ABSTRACT

Title of dissertation: DISCRIMINATIVE LEARNING AND RECOGNITION USING DICTIONARIES

Yi-Chen Chen, Doctor of Philosophy, 2013

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

In recent years, the theory of sparse representation has emerged as a powerful tool for efficient processing of data in non-traditional ways. This is mainly due to the fact that most signals and images of interest tend to be sparse or compressible in some dictionary. In other words, they can be well approximated by a linear combination of a few elements (also known as atoms) of a dictionary. This dictionary can either be an analytic dictionary composed of wavelets or Fourier basis or it can be directly trained from data. It has been observed that dictionaries learned directly from data provide better representation and hence can improve the performance of many practical applications such as restoration and classification.

In this dissertation, we study dictionary learning and recognition under supervised, unsupervised, and semi-supervised settings. In the supervised case, we propose an approach to recognize humans in unconstrained videos, where the main challenge is exploiting the identity information in multiple frames and the accompanying dynamic signature. These identity cues include face, body, and motion.

Our approach is based on video-dictionaries for face and body. We design video-dictionaries to implicitly encode temporal, pose, and illumination information. Next, we propose a novel multivariate sparse representation method that jointly represents all the video data by a sparse linear combination of training data. To increase the ability of our algorithm to learn nonlinearities, we apply kernel methods to learn the dictionaries. Next, we address the problem of matching faces across changes in pose in unconstrained videos. Our approach consists of two methods based on 3D rotation and sparse representation that compensate for changes in pose. We demonstrate the superior performance of our approach over several state-of-the-art algorithms through extensive experiments on unconstrained video datasets.

In the unsupervised case, we present an approach that simultaneously clusters images and learns dictionaries from the clusters. The method learns dictionaries in the Radon transform domain. The main feature of the proposed approach is that it provides in-plane rotation and scale invariant clustering, which is useful in many applications such as Content Based Image Retrieval (CBIR). We demonstrate through experiments that the proposed rotation and scale invariant clustering provides not only good retrieval performances but also substantial improvements and robustness compared to traditional Gabor-based and several state-of-the-art shape-based methods.

We then extend the dictionary learning problem to a generalized semi-supervised formulation, where each training sample is provided with a set of possible labels and only one label among them is the true one. Such applications can be found in image and video collections where one often has only partially labeled data. For instance,

given an image with multiple faces and a caption specifying the names, we can be sure that each of the faces belong to one of the names specified, while the exact identity of each face is not known. Labeling involves significant amount of human effort and is expensive. This has motivated researchers to develop learning algorithms from partially labeled training data. In this work, we develop dictionary learning algorithms that utilize such partially labeled data. The proposed method aims to solve the problem of ambiguously labeled multiclass-classification using an iterative algorithm. The dictionaries are updated using either soft (EM-based) or hard decision rules. Extensive evaluations on existing datasets demonstrate that the proposed method performs significantly better than state-of-the-art approaches for learning from ambiguously labeled data.

As sparsity plays a major role in our research, we further present a sparse representation-based approach to find the salient views of 3D objects. The salient views are categorized into two groups. The first are *boundary representative views* that have several visible sides and object surfaces that may be attractive to humans. The second are *side representative views* that best represent side views of the approximating convex shape. The side representative views are class-specific views and possess the most representative power compared to other within-class views. Using the concept of *characteristic view* class, we first present a sparse representation-based approach for estimating the boundary representative views. With the estimated boundaries, we determine the side representative views based on a minimum reconstruction error criterion. Furthermore, to evaluate our method, we introduce the notion of geometric dictionaries built from salient views for ap-

plications in 3D object recognition, retrieval and sparse-to-full reconstruction. By a series of experiments on four publicly available 3D object datasets, we demonstrate the effectiveness of our approach over state-of-the-art algorithms and baseline methods.

DISCRIMINATIVE LEARNING AND RECOGNITION USING DICTIONARIES

by

Yi-Chen Chen

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry Davis
Professor Min Wu
Dr. P. Jonathon Phillips
Professor John J. Benedetto, Dean's Representative

© Copyright by
Yi-Chen Chen
2013

Dedication

To my parents, Shou-Chih Chen and Fang-Mei Lu

Acknowledgments

I sincerely thank my advisor, Professor Rama Chellappa, for giving me a chance to work for him two and half years ago, for his advices guiding me toward timely research topics and the completion of my dissertation, and for his patience and encouragement, which made me confident in myself and in my heart.

I sincerely thank Dr. Vishal Patel and Dr. Jonathon Phillips, for their great support and precious advices, which helped me be on the right track for my research. I also sincerely thank Professor Larry Davis, Professor Min Wu and Professor John Benedetto, for being committee members of my dissertation defense, and providing valuable comments and suggestions on my presentation.

I sincerely thank Dr. Tracy Chung, Melanie Prange, Melanie Lynn, and Vivian Lu, for giving me teaching assistantship opportunities, English conversation training, and helpful advices on academic regulations and guidelines.

I sincerely thank sweet Akikawa-sensei, Seya-sensei, Inoue-sensei and Yamakita-sensei, for patiently teaching me Japanese and Japanese culture, from which I understood the meaning of showing people politeness and modesty. I also sincerely thank my class instructors, seniors, lab mates, colleagues, mentors and friends, who have definitely been resourceful consultants on many aspects of my life.

Finally, I sincerely thank my parents, my sister and my brother-in-law, for their unceasing concern, care and thoughtfulness. Without their support, I can never finish my study here at the University of Maryland.

August 2013

Table of Contents

List of Figures

Chapter 1: Introduction

In this dissertation, we propose methods for dictionary learning and recognition for three cases: 1. supervised, 2. unsupervised, and 3. semi-supervised. In the first case, we study the video-based person recognition problem using dictionaries, where video-dictionaries are learned from labeled training video sequences and evaluated on testing videos for identification and verification. In the second case, we study rotation and scale-invariant, simultaneous dictionary learning and clustering from unlabeled training images. The learned dictionaries and clusters are used for content based image retrieval. In the third case, we extend the dictionary learning approach to address the ambiguously labeled problem, where each training sample is provided with a set of possible labels and only one label among them is the true one. Such applications can be found in image and video collections where one often has only partially labeled data.

Based on the concept of characteristic view class, we further present a sparse representation-based approach to find the salient views of 3-dimensional (3D) objects. We categorize salient views into two types: *boundary representative views* (BRVs) that have several visible sides and object surfaces that may be attractive to human perceivers, and *side representative views* (SRVs) that best represent views

from each side of a object. Based on the salient views, we build geometric dictionaries for 3D object recognition and evaluate their sparse-to-full reconstruction power.

In this chapter we briefly describe these topics.

1.1 Dictionary-based Person Recognition from Unconstrained Video

Face recognition research has traditionally concentrated on recognition from still images [6], [7], [8], [9]. With inclusion of video cameras in mobile devices, face recognition from video is gaining attention. In unconstrained videos, recognition purely from face uses only partial information, as in reality, recognition of people in video requires fusing identity-cues from the face and body, as well as their motion [2].

While the advantage of using motion information in face videos has been widely recognized, computational models for video-based face recognition have only recently received attention [10], [6]. In video-based face and person recognition, a key challenge is in exploiting all the available identity-cues in video. In addition, different video sequences of the same subject may contain variations in resolution, illumination, pose, and facial expressions. These variations contribute to the challenges in designing an effective video-based face recognition algorithm.

To address the challenges in recognizing people from unconstrained videos, we present a generative approach based on dictionary learning methods. From cropped images of faces, bodies, or motion identity cues extracted from a video sequence, we first partition the video sequence so that images with the same or close pose and

illumination conditions are in one partition. This step removes the temporal redundancy while capturing variations due to changes in pose and illumination. For each partition, a sub-dictionary is learned where the representation error is minimized under a sparseness constraint. These sub-dictionaries are combined to form a video dictionary. In the recognition phase, images of faces, bodies, or motion identity cues from a given query video sequence are projected onto the span of atoms in every video dictionary. From the projection onto the atoms, the residuals are computed and combined for recognition. Using video dictionaries, we next propose a joint sparsity-based approach, which simultaneously takes into account correlations as well as coupling information between frames of a video while enforcing joint sparsity within each frame's observation. We kernelize the dictionary learning algorithm to handle the non-linearities present in the video data. In addition, we combine the face features with the upper body features or motion identity cues, to improve the recognition accuracy. We demonstrate the effectiveness of the proposed dictionary approach through comparisons with other recently proposed state-of-the-art methods, and with human performance.

1.2 Adaptive Representations for Video-based Face Recognition Across Pose

Though significant efforts have gone into understanding the different sources of variations affecting facial appearance, the accuracy of video-based face recognition algorithms in completely uncontrolled scenarios is still far from satisfactory. Pose

and illumination variations still remain one of the biggest challenges. Some of the existing methods [11], [1], [12], [13], [14], [3], rely on the pose diversity contained in the gallery videos to handle pose variations. When there are pose differences between the videos, the robustness of these methods is limited.

We hence consider matching faces across different poses between the probe and the gallery videos. We propose two methods that compute pose aligned features based on 3D rotation and sparse representation to compensate for changes in pose. The first method is referred to as Sparse Representation-based Alignment (SRA) method. This method generalizes traditional methods to the use of novel face datasets. These datasets function as reference sets for pose alignment, and are independent of the gallery and probe sets specified by the protocol. The second method is an adaptation of the SRA method that rotates the video dictionary atoms to align the pose prior to recognition. It is referred to as the Dictionary Rotation (DR) method.

1.3 In-plane Rotation and Scale Invariant Clustering using Dictionaries

Dictionary learning techniques for unsupervised clustering have also gained some traction in recent years. In [15], a method for simultaneously learning a set of dictionaries that optimally represent each cluster is proposed. To improve the accuracy of sparse coding, this approach was later extended by adding a block incoherence term in their optimization problem [16]. Some of the other sparsity

motivated subspace clustering methods include [17], [18], [19].

Invariance to rotation and scale are desirable in many practical applications such as image classification and retrieval where one wants to classify or retrieve images having the same content but different orientation and scale. For instance, in content based image retrieval (CBIR), images are retrieved from a database using features that best describe the orientation and scale of objects in the query image.

We present an in-plane rotation and scale invariant clustering approach, extending the dictionary learning and sparse representation framework for clustering and retrieval of images. Our method uses Radon transformation to find scale and rotation invariant features. It then uses sparse representation methods to simultaneously cluster the data and learn dictionaries for each cluster. One of the main features of our method is that it is effective for both texture and shape-based features. We demonstrate through experimental results that the proposed rotation and scale invariant clustering provides not only good retrieval performances but also substantial improvements and robustness compared to standard Gabor-based and several state-of-the-art shape-based methods.

1.4 Dictionary Learning from Ambiguously Labeled Data

In image and video collections, one often has only partially labeled data. For instance, given an image with multiple faces and a caption specifying the names, we can be sure that each of the faces belong to one of the names specified. But the exact identity of each face is not known. Labeling involves significant amount of

human effort and is time consuming and expensive. This has motivated researchers to develop learning algorithms from partially labeled training data.

While dictionary learning techniques for unsupervised clustering [15], [16], [20] have recently demonstrated tremendous success in tackling image understanding problems, their performance is often limited by the amount of labeled data available for training. We therefore consider a dictionary learning problem where each training sample is provided with a set of possible labels and only one label among them is the true one. We develop dictionary learning algorithms that process ambiguously labeled data. In particular, the dictionary learning problem is solved using an iterative alternating algorithm. At each iteration of the algorithm, two alternating steps are performed: a confidence update and a dictionary update. The confidence of each sample is defined as the probability distribution on its ambiguous labels. The dictionaries are updated using either soft (EM-based) or hard decision rules.

1.5 Salient Views and Geometric Dictionaries for Object Recognition

The selection of salient views of 3D objects has drawn researcher's interest for several years. There are a number of approaches for describing what is contained in a view [21], [22]. For view-based representations, human perceivers are influenced by factors such as familiarity with the object being viewed, the similarity of a given view to known views of visually-similar objects and the pose of the object [21]. Three-quarter views with all visible front, top and side, are often used as candidate views. As noted in [23], three-quarter views are essentially the views that most humans

prefer when looking at an object. These views are also known as the *canonical views* [22].

We propose a sparse representation-based approach to select the salient views of an object [22], [23]. Using the concept of *characteristic view* class, we present a sparse representation-based approach for estimating the boundary representative views (BRVs). With the estimated boundaries, we determine the side representative views (SRVs) based on minimum reconstruction error. To evaluate our method, we introduce the notion of geometric dictionaries that are built from the SVs and SVCs for 3D object recognition and retrieval. We demonstrate the effectiveness of our approach over two existing state-of-the-art algorithms and baseline methods through the performances on object recognition, retrieval and sparse-to-full reconstruction.

1.6 Contributions

Contributions of this dissertation are summarized as follows:

- Video-based person recognition

    1. We introduce video-dictionaries for video-to-video face recognition. Through video partitioning, the learned dictionaries implicitly encode face pose and illumination information [3].

    2. We propose a multivariate sparse representation method that simultaneously takes into account correlations, as well as coupling information among the video frames [24].

3. The dictionary learning algorithms in 1. and 2. are kernelized to handle non-linearities in the data samples [25], [24].

4. The video dictionaries are further extended to encode the upper body features and motion identity cues. The face features are combined with the upper body features or motion identity cues, to enhance the recognition accuracy [25].

5. To match faces across changes in pose from unconstrained videos, we propose two methods based on 3D rotation and sparse representation that compensate for changes in pose [26].

6. We demonstrate the effectiveness of our approaches over several state-of-the-art algorithms on unconstrained video datasets [3], [25], [24], [26].

- Unsupervised clustering and retrieval

7. We propose a rotation invariant clustering algorithm suitable for applications such as content based image retrieval (CBIR) [20].

8. We propose a normalization method validated by a mathematical proof, to achieve scale invariance in the Radon domain [27].

9. We propose a method to obtain initial classes and class dictionaries in a deterministic way to improve the clustering performance [27].

10. We demonstrate by experiments on shape-based and texture-based datasets the effectiveness of the proposed method for CBIR applications, and performance improvements compared to other Gabor-based and shape-based

methods [20], [27].

- Learning from ambiguously labeled data

  11. We extend dictionary learning to the case of ambiguously labeled learning, a general kind of semi-supervised learning where each example is supplied with multiple labels, only one of which is correct [28].

  12. We present two effective approaches for updating the dictionary: dictionary learning with hard decision (DLHD) and dictionary learning with soft decision (DLSD) [28].

  13. We show that our DLSD rule is an EM-based dictionary learning method. It is a weighted K-SVD algorithm to weigh the importance of samples according to their confidences during the learning process [28].

- Sparse view/geometry

  14. We propose a sparse representation-based approach for selecting the salient views of an object. Our method is based on characteristic views. It selects representative views of visible sides and object surfaces [29].

  15. We introduce the notion of geometric dictionaries based on the salient views for object recognition, retrieval [30] and sparse-to-full reconstruction.

  16. Through a series of experiments on four publicly available 3D object datasets, we demonstrate the effectiveness of our approach over two existing state-of-the-art algorithms and baseline methods [30].

1.7 Dissertation Organization

The rest of the dissertation is organized as follows. In chapter 2, we present our dictionary-based approach for person recognition from unconstrained video. In chapter 3, we present two methods to compute pose aligned features based on 3D rotation and sparse representation to match faces across different poses from video. In chapter 4, we present our in-plane rotation and scale invariant clustering approach using dictionaries. In chapter 5, we present our approach on dictionary learning from ambiguously labeled data. In chapter 6, we present our sparse representation-based approach for salient view selection, as well as geometric dictionaries for object recognition and sparse-to-full reconstruction. Finally, we conclude the dissertation in chapter 7.

Chapter 2: Dictionary-based Person Recognition from Unconstrained Video

Face recognition research has traditionally concentrated on recognition from still images [6], [7], [8], [9]. With inclusion of video cameras in mobile devices, face recognition from video is gaining attention. In unconstrained videos, recognition purely from face uses only partial information, as in reality, recognition of people in video requires fusing identity-cues from the face and body, and their motion [2].

While the advantage of using motion information in face videos has been widely recognized, computational models for video-based face recognition have only recently received attention [10], [6]. In video-based face and person recognition, a key challenge is in exploiting all the available identity-cues in video. In addition, different video sequences of the same subject may contain variations in resolution, illumination, pose, and facial expressions. These variations contribute to the challenges in designing an effective video-based face recognition algorithm.

Existing approaches to the problem include multi-still face recognition [31], extracting joint appearance and behavioral features from a video [32], or explicitly modeling the temporal correlations between faces in two videos [10]. A major drawback of frame-based fusion approaches is that they do not exploit the tem-

poral information present in video sequences [33]. It has been shown that for a generic video-face recognition algorithm, performance can be significantly improved by simultaneously performing recognition and tracking [32].

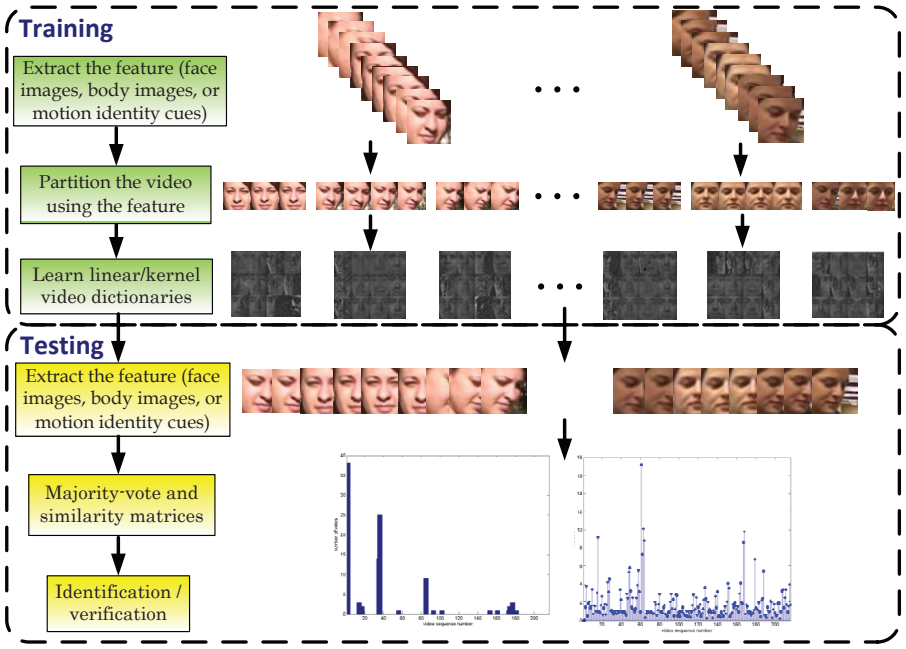


Figure 2.1: Overview of the proposed approach. The extracted feature can be face images, body images, or motion identity cues. For illustration purpose only, here we just show cropped face images as the feature.

Dictionaries have been observed to provide better representation and hence improve the performance on many practical applications such as restoration and classification [34], [35], [36], [8]. Dictionaries can be learned for both reconstruction and discrimination applications. In the late nineties, Etemad and Chellappa proposed a linear discriminant analysis (LDA) based basis selection and feature extraction algorithm for classification using wavelet packets [37], and Phillips proposed a dictionary method for face recognition [7]. Recently, algorithms for simultaneous

sparse signal representation and discrimination have been proposed [38], [39], [40], [41], [42], [43], [35], [44] and [45]. Additional techniques may be found within these references.

To address the challenges in recognizing people from unconstrained videos, we present a generative approach based on dictionary learning methods. This approach is robust to changes in illumination and pose. Fig. 2.1 shows an overview of this approach. While Fig. 2.1 illustrates our approach for faces, we apply the same method for recognition using the body. From cropped images of faces, bodies, or motion identity cues extracted from a video sequence, we first partition the video sequence so that images with the same or close pose and illumination conditions are in one partition. This step removes the temporal redundancy while capturing variations due to changes in pose and illumination. For each partition, a sub-dictionary is learned where the representation error is minimized under a sparseness constraint. These partition-specific sub-dictionaries are combined to form a sequence-specific dictionary (i.e. a video dictionary). In the recognition phase, images of faces, bodies, or motion identity cues from a given query video sequence are projected onto the span of atoms in every sequence-specific dictionary. From the projection onto the atoms, the residuals are computed and combined for recognition.

Motivated by the success of sparse representation and dictionary learning in biometrics recognition, we further propose a joint sparsity-based approach for unconstrained video-to-video face recognition. This method is based on a well known regularized regression method, multi-task multivariate Lasso [46], [47]. It simultaneously takes into account correlations as well as coupling information between frames

of a video while enforcing joint sparsity within each frame's observation. The proposed dictionary learning algorithms are then kernelized to handle the non-linearities present in video data.

There are number of approaches for fusing face and gait for person recognition [48], [49]. However, there is a substantial difference between the composition of gait videos and unconstrained videos. In the vast majority of gait videos, people are walking across the field of view and the complete body is visible [50]. In unconstrained videos, people can be performing any action and only a portion of a person's body could be visible. Most gait recognition algorithms base recognition on how a person walks, which does not generalize to unconstrained videos. Instead, our approach concentrates identity cues present in the body. It combines the face features with the upper body features or motion identity cues, to improve the recognition accuracy. We demonstrate the effectiveness of the proposed dictionary approach through comparisons with other recently proposed state-of-the-art methods, and with human performance on the challenging the Multiple Biometric Grand Challenge (MBGC) [51], [52], Face and Ocular Challenge Series (FOCS) [2], [53], Honda/UCSD [32], and UMD [54] datasets.

Key contributions of this work are:

1. We introduce video-dictionaries for video-to-video face recognition. Through video partitioning, the learned dictionaries implicitly encode face pose and illumination information.

2. We propose a multivariate sparse representation method that simultaneously

takes into account correlations, as well as coupling information among the video frames.

3. The dictionary learning algorithms in **1.** and **2.** are kernelized to handle non-linearities in the data samples.

4. The video dictionaries are further designed to encode the upper body features and motion identity cues. The face features are combined with the upper body features or motion identity cues, to enhance the recognition accuracy.

5. We demonstrate performance improvements over other state-of-the-art methods on several video datasets.

The rest of the chapter is organized as follows: In Section 2.1 we review some recent video-based face recognition methods. Section 2.2 describes the proposed dictionary-based video face recognition algorithm. Section 2.3 describes our joint sparsity-based method for video-to-video face recognition. In both Section 2.2 and Section 2.3, kernel methods for dictionary learning are presented as well. In Section 2.4, we present results on four challenging video datasets. Section 2.5 concludes the chapter with a summary.

2.1 Related work

In this section, we review some of the recent video-based face recognition methods. In video-based face recognition, given a test video of a moving face, the first step is to track a set of facial features across all the frames in the video.

From the tracked features, one can extract a few key frames that are used for matching. Significant work has been done on face tracking using two-dimensional (2D) appearance-based models [55], [56], [57]. The 2D approaches, however, do not account the three-dimensional (3D) configuration of the head, and are not robust to large changes in pose or viewpoint. To deal with this problem, several methods have been developed for 3D face tracking. Cascia *et al.* [58] proposed a cylindrical face model for face tracking. An extension of this work was proposed by Aggarwal *et al.* in [59] based on a particle filter for state estimation.

Temporal information in videos can be exploited for simultaneous tracking and recognition of faces without the need to perform these tasks in a sequential manner. One such method was proposed by Zhou *et al.* in [60]. Their tracking-and-recognition approach resolves uncertainties in tracking and recognition simultaneously in a unified probabilistic framework. Another method was proposed by Lee *et al.* [32], where a model of a subject is represented by a complex nonlinear appearance manifold. All frames in a video sequence are samples from an appearance manifold. To simplify the problem, the manifold is approximated by a collection of linear subspaces. Each subspace consists of nearby poses and is obtained by principle component analysis (PCA) of frames from training video sequences. If sufficient 3D view variations and illumination variations are available in the training set, this method can be robust to large changes in appearance.

In a related work, Arandjelovic and Cipolla [11] represent the appearance variations due to shape and illumination on faces by assuming that the shape-illumination manifold of all possible illuminations and poses is generic for faces.

This in turn implies that the shape-illumination manifold can be estimated using a set of subjects independent of the test set. It was shown that the effects of face shape and illumination can be learned using PCA from a small, unlabeled set of video sequences of faces acquired in randomly varying lighting conditions [5]. Given a novel sequence, the learned model is used to decompose the face appearance manifold into albedo and shape-illumination manifolds. Then a classification decision is made using robust likelihood estimation.

Recently, Turaga *et al.* [1] presented a statistical method for video-based face recognition, which uses subspace-based models and tools from Riemannian geometry of the Grassmann manifold. Intrinsic and extrinsic statistics are derived for designing maximum-likelihood classification rules. An image set classification method for the video-based face recognition problem was recently proposed by Hu *et al.* [13]. This method is based on a measure of between-set dissimilarity, which is the distance between sparse approximated nearest points of two image sets and is found by a scalable accelerated proximal gradient method for optimization.

2.2 Dictionary Video Algorithm

In this section, we present the details of our dictionary-based video face and person recognition algorithm. The details of our approach are described for face video dictionaries. The approach is exactly the same for learning dictionaries for bodies. We first describe how the video sequence is partitioned into sub-sequences in section 2.2.1, and how we build sequence-specific dictionaries in section 2.2.2.

Identification and verification are described in sections 2.2.3 and 2.2.4, respectively.

2.2.1 Video Sequence Partition

For each frame in a video sequence, we first detect and crop the face regions. We then partition all the cropped face images into K different partitions. We partition the cropped faces by a k-means clustering type of algorithm that is inspired by a video summarization algorithm [61]. Let $S = \{\mathbf{f}_1, ..., \mathbf{f}_n\}$ be the set of all n cropped faces from a video sequence. The following steps summarize our video sequence partition approach.

One major difference between our method and [61] is that the overall cost $J(S) \triangleq \alpha \times err(S) + (1 - \alpha) \times (D - div(S))$ used in [61], is now replaced with

$$M(S) \triangleq \frac{div(S)}{err(S)},$$

where $err(S)$, $div(S)$ and D are the square error, diversity and an upper bound of diversity of summary $S(\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_K)$, respectively [61], and $\mathbf{s}_i$'s are representatives. The terms $err(S)$ and $div(S)$ are *square error* and *diversity*, respectively [61]. They are defined as follows

$$err(S) \triangleq tr\left[\sum_{i=1}^{K}\sum_{\mathbf{s}\in S_i}(\mathbf{s} - \mathbf{s}_i)(\mathbf{s} - \mathbf{s}_i)^T\right],$$

and

$$div(S) \triangleq tr\left[\sum_{i=1}^{K}(\mathbf{s}_i - \bar{\mathbf{s}})(\mathbf{s}_i - \bar{\mathbf{s}})^T\right],$$

where $\bar{\mathbf{s}} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{s}_i$ and $tr(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$. The *diversity* represents the scatter of representatives to their mean, while the *square error* represents

Algorithm 1: Video sequence partition algorithm.

Initialization of sets:

$S = \{\mathbf{f}_1, ..., \mathbf{f}_n\}, I = \{1, 2, ..., n\}, T = \phi$.

