omega|}\sum_{n=1}^{N} |H_n| Cov(H_n)$ as defined above. Let r denote its effective rank (the number of singular values of S_{ch} which are significantly larger than 0).
2. Compute the total covariance (scatter) matrix of the original data S_T, and project the data using PCA to its r largest dimensions.
3. Project S_{ch} onto the reduced dimensional space, and compute the corresponding whitening transformation W.
4. Apply W to the original data (in the reduced space).

Step 2 is meant to insure that while the whitening transformation rescales the variability in all directions so as to equalize them, it will not rescale directions whose corresponding eigenvalues are effectively zero. We note that in step 3 above we assume that the rank of S_{ch} remains r after projection. This may not always be the case: the rank of of S_{ch} may become smaller than r after projection.

Our algorithm is based on the use of S_{ch} as an approximation to S_W. This approximation may be used in ways other than whitening. We are currently investigating its use in approximating the FDA technique, which will be presented in Section 4.2.

3.4 Chunklet analysis: what makes a chunklet good or bad

In the discussion above chunklets are used to approximate the scatter matrix S_W, which depends solely on the class means. It therefore follows that the difference between S_W and S_{ch} depends on the difference between the class mean $\hat{\mu}_{m_n}$ and the chunklet's mean $\hat{\mu}_n$. More specifically, it follows from (1) and (3) that

$$S_W - S_{ch} = \frac{1}{|\Omega|} \sum_{n=1}^{N} \sum_{j=1}^{|H_n|} (\hat{\mu}_n - \hat{\mu}_{m_n})(\hat{\mu}_n - \hat{\mu}_{m_n})^T \qquad (4)$$

Hence a "good" chunklet is a chunklet which approximates the mean value of a class well, regardless of the chunklet's size. However, size matters because as the size of the chunklet increases, the likelihood that its mean approximates correctly the class mean also increases.

4 Experimental results: face recognition

We compare the performance of three linear classifiers for face recognition: the Eigenface method [5], the Fisherface method [4], and the RCA method proposed here. The task is to classify facial images with respect to the person photographed. We analyze two paradigms:

1. A retrieval paradigm using nearest neighbor classification, in which a query image is classified using its nearest neighbors or its k-nearest neighbors in the dataset.
2. Classification using clustering, which extracts structure from the data. Clustering results can then be used for exemplar-based retrieval.

4.1 The test data: yaleA database

We used the yaleA database [4], which includes 155 facial images of 15 subjects under varying lighting conditions and different facial expressions. Fig. 2 shows ten images of one subject. As noted in the introduction, these images vary greatly from one another due to these changes [10].

4.2 Methods

We represent each image as a vector in a high-dimensional pixel space (R^{32k}) as in [4]. All three methods compute a global linear transformation of the data space. We shall now describe the two additional methods used in our comparative study.

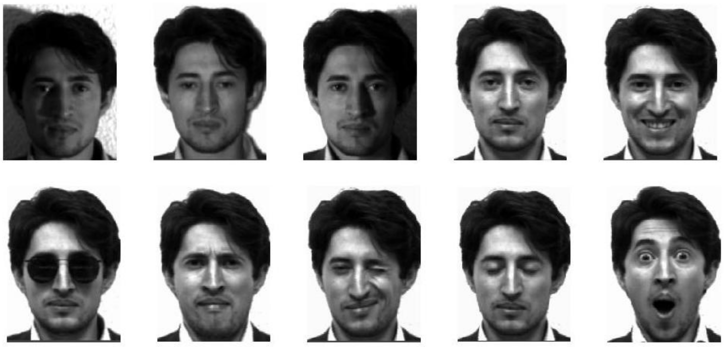

Fig. 2. The YaleA database contains 155 frontal face images of fifteen individuals (males and females) taken under different lighting conditions. Five images were taken under different point light sources, one with or without glasses, and about four with different facial expressions. The number of images per subject varied between 9-11. The images were manually centered and cropped so as to include the full face and part of the background.

Eigenfaces: The Eigenface method is based on PCA. In PCA the optimal projection transformation is chosen to maximize the determinant of the total scatter matrix of the projected samples. In [4] the method was augmented by throwing out the first three principal components in order to reduce variation due to lighting. In agreement we found that throwing out the three largest principal components yielded better results, and we therefore adopted this strategy as well.