Procedure:

1. Find $(i^*, j^*) = \underset{i,j \in I, i \neq j}{\operatorname{argmax}} \|\mathbf{f}_i - \mathbf{f}_j\|_2$.
2. Update of sets: $t_1 \leftarrow i^*, t_2 \leftarrow j^*, T \leftarrow T \bigcup \{t_1, t_2\}$, $I \leftarrow I \setminus \{i^*, j^*\}$.
3. Find $k^* = \underset{k \in I}{\operatorname{argmax}} \prod_{l=1}^{|T|} \|\mathbf{f}_{t_l} - \mathbf{f}_k\|_2$.
4. Update of sets: $t_{|T|+1} \leftarrow k^*, T \leftarrow T \bigcup \{t_{|T|+1}\}$, $I \leftarrow I \setminus \{k^*\}$.
5. Repeat steps 3 and 4 until $|T| = K$.
6. Given $\{\mathbf{f}_{t_1}, ..., \mathbf{f}_{t_K}\}$, use the nearest neighbor criterion to partition S into K partitions, denoted by $S(\mathbf{f}_{t_1}, ..., \mathbf{f}_{t_K}) = \bigcup_{i=1}^{K} S_i$. $S(\mathbf{f}_{t_1}, ..., \mathbf{f}_{t_K})$ is the initial partitions which are followed by N iterations of updating described in step 7 and 8.
7. Randomly select s_i from S_i, $i = 1, 2, ..., K$, as representatives. Find the corresponding nearest neighbor partitions which are denoted by $S(s_1, s_2, ...s_K)$, and calculate the corresponding score $M(S(s_1, s_2, ...s_K))$.
8. Repeat step 7, and keep updating for $\{\mathbf{s}_1^*, \mathbf{s}_2^*, ..., \mathbf{s}_K^*\}$ which gives the highest score M, until the number of repeating iterations for step 7 reaches N. In other words,

$$\{\mathbf{s}_1^*, \mathbf{s}_2^*, ..., \mathbf{s}_K^*\} = \underset{\mathbf{s}_i \in S_i, i=1,2,...,K, \text{ in } N \text{ iterations}}{\operatorname{argmax}} M(S(\mathbf{s}_1, \mathbf{s}_2, ...\mathbf{s}_K)).$$

Output:

K partitions, $S(\mathbf{s}_1^*, \mathbf{s}_2^*, ..., \mathbf{s}_K^*)$.

the total summation of partition-specific scatters, over the K partitions. The maximization of $M(S)$ is achieved through maximizing the *diversity* while minimizing the *square error*. Using this score, there is no need to set the weighting factor α [61], and the original cost minimization problem becomes an equivalent score maximization problem. The other major difference is that we initialize the partitions deterministically (steps 1 to 6 above). As seen in steps 1 and 3, these K initial representatives are chosen so they are separated as far apart as possible. The corresponding initial K partitions are then determined by the nearest neighbor criterion. Under the assumption that there exist K exemplars, each of the K initial partitions determined by finding the nearest neighbor among the K initial representatives contains exactly one exemplar. In Appendix A, we present a proof of this claim by contradiction. For all subsequent iterations steps (7 and 8), K distinct representatives are chosen always from the predetermined K initial partitions, and are used to calculate the associated score. As long as each of the K exemplars fall in a distinct initial partition, they can be found after sufficient number of iterations. The representatives that give the maximum $M(S)$ among, say N iterations, are recorded as exemplars. The corresponding final partitions are obtained by the nearest neighbor criterion.

2.2.2 Building Sequence-specific Dictionaries

By partitioning the original video sequence, we obtain K separate sequences each containing images with specific pose and/or lighting conditions. To remove the temporal redundancy while capturing variations due to changes in pose and

illumination, we construct a dictionary for each partition. A dictionary is learned with the minimum representation error under a sparseness constraint. Thus, there will be K sub-dictionaries built to represent a video sequence. Due to changes in pose and lighting in a video sequence, the number of face images in a partition will vary. For partitions with very few images, before building the corresponding dictionary, we augment the partition by introducing synthesized face images. This is done by creating horizontally, vertically or diagonally position shifted face images, or by in-plane rotated face images. We assume that each partition contains B images.

Let $\mathbf{G}^i_{j,k}$ be the augmented gallery matrix of the kth partition of the jth video sequence of subject i. In

$$\mathbf{G}^i_{j,k} = [\mathbf{g}^i_{j,k,1}\mathbf{g}^i_{j,k,2}, \cdots, \mathbf{g}^i_{j,k,B}] \in I\!R^{L\times B},$$

each column is a vectorized form of the corresponding cropped grayscale face image of size L. Given $\mathbf{G}^i_{j,k}$, a dictionary $\mathbf{D}^i_{j,k} \in \mathbb{R}^{L\times K_0}$ is learned such that the columns of $\mathbf{G}^i_{j,k}$ are best represented by linear combinations of K_0 atoms of $\mathbf{D}^i_{j,k}$. This can be done by solving the following optimization problem

$$(\hat{\mathbf{D}}^i_{j,k}, \hat{\mathbf{\Gamma}}^i_{j,k}) = \underset{\mathbf{D}^i_{j,k},\mathbf{\Gamma}^i_{j,k}}{\operatorname{argmin}} \|\mathbf{G}^i_{j,k} - \mathbf{D}^i_{j,k}\mathbf{\Gamma}^i_{j,k}\|^2_F,$$

$$\text{subject to } \|\boldsymbol{\gamma}_l\|_0 \le T_0, \ \forall l, \tag{2.1}$$

where $\boldsymbol{\gamma}_l$ is the lth column of the coefficient matrix $\mathbf{\Gamma}^i_{j,k}$ and T_0 is a sparsity parameter. The ℓ_0 sparsity measure $\|\cdot\|_0$ counts the number of nonzero elements in the representation and $\|\mathbf{G}\|_F$ is the Frobenius norm of the matrix $\mathbf{G}$ defined as $\|\mathbf{G}\|_F = \sqrt{\sum_i \sum_j |\mathbf{G}(i,j)|^2}$. Many approaches have been proposed in the literature

for solving such optimization problems. In our work, we adapt the K-SVD algorithm [62] for solving (2.1) due to its simplicity and fast convergence[1]. The K-SVD algorithm alternates between sparse-coding and dictionary update steps. In the sparse-coding step, the dictionary $\mathbf{D}^i_{j,k}$ is fixed and the representation vectors $\boldsymbol{\gamma}_l$ are found for each example $\mathbf{g}^i_{j,k,l}$. Then, the dictionary is updated atom-by-atom in an efficient way [62]. Please refer to Section 4.2.2.1 for more on the K-SVD principle.

The video-specific dictionary is constructed by concatenating partition-level sub-dictionaries. In other words, the jth dictionary of a subject i is

$$\mathbf{D}^i_j = [\mathbf{D}^i_{j,1}\ \mathbf{D}^i_{j,2}\ ...\ \mathbf{D}^i_{j,k}]. \tag{2.2}$$

2.2.3 Identification

Let Q denote the total number of query video sequences. Given the mth query video sequence $\mathbf{Q}^{(m)}$, where $m = 1, 2, ..., Q$, we can write $\mathbf{Q}^{(m)} = \bigcup_{k=1}^{K} \mathbf{Q}^{(m)}_k$. Partitions $\mathbf{Q}^{(m)}_k$ are expressed by $\mathbf{Q}^{(m)}_k = [\mathbf{q}^{(m)}_{k,1}\ \mathbf{q}^{(m)}_{k,2}\ ...\ \mathbf{q}^{(m)}_{k,n_k}]$, where $\mathbf{q}^{(m)}_{k,l}$ is the vectorized form of the lth of the total n_k cropped face images belonging to the kth partition. Assume that there are a total of P gallery video sequences. We can write the associated dictionaries $\mathbf{D}_{(p)}$ for $p = 1, 2, ..., P$, where each $\mathbf{D}_{(p)}$ corresponds to $\mathbf{D}^i_j$ for some subject i and its jth video sequence. Image $\mathbf{q}^{(m)}_{k,l}$ votes for sequence $\hat{p}$ with the minimum residual. In other words,

$$\hat{p} = \underset{p}{\operatorname{argmin}} \|\mathbf{q}^{(m)}_{k,l} - \mathbf{D}_{(p)}\mathbf{D}^{\dagger}_{(p)}\mathbf{q}^{(m)}_{k,l}\|_2, \tag{2.3}$$

[1]Here "K" in "K-SVD" equals number of atoms K_0 in a learned dictionary, not number of partitions K of a video sequence.

where $\mathbf{D}_{(p)}^{\dagger} = (\mathbf{D}_{(p)}^T \mathbf{D}_{(p)})^{-1} \mathbf{D}_{(p)}^T$ is the pseudoinverse of $\mathbf{D}_{(p)}$ and $\mathbf{D}_{(p)} \mathbf{D}_{(p)}^{\dagger} \mathbf{q}_{k,l}^{(m)}$ is the projection of $\mathbf{q}_{k,l}^{(m)}$ onto the span of atoms in $\mathbf{D}_{(p)}$.

To make the sequence-level decision, we select p^* such that

$$p^* = \underset{p}{\operatorname{argmax}} \left(\sum_{k=1}^{K} C_{p,k} \right), \tag{2.4}$$

where $C_{p,k}$ is the total number of votes from partition k for sequence p. Finally, using the knowledge of the correspondence $\boldsymbol{m}(\cdot)$ between subjects and sequences, we assign the query video sequence $\mathbf{Q}^{(m)}$ to subject $i^* = \boldsymbol{m}(p^*)$.

2.2.4 Verification

For verification, given a query video sequence and any gallery video sequence, the goal is to correctly determine whether these two belong to the same subject. The well-known receiver operating characteristic (ROC) curve, which describes the relationship between false acceptance rates (FARs) and true acceptance rates (TARs), is used to evaluate the performance of verification algorithms. As the TAR increases, so does the FAR. Therefore, one would expect an ideal verification framework to have TARs all equal to 1 for any FARs. The ROC curves can be computed given a similarity matrix. In the proposed dictionary-based method, the residual between a query $\mathbf{Q}^{(m)}$ and a dictionary $\mathbf{D}_{(p)}$, is used to fill in the (m, p) entry of the similarity matrix. Denoting the residual by $\mathbf{R}^{(m,p)}$, we have

$$\mathbf{R}^{(m,p)} = \min_{k \in \{1,2,\ldots,K\}} \mathbf{R}_k^{(m,p)}, \tag{2.5}$$

where

$$\mathbf{R}_k^{(m,p)} \triangleq \min_{l \in \{1,2,\ldots,n_k\}} \| \mathbf{q}_{k,l}^{(m)} - \mathbf{D}_{(p)} \mathbf{D}_{(p)}^{\dagger} \mathbf{q}_{k,l}^{(m)} \|_2. \tag{2.6}$$

In other words, we select the minimum residual among all $l \in \{1, 2, ..., n_k\}$, and all $k \in \{1, 2, ..., K\}$, as the similarity between the query video sequence $\mathbf{Q}^{(m)}$ and dictionary $\mathbf{D}_{(p)}$. We denote the resulting dictionary-based face recognition algorithm as DFRV.

2.2.5 Non-linear kernel dictionaries for video-based face recognition

The class identities in the face dataset may not be linearly separable. Hence, we also extend the DFRV framework to the kernel space. This essentially requires the dictionary learning model to be non-linear [63].

Let $\mathbf{\Phi} : \mathbb{R}^L \rightarrow \mathcal{H}$ be a non-linear mapping from the L dimensional space into a dot product space $\mathcal{H}$. A non-linear dictionary can be trained in the feature space $\mathcal{H}$ by solving the following optimization problem

$$(\hat{\mathbf{A}}^i_{j,k}, \hat{\mathbf{\Gamma}}^i_{j,k}) = \arg \min_{\mathbf{A}^i_{j,k}, \mathbf{\Gamma}^i_{j,k}} \|\mathbf{\Phi}(\mathbf{G}^i_{j,k}) - \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k}\mathbf{\Gamma}^i_{j,k}\|^2_F,$$
$$\text{subject to } \|\boldsymbol{\gamma}_l\|_0 \leq T_0, \quad \forall l, \tag{2.7}$$

where

$$\mathbf{\Phi}(\mathbf{G}^i_{j,k}) = [\mathbf{\Phi}(\mathbf{g}^i_{j,k,1}), \mathbf{\Phi}(\mathbf{g}^i_{j,k,2}), \cdots, \mathbf{\Phi}(\mathbf{g}^i_{j,k,B})].$$

Since the dictionary lies in the linear span of the samples $\mathbf{\Phi}(\mathbf{G}^i_{j,k})$, in (2.7) we have used the following model for the dictionary in the feature space,

$$\mathbf{D}^i_{j,k} = \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k},$$

where $\mathbf{A}^i_{j,k} \in \mathbb{R}^{B \times K_0}$ is a matrix with K_0 atoms [63]. This model provides adaptivity via modification of the matrix $\mathbf{A}^i_{j,k}$. Through some algebraic manipulations, the cost

function in (2.7) can be rewritten as,

$$\begin{aligned}&\|\mathbf{\Phi}(\mathbf{G}^i_{j,k}) - \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k}\mathbf{\Gamma}^i_{j,k}\|^2_F \\ &= \mathbf{tr}((\mathbf{I} - \mathbf{A}^i_{j,k}\mathbf{\Gamma}^i_{j,k})^T\boldsymbol{\mathcal{K}}(\mathbf{G}^i_{j,k}, \mathbf{G}^i_{j,k})(\mathbf{I} - \mathbf{A}^i_{j,k}\mathbf{\Gamma}^i_{j,k})),\end{aligned} \tag{2.8}$$

where $\boldsymbol{\mathcal{K}}(\mathbf{G}^i_{j,k}, \mathbf{G}^i_{j,k})$ is a kernel matrix whose elements are computed from

$$\kappa(r, s) = \mathbf{\Phi}(\mathbf{g}^i_{j,k,r})^T\mathbf{\Phi}(\mathbf{g}^i_{j,k,s}).$$

It is apparent that the objective function is feasible since it only involves a matrix of finite dimension $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{B\times B}$, instead of dealing with a possibly infinite dimensional dictionary.

An important property of this formulation is that the computation of $\boldsymbol{\mathcal{K}}$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\mathbf{\Phi}$. Some commonly used kernels include polynomial kernels

$$\kappa(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$$

and Gaussian kernels

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma}\right),$$

where c, d and σ are the parameters.

Similar to the optimization of (2.1) using the linear K-SVD [62] algorithm, the optimization of (2.7) involves sparse coding and dictionary update steps in the feature space which results in the kernel K-SVD algorithm [63]. Details of the optimization algorithm can be found in [63].

2.2.5.1 Feature space identification

Let $\mathbf{A}_j^i = \text{diag}[\mathbf{A}_{j,1}^i, \mathbf{A}_{j,1}^i, \cdots, \mathbf{A}_{j,K}^i]$ denote the jth learned coefficient matrix of subject i. Assuming that there are a total of P gallery video sequences, we can write the associated coefficient matrices $\mathbf{A}_{(p)}$ for $p = 1, 2, ..., P$, where each $\mathbf{A}_{(p)}$ corresponds to $\mathbf{A}_j^i$ for some subject i and its jth video sequence. We first find the coefficient vectors, $\mathbf{x}_{k,l}^{(m)}$ with at most T_0 non-zero elements such that $\Phi(\mathbf{G}_{(p)})\mathbf{A}_{(p)}\mathbf{x}_{k,l}^{(m)}$ approximates $\mathbf{q}_{k,l}^{(m)}$ by minimizing the following problem

$$\min_{\mathbf{x}_{k,l}^{(m)}} \|\Phi(\mathbf{q}_{k,l}^{(m)}) - \Phi(\mathbf{G}_{(p)})\mathbf{A}_{(p)}\mathbf{x}_{k,l}^{(m)}\|_2 \quad \text{such that} \quad \|\mathbf{x}_{k,l}^{(m)}\|_0 \leq T_0, \tag{2.9}$$

where $\mathbf{G}_{(p)} = \mathbf{G}_j^i$ for the jth video sequence of the subject i. The above problem can be solved by the Kernel Orthogonal Matching Pursuit (KOMP) algorithm [63].

Similar to (2.3), image $\mathbf{q}_{k,l}^{(m)}$ votes for sequence $\hat{p}$ such that

$$\begin{aligned}
\hat{p} &= \underset{p}{\text{argmin}} \; r(\mathbf{q}_{k,l}^{(m)}, \mathbf{A}_{(p)}) \\
&= \underset{p}{\text{argmin}} \, \|\Phi(\mathbf{q}_{k,l}^{(m)}) - \Phi(\mathbf{G}_{(p)})\mathbf{A}_{(p)}\mathbf{x}_{k,l}^{(m)}\|_2^2 \\
&= \underset{p}{\text{argmin}} \, \boldsymbol{\mathcal{K}}(\mathbf{q}_{k,l}^{(m)}, \mathbf{q}_{k,l}^{(m)}) - 2\boldsymbol{\mathcal{K}}(\mathbf{q}_{k,l}^{(m)}, \mathbf{G}_{(p)})\mathbf{A}_{(p)}\mathbf{x}_{k,l}^{(m)} \\
&\quad + \mathbf{x}_{k,l}^{(m)T}\mathbf{A}_{(p)}^T\boldsymbol{\mathcal{K}}(\mathbf{G}_{(p)}, \mathbf{G}_{(p)})\mathbf{A}_{(p)}\mathbf{x}_{k,l}^{(m)},
\end{aligned} \tag{2.10}$$

where

$$\begin{aligned}
&\boldsymbol{\mathcal{K}}(\mathbf{q}_{k,l}^{(m)}, \mathbf{G}_{(p)}) = \\
&\quad [\kappa(\mathbf{q}_{k,l}^{(m)}, \mathbf{g}_{(p),1}), \kappa(\mathbf{q}_{k,l}^{(m)}, \mathbf{g}_{(p),2}), \cdots, \kappa(\mathbf{q}_{k,l}^{(m)}, \mathbf{g}_{(p),KB})].
\end{aligned} \tag{2.11}$$

To make the sequence-level decision for identification, we select p^* by (2.4), with $C_{p,k}$ replaced by $\tilde{C}_{p,k}$, the total number of votes from the kth partition of the mth query video for the pth target video sequence according to (2.10).

2.2.5.2 Feature space verification

For verification using the kernel dictionaries, we construct the similarity matrix $\tilde{\mathbf{R}}^{(m,p)}$ by

$$\tilde{\mathbf{R}}^{(m,p)} = \min_{k \in \{1,2,\ldots,K\}} \tilde{\mathbf{R}}_k^{(m,p)}, \tag{2.12}$$

where $\tilde{\mathbf{R}}_k^{(m,p)}$ is the residual between $\mathbf{Q}_k^{(m)}$ and the kernel dictionary built from the pth target video sequence. It is computed by

$$\tilde{\mathbf{R}}_k^{(m,p)} = \min_{l \in \{1,2,\ldots,n_k\}} r(\mathbf{q}_{k,l}^{(m)}, \mathbf{A}_{(p)}). \tag{2.13}$$

We denote the resulting kernel DFRV algorithm as KDFRV. Both linear DFRV and non-linear KDFRV algorithms are summarized in Algorithm 2.

2.3 Video-based Face Recognition via Joint Sparse Representation

Based on the framework presented in Section 2.2, we present our joint sparsity-based approach for unconstrained video-to-video face recognition. Note that the same approach for recognition using the body can be applied as well. This approach is based on the well known regularized regression method, multi-task multivariate Lasso [46], [47]. It simultaneously takes into account correlations as well as coupling information between frames of a video while enforcing joint sparsity within each

Algorithm 2: Video-based Face Recognition (DFRV & KDFRV)

Training:

1. Given a sequence - the jth video of subject i, extract all the frames from it. Detect and crop face regions to form a set S_j^i.

2. Separate S_j^i into K partitions. Augment each partition by adding artificial images and obtain the resulting augmented gallery matrix from the kth partition, $\mathbf{G}_{j,k}^i, \forall k = 1, 2, ..., K$.

3. Use (2.1) for DFRV (and (2.7) for KDFRV) to learn the partition-specific sub-dictionary $\mathbf{D}_{j,k}^i, \forall k = 1, 2, ..., K$. Construct the sequence-specific dictionary $\mathbf{D}_j^i$ as in (2.2).

Testing:

1. Partition the mth query video sequence $\mathbf{Q}^{(m)} = \bigcup_{k=1}^{K} \mathbf{Q}_k^{(m)}$, where $\mathbf{Q}_k^{(m)} = [\mathbf{q}_{k,1}^{(m)} \ \mathbf{q}_{k,2}^{(m)} \ ... \ \mathbf{q}_{k,n_k}^{(m)}]$.

2. (Identification) Use (2.3) for DFRV (and (2.10) for KDFRV) to determine the vote from $\mathbf{q}_{k,l}^{(m)}, \forall k, l$. Then, use (2.4) and subject-sequence correspondence $\boldsymbol{m}(\cdot)$ to make the final decision.

3. (Verification) Find the similarity matrix between $\mathbf{Q}^{(m)}$ and $\mathbf{D}_{(p)}$ by (2.5) for DFRV (and (2.12) for KDFRV). The ROC curve can be obtained from the similarity matrix.

frame's observation. The algorithm is then kernelized to enable it to handle the non-linearities present in video data.

Fig. 2.2 shows an overview of this approach. In the training stage, from cropped face images, we partition the pth video sequence so that frames with the same pose and illumination condition are in one partition, where $p \in \{1, 2, ..., P\}$ corresponds to the jth video of subject i for some i, j. We then find the best representation for each member in these partitions by learning dictionaries under strict sparsity constraints. Each learned sub-dictionary $\mathbf{D}^k_{(p)}$ for $k = 1, 2, 3$, and $p = 1, 2, ..., P$, represents the pth video's kth face feature that is under a particular pose and/or illumination condition.

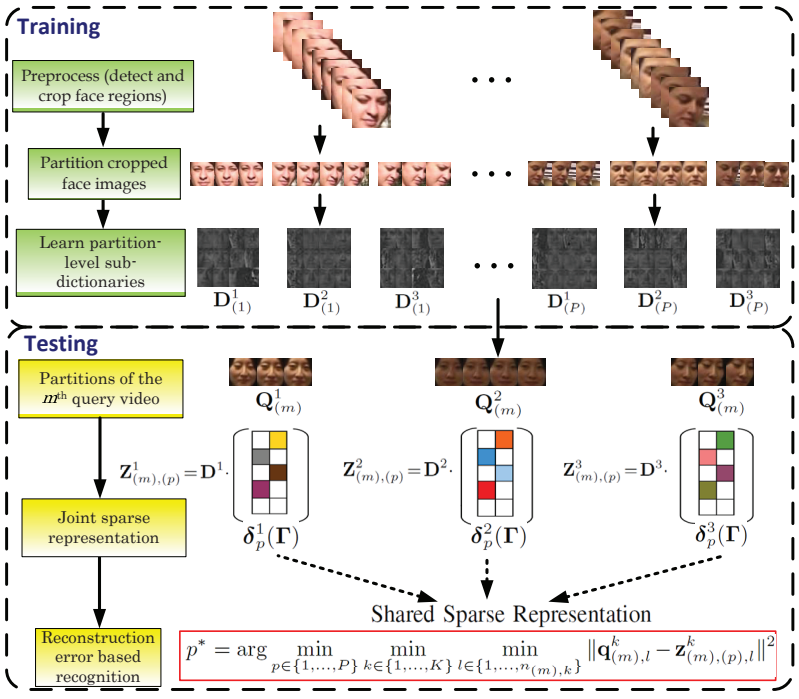


Figure 2.2: Overview of our sparsity-based approach.

In the testing stage, the same partition step is applied on the mth query

video sequence to acquire partitions, $\mathbf{Q}_{(m)}^k$, $k = 1, 2, 3$. Then, for each $\mathbf{Q}_{(m)}^k$, sub-dictionaries from all target videos are found and concatenated to form the dictionary $\mathbf{D}^k$. Using $\mathbf{D}^k, k = 1, 2, 3$, and a query sample, the joint sparse representation $\mathbf{\Gamma} = [\mathbf{\Gamma}^1\ \mathbf{\Gamma}^2\ \mathbf{\Gamma}^3]$ is found to make decisions for recognition under the minimum class reconstruction error criterion.

2.3.1 Sparse representation for video-based face recognition (SRV)

We exploit the joint sparsity of coefficients from different partitions to make a joint decision. Let the augmented gallery matrix of the kth partition of the pth video sequence be denoted by $\mathbf{G}_{(p)}^k$. Given $\mathbf{G}_{(p)}^k$, a dictionary $\mathbf{D}_{(p)}^k \in \mathbb{R}^{L \times \tilde{K}}$ is learned by (2.1). Let $\mathbf{D}^k = [\mathbf{D}_{(1)}^k\ \mathbf{D}_{(2)}^k\ ...\ \mathbf{D}_{(P)}^k]$ be the concatenation of the kth sub-dictionaries from all target videos. Letting $\mathbf{\Gamma} = [\mathbf{\Gamma}^1\ \mathbf{\Gamma}^2 \dots \mathbf{\Gamma}^K] \in I\!R^{d \times n_{(m)}}$ be the matrix formed by concatenating the coefficient matrices with $d = \sum_{j=1}^{P} \tilde{K}$ and $n_{(m)} = \sum_{k=1}^{K} n_{(m),k}$, we seek the row-sparse matrix $\mathbf{\Gamma}$ by solving the following ℓ_1/ℓ_q-regularized least square problem

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{Q}_{(m)}^k - \mathbf{D}^k \mathbf{\Gamma}^k\|_F^2 + \lambda \|\mathbf{\Gamma}\|_{1,q} \tag{2.14}$$

where λ is a positive parameter and q is set greater than 1 to make the optimization problem convex. Here, $\|\mathbf{\Gamma}\|_{1,q}$ is a norm defined as $\|\mathbf{\Gamma}\|_{1,q} = \sum_{i=1}^{d} \|\boldsymbol{\gamma}^i\|_q$ where $\boldsymbol{\gamma}^i$'s are the row vectors of $\mathbf{\Gamma}$. Problem (2.14) can be solved using the classical Alternating Direction Method of Multipliers (ADMM) [64], [65], [66]. See [64], [65] for more details on ADMM. For our experiments, we choose $q = 2$.

2.3.1.1 Identification

For identification, we use the knowledge of the correspondence $\boldsymbol{f}(\cdot)$ between subjects and sequences to assign the query video sequence $\mathbf{Q}_{(m)}$ to subject $i^* = \boldsymbol{f}(p^*)$, where p^* is the sequence-level decision. Once $\hat{\mathbf{\Gamma}}$ is obtained, p^* is declared as the one that produces the smallest approximation error.

$$p^* = \arg\min_{p} \min_{k\in\{1,\ldots,K\}} \min_{l\in\{1,\ldots,n_{(m),k}\}} \|\mathbf{q}^k_{(m),l} - \mathbf{D}^k \boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})\|^2, \tag{2.15}$$

where $\boldsymbol{\delta}^k_{p,l}(\cdot)$ is the indicator function defined by keeping the coefficients corresponding to the kth partition from pth target video for the lth query image, and setting coefficients in all other rows and columns equal to zero.

2.3.1.2 Verification

For verification, given a query video sequence and any gallery video sequence, the goal is to correctly determine whether these two belong to the same subject. The well-known receiver operating characteristic (ROC) curve, which describes relations between false acceptance rates (FARs) and true acceptance rates (TARs), is used to evaluate the performance of verification algorithms. As the TAR increases, so does the FAR. Therefore, one would expect an ideal verification framework to have all TARs equal to 1 for any FARs. The ROC curves can be computed given a similarity matrix. In the proposed method, the residual between a query $\mathbf{Q}^{(m)}$ and the pth target video, is used to fill in the (m, p) entry of the similarity matrix. Denoting the

residual by $\mathbf{R}^{(m,p)}$, we have

$$\mathbf{R}^{(m,p)} = \min_{k\in\{1,2,...,K\}} \mathbf{R}_k^{(m,p)}, \tag{2.16}$$

where

$$\mathbf{R}_k^{(m,p)} \triangleq \min_{l\in\{1,...,n_{(m),k}\}} \|\mathbf{q}_{(m),l}^k - \mathbf{D}^k \boldsymbol{\delta}_{p,l}^k(\hat{\boldsymbol{\Gamma}})\|^2. \tag{2.17}$$

In other words, we select the minimum residual among all $l \in \{1, 2, ..., n_{(m),k}\}$, and all $k \in \{1, 2, ..., K\}$, as the similarity between the query video sequence $\mathbf{Q}^{(m)}$ and the pth target video.

The SRV algorithm is summarized in Algorithm 3.

Algorithm 3: Sparse representation for video-based face recognition (SRV)

Input: Partition-level sub-dictionaries $\{\mathbf{D}^k\}_{i=k}^K$ and query videos $\{\mathbf{Q}_{(m)}^k\}_{k=1}^K$.

Procedure: Obtain $\hat{\boldsymbol{\Gamma}}$ by solving

$$\hat{\boldsymbol{\Gamma}} = \arg\min_{\boldsymbol{\Gamma}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{Q}_{(m)}^k - \mathbf{D}^k \boldsymbol{\Gamma}^k\|_F^2 + \lambda_1 \|\boldsymbol{\Gamma}\|_{1,q},$$

Output:

(Identification) `video` $p^* =$

$$\arg\min_p \min_{k\in\{1,...,K\}} \min_{l\in\{1,...,n_{(m),k}\}} \|\mathbf{q}_{(m),l}^k - \mathbf{D}^k \boldsymbol{\delta}_{p,l}^k(\hat{\boldsymbol{\Gamma}})\|^2,$$

`subject` $i^* = \boldsymbol{f}(p^*)$.

(Verification) Use the similarity $\mathbf{R}^{(m,p)}$ computed by (2.16) and (2.17) to construct the similarity matrix, from which the ROC curves can be obtained.

2.3.2 Finding aligned sub-dictionaries for unconstrained videos

The formulation presented above is made under the assumption that $\mathbf{D}^k$ is a concatenation of sub-dictionaries that are aligned with $\mathbf{Q}_{(m)}^k$. In other words, if $\mathbf{Q}_{(m)}^k$

collects a subject's left side face images from the mth video, then $\mathbf{D}^k$ must also collect sub-dictionaries of left side faces from all target videos. In practical situations, unlike constrained videos, illumination and pose conditions in an unconstrained video vary. For example, some query videos contain left side face images only, while some target videos contain frontal face images only. In addition, no information on which partition represents which specific pose and illumination condition is available. To overcome these difficulties before finding the joint sparse representation, we find approximately aligned dictionaries $\mathbf{D}^k$ such that $\mathbf{D}^k_{(p)}, p = 1, 2, ..., P$ are obtained by:

$$\mathbf{D}^k_{(p)} = \arg \max_{\hat{\mathbf{D}}^u_{(p)}, u \in \{1,2,...,K\}} C_u, \tag{2.18}$$

where C_u is the number of votes for the uth sub-dictionary of the pth target video collected from each $\mathbf{q}^k_{(m),l}$ in $\mathbf{Q}^k_{(m)}$. In other words, the aligned sub-dictionaries are determined by the majority vote criterion. Each query image $\mathbf{q}^k_{(m),l}$ in the kth partition of the mth query video $\mathbf{Q}^k_{(m)}$ votes for $\hat{\mathbf{D}}^u_{(p)}$ such that it has the minimum reconstruction error from its projection on $\hat{\mathbf{D}}^v_{(p)}$:

$$u = \arg \min_v \|\mathbf{q}^k_{(m),l} - \hat{\mathbf{D}}^v_{(p)} \hat{\mathbf{D}}^{v\dagger}_{(p)} \mathbf{q}^k_{(m),l}\|^2, \tag{2.19}$$

where $\hat{\mathbf{D}}^{v\dagger}_{(p)}$ is the pseudo-inverse of $\hat{\mathbf{D}}^v_{(p)}$.

2.3.3 Kernel sparse representation for video-based face recognition (KSRV)

The class identities in different partitions may not be linearly separable. Hence, we also extend the joint sparse representation framework to the non-linear kernel

space. The kernel function, $\kappa : \mathbb{R}^n \times \mathbb{R}^n$, is defined as the inner product

$$\kappa(\mathbf{d}_i, \mathbf{d}_j) = \langle \phi(\mathbf{d}_i), \phi(\mathbf{d}_j) \rangle$$

where, ϕ is an implicit mapping projecting the vector $\mathbf{d}$ into a higher dimensional space.

Considering the general case of K partitions of the mth query video with $\{\mathbf{Q}_{(m)}^k\}_{k=1}^K$ as a set of $n_{(m),k}$ observations, the feature space representation can be written as:

$$\mathbf{\Phi}(\mathbf{Q}_{(m)}^k) = [\phi(\mathbf{q}_{(m),1}^k)\ \phi(\mathbf{q}_{(m),2}^k)\ ...\ \phi(\mathbf{q}_{(m),n_{(m),k}}^k)]$$

Similarly, the dictionary of training samples for the kth partition can be represented in feature space as

$$\mathbf{\Phi}(\mathbf{D}^k) = [\phi(\mathbf{D}_1^k), \phi(\mathbf{D}_2^k), \cdots, \phi(\mathbf{D}_P^k)]$$

As in joint linear space representation, we have:

$$\mathbf{\Phi}(\mathbf{Q}_{(m)}^k) = \mathbf{\Phi}(\mathbf{D}^k)\mathbf{\Gamma}^k$$

where, $\mathbf{\Gamma}^k$ is the coefficient matrix associated with partition k. Incorporating information from all the partitions, we solve the following optimization problem similar to the linear case:

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{\Phi}(\mathbf{Q}_{(m)}^k) - \mathbf{\Phi}(\mathbf{D}^k)\mathbf{\Gamma}^k\|_F^2 + \lambda \|\mathbf{\Gamma}\|_{1,q} \tag{2.20}$$

where, $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \cdots, \mathbf{\Gamma}^K]$. It is clear that the information from all the partitions of a video are integrated via the shared sparsity pattern of the matrices $\{\mathbf{\Gamma}^k\}_{k=1}^K$.

This can be reformulated in terms of kernel matrices as:

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{k=1}^{K} \big(\text{trace}(\mathbf{\Gamma}^{k^T} \mathbf{K}_{\mathbf{D}^k, \mathbf{D}^k} \mathbf{\Gamma}^k) - 2\text{trace}(\mathbf{K}_{\mathbf{D}^k, \mathbf{Q}^k_{(m)}} \mathbf{\Gamma}^k) \big) + \lambda \|\mathbf{\Gamma}\|_{1,q} \tag{2.21}$$

where, the kernel matrix $\mathbf{K}_{\mathbf{X},\mathbf{Y}}$ is defined as:

$$\mathbf{K}_{\mathbf{X},\mathbf{Y}}(i,j) = \kappa(\mathbf{x}_i, \mathbf{y}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{y}_j) \rangle, \tag{2.22}$$

with $\mathbf{x}_i$ and $\mathbf{y}_j$ being i^{th} and j^{th} columns of $\mathbf{X}$ and $\mathbf{Y}$ respectively. Similar to the linear case, (2.20) can be solved using the ADMM type of algorithm.

2.3.3.1 Identification

Once $\hat{\mathbf{\Gamma}}$ is obtained, we assign $\mathbf{Q}_{(m)}$ to subject $i^* = \boldsymbol{f}(p^*)$, where p^* is obtained as follows.

$$\begin{aligned} p^* &= \arg\min_{p} \min_{k} \min_{l \in \{1,\ldots,n_{(m),k}\}} \|\phi(\mathbf{q}^k_{(m),l}) - \mathbf{\Phi}(\mathbf{D}^k_{(p)}) \boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})\|^2 \\ &= \arg\min_{p} \min_{k} \min_{l \in \{1,\ldots,n_{(m),k}\}} \Big\{ \text{trace}(\mathbf{K}_{\mathbf{Q}^k_{(m)} \mathbf{Q}^k_{(m)}}) \\ &\quad -2 \text{ trace}(\boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})^T \mathbf{K}_{\mathbf{D}^k_{(p)}} \mathbf{K}_{\mathbf{Q}^k_{(m)}} \boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})) \\ &\quad +\text{trace}(\boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})^T \mathbf{K}_{\mathbf{D}^k_{(p)} \mathbf{D}^k_{(p)}} \boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})) \Big\}. \end{aligned} \tag{2.23}$$

2.3.3.2 Verification

Similar to the linear case in section 2.3.1.1, we use (2.16) to construct the similarity $\mathbf{R}^{(m,p)}$, with $\mathbf{R}^{(m,p)}_k$ in (2.17) replaced with

$$\mathbf{R}^{(m,p)}_k \triangleq \min_{l \in \{1,\ldots,n_{(m),k}\}} \|\phi(\mathbf{q}^k_{(m),l}) - \mathbf{\Phi}(\mathbf{D}^k_{(p)}) \boldsymbol{\delta}^k_{p,l}(\hat{\mathbf{\Gamma}})\|^2. \tag{2.24}$$

The KSRV algorithm is summarized in Algorithm 4.

Algorithm 4: Kernel sparse representation for video-based face recognition (KSRV)

Input: Partition-level sub-dictionaries $\{\mathbf{D}^k\}_{i=k}^K$ and query videos $\{\mathbf{Q}_{(m),k}\}_{k=1}^K$.

Procedure: Obtain $\hat{\mathbf{\Gamma}}$ by solving

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{\Phi}(\mathbf{Q}_{(m)}^k) - \mathbf{\Phi}(\mathbf{D}^k)\mathbf{\Gamma}^k\|_F^2 + \lambda\|\mathbf{\Gamma}\|_{1,q} \tag{2.25}$$

Output:

(Identification) `video` $p^* =$

$$\arg\min_p \min_k \min_{l\in\{1,\ldots,n_{(m),k}\}} \|\phi(\mathbf{q}_{(m),l}^k) - \mathbf{\Phi}(\mathbf{D}_{(p)}^k)\boldsymbol{\delta}_{p,l}^k(\hat{\mathbf{\Gamma}})\|^2,$$

`subject` $i^* = \boldsymbol{f}(p^*)$.

(Verification) Use the similarity $\mathbf{R}^{(m,p)}$ computed by (2.16) and (2.24) to construct the similarity matrix, from which the ROC curves can be obtained.

2.4 Experimental Results

To illustrate the effectiveness of our method, we present experimental results on four publicly available datasets for video-based face recognition: the Multiple Biometric Grand Challenge (MBGC) [51], [52], the Face and Ocular Challenge Series (FOCS) [2], [53], the Honda/UCSD [32], and the UMD Comcast10 [54] datasets. For MBGC and FOCS videos, we use the upper body information in addition to faces for recognizing humans. All cropped face and upper body images were resized to $L = 20\times 20$ pixels. Unless otherwise stated[2], we took histogram equalized grayscales

[2]For experiments on FOCS UT-Dallas dataset in section 2.4.2, we took concatenations of images showing sequential differences among bodies (in grayscale) as motion identity cues.

from resized images, and use them as the feature for recognition. We summarize in Table 2.1 the number of partitions per video (K) and the number of atoms per sub-dictionary (K_0) used in our experiments on the four datasets. For kernel dictionaries, we choose the Gaussian kernel with parameter $\sigma = 32$.

We compare the performance of our method with several state-of-the-art video-based face recognition methods, including the Wrapped Gaussian Common Pole (WGCP) method [1], [12], and the Sparse Approximated Nearest Points (SANP) method [13]. When reporting the experimental results on face and upper body parts using the DFRV based methods, we use the following naming convention:

- DFRV-f: DFRV on face images
- DFRV-b: DFRV on upper body images
- DFRV-bf: Score-level fusion of DFRV on both face and upper body images
- KDFRV-f: KDFRV on face images
- SRV-f: SRV on face images
- KSRV-f: KSRV on face images

2.4.1 MBGC Video version 1

The MBGC Video version 1 dataset (Notre Dame dataset) contains 399 walking (frontal-face) and 371 activity (profile-face) video sequences of 146 subjects. Both types of sequences were collected in standard definition (SD) format (720×480

datasets	MBGC	FOCS	Honda/UCSD	UMD Comcast10
K	3	5	3	3
K_0	40	25	14	5

Table 2.1: Summary of number of partitions per video (K) and number of atoms per sub-dictionary (K_0) in our experiments.

pixels) and high definition (HD) format (1440×1080 pixels). The 399 walking sequences consist of 201 sequences in SD and 198 in HD. For the 371 walking video sequences, 185 are in SD and 186 are in HD. The top row of Fig. 2.3(a) shows example frames from four different walking sequences, where each subject walks toward the video camera with a frontal pose for most of the time and turns to the left or right showing the profile face at the end. The bottom row of Fig. 2.3(a) shows example frames from four different activity sequences, where each subject reads from a paper, and the sequences consists of non-frontal views of the subject. There exist several challenging conditions including frontal and profile faces in shadow, and the profile faces sometimes are heavily covered by one's hair.

Fig. 2.4 shows an example of the output from the video partitioning stage. For results in Fig. 2.4, the number of partitions is set equal to $K = 3$. Results are presented for 2 subjects for both walking and activity sequences[3]. For subject faces from walking videos shown in Fig. 2.4(a), the corresponding cropped upper body images from activity videos are shown in Fig. 2.4(b)[4]. Each row shows up to

[3]For the illustration purpose only, here we just show results of 2 subjects.

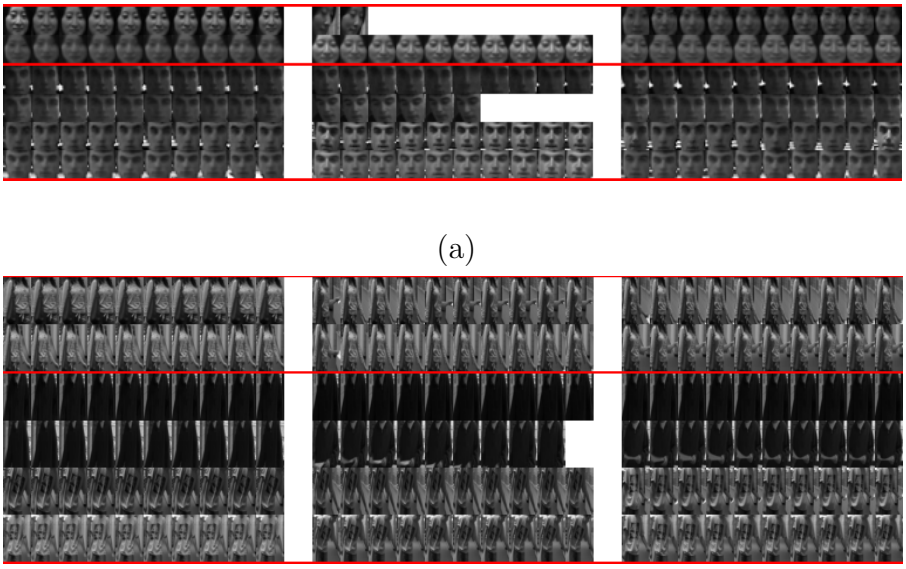
[4]As lower body parts are not available for some videos, in our work only face and upper body

(a)

(b)

Figure 2.3: Examples of MBGC and UT-Dallas video sequences. (a) MBGC walking (top row) and activity (bottom row) sequences. (b) UT-Dallas walking (top row) and activity (bottom row) sequences.

30 partitioned cropped face (or upper body) images from the same video sequence. The red lines separate different subjects. It can be seen that each partition from a video sequence encodes a particular pose and/or illumination condition, and different partitions represent different conditions.



(a)

(b)

Figure 2.4: Partition results of example face and upper body images from MBGC videos: (a) Face images from walking videos. (b) The corresponding upper body images from activity videos. Red lines separate different subjects. A subject has at least two video sequences. Face (or upper body) images from a video sequence are shown in a row, and are further divided into three partitions. Each partition shows up to 10 face (or upper body) images. A partition represents a particular pose and illumination condition.

images were used for recognition.

2.4.1.1 Identification results on the MBGC dataset

Following the experiment design in [1], we conducted a leave-one-out identification experiment on 3 subsets of the cropped face and upper body images from walking videos performed. These 3 subsets are $S2$ (subjects which have at least two video sequences: 144 subjects, 397 videos), $S3$ (subjects which have at least three video sequences: 55 subjects, 219 videos) and $S4$ (subjects which have at least four video sequences: 54 subjects, 216 videos).

Table 2.2 lists the percentages of correct identifications for this experiment. The proposed DFRV and sparsity-based methods outperform the other state-of-the-art methods [1], [12] and [13]. For most subjects in this dataset, videos of the same subject wearing the same cloth and performed similar activities, were recorded in the same day. As different subjects possess different body appearance, compared to DFRV-f, the use of body information in DFRV-b and DFRV-bf enhance the discriminative identification rate. Comparing DFRV-f and KDFRV-f (or SRV-f and KSRV-f), we observe that kernel dictionaries obtained higher average identification rates on this dataset. This may be the case due to the fact that kernel dictionaries are able to capture the non-linearities in data. Hence, with the proper choice of kernel and parameters, the performance obtained using the kernel dictionaries is in general better than that given by the linear dictionaries.

We further compared our method on face images with a baseline method where the dictionary learning stage in the DFRV method is omitted and the cropped images in each partition are directly used as dictionaries. This method is denoted

as "no DL". As shown in Table 2.2, omitting the dictionary learning stage results in a poor performance compared to the DFRV-f method. This baseline, however, remains better than SANP [13] as it keeps the video partitions that account for the pose and illumination variations.

MBGC walking videos	Procrustes Metric [1], [12]	Kernel Density [1], [12]	WGCP [1]	SANP [13]	Baseline (no DL)	DFRV-f
$S2$	43.79	39.74	63.79	83.88	78.09	85.64
$S3$	53.88	50.22	74.88	84.02	77.63	88.13
$S4$	53.70	50.46	75	84.26	77.78	88.43
Average	50.46	46.81	71.22	84.05	77.83	87.40
MBGC walking videos	KDFRV-f	SRV-f	KSRV-f	DFRV-b	DFRV-bf	-
$S2$	84.89	86.65	86.65	94.71	**95.97**	-
$S3$	89.50	87.67	88.58	94.98	**95.89**	-
$S4$	89.81	87.96	88.89	95.37	**96.30**	-
Average	88.07	87.43	88.04	95.02	**96.05**	-

Table 2.2: Identification rates (%) of leave-one-out testing experiments on the MBGC walking videos. The proposed DFRV and sparsity-based methods outperform statistical methods and the SANP method, recently proposed in [1] and [13], respectively.

In the second set of experiments, we selected videos of subjects that are in at least two videos (i.e., $S2$). We divide all these videos into SD and HD videos, to conduct "SD vs HD" (SD as probe; HD as gallery) and "HD vs SD" (HD as probe; SD as gallery) experiments. Correct identification rates are shown in Table 2.3. The DFRV and sparsity-based methods outperformed the other methods significantly. The WGCP [1] method finds projections of training samples on a Grassmann manifold on its tangent plane and uses them to learn a pre-assumed Gaussian model. While the geodesic distance of any point on the manifold to the pole (i.e., the tangent point of the manifold and the corresponding tangent plane) is maintained, this property does not always apply to the geodesic distance between any pair of points on the manifold. Also, the pre-assumed Gaussian model may not be appropriate to model the training samples. On the other hand, the SANP [13] method is based on image set classification. The major limitation of this method is that it relies on the unseen appearances of a set to be modeled by affine combinations of samples. While this may be true for some variations in facial illumination, it does not hold for extreme variations especially in the presence of shadows, pose and expression variations. The proposed DFRV-based methods overcome this limitation by video partitioning and effectively combining different partition-level sub-dictionaries.

2.4.1.2 Verification results on the MBGC dataset

Fig. 2.5(a) and (b) show the ROC curves of WGCP and the proposed DFRV methods for "SD vs HD" and "HD vs SD" verification experiments, respectively.

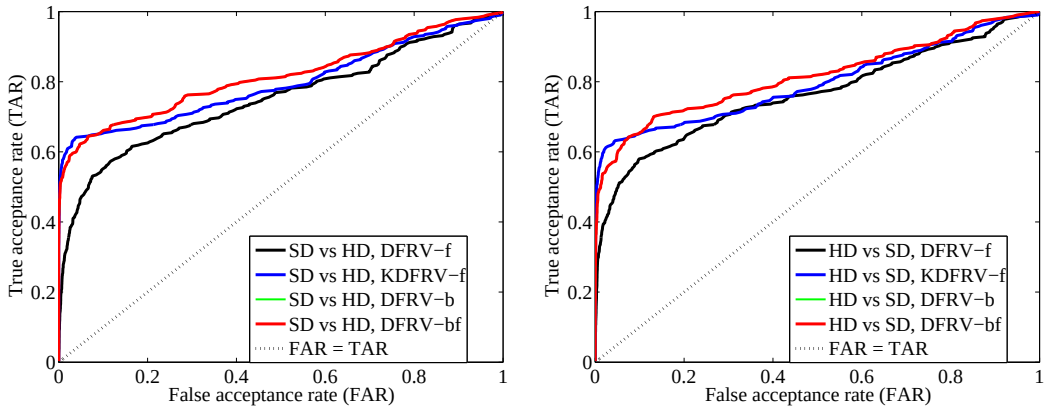
MBGC walking videos	Procrustes Metric [1], [12]	Kernel Density [1], [12]	WGCP [1]	SANP [13]	Baseline (no DL)	DFRV-f
SD vs HD	61.31	55.78	30.15	41.71	77.39	86.93
HD vs SD	68.69	56.06	30.30	45.96	85.35	91.41
Average	65	55.92	30.23	43.84	81.37	89.17
MBGC walking videos	KDFRV-f	SRV-f	KSRV-f	DFRV-b	DFRV-bf	-
SD vs HD	89.45	91.96	91.46	95.48	**96.48**	-
HD vs SD	89.90	90.40	91.41	**95.96**	**95.96**	-
Average	89.68	91.18	91.44	95.72	**96.22**	-

Table 2.3: Identification rates (%) of "SD vs HD" and "HD vs SD" experiments on the MBGC walking video subset $S2$ (the subset that contains subjects who have at least two video sequences). In this experiment, most subjects (89 out of 144) have only one video per subject available for training. The DFRV-bf method achieves the best identification rates.

As shown in both figures, one could hardly see the difference between DFRV-b (body only, in color green) and DFRV-bf (body and face, in color red) curves as the body feature dominates the overall performance. In addition, both DFRV-b and DFRV-bf obtained better verification performances than the DFRV-f method. For both identification and verification, the HD test samples had better performances than the SD test samples. Fig. 2.6(a)(b) show ROC curves of WGCP and the proposed sparsity-based methods. Both SRV-f and KSRV-f methods have similar performances. They give better ROC curves than the WGCP method in Fig. 2.6(a) for all FARs, and in Fig. 2.6(b) for low FARs.

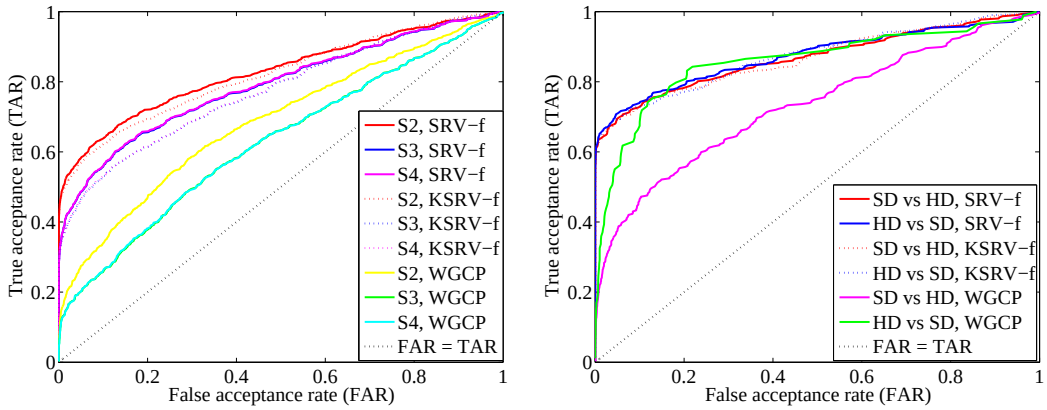
We further examine the effect on the performance of varying the number of video sequences per person in the gallery. We divide the videos into two groups beforehand either as probe, or as gallery. For most subjects (89 out of 144), this setting allows only one video per subject for training, unlike the previous leave-one-out test in which there are always at least two training video sequences per subject (the subject whose video is currently used as the probe is excluded). Results presented above show that the WGCP method in this setting does not perform so well. We observe that the WGCP method is able to give satisfactory performance only when there are enough video sequences for training, which allows one to obtain more discriminative metrics for different subjects.

In the MBGC [51] protocol, verifications are specified by two sets: target and query. The protocol requires the algorithm to match each target sequence with all query sequences. We performed three verification experiments: walking vs walking (WW), activity vs walking (AW), activity vs activity (AA). Fig. 2.7(a) shows the



(a) (b)

Figure 2.5: ROC curves of DFRV-based methods on the MBGC walking videos: (a) "SD vs HD". (b) "HD vs SD". There is no difference between DFRV-b and DFRV-bf curves. Both DFRV-b and DFRV-bf obtained better verification performances than DFRV-f.



(a) (b)

Figure 2.6: ROC curves of WGCP and sparsity-based methods on the MBGC walking videos: (a) S2, S3 and S4. (b) "SD vs HD", and "HD vs SD". The proposed sparsity-based methods give better ROC curves than the WGCP method shown in (a), and in (b) for low FARs.

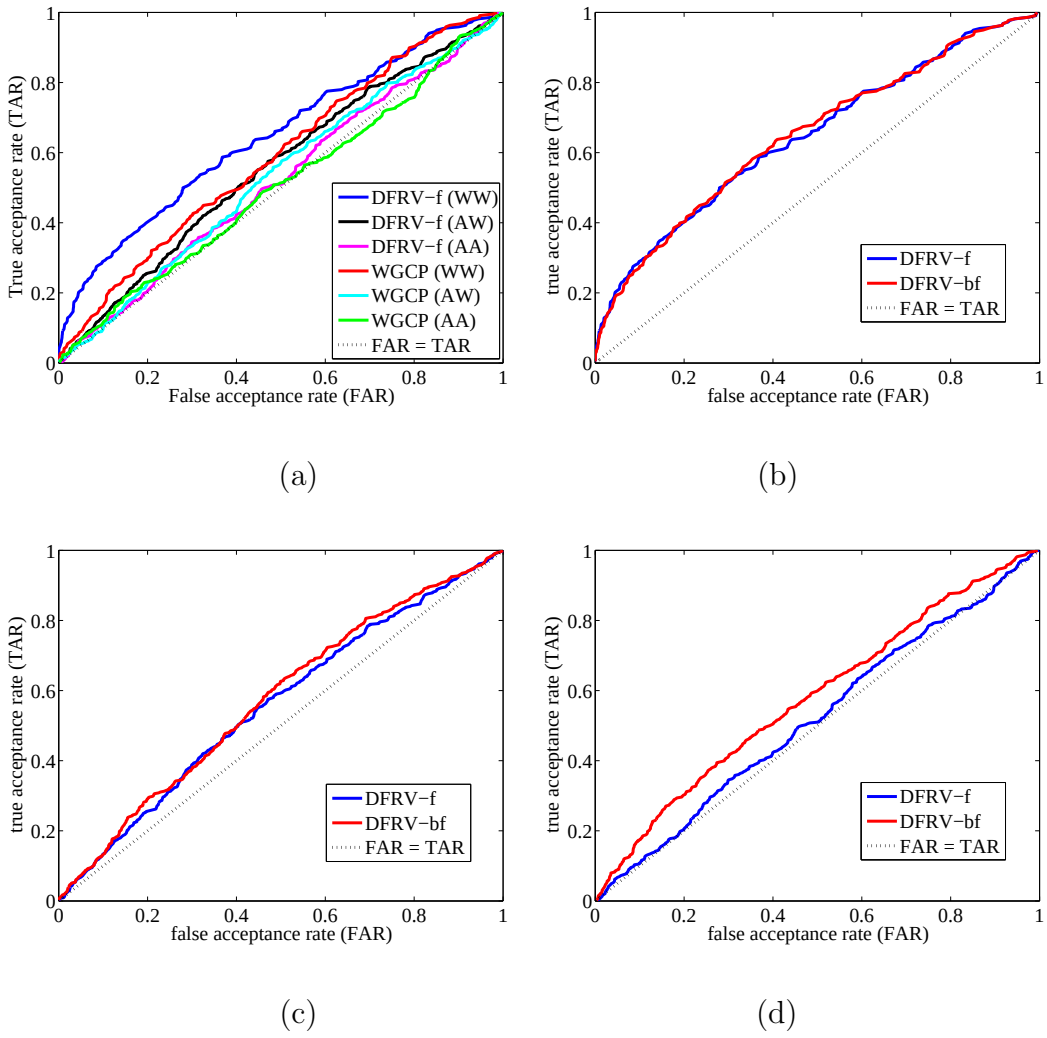


Figure 2.7: ROC curves of the MBGC experiments on walking and activity videos: (a) Comparing DFRV-f with WGCP in WW, AW and AA experiments. The proposed DFRV-f method gives better ROC curves than WGCP in WW experiments. Both curves are close to the random guess in the challenging AW and AA experiments. (b) Comparing DFRV-f and DFRV-bf in WW experiments. (c) Comparing DFRV-f and DFRV-bf in AW experiments. (d) Comparing DFRV-f and DFRV-bf in AA experiments, where a better improvement of DFRV-bf over DFRV-f is obtained.