Fisherfaces: The Fisherface method proposed in [4] is a variant of Fisher's Linear Discriminant (FLD) [1]. It is a supervised method which searches for projection directions that "shape" the scatter in order to improve classification results. This method selects the projection matrix W that maximizes the ratio of the between-class scatter and the within-class scatter.

The drawback of the eigenface method is that it maximizes the total scatter of the data. The total scatter depends not only on between-class scatter, which is useful for classification, but also on the within-class scatter. In the task of classification, the within-class scatter represents the irrelevant variability that we seek to eliminate. This is essentially the problem that the Fisherface method tries to overcome using labeled data.

Data preprocessing: Each of the three methods require different preprocessing steps. For the Eigenface method, the images were normalized to have zero mean and unit variance, as this improved performance. The Fisherface method requires dimensionality reduction of the data using PCA. The RCA method also reduces the dimensionality of the data with respect to the effective rank of the chunklet scatter matrix S_{ch}.

Our RCA algorithm requires the use of chunklets. Although, as claimed above, chunklets may be obtained from the data automatically in some applications, when using the yaleA database chunklets were chosen randomly. We tested our method using different chunklet sizes ($s = 3, 4, 5, 10$). For each chunklet size s, 1000 realizations were sampled. We tried two schemes: First, in each realization we randomly divided each class into chunklets of approximately s data points. In Section 4.5 we will address the effect of chunklet size on performance. Second, in each realization we randomly chose 10-20 chunklets of size s from all classes, and left the remaining data points unassigned to chunklets. This scheme corresponds to a scenario in which only part of the data set is given in chunklets. We will address the effects of such limited assignments in Section 4.4.

4.3 Results with nearest neighbor classification

Scheme *A*. **All data points are assigned to chunklets:** We first provide the results of a retrieval task using the "leave one out" paradigm [8]: To classify an image of a particular person, that image is removed from the data set and consequently the transformation matrices of the three methods are computed. Recognition by exemplar is then performed using nearest neighbor classification.

In the Eigenface method, performance varied with the number of principal components[1]. The left plot in Fig. 3 shows the relative performance of the three methods using nearest neighbor classification, each using its optimal number of principal components. Both the RCA and Fisherface methods outperform the Eigenface method. While the Fisherface method achieves such performance using a label for each of the images in the training set, RCA achieves similar error rates relying on chunklets with unknown class label.

Another interesting measure of performance is the total number of errors on all neighbors. Using the "leave one out" strategy once more, we compute the k-nearest neighbors of the image x_i, where $k = |C_i| - 1$ ($|C_i|$ is the size of the class that the image x_i belongs to). We then compute the number of neighbors which were not retrieved correctly. The right plot in Fig. 3 shows the mean error rate on all neighbors using all three methods. Once more, both RCA and Fisherface achieve almost similar results.

Scheme *B*. **Only part of the data points are assigned to chunklets:** Fig. 4 shows the same classification measures as in Fig. 3, but with varying number of chunklets (see figure caption); results are compared with the assigment of all data points to chunklets, and with Eigenfaces. Clearly RCA always outperforms the Eigenface method, but degradation from the fully assigned case is evident.

4.4 Clustering Results

Our goal here is to study how much each pre-processing (via the corresponding rescaling transformation) helps in revealing the class' structure of the data. To

[1] This is also true for the Fisherface and RCA methods, but in both the optimal number of principal components can be determined automatically from the data.

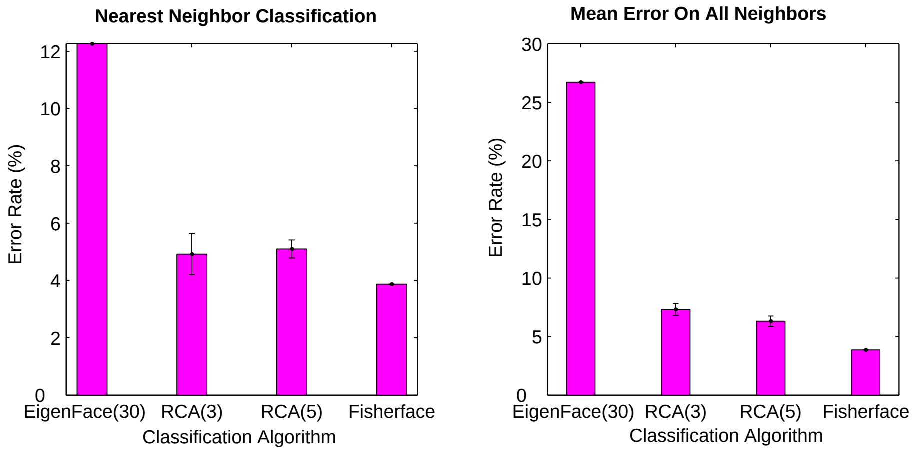

Fig. 3. Classification Error rates of the three methods. For Eigenface we used the 30 largest eigen-vectors. For RCA we show two results, depending on the size of the randomly sampled chunklet: 3 (middle left bar) or 5 (middle right bar). In both cases **all** data points are assigned to chunklets. Error bars reflect the variability in the results due to the sampling of chunklets (1000 realizations). On the left results of nearest neighbor classification are shown. On the right the mean error rate on all neighbors is shown. As can be seen RCA has error rates that approximate the errors achieved by the Fisherface method.

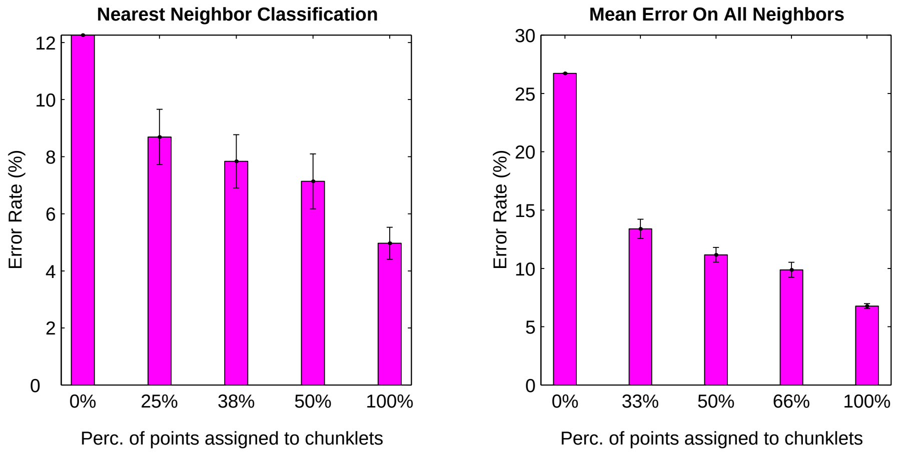

Fig. 4. The same classification errors as in Fig 3, for variable numbers of chunklets of size 4 (left figure) and of size 5 (right figure). Labels display the percentage of points in chunklets, *e.g.*, the 25% bar in the left figure corresponds to 10 chunklets of size 4. Assignment of 0% corresponds to Eigenfaces.

this end we used two clustering algorithms to cluster the data after it had been transformed by each of the three methods. The clustering algorithms used were

k-means [8] and the typical-cut algorithm [7]. Table 5 shows the results of both algorithms. We tested the effect of full and partial assignment of data points to chunklets. We also tested the effect of incorporating chunklet information as constraints to the clustering algorithms themselves. This was done by forcing data points which belong to a chunklet to be assigned to the same cluster[2].

Clustering results are displayed using two common measures: purity and accuracy (efficiency). Purity measures the frequency whereby data points that belong to the same cluster share the same class label, while accuracy measures the frequency whereby all data points from the same class label reside in one cluster.

Table 5 clearly shows that both clustering methods (RCA and Fisherface) achieve comparable clustering results. Eigenfaces also improves when applying 'chunklet' constraints. We also note that applying chunklet constraints to RCA does not improve results; this is probably due to the fact that after RCA the constraints are typically satisfied.