ROC curves. We observe that DFRV-f gives better ROC curve than WGCP for almost all FARs, in WW experiments. In AW and AA experiments; however, all curves are pretty close to random performance. These two experiments are very challenging. According to the MBGC website [52], for the AW and AA experiments, no results have been reported that are better than random.

Fig. 2.7(b)(c)(d) show the comparisons between DFRV-f and DFRV-bf in WW, AW and AA experiments, respectively. As the MBGC verification protocol is designed to exclude matching videos of the same subject recorded in the same day, the body feature no longer contributes as much as it does in the identification experiments. Therefore, the gain obtained from the DFRV-bf is limited. A slightly larger improvement of DFRV-bf over DFRV-f can be observed in AA experiments (Fig. 2.7(d)) only.

MBGC v1 experiments	score-level fusion
$S2$, $S3$, $S4$ identification	$0.55\mathbf{C}_b+0.45\mathbf{C}_f$
HD(SD) vs SD(HD) identification	$0.5\mathbf{C}_b+0.5\mathbf{C}_f$
HD(SD) vs SD(HD) verification	$0.95\left(\frac{\mathbf{R}_b-\text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right)+0.05\left(\frac{\mathbf{R}_f-\text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$
'walking vs walking' verification	$0.95\left(\frac{\mathbf{R}_b-\text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right)+0.05\left(\frac{\mathbf{R}_f-\text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$
'walking vs activity' verification	$0.95\left(\frac{\mathbf{R}_b-\text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right)+0.05\left(\frac{\mathbf{R}_f-\text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$
'activity vs activity' verification	$0.95\left(\frac{\mathbf{R}_b-\text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right)+0.05\left(\frac{\mathbf{R}_f-\text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$

Table 2.4: Score level fusion summary of MBGC version 1 (Notre Dame) experiments.

We regard face and body as distinct biometric modalities. Table 2.4 summarizes the score level fusion (among face and body) approach that linearly combines vote matrices ($\mathbf{C}_f$ denoted for face vote matrix and $\mathbf{C}_b$ for body vote matrix) for identification, and median & median absolute deviation (MAD) normalized distance matrices ($\mathbf{R}_f$ for face distance matrix and $\mathbf{R}_b$ for body distance matrix) for verification. The distance normalization using median and MAD are used as they are robust to outliers [67].

2.4.2 FOCS UT-Dallas Video

The video challenge of Face and Ocular Challenge Series (FOCS) [2] is designed to match "frontal vs frontal", "frontal vs non-frontal", and "non-frontal vs non-frontal" video sequences. In this section we present our experimental results on the UT Dallas video sequences contained in the FOCS video challenge. The performance of the DFRV-f algorithm on the UT Dallas dataset shows the strength of our approach on a difficult data set. In addition, it allows us to directly compare the performance of the DFRV-f algorithm to humans [2].

The FOCS UT Dallas dataset contains 510 walking (frontal face) and 506 activity (non-frontal face) video sequences recorded from 295 subjects with frame size 720×480 pixels. The top row of Fig. 2.3(b) shows key frames from four different walking sequences of one subject. The sequences were acquired on different days. In the walking sequences, the subject is originally positioned far away from the video camera, walks towards it with a frontal pose, and finally turns away from the

video camera with profile face. The bottom row of figure 2.3(b) shows key frames of four different activity sequences of the same subject. In these sequences, the subject stands and talks with another person with a non-frontal face view to the video camera. The sequences contain normal head motions that occur during a conversation; e.g., the head turning up to 90 degrees, hand raising and/or pointing somewhere.

2.4.2.1 Identification results on the FOCS dataset

We conducted the same leave-one-out tests on 3 subsets: $S2$ (189 subjects, 404 videos), $S3$ (19 subjects, 64 videos), and $S4$ (6 subjects, 25 videos) from the UT-Dallas walking videos. For body images, in order to capture both shape and temporal information in a low resolution scenario, we took the grayscale differences between a reference upper body frame and all of its subsequent frames in a cycle period (L subsequent frames). Then we resized the resulting concatenated sequential differences as a motion cue of that reference frame. Fig. 2.8 shows for the three example subjects their sequential upper body differences (in grayscale) over $L = 18$ frames, where each row captures a subject's upper body shape and information on its temporal movements. This method does not require silhouette extraction or background substraction. Table 2.5 shows the identification results. Among the compared methods, the DFRV-bf method achieved the best identification rates. Among methods other than DFRV-bf and DFRV-b (i.e., methods using face only), the KDFRV-f method, however, did not obtain better identification performance

Figure 2.8: Sequential upper body differences in grayscale: the grayscale differences between a reference upper body frame and its subsequent frames in a cycle period of $L = 18$ frames. For each subject, the corresponding upper body differences computed from a reference frame are shown in a row as a motion cue of that reference frame. Here there are three rows shown for three different subjects. This feature captures both the shape and its temporal movement information, while not requiring either silhouette extraction or background substraction.

than DFRV-f and WGCP. This may be due to the fact that the choice of the kernel and its parameter(s) for learning kernel dictionaries may not be optimal for this experimental setup. Optimizing the choice of kernels and parameters is one of our future research directions.

2.4.2.2 Verification results on the FOCS dataset

Like MBGC, FOCS specifies a verification protocol: **1A** (walking vs walking), **2A** (activity vs walking), and **3A** (activity vs activity). In these experiments, 481 walking videos and 477 activity videos are chosen as query videos. The size of target sets ranges from 109 to 135 video sequences. Fig. 2.9 shows ROC curves of verification experiments. In Fig. 2.9(a), we compare the proposed algorithm with WGCP [1]. In all three experiments, the DFRV-f algorithm is superior to the

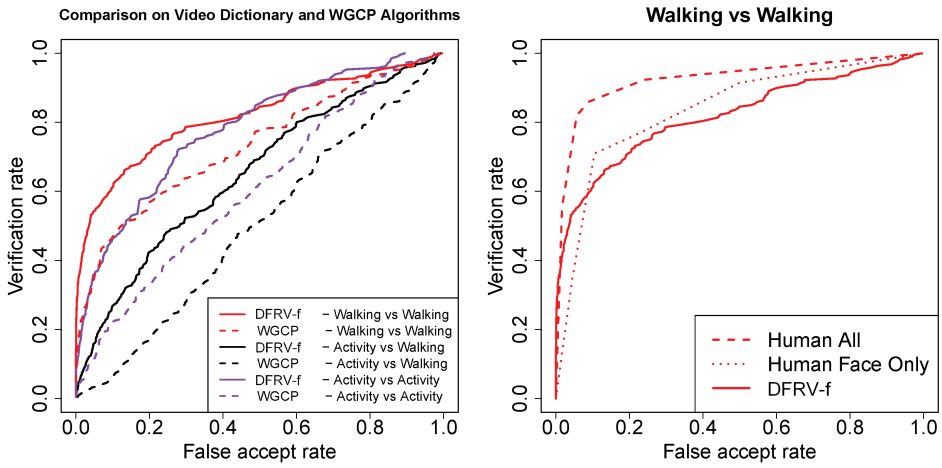
UT-Dallas walking videos	Procrustes Metric [1], [12]	Kernel Density [1], [12]	WGCP [1]	SANP [13]	Baseline (no DL)
$S2$	38.12	40.84	53.22	48.27	45.05
$S3$	60.94	64.06	70.31	60.94	67.19
$S4$	64	64	76.00	68.00	76.00
Average	54.35	54.97	66.51	59.07	62.75
UT-Dallas walking videos	DFRV-f	KDFRV-f	DFRV-b	DFRV-bf	-
$S2$	59.90	46.53	20.30	**61.14**	-
$S3$	78.13	71.88	42.19	**79.69**	-
$S4$	80.00	76.00	60.00	**84.00**	-
Average	72.68	64.80	40.83	**74.94**	-

Table 2.5: Identification rates (%) of leave-one-out testing experiments on the FOCS UT-Dallas walking videos. The DFRV-bf method performs the best.

WGCP algorithm.

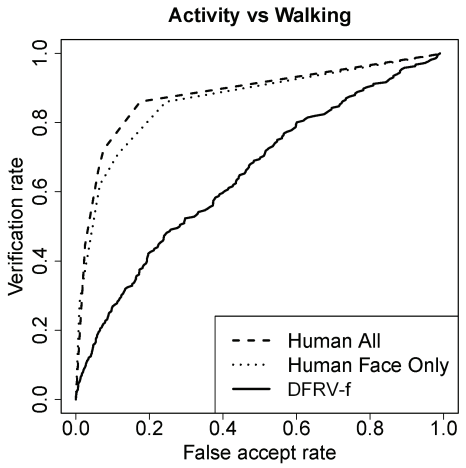
O'Toole *et al.* [2] evaluated the accuracy of humans recognizing people in the UT Dallas data set. Human performance was reported for both static and dynamic presentations of faces and bodies. Performance in [2] was reported for humans viewing the original sequence and for sequences edited to contain only the head. Since the DFRV-f algorithm only encodes face information, it is reasonable to compare the DFRV-f with human performance on the original sequences and the edited face only sequences. In Fig. 2.9(b)(c)(d) we compare the performance of the DFRV-f algorithm and humans for experiments **1A**, **2A**, and **3A**. In Fig. 2.9(b) and (d), we observe that the performance of the DFRV-f algorithm is very close to humans on the face only matching task. Experiments **1A** and **3A** are within pose matching tasks; whereas, **2A** is cross pose. Reported performance is better than random; however, not near human level of performance.

In Fig. 2.10(a)(b)(c), we compare DFRV-f and DFRV-bf in 1A, 2A and 3A experiments, respectively. As shown, there is not much difference between the two methods. In fact, unlike MBGC, a subject with different cloth and facial appearances (as shown in Fig. 2.3(b)) was recorded in different days. The body feature becomes much less discriminative and DFRV-b no longer gives satisfactory identification results. Therefore, for this challenging dataset, as the face feature dominates the performance, both DFRV-f and DFRV-bf obtained similar identification and verification results. The score level fusion between face and body for DFRV-bf is summarized in Table 2.6, where scores of the face feature weigh more as the face features are more discriminative on this dataset.

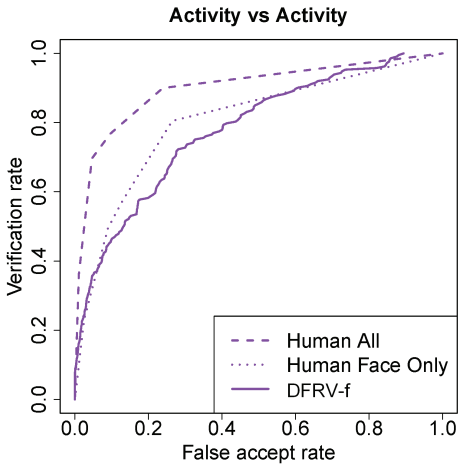


(a)

(b)

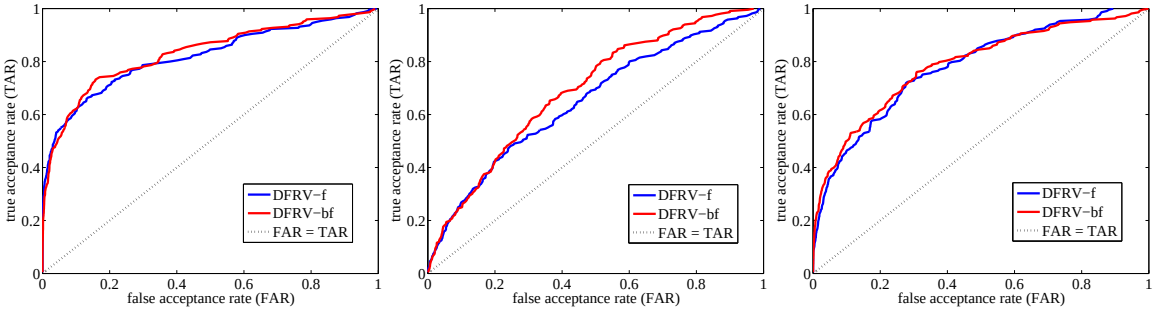


(c)



(d)

Figure 2.9: ROC curves of FOCS experiments on UT-Dallas videos: (a) comparison between DFRV-f and WGCP [1]; (b)(c)(d) comparison between DFRV-f and human perception [2]: (b) walking vs walking (c) activity vs walking (d) activity vs activity. Compared to WGCP, our DFRV-f method gives better ROC curves, which also stay very close to those of face-only human perception in (b)(d) cases.



(a) (b) (c)

Figure 2.10: ROC curves of DFRV-f and DFRV-bf on the UT-Dallas videos. (a) walking vs walking. (b) activity vs walking. (c) activity vs activity. DFRV-bf obtained higher detection rates than DFRV-f (for FARs > 0.3) in the activity vs activity experiment.

FOCS (UT-Dallas) experiments	score-level fusion
$S2$, $S3$, $S4$ identification	$0.4\mathbf{C}_b + 0.6\mathbf{C}_f$
'walking vs walking' verification	$0.15\left(\frac{\mathbf{R}_b - \text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right) + 0.85\left(\frac{\mathbf{R}_f - \text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$
'activity vs walking' verification	$0.15\left(\frac{\mathbf{R}_b - \text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right) + 0.85\left(\frac{\mathbf{R}_f - \text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$
'activity vs activity' verification	$0.15\left(\frac{\mathbf{R}_b - \text{MED}(\mathbf{R}_b)}{\text{MAD}(\mathbf{R}_b)}\right) + 0.85\left(\frac{\mathbf{R}_f - \text{MED}(\mathbf{R}_f)}{\text{MAD}(\mathbf{R}_f)}\right)$

Table 2.6: Score level fusion summary of FOCS (UT-Dallas) experiments.

2.4.3 Honda/UCSD Dataset

The third set of experiments is conducted on the Honda/UCSD Dataset [32]. The Honda Dataset consists of 59 video sequences from 20 distinct subjects. We follow the same experiment procedure in [13]. The experiments are done in three cases of the maximum set length (available number of cropped-face images per video sequence) as defined in [13]: 50, 100 and full length frames. Table 2.7 shows identification rates of our methods and other state-of-the-art methods. Both SRV-f and KSRV-f obtained the highest average identification rates. The proposed DFRV and sparsity-based methods ranked the second and tied with the MDA method [68] for the full length case.

2.4.4 UMD Comcast10 dataset

The UMD Comcast10 dataset contains 12 videos recorded of a group of 16 subjects. The videos were collected in a high definition format (1920×1088 pixels). They contain sequences of subjects standing without walking toward the camera, which we refer to standing sequences, and sequence(s) of each subject walking toward the camera, which we refer to walking sequences. After segmenting the videos according to subjects and sequence types, we obtained 93 sequences in total: 70 standing sequences and 23 walking sequences. Fig. 2.11(a) shows example frames from four different standing sequences, where most subjects are standing in a group. As some subjects were having conversations and others were looking elsewhere, their faces were sometimes non-frontal or partially occluded. Fig. 2.11(b) shows example

Set length	DCC [69]	MMD [70]	MDA [68]	AHISD [71]	CHISD [71]
50 frames	76.92	69.23	74.36	87.18	82.05
100 frames	84.62	87.18	94.87	84.62	84.62
full length	94.87	94.87	97.44	89.74	92.31
Average	85.47	83.76	88.89	87.18	86.33
Set length	SANP [13]	DFRV-f	KDFRV-f	SRV-f	KSRV-f
50 frames	84.62	89.74	92.31	**94.87**	**94.87**
100 frames	92.31	**97.44**	94.87	**97.44**	**97.44**
full length	**100**	97.44	97.44	97.44	97.44
Average	92.31	94.87	94.87	**96.58**	**96.58**

Table 2.7: Identification rates (%) on Honda/UCSD Dataset. Both SRV-f and KSRV-f obtained the highest average identification rates.

frames from four different walking sequences, in each of which a single subject was walking toward the camera, with a frontal face for most of the time. However, the walking subject's head sometimes turned to the right or left showing a profile face. Furthermore, for both types of sequences, the camera was not always static. In fact quite often it switched back and forth, to create more challenging conditions in these unconstrained video sequences. Fig. 2.11(c) shows example frames with blurred subjects due to the movement of the camera.

(a)

(b)

(c)

Figure 2.11: Example frames from the UMD Comcast10 videos. (a) standing sequences. (b) walking sequences. (c) Frames with blurred subjects due to the moving camera. Faces in standing sequences were sometimes non-frontal or partially occluded, while faces in walking sequences were frontal for most of the time. Camera's movement raises the difficulty of face tracking and recognition.

Following the experiment design in [1], we conducted a leave-one-out identification experiment on 3 subsets of the cropped face images from walking videos.

These 3 subsets are $S2$ (subjects which have at least two video sequences: 16 subjects, 93 sequences), $S3$ (subjects which have at least three sequences: 15 subjects, 91 sequences) and $S6$ (subjects which have at least four sequences: 7 subjects, 51 sequences). Note that for these particular segmented sequences, the three sets $S3$, $S4$ and $S5$ are identical. Table 2.8 lists the percentages of correct identifications for this experiment. The proposed DFRV-f, KDFRV-f, SRV-f and KSRV-f outperformed the other compared methods. In particular, KDFRV-f achieved 100% identification rates on $S3 \sim S6$ video subsets.

UMD videos	Procrustes Metric [1], [12]	Kernel Density [1], [12]	WGCP [1]	SANP [13]	Baseline (no DL)
$S2$	82.80	81.72	82.97	92.47	91.40
$S3, S4, S5$	84.62	83.52	83.52	93.41	92.31
$S6$	98.04	96.08	88.23	98.04	92.31
Average	88.49	87.11	84.91	94.64	92.01
UMD videos	DFRV-f	KDFRV-f	SRV-f	KSRV-f	-
$S2$	94.62	**95.70**	92.47	93.55	-
$S3, S4, S5$	96.70	**100**	94.51	94.51	-
$S6$	96.70	**100**	98.04	98.04	-
Average	96.00	**98.57**	95.01	95.37	-

Table 2.8: Identification rates (%) of leave-one-out testing experiments on the UMD Comcast10 dataset. The KDFRV-f method outperforms other compared methods.

Fig. 2.12(a)(c) show the verification performances in $S2$, $S3$ and $S6$ experiments through ROC curves. From this figure, ROC curves of $S2$, $S3$ and $S6$ are indistinguishable. The proposed DFRV and sparsity-based methods give better ROC curves than the WGCP method. Fig. 2.12(b)(d) show the ROC curves for "Standing vs Walking" (standing sequences as probe; walking sequences as gallery) and "Walking vs Standing" (walking sequences as probe; standing sequences as gallery) experiments. Similar to the identification results, the proposed KDFRV-f performs slightly better than DFRV-f, and KSRV-f performs better than SRV-f. They all outperform the WGCP method.

2.5 Summary

We presented a video dictionary-based family for unconstrained video-to-video human identification and verification. We kernelized the dictionary learning algorithm to handle the non-linearities present in the video data. We combined the face features with the upper body features or motion identity cues, to improve the recognition accuracy. Using video dictionaries, we further proposed a joint sparsity-based approach, which simultaneously takes into account correlations as well as coupling information between frames of a video while enforcing joint sparsity within each frame's observation. Extensive experiments on four unconstrained video datasets show that our approach performs better than several well known video-based face recognition methods discussed in the literature.

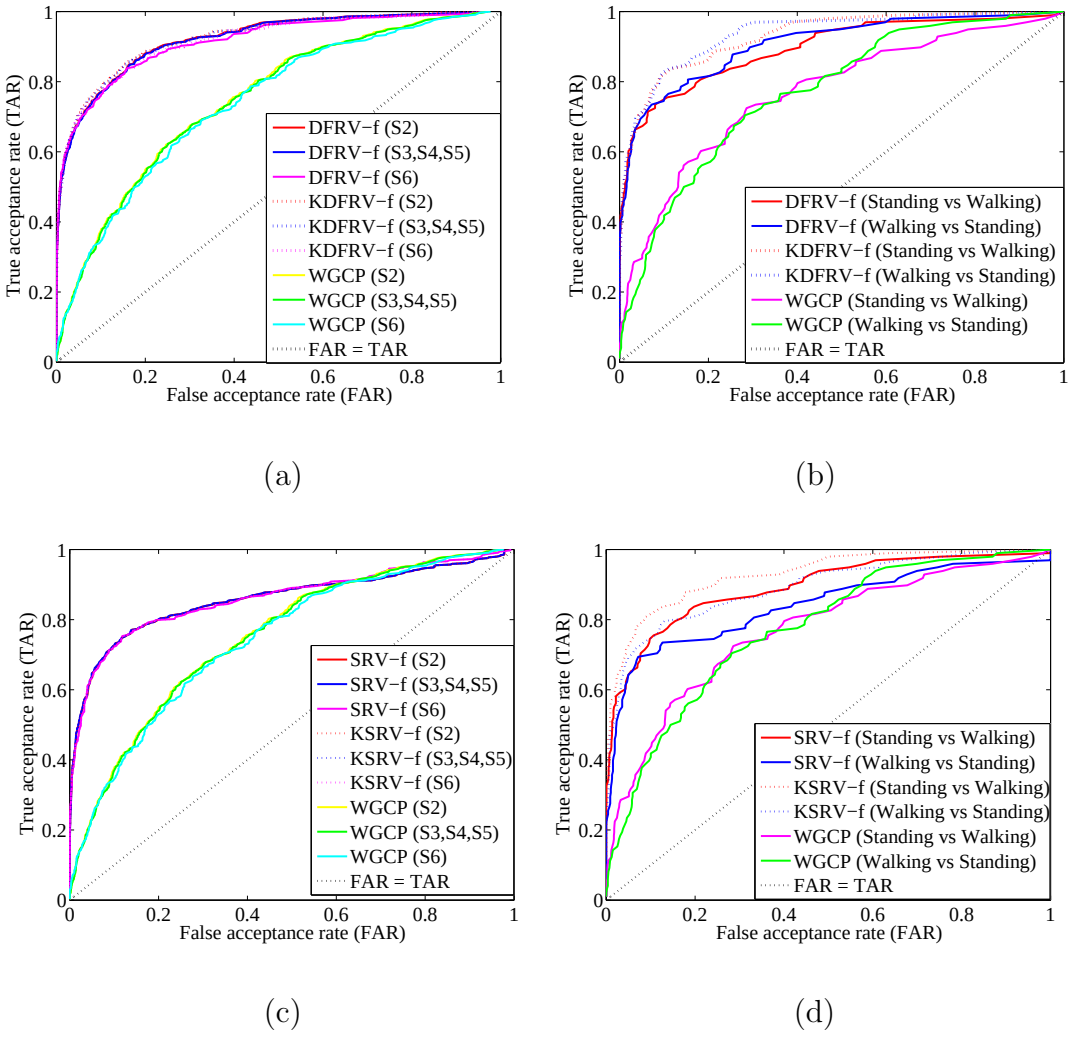


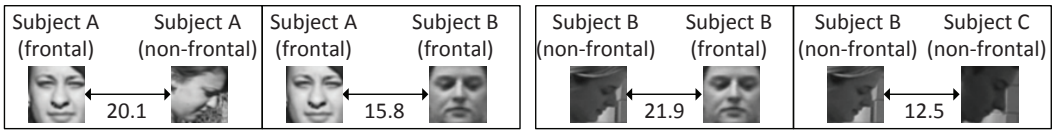
Figure 2.12: ROC curves of verification experiments on the UMD Comcast10 dataset. (a)(c) S2, S3, and S6. (b)(d) "standing vs walking" and "walking vs standing".

Chapter 3: Adaptive Representations for Video-based Face Recognition Across Pose

In video-based face recognition, pose and illumination variations still remain one of the biggest challenges. Though significant efforts have gone into understanding the different sources of variations affecting facial appearance, the accuracy of video-based face recognition algorithms in completely uncontrolled scenarios may be far from satisfactory. Some of existing methods [11], [1], [12], [13], [14], [3], rely on the pose diversity contained in the gallery videos to handle pose variations. When there are pose differences between the videos, the robustness of these methods is limited.

Fig. 3.1 shows two typical examples of face mismatching across pose. In Fig. 3.1(a), the first face pair compares frontal and non-frontal images of subject A; the second pair compares frontal images of subjects A and B. In this case, the distance[1] shows a better match between the two frontal images than the true match across pose. Fig. 3.1(b) gives another example where the distance shows a better match between the two non-frontal images than the true match. Whenever the gallery videos contain only frontal poses and the probe videos contain only side-poses

[1]Here, we take the ℓ_2-norm distance between two images.



(a) (b)

Figure 3.1: Illustration of common errors when matching faces across changes in pose. (a) The first face pair compares frontal and non-frontal images of subject A; the second pair compares frontal images of subjects A and B. (b) The first face pair compares non-frontal and frontal images of subject B; the second pair compares non-frontal images of subject B and C. In both cases, the distance shows a better match between the two in-pose images than the true match across pose.

(and vice versa)[2], the above methods can perform poorly.

In this chapter, we consider matching faces across very different poses between the probe and gallery videos. Our reference sets are independent of the gallery and probe sets specified by the protocol. We propose two methods to compute pose aligned features based on 3-dimensional (3D) rotation and sparse representation. The first method is referred to as Sparse Representation-based Alignment (SRA) method. The pose aligned images obtained through this method are referred to as the SRA images. The second method is an adaptation of the SRA method that rotates the video dictionary atoms to align the pose prior to recognition. It is referred to as the Dictionary Rotation (DR) method.

The proposed SRA method consists of three steps (see Fig. 3.2). In the

[2]We refer to gallery videos as enrolled videos for training, and probe videos as videos to be recognized for testing.