	K-Means		K-Means constrained		Typical-Cut		Typical-Cut constrained	
Algorithm	Purity	Accuracy	Purity	Accuracy	Purity	Accuracy	Purity	Accuracy
Eigenface(30)	$73 \pm 28\%$	$82 \pm 13\%$	$74 \pm 28\%$	$90 \pm 12\%$	100%	84%	$99 \pm 1\%$	$92 \pm 2\%$
RCA (50%)	$79 \pm 26\%$	$92 \pm 12\%$	$79 \pm 26\%$	$93 \pm 10\%$	$99 \pm 1\%$	$96 \pm 2\%$	$99 \pm 1\%$	$96 \pm 2\%$
RCA (100%)	$81 \pm 26\%$	$95 \pm 12\%$	$82 \pm 26\%$	$96 \pm 10\%$	100%	100%	100%	100%
Fisherfaces	$77 \pm 28\%$	$96 \pm 11\%$	$80 \pm 27\%$	$97 \pm 10\%$	100%	100%	100%	100%

Fig. 5. Results of the two clustering methods (with and without chunklet constraints) used on the data after applying each of the three transformations under investigation. RCA was based on chunklets of size 5, either assigning 50% or 100% of the points to chunklets. In constraining the clustering algorithms we used the chunklets of RCA-50%. Results are averaged over 1000 chunklet realizations (and over 1000 random initial centroid configurations in the case of k-means).

4.5 The effect of different Chunklet sizes

We tested the RCA method on the yaleA database using different chunklet sizes (when all data points were assigned to chunklets). In Section 3.4 we claimed that a "good" chunklet is defined by how well its mean approximates the mean of the class to which it belongs. However, it is clear that as the chunklet size increases, its probability to become "good" increases. In the limit, when the chunklet size equals the class size, $S_{ch} = S_W$. Fig. 6 shows a plot of the mean

[2] For the typical-cut algorithm we assigned an infinite value to edges connecting chunklet points. For k-means we replaced each data point which belong to a chunklet by the chunklet's mean.

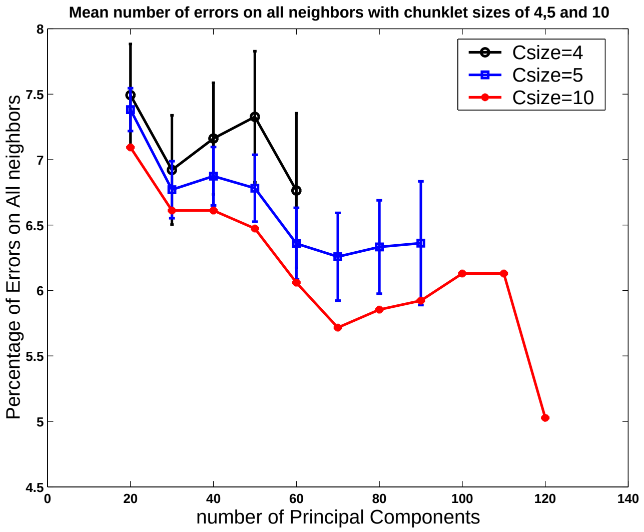

Fig. 6. Comparison of mean error on all neighbors with varying chunklet sizes (all data points were assigned to chunklets): error rates vary only slightly for different chunklet sizes.

error on all neighbors with respect to the number of principal components used, for different chunklets sizes. The plot clearly shows that changing the chunklet size also changes the effective rank of the chunklet distribution and of S_{ch}. As before the results are an average of 1000 realizations using the same chunklet sizes.

5 Experimental results: surveillance

We tested our approach on a second application where one of the main challenges was to achieve color constancy. In this application the data was taken by a stationary indoor surveillance camera, and was originally divided into short video clips. The beginning and end of each clip were automatically determined by the appearance and disappearance of a moving target. The database included many clips each displaying only one person; however, the identity of the person was unknown. The system's task was to cluster together all clips in which a certain person appeared. The application is briefly described in the following section.

5.1 The task and our approach

The characteristics of the video clips are highly complex and diversified, for several reasons. First, clips are entirely unconstrained; a person may walk all over the scene, coming closer to the camera or walking away from it. Therefore the size and resolution of each image vary dramatically. In addition, since the