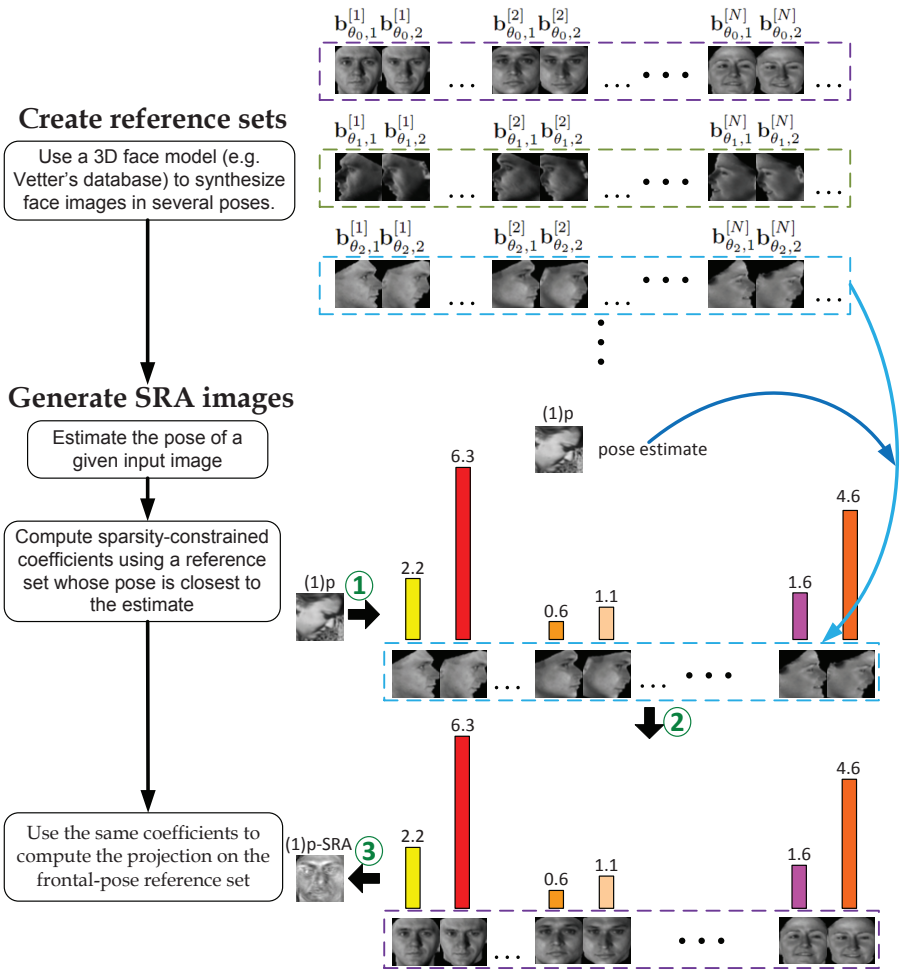


Figure 3.2: Illustration of creating reference sets and generating the SRA images.

first step, we obtain candidate reference sets for pose alignment from independent sources. The reference set does not contain videos in the gallery or probe sets specified in the protocol. Candidate reference sets can be other face datasets that contain images of many subjects in various poses, or generated from 3D face models through synthesizing face images. The second step is to generate the SRA images. Given a test image, we estimate its pose, compute the sparsity-constrained coefficient vector on the reference set for the estimated pose, and map the coefficient vector back onto the frontal-pose reference set to obtain the SRA image of the test image. Fig. 3.2 illustrates the first two steps of the proposed method. In the third step, we build the SRA video dictionaries and the base video dictionaries (DFRV [3])[3], and then effectively fuse both video dictionaries to construct the distance matrix for recognition. The SRA video dictionaries enable face recognition across changes in poses. Fig. 3.3 (a) and (b) illustrate the training and testing stages for building video dictionaries and constructing the distance matrix, respectively.

The rest of the chapter is organized as follows. Section 3.1 details the proposed SRA and DR methods. Section 3.2 describe the proposed approach to pose estimation from videos. We present experimental results with discussions in Section 3.3. Section 3.4 concludes the chapter with a brief summary.

[3]We refer to the video dictionaries by DFRV [3] as "base video dictionaries".

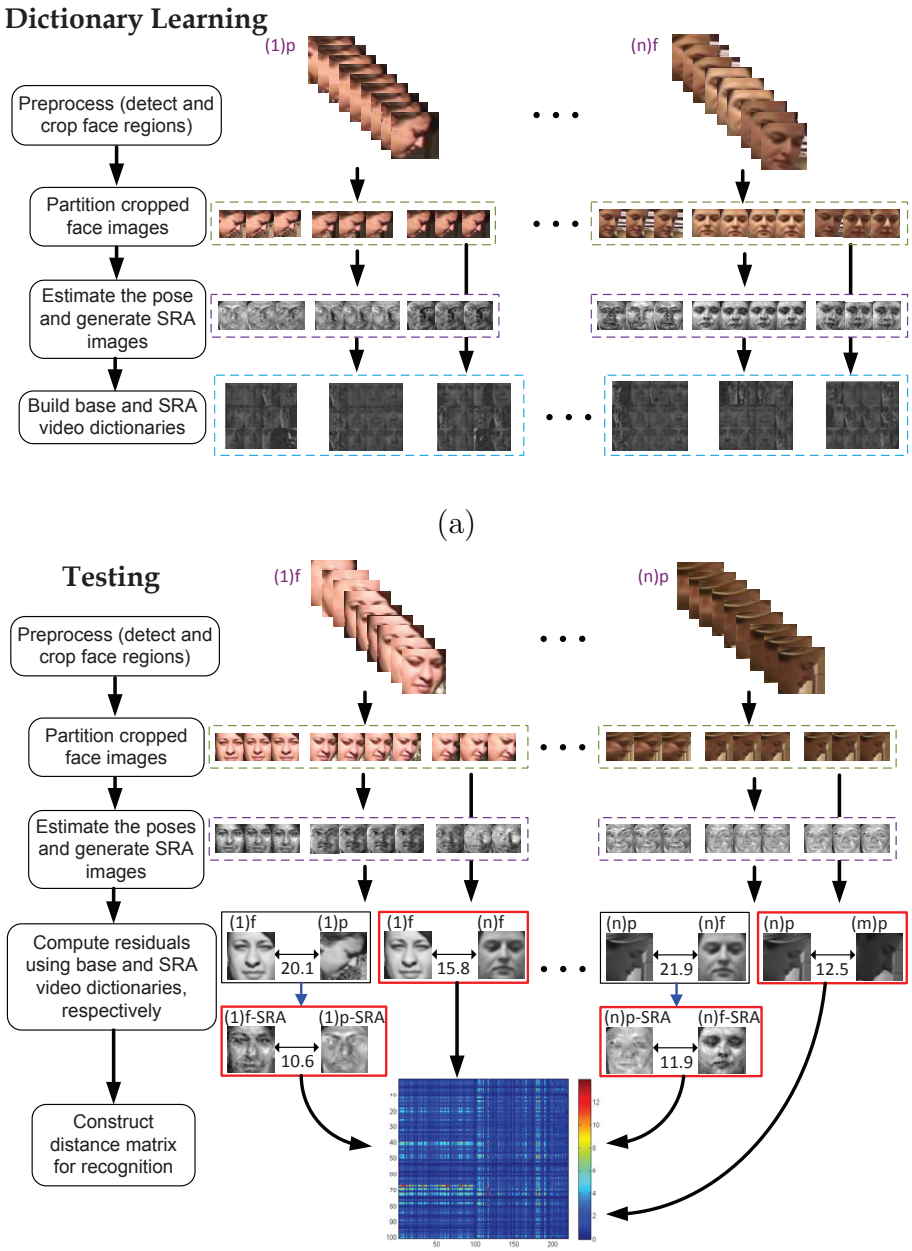


Figure 3.3: Illustration of training and testing stages. (a) Training stage: build the base video dictionaries [3] and SRA video dictionaries. (b) Testing stage: compute residuals using both the base video dictionaries and SRA video dictionaries for recognition.

3.1 Sparse Representation-based Alignment

The proposed SRA method computes the pose aligned feature as the re-projection of each face image under an arbitrary pose onto a fixed pose (e.g. frontal) reference set, and then measures the pairwise distances among these re-projections for recognition. The underlying assumption is that whenever a face image under an arbitrary pose θ_1 is represented using a reference set under pose θ_1 weighed by a set of sparse coefficients, then the face image of the same subject under another pose θ_2 can be approximately represented by the re-projection using the *same* set of sparse coefficients on the reference set under pose θ_2 [72].

Without loss of generality, let $\mathbf{y}_\theta$ be a d-dimensional vector representing an input face image under a non-frontal pose θ in its column-vectorized form, where θ_a, θ_e and θ_z stand for azimuth angle (wrt the y axis), elevation angle (wrt the x axis) and the rotation angle wrt the z axis, respectively. The input image can be a probe image for test, or a gallery image for training specified by the protocol.

The proposed SRA method consists of three steps. In what follows, we present details of these steps.

3.1.1 Obtain Reference Sets for Alignment

In the first step, we obtain the reference sets for pose alignment. Ideally, the reference sets are independent datasets from the protocol with face images from various subjects in different pose and illumination conditions. The poses of the reference sets should cover those in the probe and gallery videos. In practice, when

these datasets are not available, or lack sufficient pose and/or subject variability, one alternative is to use a 3D face model (e.g. Vetter's database [73]) to synthesize face images in several poses with illumination changed accordingly [74]. The reference sets can then be built from the synthesized images. Let the resulting reference set from V subjects under a particular pose θ be denoted by

$$\mathbf{B}_\theta = [\mathbf{b}_{\theta,0}^{[1]} ... \ \mathbf{b}_{\theta,U-1}^{[1]} | \cdots | \ \mathbf{b}_{\theta,0}^{[V]} ... \ \mathbf{b}_{\theta,U-1}^{[V]}], \tag{3.1}$$

where $\mathbf{b}_{\theta,u}^{[v]}$ denotes the uth synthetically created variation of face image of the vth subject under pose θ in its column-vectorized form. The variations include slight changes in pose (including θ_a, θ_e and θ_z), illumination or spatial locations. These are created to account for variations among images that are non-ideally cropped from unconstrained videos, and for the pose errors due to non-ideal estimation.

In particular, there are in total only $U-1$ synthetic variations, appearing in the same sequence for all subjects and all poses. In other words, the uth synthetic variation applied to yield $\mathbf{b}_{\theta,u}^{[v]}$ is the same operation for all v and θ. This constraint is required to generate final aligned images in the frontal pose using sparsity constraint coefficients, as discussed in section 3.1.2. For simplicity of notation, we use $\mathbf{B}_{\theta_0}$ to denote the reference set from V subjects under the frontal pose.

3.1.2 Generate SRA Images

In the second step, we generate the SRA images using the reference sets presented in section 3.1.1. We present the motivation of using the sparse representation-based pose aligned feature as follows.

Under the assumption that $\mathbf{y}_\theta$ can be approximated by a sparse linear combination of vectors from $\mathbf{B}_\theta$, we compute the sparse coefficient vector $\hat{\boldsymbol{\gamma}}$ by solving the following optimization problem

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \|\boldsymbol{\gamma}\|_1 \text{ such that } \|\mathbf{y}_\theta - \mathbf{B}_\theta \boldsymbol{\gamma}\|_2^2 \leq \varepsilon, \tag{3.2}$$

where $\|\cdot\|_1$ is the ℓ_1-norm. Let $\mathbf{y}_{\theta_0}$ denote $\mathbf{y}_\theta$'s frontal image, and $\hat{\boldsymbol{\gamma}}_0$ be the solution to (3.2) with $\mathbf{y}_\theta$ and $\mathbf{B}_\theta$ replaced by $\mathbf{y}_{\theta_0}$ and $\mathbf{B}_{\theta_0}$, respectively. We can relate $\mathbf{y}_\theta$ and $\mathbf{y}_{\theta_0}$ to $\mathbf{B}_\theta$ and $\mathbf{B}_{\theta_0}$ by

$$\mathbf{y}_\theta = \mathbf{B}_\theta \hat{\boldsymbol{\gamma}} + \mathbf{e}, \tag{3.3}$$

$$\mathbf{y}_{\theta_0} = \mathbf{B}_{\theta_0} \hat{\boldsymbol{\gamma}}_0 + \mathbf{e}_0, \tag{3.4}$$

where $\mathbf{e}$ and $\mathbf{e}_0$ are error terms. Now, consider the two re-projections: $\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0$, and $\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}$. In the following, we show the distance $\|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}\|_2$ can be made small if $\mathbf{B}_\theta\hat{\boldsymbol{\gamma}}$ and $\mathbf{B}_{\theta_0}\hat{\gamma}_0$ can well approximate $\mathbf{y}_\theta$ and $\mathbf{y}_{\theta_0}$, respectively.

Rotating an input image $\mathbf{y}_\theta$ by δ according to the 3D face model can be approximated through the completion of the following two steps: (1) Perform δ-rotation on the harmonic basis of $\mathbf{y}_\theta$ [74]. (2) Apply spatial translation and interpolation according to the 3D δ-rotation matrix. In step (1), the harmonic basis of $\mathbf{y}_\theta$ is changed in accordance with the azimuth, elevation and z axis rotations [74]. We denote the resulting intermediate image vector by $\tilde{\mathbf{y}}_{\theta+\delta}$. In step (2), a spatial translation and interpolation operator $\mathcal{R}_\delta(\cdot)$ determined by the 3D rotation matrix, is applied on $\tilde{\mathbf{y}}_{\theta+\delta}$ to obtain the output image $\mathbf{y}_{\theta+\delta}$. It can be shown that

$$\mathbf{y}_{\theta+\delta} \approx \mathbf{B}_{\theta+\delta}\hat{\boldsymbol{\gamma}} + \mathcal{R}_\delta(\mathbf{e}), \tag{3.5}$$

$$\|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}\|_2 \approx \|\mathcal{R}_{-\theta}(\mathbf{e}) - \mathbf{e}_0\|_2 \leq \|\mathcal{R}_{-\theta}(\mathbf{e})\|_2 + \|\mathbf{e}_0\|_2. \tag{3.6}$$

In Appendix B, we present more details on the harmonic basis rotation, as well as the derivations for (3.5) and (3.6).

Based on (3.6), $\|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}\|_2$ can be made small if the errors $\|\mathbf{e}\|_2$ and $\|\mathbf{e}_0\|_2$ are both small. Even if $\|\mathbf{e}\|_2$ and $\|\mathbf{e}_0\|_2$ cannot be ignored, since $\mathbf{e}_0$ is the reconstruction error of $\mathbf{y}_{\theta_0}$ under frontal pose θ_0, and $\mathbf{e}$ is the reconstruction error of $\mathbf{y}_\theta$ under pose θ, $\mathbf{e}_0$ and $\mathcal{R}_{-\theta}(\mathbf{e})$ should stay close to each other whenever θ is not large. In this case, $\|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}\|_2$ remains close to zero, and hence in general less than $\|\mathbf{y}_{\theta_0} - \mathbf{y}_\theta\|_2$, the distance between two original images under different poses.

Based on this reasoning, we define the SRA image of $\mathbf{y}_\theta$, denoted by $\mathbf{y}_{\theta,\mathrm{SRA}}$, as the re-projection on the frontal reference set $\mathbf{B}_{\theta_0}$ using $\hat{\boldsymbol{\gamma}}$ in (3.2). It is a synthesized face image in the frontal pose:

$$\mathbf{y}_{\theta,\mathrm{SRA}} = \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}. \tag{3.7}$$

Similarly, the SRA image of $\mathbf{y}_{\theta_0}$, denoted by $\mathbf{y}_{\theta_0,\mathrm{SRA}}$, is obtained by replacing $\hat{\boldsymbol{\gamma}}$ with $\hat{\boldsymbol{\gamma}}_0$ in (3.7). The top part of Fig. 3.2 illustrates example reference sets in different poses created from the Vetter's 3D face model [73]. We assume that images in the frontal pose are initially available. Using the frontal images, we create synthetic variations and then images in different poses $\{\theta_l\}_{l=1}^L$, from which reference sets are constructed. Algorithm 5 describes the details for generating the reference sets, $\{\mathbf{B}_{\theta_l}\}_{l=1}^L$. The bottom part of Fig. 3.2 illustrates how the SRA feature of an input image is computed. With the pose estimate of the input image, we select the reference set whose pose is closest to the estimate among all available poses. The

coefficient vector is computed with the selected reference set using (3.2) and then mapped back to $\mathbf{B}_{\theta_0}$ in (3.7), where the projection onto $\mathbf{B}_{\theta_0}$ is computed as the output SRA image.

Algorithm 5: Generate reference sets for poses $\{\theta_l\}_{l=1}^{L}$.

Input: Properly cropped frontal face images from V subjects $\{\mathbf{b}_{\theta_0,0}^{[v]}\}_{v=1}^{V}$, a set of possible poses $\{\theta_l\}_{l=1}^{L}$, and Vetter's 3D face model [73].

Algorithm:

1. Apply predefined $(U-1)$ synthetic variations on each $\mathbf{b}_{\theta_0,0}^{[v]}$ to obtain $\{\mathbf{b}_{\theta_0,u}^{[v]}\}_{u=1}^{U-1},\ \forall v \in \{1,...,V\}$. Form $\mathbf{B}_{\theta_0}$ by concatenating $\mathbf{b}_{\theta_0,u}^{[v]}$'s.

2. Estimate the basis harmonics [75]. Repeat **3** and **4** $\forall\ v \in \{1,...,V\},\ u \in \{1,...,U-1\},\ l \in \{1,...,L\}$:

3. Let $\delta_l = \theta_l - \theta_0$. Given $\mathbf{b}_{\theta_0,u}^{[v]}$, rotate the basis harmonics and compute the intermediate image $\tilde{\mathbf{b}}_{\theta_l,u}^{[v]}$, where the illumination is changed accordingly with rotation δ_l [74].

4. Compute the 3D rotation matrix $\mathbf{R}_{\delta_l}$. Obtain the final rotated image $\mathbf{b}_{\theta_l,u}^{[v]}$ for each pixel using either direct mapping from the corresponding source pixel, or interpolation from neighboring pixels.

5. Collect $\{\mathbf{b}_{\theta_l,u}^{[v]}\}_{l=1}^{L}$'s and obtain $\{\mathbf{B}_{\theta_l}\}_{l=1}^{L}$.

Output: $\mathbf{B}_{\theta_0}$ and $\{\mathbf{B}_{\theta_l}\}_{l=1}^{L}$

3.1.3 Building video dictionaries and computing distances

In this section, we describe how the video dictionaries are built and used to compute distances. We refer to the video dictionaries proposed in [3] as the base video dictionaries, and the video dictionaries built using the SRA images as the SRA video dictionaries. The SRA approach extends the DFRV method [3] to effectively combine both base and SRA video dictionaries in such a way that base

video dictionaries are used only when there is a small difference in pose across the the probe and gallery videos, otherwise, the SRA video dictionaries are used to account for the large pose difference across the the probe and gallery videos.

In the DFRV method [3] presented in Chapter 2, given the gth video sequence in the training stage, the cropped face images extracted from all frames form a set denoted by $S^{(g)}$, and the video partition algorithm [3] to separate $S^{(p)}$ into K partitions is used. Let $\mathbf{G}_k^{(g)}$ denote the resulting gallery matrix from the kth partition, $\forall k = 1, ..., K$. In the SRA method, to further obtain the SRA images of $\mathbf{G}_k^{(g)}$, we assume that all images belonging to partition $\mathbf{G}_k^{(g)}$ are in close poses. Let $\hat{\theta}$ be the estimated pose of $\mathbf{G}_k^{(g)}$. Among all available $\mathbf{B}_\theta$'s, we choose $\mathbf{B}_{\bar{\theta}}$ such that $\bar{\theta}$ is the closest pose to $\hat{\theta}$ among the other poses in the reference sets. For each column in $\mathbf{G}_k^{(g)}$, we use (3.2) and (3.7) (with $\mathbf{B}_\theta$ replaced by $\mathbf{B}_{\bar{\theta}}$ accordingly) to compute its SRA image, and concatenate the columns of SRA images to form $\mathbf{G}_{k,\text{SRA}}^{(g)}$. Next, from $\mathbf{G}_k^{(g)}$ and $\mathbf{G}_{k,\text{SRA}}^{(g)}$, we use the K-SVD algorithm [62] to learn the partition-level sub-dictionaries $\mathbf{D}_{(g),k}$, $\mathbf{D}_{(g),k,\text{SRA}}$, $\forall k = 1, ..., K$. Then the base video dictionaries $\mathbf{D}_{(g)}$ [3], and SRA video dictionaries $\mathbf{D}_{(g),\text{SRA}}$ are constructed by concatenating the corresponding sub-dictionaries.

In the testing stage, we partition the pth probe video sequence denoted by $\mathbf{Q}^{(p)} = \bigcup_{k=1}^{K} \mathbf{Q}_k^{(p)}$, where $\mathbf{Q}_k^{(p)} = [\mathbf{q}_{k,1}^{(p)} \ ... \ \mathbf{q}_{k,n_k}^{(p)}]$ as in [3], and then use (3.2) and (3.7) to compute the SRA partition $\mathbf{Q}_{k,\text{SRA}}^{(p)}, \forall k = 1, ..., K$. These partitions are collected as $\mathbf{Q}_{\text{SRA}}^{(p)}$.

Let $\mathbf{R}$ be the distance matrix with entry $\mathbf{R}^{(p,g)}$ denoting the residual between the pth probe video and the gth gallery video. The proposed method to compute

$\mathbf{R}^{(p,g)}$ requires using SRA images and SRA video dictionaries only when a gallery video dictionary and partitions of a probe video appear in very different poses. In particular, when poses of $\mathbf{Q}_k^{(p)}$ and $\mathbf{D}_{(g)}$ are close, the corresponding $\mathbf{R}^{(p,g)}$ remains computed from $\mathbf{Q}_k^{(p)}$ and base $\mathbf{D}_{(g)}$ [3]. On the other hand, when their poses are very different, $\mathbf{R}^{(p,g)}$ is computed using their $\mathbf{Q}_{\text{SRA}}^{(p)}$ and $\mathbf{D}_{(g),\text{SRA}}$. Therefore,

$$\mathbf{R}^{(p,g)} = \min_{k\in\{1,\ldots,K\}} \mathbf{R}_k^{(p,g)}, \tag{3.8}$$

where $\mathbf{R}_k^{(p,g)} =$

$$\begin{cases} \min\limits_l \|\mathbf{q}_{k,l}^{(p)} - \mathbf{D}_{(g)}\mathbf{D}_{(g)}^{\dagger}\mathbf{q}_{k,l}^{(p)}\|_2, & \text{if } \eta(\mathbf{Q}_k^{(p)}, \mathbf{D}_{(g)}) = 1, \\ \min\limits_l \|\mathbf{q}_{k,l,\text{SRA}}^{(p)} - \mathbf{D}_{(g),\text{SRA}}\mathbf{D}_{(g),\text{SRA}}^{\dagger}\mathbf{q}_{k,l,\text{SRA}}^{(p)}\|_2, & \text{else.} \end{cases}$$

In (3.8), $\mathbf{D}^{\dagger}$ denotes the pseudo-inverse of $\mathbf{D}$, and $\eta(\mathbf{Q}_k^{(p)}, \mathbf{D}_{(g)})$ is an indicator function such that $\eta = 1$ if $\mathbf{Q}_k^{(p)}$ and $\mathbf{D}_{(g)}$ are in close poses, and $\eta = 0$ otherwise. Fig. 3.3(a) and (b) are illustrations of building base video dictionaries and SRA video dictionaries, and constructing the distance matrix, respectively. Algorithm 6 summarizes the SRA method.

3.1.4 Dictionary Rotation

The second method for pose alignment is an adaptation from the SRA algorithm, which rotates the video dictionary atoms in both their harmonic basis and 3D geometry. In other words, it performs 3D rotation on atoms of video dictionaries to match the pose prior to recognition. We refer to this method as Dictionary Rotation (DR). We first obtain the pose estimate for the kth partition of the pth probe video $\mathbf{Q}_k^{(p)}$, and then use steps $2\sim4$ of Algorithm 5 to rotate each column of $\mathbf{D}_{(g)}$ to the

Algorithm 6: The SRA algorithm.

Training:

1. Given a sequence - the gth video, extract all the frames from it. Detect and crop face regions to form a set $S^{(g)}$.

2. Separate $S^{(g)}$ into K partitions. Augment each partition by adding synthetic images and obtain the resulting augmented gallery matrix from the kth partition, $\mathbf{G}_k^{(g)}, \forall k = 1, ..., K$.

3. For each column in $\mathbf{G}_k^{(g)}$, use (3.2) and (3.7) to compute its SRA image. The resulting $\mathbf{G}_{k,\text{SRA}}^{(g)}$ is formed by concatenating columns of the corresponding SRA images.

4. From $\mathbf{G}_k^{(g)}$ and $\mathbf{G}_{k,\text{SRA}}^{(g)}$, use the K-SVD algorithm to learn the corresponding partition-level sub-dictionaries $\mathbf{D}_{(p),k}$, $\mathbf{D}_{(p),k,\text{SRA}}$, $\forall k = 1, ..., K$, and video dictionaries $\mathbf{D}_{(g)}$, $\mathbf{D}_{(g),\text{SRA}}$.

Testing:

1. Partition the pth probe video sequence $\mathbf{Q}^{(p)} = \bigcup_{k=1}^{K} \mathbf{Q}_k^{(p)}$, where $\mathbf{Q}_k^{(p)} = [\mathbf{q}_{k,1}^{(p)}\ \mathbf{q}_{k,2}^{(p)}\ ...\ \mathbf{q}_{k,n_k}^{(p)}]$ as in [3].

2. Use (3.2) and (3.7) to compute the SRA images of $\mathbf{Q}_k^{(m)}$, denoted by $\mathbf{Q}_{k,\text{SRA}}^{(p)}$. Then obtain the corresponding $\mathbf{Q}_{\text{SRA}}^{(p)}$.

3. Using $\mathbf{D}_{(g)}$, $\mathbf{D}_{(g),\text{SRA}}$, $\mathbf{Q}^{(p)}$ and $\mathbf{Q}_{\text{SRA}}^{(p)}$, construct the distance matrix $\mathbf{R}^{(p,g)}$ by (3.8).

pose estimate[4]. Let the resulting video dictionary be denoted by $\mathbf{D}^{(p),k}_{(g),DR}$. The same steps are repeated for all K partitions of $\mathbf{Q}^{(p)}$. Next, we use (3.8) to find $\mathbf{R}^{(p,g)}$, with $\mathbf{R}^{(p,g)}_k$ replaced by

$$\mathbf{R}^{(p,g)}_k = \min_l \|\mathbf{q}^{(p)}_{k,l} - \mathbf{D}^{(p),k}_{(g),DR}\left(\mathbf{D}^{(p),k}_{(g),DR}\right)^{\dagger}\mathbf{q}^{(p)}_{k,l}\|_2. \tag{3.9}$$

Prior to computing the distance, the pose alignment is done by directly rotating dictionary atoms to the estimated pose from each partition of a given probe video. The underlying motivation of this method is based on the fact that if a probe image $\mathbf{q}_\theta$ is represented as a linear combination of video dictionary atoms plus an error term

$$\mathbf{q}_\theta = \mathbf{D}_{(g)}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{3.10}$$

then from (3.3) and (3.5), the δ-rotated copy of $\mathbf{q}_\theta$ is

$$\mathbf{q}_{\theta+\delta} \approx \mathcal{R}_\delta(\check{\mathbf{D}}_{(g)})\boldsymbol{\beta} + \mathcal{R}_\delta(\boldsymbol{\epsilon}), \tag{3.11}$$

where $\check{\mathbf{D}}_{(g)}$ is $\mathbf{D}_{(g)}$ with δ-rotated harmonic basis, and $\mathcal{R}_\delta(\check{\mathbf{D}}_{(g)})$ is the δ-rotated $\mathbf{D}_{(g)}$. Each column of $\mathcal{R}_\delta(\check{\mathbf{D}}_{(g)})$ is a δ-rotated version of the corresponding column of $\mathbf{D}_{(g)}$. Therefore, δ-rotation of an image can be approximated by a linear combination of the corresponding δ-rotated dictionary atoms weighed by the same coefficient vector.

3.2 Pose Estimation

Various geometric approaches have been proposed in the literature for pose estimation using facial landmarks, such as the location of the eyes, nose, and

[4]Here, the frontal pose θ_0 in steps 2∼4 of Algorithm 5 is replaced by a general pose θ.

mouth [76], [77], [78], [79], [74]. Unlike constrained still images, face images extracted from unconstrained videos may suffer from low resolution or bad illumination. This makes automatic detection of landmarks much more difficult. We present a semi-automatic method for estimating poses in videos. First, we select face images of V_1 out of V subjects from the reference set with various poses. For each face image, we manually locate T landmarks. Let $\mathbf{l}_{t,\theta}^{v}$ be the resulting two dimensional vector representing the spatial location the tth landmark of subject v under pose θ. Let s_k^* be the exemplar of the kth partition obtained through the video sequence partition algorithm presented in [3], with the corresponding vector of the tth landmark denoted by $\mathbf{l}_t(s_k^*)$. For the given test video, we assume that all the images in a partition have approximately similar pose. Due to the fact that a video may contain a large number of frames, instead of locating landmarks for all the frames, we manually locate the landmarks on the K exemplar images only. The pose estimate of each exemplar is used to represent the pose of the corresponding partition. Using nearest neighbor criterion, we select the pose with landmark vectors $\{\mathbf{l}_{t,\theta}^{v}\}_{v=1,t=1}^{V_1,T}$ that gives the minimum average distance to $\{\mathbf{l}_t(s_k^*)\}_{t=1}^{T}$ as the pose estimate $\hat{\theta}$ of the partition. In other words,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{\sum_{v=1}^{V_1} \sum_{t=1}^{T} \|\mathbf{l}_{t,\theta}^{v} - \mathbf{l}_t(s_k^*)\|_2}{V_1 T}. \tag{3.12}$$

For images from unconstrained videos, however, sometimes even manually locating the landmarks is impossible due to the image's extremely poor resolution and illumination. In this case, we simply examine the face image and roughly estimate the pose directly.

3.3 Experimental results

We evaluate the proposed methods on three challenging datasets: the video challenge of the Face and Ocular Challenge Series (FOCS) [2], the Multiple Biometrics Grand Challenge (MBGC) [51], and the Human ID [80] datasets. For FOCS and MBGC datasets, we created the reference sets using the 3D face model [73] from 100 subjects in the Vetter's database. For the Human ID dataset, as it contains facial moving mug shot videos with face poses in θ_a ranging from $-90^\circ \sim 90^\circ$, we collected frames directly from these videos as reference sets. There is no overlap between subjects whose videos are used for reference sets and subjects whose videos are used as probe and gallery videos for testing.

3.3.1 FOCS UT-Dallas Video

The FOCS UT-Dallas Video is described in Section 2.4.2. We resized the faces to 20×20 pixels and conducted leave-one-out tests on 3 subsets: $S2$ (294 subjects, 1014 videos), $S4$ (183 subjects, 782 videos), and $S6$ (19 subjects, 126 videos)[5]. Unlike DFRV [3] presented in Chapter 2, where only walking videos were chosen for identification tests, we conduct experiments across *both* walking and activity videos. Table 3.1 shows identification rates. Our SRA and DR methods perform better than state-of-the-art algorithms including SANP [13] and DFRV [3]. The SRA approach is better than the DR method on 2 of 3 cases, and tied on 1 case. Statistics-based approaches including PM, KD and WG [1], [12], no longer give satisfactory results.

[5]We refer to Sn as subjects that have at least n video sequences.

The 'no DL' is a baseline method that represents each video partition directly as a set of basis atoms without dictionary learning.

UT-Dallas all (W & A) videos	PM [1], [12]	KD [1], [12]	WGCP [1]	SANP [13]	baseline (no DL)	DFRV [3]	DR	SRA
$S2$	17.46	14.89	8.48	25.54	22.98	23.67	**28.40**	**28.40**
$S4$	24.30	20.33	11.64	33.38	29.80	31.59	36.70	**38.36**
$S6$	47.62	43.65	30.16	51.59	50.79	55.56	59.52	**62.70**
Average	29.79	26.29	16.76	36.84	34.52	36.94	41.54	**43.15**

Table 3.1: Identification rates of leave-one-out testing experiments on the FOCS UT-Dallas (both walking and activity) videos.

Fig. 3.4 compares ROC curves of DFRV [3] and the SRA method. As shown, while there is no difference between both methods under "W vs W" (walking vs walking[6]) and "A vs A" (activity vs activity) verification protocols, the SRA method outperforms DFRV under "A vs W" (activity vs walking) and "W vs A" (walking vs activity) protocols[7]. This is explained by the fact that the SRA method takes the same distances as DFRV [3] when matching in-pose videos ("W vs W" and "A vs A"), while it uses pose aligned feature (i.e. SRA image) to measure the distance between videos across different poses ("A vs W" and "W vs A"). Based on Table 3.1

[6]This means walking videos as probe, and walking videos as gallery. "A vs A", "A vs W" and "W vs A" can be explained in the same manner.

[7]This is true when the false acceptance rate (FAR) is less than 0.5, which covers the upper limit of FAR for most applications.

and Fig. 3.4, the SRA outperforms other methods through the use of pose aligned feature in matching out-of-pose videos.

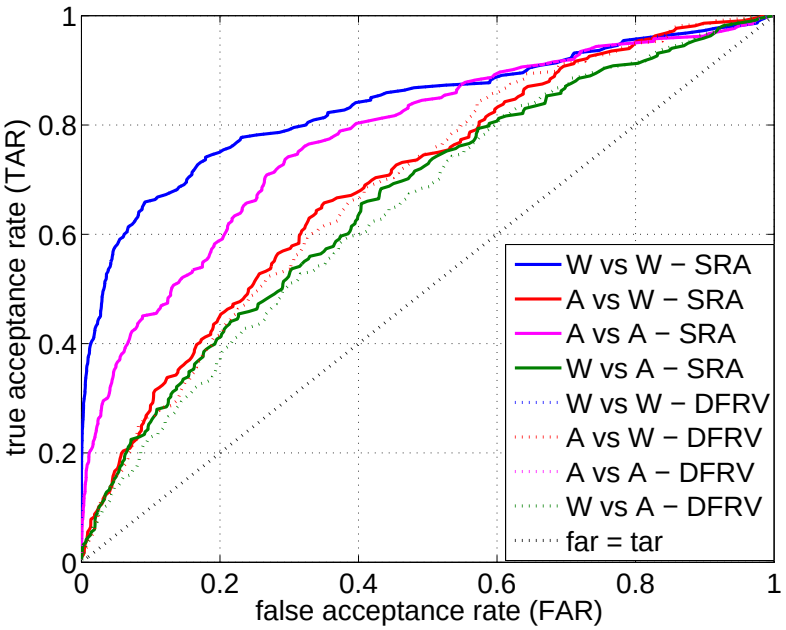


Figure 3.4: ROC curves of the DFRV and SRA methods on the FOCS UT-Dallas videos. The SRA method takes the same distances as DFRV [3] when matching in-pose videos ("W vs W" and "A vs A"), and uses the pose aligned feature to measure distances between out-of-pose videos ("A vs W" and "W vs A"). As shown, it outperforms DFRV in out-of-pose scenarios.

3.3.2 MBGC Video version 1

The MBGC Video version 1 dataset is described in Section 2.4.1. In the experiments, each cropped face image was resized to 20×20 pixels. We conducted leave-one-out tests on 3 subsets: $S2$ (145 subjects, 769 videos), $S5$ (55 subjects, 426 videos), and $S8$ (48 subjects, 384 videos). Similar to section 3.3.1, the identification

experiments are performed across *both* walking and activity videos. Table 3.2 shows identification results. As shown, the SRA and DR method obtained improved identification rates over comparable algorithms. In addition, the MBGC dataset contains videos in both HD and SD formats for the same subject recorded in the same day, while videos of each subject in the FOCS dataset were recorded on different days, during which the subjects may have changed style in their hair, facial hair, expression, pose and illumination. This explains the overall much higher recognition rates on the MBGC dataset than those on the FOCS dataset.

MBGC v1 all (W & A) videos	PM [1], [12]	KD [1], [12]	WGCP [1]	SANP [13]	baseline (no DL)	DFRV [3]	DR	SRA
$S2$	41.48	31.86	14.17	68.79	62.55	69.70	80.88	**82.18**
$S5$	43.90	35.68	17.84	69.25	63.38	70.66	80.28	**81.22**
$S8$	44.53	35.94	18.49	71.09	64.32	71.35	81.51	**82.55**
Average	43.30	34.49	16.83	69.71	63.42	70.57	80.89	**81.98**

Table 3.2: Identification rates of leave-one-out testing experiments on the MBGC v1 (both walking and activity) videos.

3.3.3 Human ID database

The Human ID database [80] contains videos of human faces and people, which is useful for testing algorithms for face and person recognition. For each selected subject, there are videos of moving facial mug shots, facial speech, dynamic facial

expressions, walking on the same day, and walking on a different day. A complete set of videos is available for 284 subjects. We selected videos of 60 out of 284 subjects from the Human ID database. The face region was properly cropped and resized to 30×24 pixels. The first three rows of Fig. 3.5 show cropped face images of one subject from its moving facial mug shot, facial speech and dynamic facial expression videos, respectively. The last row of Fig. 3.5 shows the walking video frames of the same subject recorded on the same day (left) as the first three videos, and on a different day (right). The facial mug shot video contains poses from the left side pose to the right side pose (θ_a from $-90^\circ \sim 90^\circ$ wrt the y axis), incremented in a step of 22.5°. As the facial moving mug shot videos already contain face images in variant poses, we collected frames from these facial moving mug shot videos of 30 subjects as reference sets. Videos of the remaining 30 subjects were used for testing. In particular, each gallery video is a trimmed facial moving mug shot video that contains face images with poses in θ_a ranging from about $0^\circ \sim 90^\circ$, while probe videos of the same subject contains facial speech, expression, walking (on the same day) and walking (on a different day) videos. Cropped face images from the probe videos are almost always frontal.

Table 3.3 shows recognition rates on the four types of probe videos. Both the DFRV [3] and DR obtained the best average results. While the SRA ranks the second, it far outperforms the remaining state-of-the-art algorithms. As each gallery video may contain few face images in the frontal pose ($\theta_a = 0^\circ$), the in-pose video-to-video matching favors the DFRV[8] and hence the difference between DFRV and

[8]Here we refer to the minimum residual based video-to-video matching method of the DFRV.

Figure 3.5: Example frames from the Human ID database. Videos include: moving facial mug shots (1st row), facial speech (2nd row), dynamic facial expression (3rd row), walking on the same day (4th row left), and walking on a different day (4th row right).

DR is not obvious. For SRA, the images used to construct the reference sets were chosen from a fix set of indices for all 30 selected subjects. When recorded, however, the timing of head turning may vary among the different subjects. Therefore, unlike Vetter's face reference sets, poses and their variations were in fact not aligned across different subjects. The resulting projection error may make the SRA distances even greater than the original out-of-pose distances for the same subject. This may be the main reason why the SRA did not obtain the highest rates, and it shows the importance of the choice of reference sets.

3.4 Summary

We extended the existing unconstrained video-to-video face recognition frameworks to the one that explicitly addresses the challenge of matching probe and gallery videos in different poses. The proposed approaches include a sparse representation-

Human ID video types	PM [1], [12]	KD [1], [12]	WGCP [1]	SANP [13]	baseline (no DL)	DFRV [3]	DR	SRA
Facial speech	40.00	33.33	20.00	43.33	36.67	63.33	**73.33**	63.33
Facial expression	33.33	13.33	10.00	36.67	26.67	**56.67**	53.33	53.33
Walking (same day)	3.33	3.33	6.67	**20.00**	16.67	**20.00**	13.33	**20.00**
Walking (different day)	10.00	6.67	6.67	6.67	10.00	**13.33**	**13.33**	10.00
Average	21.67	14.17	10.84	26.67	22.50	**38.33**	**38.33**	36.67

Table 3.3: Identification rates of matching 4 types of probe videos with the moving facial mug shot gallery videos on the Human ID database.

based alignment method that generates pose aligned features through pre-designed reference sets under a sparsity constraint, and a dictionary rotation method that directly rotates gallery video dictionary atoms in both their harmonic basis and 3D geometry to match the poses of the probe videos. Through extensive experiments on three challenging unconstrained video datasets across poses, illuminations and facial changes, the proposed SRA and DR have been shown to achieve better recognition performances than several state-of-the-art methods.

Chapter 4: In-plane Rotation and Scale Invariant Clustering using Dictionaries

Invariance to rotation and scale are desirable in many practical applications. One important application is image classification and retrieval where one wants to classify or retrieve images having the same content but at different orientation and scale. For instance, in content based image retrieval (CBIR), images are retrieved from a database using features that best describe the orientation and scale of objects in the query image. Gabor filters have been used to extract features for retrieval and classification [81]. However, the chosen directions of Gabor filters may not correspond to the orientation of the content in the query image. Hence, a feature extraction method that is independent of orientation and scale in the image is desirable [82]. Wavelet-based methods have been proposed to achieve rotation invariance for image classification and retrieval [83], [84]. There have also been methods proposed to learn invariant dictionaries in the image domain [41], [85], [86]. Recently, a shift, scale and rotation invariant dictionary learning method for multivariate signals was proposed in [87]. Hierarchical dictionary learning methods for invariant classification have also been proposed in [88] and [44]. These methods learn a dictionary in a log-polar domain to be invariant to scale and rotation. A cellular neural

network-based method for rotation invariant texture has also been proposed in [89].

Numerous descriptors have been proposed in the literature that are invariant to image transformations [90], [91], [92], [93], [94]. A shape matching approach based on correspondences between points on two shapes was proposed in [91]. This shape context descriptor essentially estimates the shape similarity and solves the correspondence problem. A shock graph-based feature extraction method that uses object silhouettes was proposed in [92]. The length of the shortest path within the shape boundary (called inner-distance) was used to build shape descriptors in [93]. These descriptors were shown to be robust to articulation in complicated shapes. In [94], a feature extraction method based on features that characterize the geometric relationships between each pair of images was proposed. This method was shown to be invariant to articulations and rigid transforms. Some of these methods are only shape-based and require the extraction of shape contour. These methods do not perform well on non-shape images such as textures.

In this chapter, we present an in-plane rotation and scale invariant clustering approach (box 1∼3 in Fig. 4.1), which extends the dictionary learning and sparse representation framework (box 4, 5 in Fig. 4.1) for clustering and retrieval of images. Fig. 4.1 illustrates the overview of the proposed approach. Given a database of images $\{\mathbf{x}_j\}_{j=1}^{J}$ and the number of clusters K, our method uses the Radon transform to find scale and rotation-invariant features. It then uses sparse representation methods to simultaneously cluster the data and learn dictionaries for each cluster. One of the main features of our method is that it is effective for shape-based and certain texture-based images. We demonstrate the effectiveness of our approach in image

retrieval experiments, where we report significant improvements in performance.

Key contributions of this work are:

1. We propose a rotation invariant clustering algorithm suitable content based image retrieval (CBIR).

2. We propose a normalization method validated by a mathematical proof, to achieve scale invariance in the Radon domain.

3. We propose a method to obtain initial classes and class dictionaries in a deterministic way to improve the clustering performance.

4. We demonstrate by experiments on shape-based and texture-based datasets the effectiveness and performance improvements of our approach compared to other Gabor-based and shape-based methods.

The organization of the chapter is as follows. A method based on scale and rotation invariant features that are extracted using the Radon transform is detailed in Section 4.1. Our simultaneous clustering and dictionary learning method is described in Section 4.2. Experimental results are presented in Section 4.3. In Section 4.4, we conclude the chapter with a brief summary and future research directions.

4.1 Radon-based rotation and scale invariance

In this section, we show how the Radon transform is used to achieve in-plane rotation and scale invariance (box $1 \sim 3$ in Fig. 4.1).

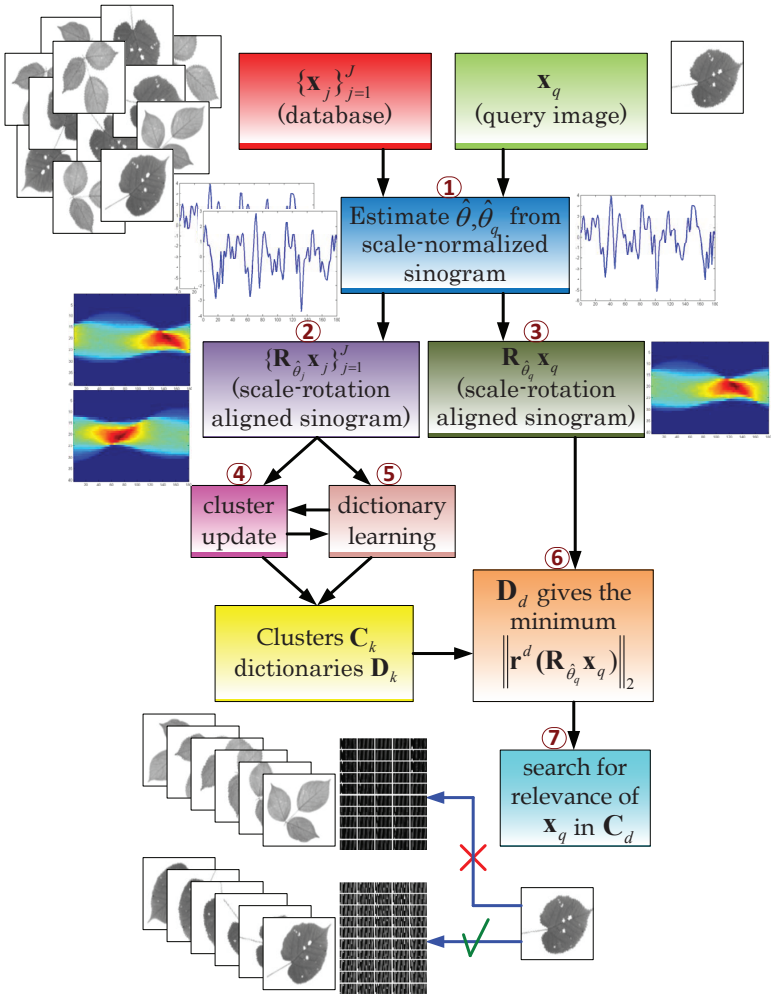
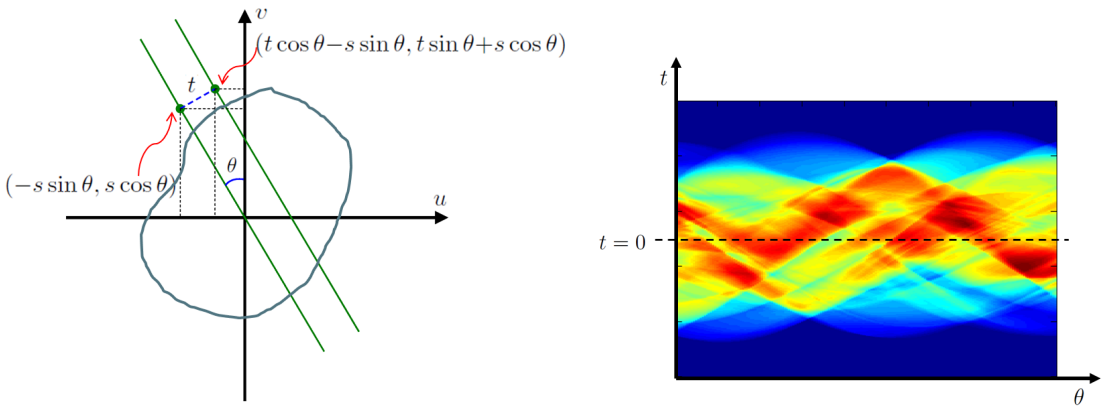


Figure 4.1: Overview of the proposed simultaneous clustering and dictionary learning method.

4.1.1 Estimating the rotation present in an image



(a) (b)

Figure 4.2: (a) Illustration of how the Radon transform is calculated. Given any point (u, v) in the image domain, we can express u and v as: $u = -s\sin\theta, v = s\cos\theta$ for some s and θ, where s is the distance between (u, v) and the origin; and θ is the angle between the positive vertical axis direction and the line passing through (u, v) and the origin. As indicated, a t-translated point from (u, v) is located at $(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)$. t is the distance between the line that passes through (u, v) and the origin, and the parallel line that passes through $(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)$. (b) In practice, the Radon transform of an image is represented as a matrix called sinogram, where the column indices correspond to discrete values of θ and row indices correspond to discrete values of t. θ and t are the two continuous variables of $R_\theta x(t)$ given in (4.1).

The Radon transform of a two variable function x is defined as

$$R_\theta x(t) = \int_{-\infty}^{\infty} x(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds, \tag{4.1}$$

where $(t, \theta) \in (-\infty, \infty) \times [0, \pi)$. Fig. 4.2(a) illustrates how the Radon transform is calculated. We use (4.1) to compute the value at any given point (θ, t) in the Radon domain by integrating along the line: $(u, v) = (t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta), \forall s \in \mathbb{R}$. If $\tilde{x}$ is a rotated copy of x by an angle $\hat{\theta}$, then a simple proof shows that their Radon transforms are related as

$$R_\theta \tilde{x}(t) = R_{\theta+\hat{\theta}} x(t), \quad \forall t, \theta. \tag{4.2}$$

For directional texture images, the principal orientation is roughly defined as the direction where there are more straight lines. The Radon transform can be used to detect linear trends in images. For general images, the principal orientation may be taken as the direction along which the Radon transform has the maximum variability. Let $\sigma_\theta \triangleq \mathbf{Var}_t[R_\theta x(t)]$ denote the variance with respect to t of the Radon transform at θ. In [95], σ_θ was found to be useful in estimating the principal orientation in an image. An important observation was that the Radon transform along $\hat{\theta}$ has larger variations with respect to t and hence the variance $\sigma_{\hat{\theta}}$ is a local maximum along the θ axis. Based on the observation, one can estimate $\hat{\theta}$ of a given image $\tilde{x}$ from the following formula[1]

$$\hat{\theta} = \arg\min_\theta \left(\frac{d^2 \sigma_\theta}{d\theta^2} \right). \tag{4.3}$$

The global minimum of the second derivative of σ_θ is computed in order to locate the angle at which the change rate of the first derivative of σ_θ is the maximum, which represents the maximum number of line trends (i.e., along the principal orientation

[1]We apply this approach not only to directional texture images, but also to other isotropic textures (any direction is the principal orientation) and shape-based images.

where the local maximum is the narrowest in shape, as indicated by Figs. 1(b) and (d) in [95]). Once the orientation is estimated, this estimate and (4.2) can be used to align the rotation in the Radon domain. Hence, we achieve rotation invariance.

In practice, the Radon transform of an image is represented as a matrix, called a *sinogram*. Fig. 4.2(b) gives the illustration of a sinogram. In a sinogram, column indices correspond to discrete values of θ, while row indices correspond to discrete values of t, where θ and t are the two continuous variables of $R_\theta x(t)$ given in (4.1). σ_θ is the variance computed from values of the column that corresponds to angle θ.

Fig. 4.3 illustrates how the sinogram is used to estimate the angle present in an image[2]. The second image shown on the first row of Fig. 4.3 is a rotated copy of the first image by 30°. The plots on the second row are the second derivatives of variances σ_θ (vertical axis) versus θ (horizontal axis) of their sinograms, where σ_θ is the variance over all entries in the column that corresponds to θ. It may be noted that the difference between the points of global minima of both curves is 30°, coinciding with the rotation present in the second image. Consequently, the estimate presented in (4.3) is useful for estimating the presence of rotation in the images.

[2]`http://www.flowersofpictures.com/wp-content/uploads/2011/07/Lily-Flowers.jpg`

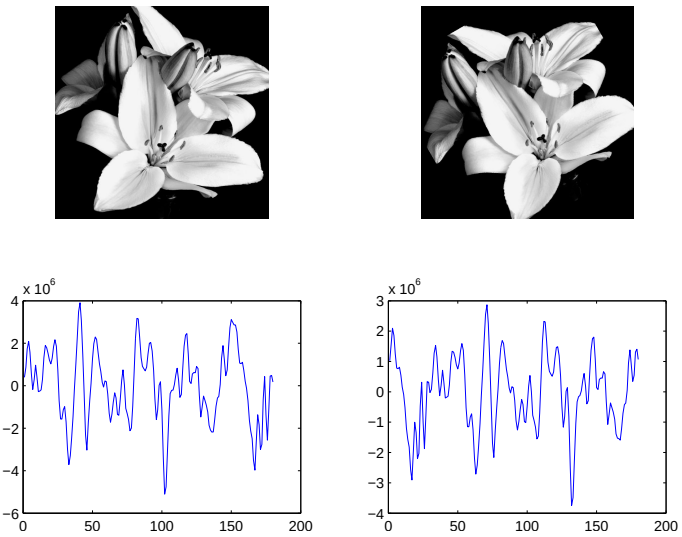


Figure 4.3: For the rotated images present on the first row, the plots on the second row show their $\frac{d^2\sigma_\theta}{d\theta^2}$ (along the vertical axis) versus θ (along the horizontal axis). The second row plots indicate that the difference between the points of global minimum of both curves preserves the rotation present in the second image.