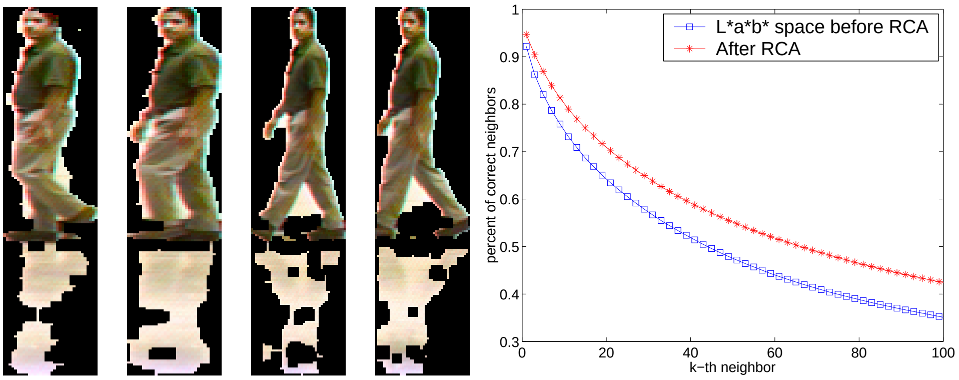

Fig. 7. Left: several images from a video clip of one intruder. Right: percent of correct retrievals as a function of the number of retrieved images.

environment was not controlled, images include varying occlusions, reflections and (most importantly from our perspective) highly variable illumination. In fact, illumination changed dramatically across the scene both in intensity (from brighter to darker regions), and in spectrum (from neon light to natural lighting). Fig. 7 displays several images from one input clip.

Our goal was to devise a representation that would enable effective clustering of clips. We found that the only low-level attribute that could be reliably used in this application was color. Therefore our task was to accomplish some sort of color constancy, i.e., to overcome the irrelevant variability created by the changing illumination.

5.2 Image representation and the application of RCA

Each image in a clip is represented using its color histogram in $L^*a^*b^*$ space (we used 5 bins for each dimension). Since a clip forms a chunklet by definition, we can naturally apply RCA without further preprocessing of the data. This transformation creates a new representation of each image.

In order to quantify the effect of RCA, we compute the *L1* distance between image representations both in the original space, and after the transformation. Using this distance we sorted the neighbors of each image, before and after applying the transformation. We used over 6000 images from 130 clips (chunklets) of 20 different people. Fig. 7 shows the percentage of 'correct' neighbors up to the k'th neighbor over all 6000 images. One can see that RCA makes a significant contribution by bringing 'correct' neighbors closer to each other (relative to other images).

6 Discussion

In the face recognition task, the difference in the results between the Eigenface and Fisherface methods is not surprising, given that the first method is unsupervised and the second is supervised. What is interesting is that the RCA method performed almost as well as the supervised Fisherface method, even though it only used slightly more prior information than the unsupervised Eigenface method. Improved performance was also demonstrated in the surveillance application, where the distance in the transformed space (via RCA) was significantly better than the distance in the original space. This gives us reason to hope that in applications where chunklets can be obtained automatically, *adjustment learning* will improve the results of unsupervised learning significantly.

Acknowledgments We would like to thank Aharon Bar-Hillel for many fruitful discussions. DW was supported in part by BSF grant prop. no. 1999037 to the Hebrew University. MP was supported in part by the NCC 2-1218 grant from NASA to the Oregon Health and Science University.

References

1. K. Fukunaga, *Introduction to statistical pattern recognition.* Academic Press, Boston, 1990.
2. M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications.* Kluwer Academic Press, 1984.
3. H. Murase and S. K. Nayar, Visual learning and recognition of 3-D objects from appearance. *IJCV*, 1995.
4. P.N Belhumeur, J. Hespanha, and D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE PAMI*, 19(7):711–720, 1997.
5. M.A. Turk and A.P Pentland, Face recognition using eigenfaces. In *Proc. IEEE CVPR*, pages 586–591, 1991.
6. P.Y. Simard, Y.A. LeCun, J.S. Denker and B. Victorri, Transformation Invariance in Pattern Recognition - Tangent Distance and Tangent Propagation. *IJIST*, 11(3), 2000.
7. Y. Gdalyahu, D. Weinshall, and M. Werman, Self organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE PAMI*, 23(10):1053–1074, 2001.
8. R. Duda, P. Hart and D.G. Stork, Pattern Classification, Wiley, New York, 2001.
9. E. Sali and S. Ullman, Recognizing novel 3-D objects under new illumination and viewing position using a small number of example views or even a single view. In *Proc. IEEE ICCV*, 1998.
10. Y. Adini, Y. Moses, and S. Ullman, Face recognition: The problem of compensating for changes in illumination direction. *IEEE PAMI*, 19(7):721–732, 1997.
11. S. Haykin, Neural Networks - a comprehensive foundation, Prentice Hall, New Jersey, 1991.