4.1.2 Scale invariance

Let $\bar{x}$ be a scaled copy of x with the scaling factor ξ such that $\bar{x}(u,v) = x(\xi u, \xi v)$. We can relate their Radon transforms as follows:

$$\begin{aligned} R_\theta \bar{x}(t) &= \int_{-\infty}^{\infty} \bar{x}(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta) ds \\ &= \int_{-\infty}^{\infty} x(\xi t\cos\theta - \xi s\sin\theta, \xi t\sin\theta + \xi s\cos\theta) ds \\ &= \frac{1}{\xi} R_\theta x(\xi t). \end{aligned}$$

From the above equations, size scaling in the image domain results in scaling and normalization of the Radon transform. From this observation, scale invariance can be achieved through the following normalization in the Radon domain:

$$\frac{1}{M_{\bar{x}}} R_\theta \bar{x}\left(T_{\bar{x}} t\right), \tag{4.4}$$

where

$$M_{\bar{x}} \triangleq \max_{t,\theta} |R_\theta \bar{x}(t)|,$$

and

$$T_{\bar{x}} = \inf\{T | R_\theta \bar{x}(t) = 0, \forall |t| > T, \theta \in [0, 2\pi)\}.$$

Based on this formulation, one can derive the following result:

Proposition 4.1.1. *Let $\mathcal{Y}$ be a set of functions related by different scales. For any pair $\bar{x}, x \in \mathcal{Y}$ related by $\bar{x}(u,v) = x(\xi u, \xi v)$ where $\xi > 0$, the following holds*

$$\frac{1}{M_{\bar{x}}} R_\theta \bar{x}\left(T_{\bar{x}} t\right) = \frac{1}{M_x} R_\theta x\left(T_x t\right).$$

Proof. For any pair $\bar{x}, x \in \mathcal{Y}$ related by $\bar{x}(u,v) = x(\xi u, \xi v)$ where $\xi > 0$, we have

$$\begin{aligned} M_{\bar{x}} &= \max_{t,\theta} |R_\theta \bar{x}(t)| \\ &= \max_{t,\theta} |\frac{1}{\xi} R_\theta x(\xi t)| \\ &= \frac{1}{\xi} \max_{t,\theta} |R_\theta x(\xi t)| \\ &= \frac{1}{\xi} \max_{\tau,\theta} |R_\theta x(\tau)| (\text{let } \tau = \xi t) \\ &= \frac{1}{\xi} M_x, \end{aligned} \tag{4.5}$$

and

$$\begin{aligned} T_{\bar{x}} &= \inf\{T | R_\theta \bar{x}(t) = 0, \forall |t| > T, \theta \in [0, 2\pi)\} \\ &= \inf\{T | \frac{1}{\xi} R_\theta x(\xi t) = 0, \forall |t| > T, \theta \in [0, 2\pi)\} \\ &= \inf\{T | R_\theta x(\xi t) = 0, \forall |t| > T, \theta \in [0, 2\pi)\} \\ &= \inf\{T | R_\theta x(\xi t) = 0, \forall |\xi t| > \xi T, \theta \in [0, 2\pi)\} \\ &= \frac{1}{\xi} \inf\{\xi T | R_\theta x(\xi t) = 0, \forall |\xi t| > \xi T, \theta \in [0, 2\pi)\} \\ &= \frac{1}{\xi} \inf\{T' | R_\theta x(\tau) = 0, \forall |\tau| > T', \theta \in [0, 2\pi)\} \\ &\qquad\qquad (\text{let } T' = \xi T, \tau = \xi t) \\ &= \frac{1}{\xi} T_x. \end{aligned} \tag{4.6}$$

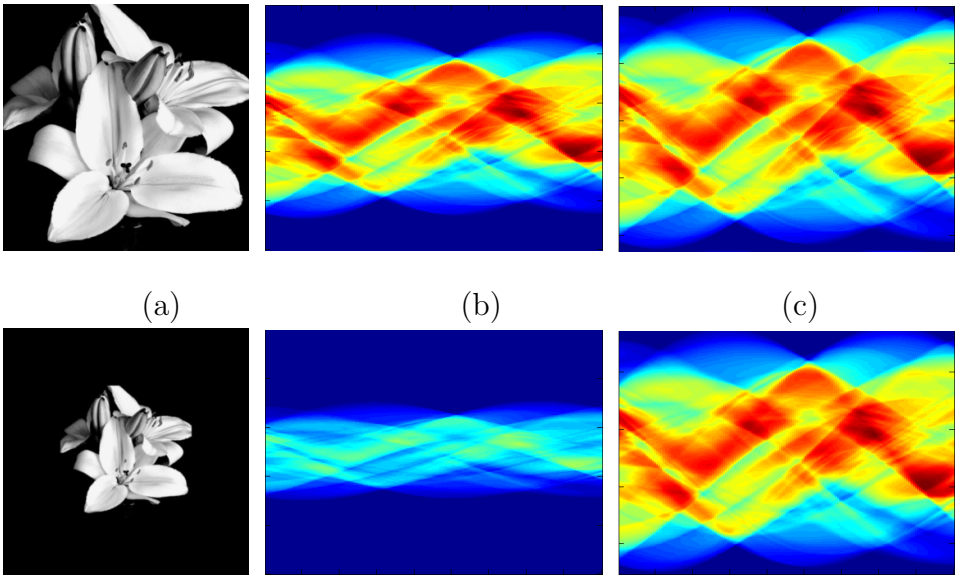
Therefore,

$$\begin{aligned} \frac{1}{M_{\bar{x}}} R_\theta \bar{x}\left(T_{\bar{x}} t\right) &= \frac{\xi}{M_x} R_\theta \bar{x}\left(\frac{T_x}{\xi} t\right) \\ &= \frac{\xi}{M_x \xi} R_\theta x\left(\frac{\xi T_x}{\xi} t\right) \\ &= \frac{1}{M_x} R_\theta x\left(T_x t\right). \end{aligned} \tag{4.7}$$

□

As the above result holds for any pair of $\bar{x}, x \in \mathcal{Y}$, we have shown the invariance of $\frac{1}{M_{\bar{x}}} R_\theta \bar{x}\,(T_{\bar{x}} t)$ over all $\bar{x} \in \mathcal{Y}$.

Fig. 4.4 illustrates an example of scale alignment in the Radon domain. Figs. 4.4(a) and (d) show the flower images with the same orientation but in different scales. The corresponding sinograms are shown in Figs. 4.4(b) and (e), respectively. The corresponding normalized sinograms obtained according to (4.4) are shown in Figs. 4.4(c) and (f), respectively. As can be seen from the figure, after the adjustment, the resulting sinograms are scale-aligned to each other.



(a) (b) (c)

(d) (e) (f)

Figure 4.4: Alignment of sinograms: (a) and (d) show the flowers images with different scales. (b) and (e) show their corresponding sinograms. Sinograms obtained after normalization are shown in (c) and (f). Note that after the adjustment, the resulting sinograms are closely scale-aligned to each other.

4.2 Simultaneous Clustering and Dictionary Learning

In this section, we present the proposed rotation and scale invariant clustering and dictionary learning framework. Our method learns dictionaries and clusters images in the Radon domain. Let $\{\mathbf{x}_j\}_{j=1}^{J}$ be the database of images represented as vectors and K be the number of clusters. $\mathbf{x}_j$ is a $Z_I \times 1$ column vector representing the jth image, where Z_I is the image size (i.e., product of width and height in pixels). Let $\mathbb{C} \triangleq \{\mathbb{C}_k\}_{k=1}^{K}$ denote the collection of K clusters such that $\mathbb{C}_k$ is a cluster that contains images belonging to the kth class. Given an image $\mathbf{x}_j$ and its estimated orientation $\hat{\theta}_j$ that is calculated from the discretized version of (4.3), we use a $Z_R \times 1$ column vector $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ to denote the (column-vectorized) vector version of the column-shifted, scale normalized sinogram, where Z_R is the sinogram size (i.e., product of width and height in pixels). This sinogram is obtained by left shifting columns of the scale normalized sinogram of $\mathbf{x}_j$ by $\hat{\theta}_j$. Let $\mathbf{C}_k$ be a $Z_R \times L_k$ matrix containing $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$s as columns, where L_k is the population size of $\mathbb{C}_k$ (i.e., number of images in $\mathbb{C}_k$). Let $\mathbf{D}_k$ be the class dictionary learned from $\mathbf{C}_k$ such that $\mathbf{D}_k$ is a $Z_R \times d_k$ matrix where d_k is the number of dictionary atoms. Define $\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_K]$ as the concatenation of class dictionaries. Note that $\mathbf{C}_k$ is the matrix with columns as vector forms of kth-class images' sinograms. Table 4.1 gives a summary of the notations.

Our objective is to simultaneously cluster the data into K groups and learn

Table 4.1: Summary of notations.

Variable	Definition	Dimensions	Domain
$\mathbf{x}_j$	a column representing the jth image	$Z_I \times 1$	image
$\mathbb{C}_k$	a cluster that contains $\mathbf{x}_j$s belonging to the kth class	$Z_I \times L_k$	image
$\mathbb{C}$	the collection all $\mathbb{C}_k$s	$Z_I \times (\sum_{k=1}^{K} L_k)$	image
$\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$	column form of the column-shifted, scale normalized sinogram	$Z_R \times 1$	Radon
$\mathbf{C}_k$	a matrix with $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$s as columns	$Z_R \times L_k$	Radon
$\mathbf{D}_k$	the class dictionary learned from $\mathbf{C}_k$	$Z_R \times d_k$	Radon
$\mathbf{D}$	the concatenation of class dictionaries $\mathbf{D}_k$s	$Z_R \times (\sum_{k=1}^{K} d_k)$	Radon

the best dictionaries for each cluster by solving the following optimization problem

$$\min_{\mathbb{C},\mathbf{D},\boldsymbol{\alpha}} \sum_{k=1}^{K} \sum_{\mathbf{x}\in\mathbb{C}_k} \min_{\theta} \left\{ \|\mathbf{R}_\theta \mathbf{x} - \mathbf{D}\delta_k(\boldsymbol{\alpha})\|_2^2 + \mu_1 \|\boldsymbol{\alpha}\|_1 + \mu_2 \left| \frac{d^2 \tilde{\sigma}_\theta}{d\theta^2} \right| \right\}, \tag{4.8}$$

where $\mu_1, \mu_2 > 0$, and $\|.\|_1$ denotes the ℓ_1 norm, $\boldsymbol{\alpha}$ is the representation vector and $\delta_k(\boldsymbol{\alpha})$ is the masked version of $\boldsymbol{\alpha}$ such that its only nonzero entries are those of $\boldsymbol{\alpha}$ that correspond to the kth dictionary. Here, $\tilde{\sigma}_\theta$ is the variance of the column corresponding to θ of the scale-normalized sinogram of $\mathbf{x}$. In other words, $\tilde{\sigma}_\theta$ is the variance of the first column of the sinogram that is column left-shifted (by θ) version of the scale normalized sinogram of $\mathbf{x}$. According to (4.3), if θ is not the principal orientation, then the last term in (4.8) can never be the minimum. Therefore, this term introduces a penalty due to rotation misalignment. It uses $\tilde{\sigma}_\theta$ to estimate the presence of rotation in images. Our approach for solving the above optimization problem essentially consists of two steps: cluster assignment and dictionary learning. Detailed descriptions of these two steps are given below.

4.2.1 Cluster assignment (box 4 in Fig. 4.1)

Given dictionary $\mathbf{D}^{(i)} = [\mathbf{D}_1^{(i)} \dots \mathbf{D}_K^{(i)}]$ at iteration i, we obtain the sparse representation of $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ in $\mathbf{D}^{(i)}$ by solving the following optimization problem:

$$\boldsymbol{\alpha}^j = \arg\min_{\omega} \|\boldsymbol{\omega}\|_1 \ \text{ subject to } \ \mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j = \mathbf{D}^{(i)}\boldsymbol{\omega}. \tag{4.9}$$

Several approaches have been suggested for solving (4.9) [96]. In our approach, we employ a highly efficient algorithm that is suitable for large-scale applications known as the spectral projected gradient (SPGL1) algorithm [97]. Once the sparse

coefficients are found, $\mathbf{x}_j$ is set to belong to cluster $\hat{k}$ if the coefficients corresponding to the $\hat{k}$th dictionary give the best representation of the sinogram $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ [9]. In other words, if

$$\hat{k} = \arg\min_k \|\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j - \mathbf{D}^{(i)}\delta_k(\boldsymbol{\alpha}^j)\|_2^2, \quad j = 1, \cdots, J, \tag{4.10}$$

then $\mathbf{x}_j$ is set to belong to $\mathbb{C}_{\hat{k}}^{(i)}$. The motivation for this consideration is that if $\mathbf{x}_j$ belongs to the kth cluster, then the dictionary corresponding to cluster k will represent $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ well.

4.2.2 Dictionary learning (box 5 in Fig. 4.1)

The K-SVD algorithm is a common dictionary learning algorithm [34]. Given clusters $\{\mathbb{C}_k^{(i)}\}_{k=1}^K$, we use the K-SVD algorithm to learn the dictionary $\mathbf{D}^{(i+1)} = [\mathbf{D}_1^{(i+1)} \dots \mathbf{D}_K^{(i+1)}]$. In this section, we detail the K-SVD principle, and then we show how the K-SVD principle is included in the dictionary learning step of our RSICD method.

4.2.2.1 The K-SVD algorithm

Given a set of input examples (in a column-vectorized form) $\{\mathbf{y}_l^k\}_{l=1}^{n_k}$ belonging to the kth class, the K-SVD algorithm finds a dictionary $\hat{\mathbf{B}}_k$ that provides the best representation for each example in this set by solving the following optimization

problem:

$$(\hat{\mathbf{B}}_k, \hat{\mathbf{\Lambda}}_k) = \arg\min_{\mathbf{B}_k, \mathbf{\Lambda}_k} \|\mathbf{Y}_k - \mathbf{B}_k\mathbf{\Lambda}_k\|_F^2, \text{ s.t. } \|\boldsymbol{\lambda}_{k,l}\|_0 \leq T_0,$$
$$\forall l \in \{1, ..., n_k\}, \forall k \in \{1, ..., K\}, \tag{4.11}$$

where $\boldsymbol{\lambda}_{k,l}$ represents the l^{th} column of $\mathbf{\Lambda}_k$, $\mathbf{Y}_k$ is the matrix whose columns are $\mathbf{y}_l^k$s and T_0 is the sparsity parameter. Here, the Frobenius norm is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{ij}^2}$ and the norm $\|\boldsymbol{\lambda}\|_0$ counts the number of non-zero elements in $\boldsymbol{\lambda}$. The K-SVD algorithm alternates between sparse-coding and dictionary update steps.

In the sparse-coding step, $\mathbf{B}_k$ is fixed and the representation vectors $\boldsymbol{\lambda}_{k,l}$s are found for each example $\mathbf{y}_l^k$ by solving the following equation:

$$\min_{\boldsymbol{\lambda}_{k,l}} \|\mathbf{y}_l^k - \mathbf{B}_k\boldsymbol{\lambda}_{k,l}\|_2^2 \text{ such that } \|\boldsymbol{\lambda}_{k,l}\|_0 \leq T_0,$$
$$\forall l \in \{1, .., n_k\}, \forall k \in \{1, ..., K\}. \tag{4.12}$$

As solving (4.12) is NP-hard, approximate solutions are usually sought [98], [96]. Greedy pursuit algorithms such as matching pursuit and orthogonal matching pursuit [99] are often used to find the approximate solutions to the above sparse coding problem [100]. In the dictionary update step, the dictionary is updated atom-by-atom in an efficient way. The K-SVD algorithm has been observed to converge in a few iterations.

4.2.2.2 Learning $\mathbf{D}^{(i+1)}$

Having obtained clusters $\{\mathbb{C}_k^{(i)}\}_{k=1}^K$, we update the dictionaries $\mathbf{D}_k^{(i+1)}$ with the K-SVD algorithm. In particular, we find the best representation of the members in

$\mathbf{C}_k^{(i)}$ by solving the following optimization problem:

$$(\mathbf{D}_k^{(i+1)}, \mathbf{\Gamma}_k^{(i+1)}) = \arg\min_{\mathbf{D}_k, \mathbf{\Gamma_k}} \|\mathbf{C}_k^{(i)} - \mathbf{D}_k\mathbf{\Gamma}_k\|_F^2, \text{s.t.} \|\boldsymbol{\gamma}_l\|_0 \leq T_0,$$
$$\forall l \in \{1, ..., L_k\}, \forall k \in \{1, ..., K\}, \tag{4.13}$$

where $\mathbf{\Gamma}_k^{(i+1)}$ is a $d_k \times L_k$ coefficient matrix that contains $\boldsymbol{\gamma}_l$s as its columns. Here $\mathbf{C}_k^{(i)}$, L_k, $\mathbf{D}_k$ and $\mathbf{\Gamma}_k$ in (4.13) correspond to $\mathbf{Y}_k$, n_k, $\mathbf{B}_k$ and $\mathbf{\Lambda}_k$ in (4.11), respectively. Note that each $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ corresponds to a $\mathbf{y}_l^k$ for some k and l in (4.11).

4.2.3 RSICD algorithm

Our RSICD algorithm is an iterative approach, where there are global iterations and local iterations. Each global iteration consists of cluster assignment and dictionary learning. As the K-SVD algorithm is used for the dictionary learning step, this step further consists of local K-SVD iterations. In particular, given $\mathbf{C}_k^{(i)}$ in the beginning of the dictionary learning step in the ith global iteration, $\mathbf{D}_k$ in (4.13) is set by $\mathbf{D}_k^{(i)}$. After a few local K-SVD iterations, $\mathbf{D}^{(i+1)}$ is obtained as the updated dictionary for the next global iteration. We iteratively repeat the cluster assignment and dictionary learning steps till there is no significant change in $\{\mathbb{C}_k^{(i+1)}\}_{k=1}^K$.

Note that (4.9) is used in each global iteration under an error constraint to find the sparsest coefficients using the concatenation of all class dictionaries, while (4.12) is used in each local iteration under a sparsity constraint to find the coefficients that give the minimum representation error for each class dictionary.

4.2.4 Obtaining initial dictionaries

As one can see from the previous discussion, the performance of our algorithm depends on the choice of initial dictionaries. In this section, we describe a method for obtaining the initial dictionary $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)} \dots \mathbf{D}_K^{(0)}]$.

Let $L \triangleq \min_{k \in \{1,\dots K\}} L_k$ be the minimum cluster population size. To determine initial clusters $\{\mathbb{C}_1^{(0)}, \dots, \mathbb{C}_K^{(0)}\}$, we propose an approach that uses the Hamming distance between R_i and R_j for any pair $(i, j) \in \{1, 2, ..., J\}$, where R_j is the set that consists of $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ and its $L-1$ nearest neighbors. Algorithm 1 details our approach.

Let S be the set that consists of $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j, j = 1, 2, \dots, J$ (i.e., $S = \{\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j\}_{j=1}^{J}$). In step 1, for each $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$, we find its $L-1$ nearest neighbors and obtain the set R_j. $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ in general should be closer to other within-class members than to other different-class members. Therefore, we expect R_js that correspond to within-class members to be similar to each other, while different to those that correspond to other different-class members. In step 2, we calculate the Hamming distance between R_i and R_j, defined by

$$d(R_i, R_j) \triangleq (L - \#\text{ of common elements in } R_i \text{ and } R_j).$$

Like the Euclidean distance, the Hamming distance $d(R_i, R_j)$ is an indication of how far the feature pair $\mathbf{R}_{\hat{\theta}_i}\mathbf{x}_i$ and $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ are separated from each other. However, $d(R_i, R_j)$ for $\mathbf{R}_{\hat{\theta}_i}\mathbf{x}_i$ and $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ that belong to two different classes, is always upper bounded by L. Hence, the Hamming distance $d(R_i, R_j)$ preserves the class sepa-

rability between any different-class pair $\mathbf{R}_{\hat{\theta}_i}\mathbf{x}_i$ and $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$. Using this property, we are able to choose K initial representatives such that they belong to K different classes with high probability (as shown in steps 3 to 6). With these K initial representatives, the corresponding initial K partitions are determined by the nearest neighbor criterion (step 7). We assume each of the K initial partitions contains one exemplar. For all subsequent iterations (steps 8 and 9), K distinct representatives $\{\mathbf{s}_k\}_{k=1}^K$ are always chosen from these predetermined K initial partitions, and are used to calculate the associated score $M(S(\{\mathbf{s}_k\}_{k=1}^K))$ defined by

$$M(S(\{\mathbf{s}_k\}_{k=1}^K)) = \frac{div(S(\{\mathbf{s}_k\}_{k=1}^K))}{err(S(\{\mathbf{s}_k\}_{k=1}^K))}.$$

The terms $err(S(\{\mathbf{s}_k\}_{k=1}^K))$ and $div(S(\{\mathbf{s}_k\}_{k=1}^K))$ are *square error* and *diversity*, respectively [61]. They are defined as follows:

$$err(S(\{\mathbf{s}_k\}_{k=1}^K)) \triangleq tr\left[\sum_{k=1}^{K}\sum_{\mathbf{s}\in S_k}(\mathbf{s}-\mathbf{s}_k)(\mathbf{s}-\mathbf{s}_k)^T\right],$$

and

$$div(S(\{\mathbf{s}_k\}_{k=1}^K)) \triangleq tr\left[\sum_{k=1}^{K}(\mathbf{s}_k-\bar{\mathbf{s}})(\mathbf{s}_k-\bar{\mathbf{s}})^T\right],$$

where $\bar{\mathbf{s}} = \frac{1}{K}\sum_{k=1}^K \mathbf{s}_k$ and $tr(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$. The *diversity* represents the scatter of representatives to their mean, while the *square error* represents the total summation of partition-specific scatters, over all K partitions. The maximization of $M(S(\{s_k\}_{k=1}^K))$ is achieved through maximizing the *diversity* while minimizing the *square error*. Since these K exemplars by assumption respectively fall within the K initial partitions, they can be found after a sufficient number of iterations. The representatives that give the maximum score $M(S(\{\mathbf{s}_k\}_{k=1}^K))$ in

W_1 iterations, are recorded as exemplars. The corresponding final partitions are obtained by finding nearest neighbors of the exemplars.

Algorithm 7: Design of initial dictionary, $\mathbf{D}^{(0)}$.

Input: Scale and rotation aligned sinograms, $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j, j = 1, 2, ...J$.

Initialization of sets: $S \leftarrow \{\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j\}_{j=1}^{J}, I \leftarrow \{1, 2, ..., J\}$, $T \leftarrow \phi$. **Procedure:**

1. For each $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ in S, find its $L-1$ nearest neighbors. $\mathbf{R}_{\hat{\theta}_j}\mathbf{x}_j$ and its $L-1$ nearest neighbors form a set denoted by R_j.

2. For all pairs (i, j), calculate the Hamming distance between R_i and R_j, $d(R_i, R_j)$.

3. Find $(i^*, j^*) = \underset{i,j \in I, i \neq j}{\operatorname{argmax}} d(R_i, R_j)$.

4. Update of sets: $t_1 \leftarrow i^*, t_2 \leftarrow j^*, T \leftarrow T \bigcup \{t_1, t_2\}$, $I \leftarrow I \setminus \{i^*, j^*\}$. If $|T| = K$, goto 7.

5. Find $k^* = \underset{k \in I}{\operatorname{argmax}} \prod_{l=1}^{|T|} d(R_{t_l}, R_k)$.

6. Update of sets: $t_{|T|+1} \leftarrow k^*, T \leftarrow T \bigcup \{t_{|T|+1}\}$, $I \leftarrow I \setminus \{k^*\}$, and goto step 3.

7. Given $\{\mathbf{R}_{\hat{\theta}_{t_k}}\mathbf{x}_{t_k}\}_{k=1}^{K}$, use the nearest neighbor criterion to partition S into K partitions, denoted by $S(\{\mathbf{R}_{\hat{\theta}_{t_k}}\mathbf{x}_{t_k}\}_{k=1}^{K}) = \bigcup_{k=1}^{K} S_k$.

8. Randomly select $\mathbf{s}_k$ from S_k, $k = 1, 2, ..., K$, as representatives. Find the corresponding nearest neighbor partitions $S(\{\mathbf{s}_k\}_{k=1}^{K})$, and calculate the corresponding score $M(S(\{\mathbf{s}_k\}_{k=1}^{K}))$.

9. Repeat step 8, and keep updating for $\{\mathbf{s}_k^*\}_{k=1}^{K}$ that gives the highest score M, until the number of repeating iterations for step 9 reaches W_1. In other words,

$$\{\mathbf{s}_1^*, ..., \mathbf{s}_K^*\} = \underset{\mathbf{s}_k \in S_k, k=1,2,...,K, \text{ in } W_1 \text{ iterations}}{\operatorname{argmax}} M(S(\{\mathbf{s}_k\}_{k=1}^{K})).$$

10. Obtain K initial clusters $\{\mathbb{C}_1^{(0)}, \ldots, \mathbb{C}_K^{(0)}\}$ from $S(\{\mathbf{s}_k^*\}_{k=1}^{K})$.

Output: Initial dictionaries, $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)} \ldots \mathbf{D}_K^{(0)}]$, where $\mathbf{D}_k^{(0)} = \mathbf{C}_k^{(0)}, \;\; k = 1, \ldots, K$.

4.2.5 Application to CBIR

In this subsection, we show how the proposed simultaneous clustering and dictionary learning method is used in CBIR. Once the dictionaries have been learned

for each class in the Radon domain, given a query image $\mathbf{x}_q$, we obtain its scale and rotation normalized sinogram $\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q$. Then, we project $\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q$ onto the span of the atoms in each $\mathbf{D}_k$ using the orthogonal projector

$$\mathbf{Proj}_{\mathbf{D}_k} = \mathbf{D}_k(\mathbf{D}_k^T\mathbf{D}_k)^{-1}\mathbf{D}_k^T. \tag{4.14}$$

The approximation and residual vectors can then be calculated as

$$\mathbf{R}_{\hat{\theta}_q}^k\mathbf{x}_q = \mathbf{Proj}_{\mathbf{D}_k}\big(\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q\big), \tag{4.15}$$

and

$$\mathbf{r}^k(\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q) = \mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q - \mathbf{R}_{\hat{\theta}_q}^k\mathbf{x}_q = (\mathbf{I} - \mathbf{Proj}_{\mathbf{D}_k})\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q, \tag{4.16}$$

respectively, where $\mathbf{I}$ is the identity matrix. Since the dictionary learning step in our algorithm finds the dictionary $\mathbf{D}_k$ that leads to the best representation for each member of $\mathbb{C}_k$ in the Radon domain, we assume $\|\mathbf{r}^k(\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q)\|_2$ is small if $\mathbf{x}_q$ belongs to the kth cluster and larger for the other clusters. Based on this, if

$$d = \arg\min_{1\leq k\leq K} \|\mathbf{r}^k(\mathbf{R}_{\hat{\theta}_q}\mathbf{x}_q)\|_2, \tag{4.17}$$

we search for the relevance of $\mathbf{x}_q$ in $\mathbb{C}_d$ by means of a nearest neighbor search (box 6, 7 in Fig. 4.1). We refer to our Rotation and Scale Invariant Clustering and Dictionary learning method as *RSICD*. Algorithm 8 summarizes the overall RSICD-based CBIR procedure.

The RSICD-based CBIR algorithm consists of two main steps: cluster assignment and dictionary learning. The overall algorithm is not convex on both of these steps. It is likely that the approach may get stuck in a local minima. How-

Algorithm 8: RSICD-based CBIR.

Input: Database $\{\mathbf{x}_j\}_{j=1}^{J}$ and query image $\mathbf{x}_q$.

Algorithm:

1. Use (4.3) and (4.4) to obtain scale and rotation aligned sinograms of $\{\mathbf{x}_j\}_{j=1}^{J}$ and $\mathbf{x}_q$.

2. Use Algorithm 1 to design initial dictionaries $\{\mathbf{D}_k^{(0)}\}_{k=1}^{K}$.

3. Given $\mathbf{D}^{(i)} = [\mathbf{D}_1^{(i)} \cdots \mathbf{D}_K^{(i)}]$, assign each $\mathbf{x}_j$ to $\mathbb{C}_{\hat{k}}^{(i)}$, where i denotes the current iteration number, and $\hat{k}$ is obtained from (4.9) and (4.10).

4. Given $\{\mathbb{C}_k^{(i)}\}_{k=1}^{K}$, use the K-SVD algorithm to learn $\mathbf{D}_k^{(i+1)}$ from $\mathbb{C}_k^{(i)}, k = 1, 2, \ldots, K$. Then increment i by 1 (i.e., $i \leftarrow (i+1)$).

5. Repeat **3** and **4** until the number of repeating iterations reaches W_2.

6. Determine the closest cluster to $\mathbf{x}_q$ from (4.17), from which the relevances are found by the nearest neighbor criterion.

Output: Clusters $\{\mathbb{C}_k^{(W_2)}\}_{k=1}^{K}$, dictionaries $\{\mathbf{D}_k^{(W_2)}\}_{k=1}^{K}$, and the relevance of $\mathbf{x}_q$ in its closest cluster.

ever, experiments on various training sets have shown that it usually takes about 20 iterations for the algorithm to converge.

4.3 Experimental Results

In this section, we show the effectiveness of the proposed simultaneous clustering and dictionary approach. We report the results of empirical evaluation of our method and compare it with six state-of-the-art matching algorithms on three standard datasets: the Smithsonian isolated leaf dataset [93], Kimia's object dataset [92] and Brodatz texture dataset [101]. We compare the performance of our method with a modified Gabor-based approach [82], a local binary pattern (LBP)-based ap-

proach [102], and three recently proposed feature-based approaches [94], [93], [91]. We refer to the indexing and retrieval method presented in [94] as the BAC[3] method. Note that methods presented in [94], [93], [91] are (in-plane) rotation and scale invariant as well. In addition, we compare our method with a recently proposed unsupervised discriminative dictionary learning method [15]. We refer to the method presented in [15] as dictionary-based clustering (DC).

For all the experiments implemented using LBP [102], we first resized each image to 40×40 pixels. Each resized image consists of 25 square patches, each with 64 pixels. On each patch we implemented uniform rotation-invariant LBP with $P = 8$ and $R = 1$, where P is the member number in a circularly symmetric neighbor set, and R is the corresponding radius [102]. Results of all 25 patches are then combined to form a feature vector of the image. For the experiments implemented using the dictionary-based methods (RSICD and DC), we set the sparsity parameter T_0 to be 20.

We evaluate the performance of various methods using precision-recall curves, average retrieval performance [90], [82] and recognition rates. Recall and precision are defined as

$$\text{Precision} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}, \tag{4.18}$$

$$\text{Recall} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images}}. \tag{4.19}$$

[3]As no representative name was given for the method presented in [94], we chose the first letter of each author's last name (i.e., 'B', 'A', and 'C' in serial) and connected these letters as 'BAC' to stand for their method.

Recall is the portion of total relevant images retrieved whereas precision indicates the capability to retrieve only relevant images. An ideal retrieval should give precision rate that always equals 100% for any recall rate. Given a certain number of retrieved images, the average retrieval performance is defined as the average number of relevant retrieved images over all query images of a particular class. On the other hand, the rank-n recognition rates indicate how well the recognition performance of an approach can maintain from the best-match retrieval up to the n-th best-match retrieval. An ideal retrieval should also maintain 100% recognition rate for any rank-n retrieval.

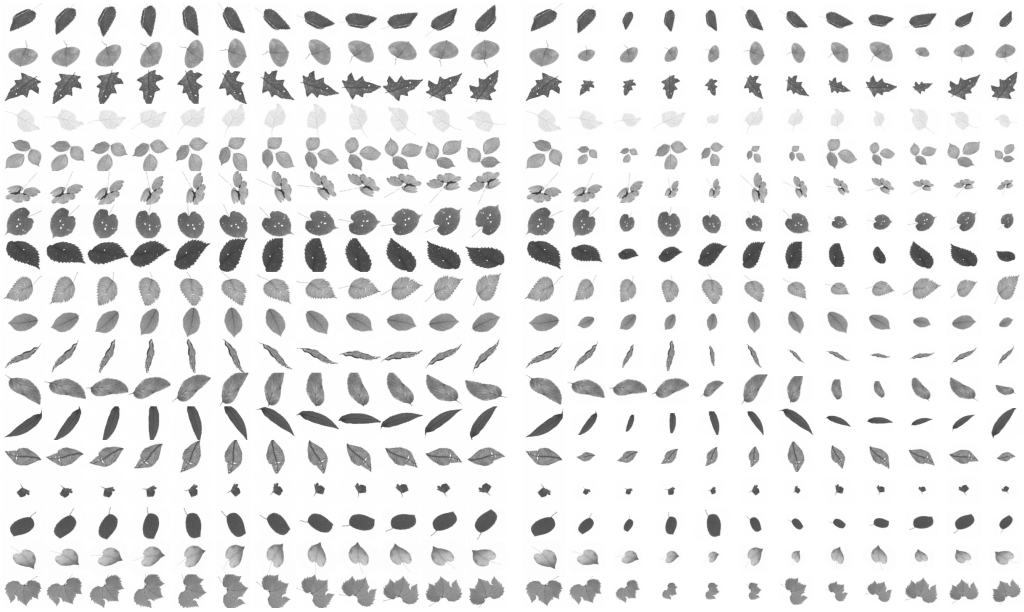
4.3.1 Smithsonian isolated leaf database

The original Smithsonian isolated leaf database consists of 93 different leaves [93]. From the original database, we created two challenging datasets, one containing rotated images and the other containing both rotated and scaled images. For the first set of experiments using this dataset, one representative image is selected from each of the last 18 leaves. We created an 18-class Smithsonian dataset by generating 11 additional in-plane rotated images with the following angles

$$18^\circ, 36^\circ, 54^\circ, 72^\circ, 90^\circ, 108^\circ, 126^\circ, 144^\circ, 162^\circ, 180^\circ, 198^\circ. \tag{4.20}$$

This sub-dataset contains rotated images of different leaves. We also created a dataset that contains both rotated and scaled images. This dataset is created by using the same rotated images as before. However, a random scaling that ranges from 0.25 to 1, is further applied to these rotated images. As a result, both of these

sub-datasets have 18 classes and each class contains 12 different images. Fig. 4.5(a) and (b) show the resulting datasets containing rotated as well as rotated and scaled images, respectively. In both of these datasets, the final images were resized to 100×80 pixels.



(a) (b)

Figure 4.5: (a) Sample images from the generated dataset containing the rotated images from the Smithsonian dataset. (b) Sample images from the Smithsonian dataset containing both scale and rotation variations.

4.3.1.1 Results on the Smithsonian dataset with 18 classes

In the first experiment using this dataset, we selected the last image (i.e., the 12-th image) of each class to form a query set (with 18 query images), and all the

other images to form an unsupervised[4] training set (with $216 - 18 = 198$ training images). The dictionary $\mathbf{D}$ is of size 288×90. Five atoms per class dictionary are learned and concatenated to form the dictionary $\mathbf{D}$. Here, 288 is the dimension of the vectorized sinogram.

The results of this experiment are shown in Fig. 4.6(a) and (c). From the precision-recall curves we see that the proposed RSICD achieves ideal precision rates for all recall rates and outperforms other competitive methods. Fig. 4.6(b) and (d) show the total average retrieval performance over all shapes. For the sub-dataset containing rotated images only, on average RSICD obtained 7.9352 out of 8 retrieved images per shape. Whereas the BAC [94], IDSC+DP [93], SC [91], DC [15], LBP [102] and Gabor-based methods [82] obtained 5.5417, 5.2593, 5.0463, 2.9630, 2.8809 and 2.5324, respectively.

For the dataset containing rotated and scaled images, our RSICD obtained 7.9815 out of 8 retrieved images per shape. The BAC, IDSC+DP, SC, DC, LBP and Gabor-based methods obtained 2.8333, 6.6389, 6.2037, 2.1343, 1.5000 and 2.5324, respectively.

In our CBIR experiments, to determine the class label of a given test image, we find its nth nearest neighbor among the training images, and then assign that training image's estimated class label (given by our RSICD algorithm) to the test image. The nth rank recognition rate is therefore defined as the ratio of the number of test images' nth nearest neighbors (among the training images) that are assigned with the same class labels as the true labels of the test images, to the total number

[4]Here 'unsupervised' means samples' class labels are unknown to the algorithm initially.

of test images. Tables 4.2 and 4.3 show the rank recognition rates for the above two 18-class datasets. Numbers in the abscissa of Table 4.2, and 4.3 are the values of n (same for Tables 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9). We observe that the proposed RSICD performs favorably in comparison to other methods.

Table 4.2: Rank recognition rates (%) corresponding the dataset containing 18-classes with rotated images from the Smithsonian isolated leaf database.

Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	60.65	27.78	31.02	31.48	30.09	25.93	25.00	21.30	22.22	15.28
LBP [102]	77.78	49.54	35.19	34.26	25.46	22.69	23.15	20.83	18.06	15.74
DC [15]	74.54	55.56	48.61	40.74	32.41	27.78	25.00	20.83	23.61	17.59
SC [91]	91.20	80.09	74.54	68.98	58.33	51.39	40.28	39.81	36.57	26.39
IDSC+DP [93]	92.13	78.70	73.15	68.98	66.67	55.09	48.15	43.06	35.19	29.63
BAC [94]	91.20	92.13	85.19	75.93	66.20	54.17	49.07	40.28	33.33	33.33
RSICD	**100**	**100**	**100**	**99.54**	**100**	**97.69**	**99.54**	**99.07**	**100**	**99.54**

4.3.1.2 Results on all 93 classes of the Smithsonian dataset

In the second set of experiments with the Smithsonian leaves dataset, we used all 93 classes to evaluate the rank recognition rates of different methods. Similar to the 18-class sub-datasets, we created rotated and scaled images for all 93 classes. Five atoms per class dictionary are learned and concatenated to form the dictionary $\mathbf{D}$ of size 288×465.

Tables 4.4 and 4.5 show the rank recognition rates of the 1st up to the 15th

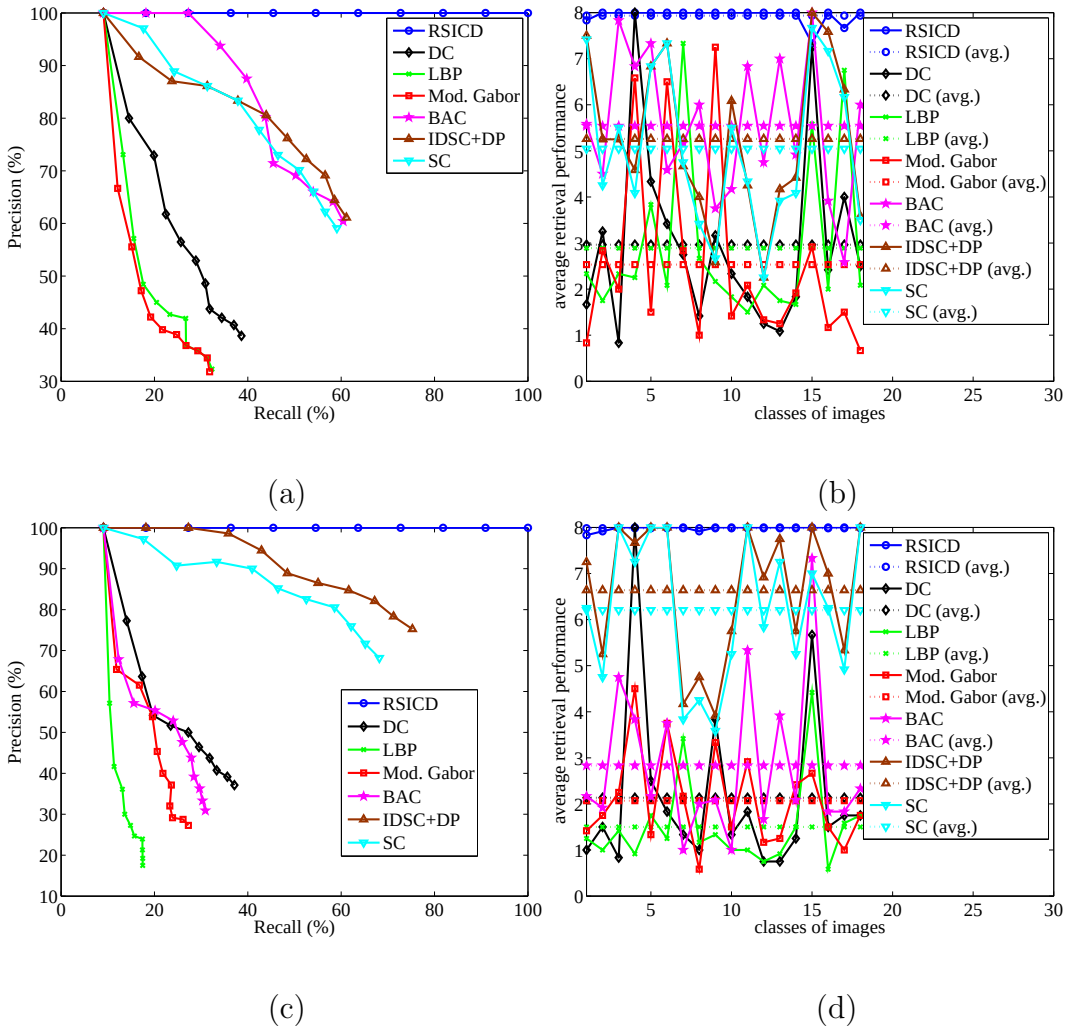


Figure 4.6: Results on rotated 18-class Smithsonian datasets. (a) Precision-recall curves and (b) the average retrieval performance corresponding to the dataset containing the rotated images. (c) Precision-recall curves and (d) the average retrieval performance corresponding to the dataset containing the rotated and scaled images. For both of these datasets, the proposed RSICD achieves the best precision rates for almost all recall rates and outperforms other methods.

Table 4.3: Rank recognition rates (%) corresponding the dataset containing 18-classes with rotated and scaled images from the Smithsonian isolated leaf database.

Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	48.61	32.87	31.48	24.54	20.83	17.59	18.06	12.96	9.72	12.96
LBP [102]	41.67	20.83	13.89	20.83	16.20	15.28	8.80	12.50	8.80	10.65
DC [15]	55.09	35.65	26.39	23.61	18.98	19.44	16.20	15.28	16.67	14.81
SC [91]	96.30	93.52	87.96	83.80	78.70	67.13	58.33	54.63	53.70	45.83
IDSC+DP [93]	99.07	99.07	93.52	86.57	79.17	73.61	69.91	62.96	63.43	56.02
BAC [94]	67.59	49.07	42.59	34.26	26.39	25.46	21.30	16.67	17.59	12.96
RSICD	**100**	**100**	**100**	**99.07**	**99.54**	**100**	**99.07**	**99.54**	**99.07**	**98.15**

rank retrieval. For both datasets, the recognition rates of the RSICD are the highest and at least 10% above those of the others for all rank retrievals. Comparing the RSICD results in Table 4.2 and Table 4.3, the average recognition rate goes from 98.99% (18-classes) to 90.95% (93-classes). This decrease is only 8.04%, which shows that the RSICD is robust in maintaining recognition performances on rotated and scaled leaf databases across different class numbers.

4.3.1.3 Robustness of RSICD to missing pixels

We compare the results obtained by different methods when pixels are randomly removed from the query and probe images. The results are shown in Fig. 4.7 where we compare the rank-1 recognition rates of different methods as we vary the percentage of missing pixels. We can see that both dictionary methods (RSICD and

Table 4.4: Rank recognition rates (%) corresponding the Smithsonian dataset containing 93-classes with rotated images.

Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	48.66	19.44	17.29	11.47	12.10	10.04	9.68	9.68	9.95	9.32
LBP [102]	56.09	31.09	24.55	16.58	17.65	11.29	11.38	10.39	9.68	7.44
DC [15]	54.75	34.59	25.27	22.13	17.74	14.78	13.98	12.99	11.83	11.38
SC [91]	85.13	72.31	64.78	59.95	51.16	43.10	37.90	31.81	28.76	22.85
IDSC+DP [93]	89.07	80.11	71.77	64.25	56.54	46.51	42.29	37.19	30.65	27.06
BAC [94]	83.87	75.72	68.46	55.73	44.09	35.93	28.32	27.33	22.94	18.91
RSICD	**99.28**	**97.49**	**95.79**	**95.25**	**93.55**	**90.41**	**91.49**	**90.23**	**87.46**	**83.78**

Table 4.5: Rank recognition rates (%) corresponding the Smithsonian dataset containing 93-classes with rotated and scaled images.

Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	26.43	16.40	12.10	10.39	10.84	10.30	9.23	9.41	6.72	6.99
LBP [102]	26.16	11.47	8.87	6.36	6.27	4.93	4.12	4.21	4.93	3.41
DC [15]	29.75	15.86	11.02	8.33	7.97	6.27	6.54	5.73	5.56	5.38
SC [91]	92.13	81.81	73.03	63.35	55.11	48.48	44.00	36.65	33.87	29.57
IDSC+DP [93]	97.58	92.83	83.96	72.85	62.01	54.84	49.73	43.28	43.37	36.83
BAC [94]	50.54	31.54	20.52	19.27	15.50	14.52	11.20	11.11	9.95	8.60
RSICD	**98.21**	**95.52**	**92.83**	**91.85**	**90.95**	**89.16**	**86.29**	**86.65**	**81.36**	**78.23**

DC) outperform the other methods. The RSICD is able to maintain its recognition rate at 75% even when 80% of the pixels are missing and it performs better than the DC.

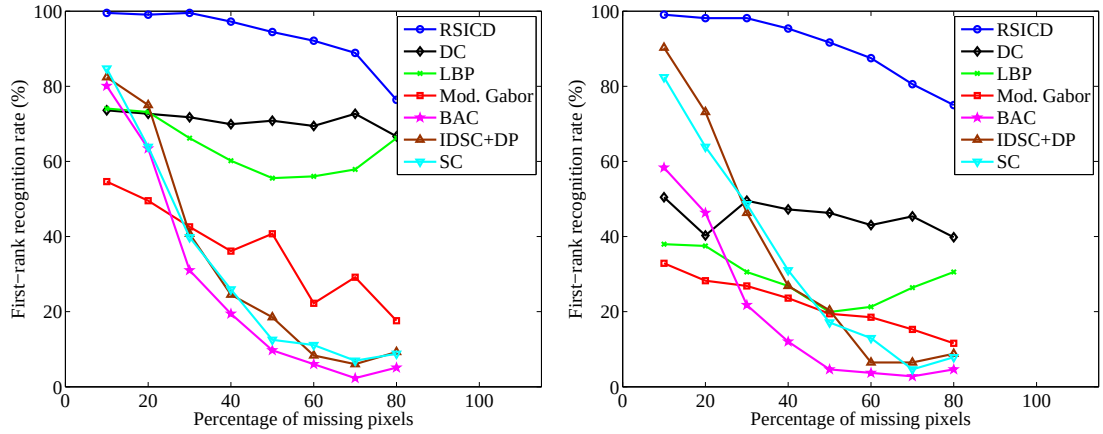


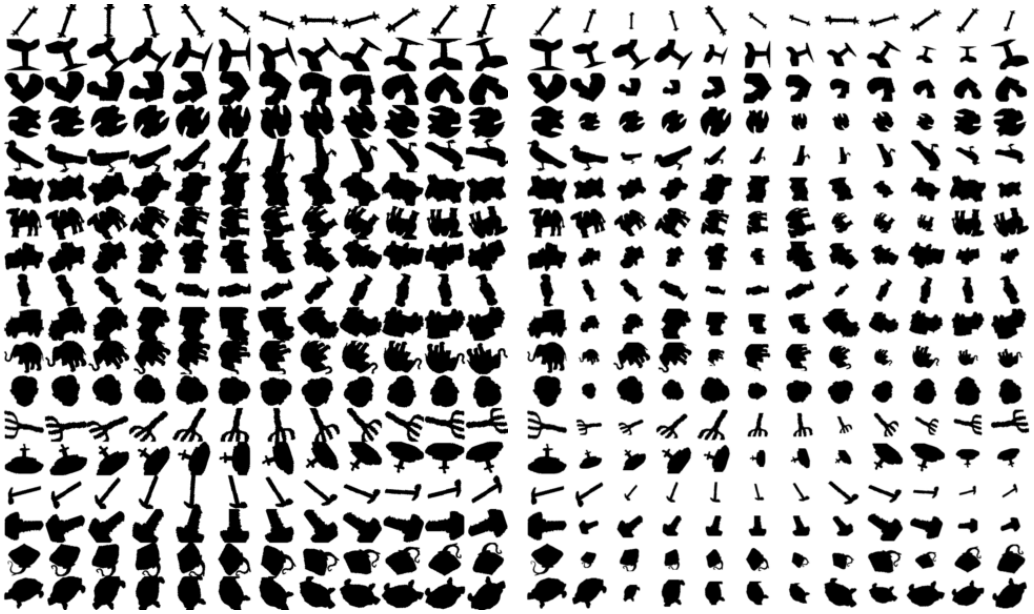
Figure 4.7: First-rank recognition rates on 18-class Smithsonian datasets with missing pixels. (a) Experiment with the dataset with rotated images. (b) Experiment with the dataset containing both rotated and scaled images. These results show the proposed RSICD is robust to effects of missing pixels.

4.3.2 Kimia shape database

The Kimia database [92] consists of 216 images, where there are 18 shapes with small rotation, and each shape has 12 different images. Similar to how we generated new datasets for the Smithsonian Leaf database in the previous experiments, we created two sub-datasets from the original Kimia database: one containing the rotated images and the other containing the rotated and scaled images.

To obtain the rotated images, we selected one representative image from each

of the 18 shapes in the original Kimia dataset. For each selected shape, 11 in-plane rotated images with the same angles as in (4.20) were created. We created a dataset that contains rotated and scaled images by scaling the rotated images as before. The resulting datasets are shown in Fig. 4.8(a) and (b), respectively. They possess more rotation and scale challenges than the original Kimia's dataset.



(a) (b)

Figure 4.8: Kimia datasets containing (a) rotated images and (b) rotated and scaled images.

We selected the last image (i.e., the 12-th image) of each class to form a query set, and used all the other images to form an unsupervised training set. The precision-recall curves are shown in Fig. 4.9(a) and the average retrieval performance curves are shown in Fig. 4.9(b) for the datasets containing the rotated images. The dictionary size is set as 288.

As can be seen from both these figures, our method outperforms other competitive methods. Regarding the overall retrieval performance, our RSICD obtained 7.0324 out of 8 retrieved images per shape. Whereas the BAC [94], IDSC+DP [93], SC [91], DC [15], LBP [102] and Gabor-based methods [82] obtained 6.5185, 4.0231, 3.8935, 2.0926, 2.3657 and 1.2778, respectively. Table 4.6 shows rank recognition rates for the above two 18-class datasets. The rank recognition rates of our RSICD remains the second while they still are close to the best results from BAC, for up to the 5th rank recognition.

Figs. 4.9(c) and (d) show the results obtained using the dataset containing both scaled and rotated images. For the overall retrieval performance, our RSICD obtained 7.0463 out of 8 retrieved images per shape. The BAC, IDSC+DP, SC, DC, LBP and Gabor-based methods obtained 6.5185, 7.8472, 7.2269, 1.2500, 1.3889 and 1.3426, respectively. Table 4.7 shows the corresponding rank recognition rates.

4.3.3 Brodatz texture database

In addition to shape-based datasets, we demonstrate the effectiveness of our RSICD on the Brodatz texture dataset [101]. We selected 25 textures and 60 textures from the Brodatz database, which are the dataset 1 and the dataset 3 defined in [95]. For each selected texture, we generated its in-plane rotated versions at the following angles: $10°, 20°, 30°, ..., 170°$. Each original texture image and its 17 rotated images form a new in-plane rotated class. Fig. 4.10 shows the resulting sample images from the 25-class dataset. The dictionaries are of size 192×300 and 192×720 for 25 and

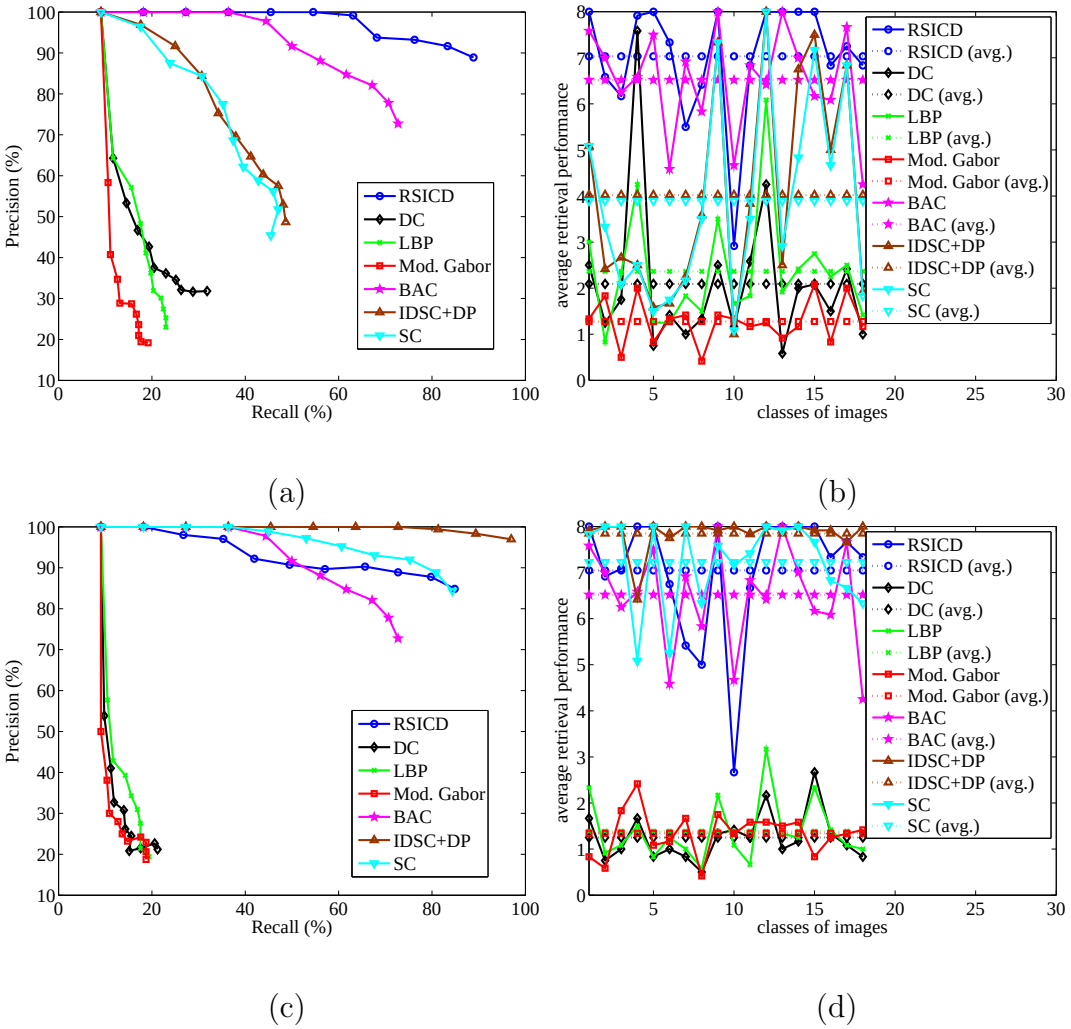


Figure 4.9: Results on Kimia dataset. (a) Precision-recall curves and (b) the average retrieval performance of the dataset containing rotated images. (c) Precision-recall curves and (d) the average retrieval performance of the dataset containing rotated and scaled images.

Table 4.6: Retrieval results (rank recognition percentage rates) on Kimia dataset containing rotated images.

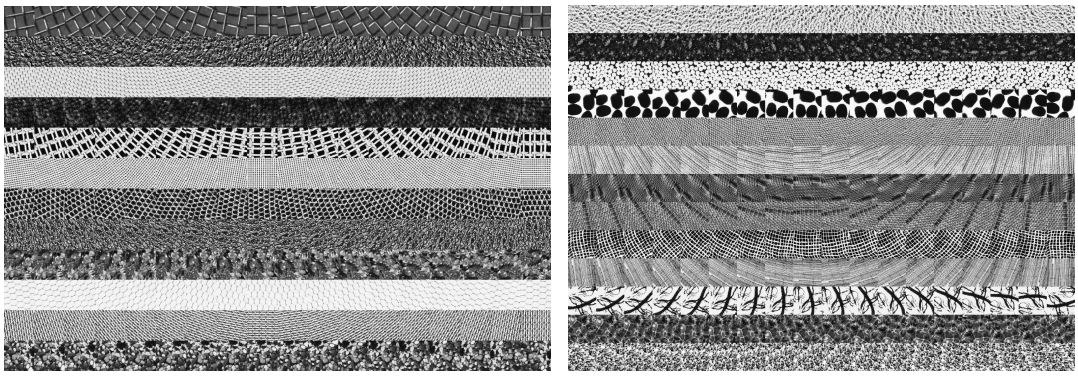
Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	42.13	21.30	15.28	13.43	7.41	12.50	8.33	7.41	4.17	4.63
LBP [102]	62.96	45.37	26.85	29.17	24.07	15.28	16.20	16.67	11.11	10.65
DC [15]	60.65	38.43	23.61	21.76	17.59	17.59	16.20	13.43	17.59	12.04
SC [91]	86.11	66.67	59.26	48.61	40.28	32.41	28.24	27.78	25.00	15.28
IDSC+DP [93]	86.57	69.91	57.41	48.61	42.13	37.50	31.94	28.24	28.24	21.30
BAC [94]	**100**	**97.69**	88.89	**90.28**	**84.72**	70.37	63.89	56.02	48.61	32.41
RSICD	98.15	92.13	**89.81**	88.43	82.87	**84.26**	**81.48**	**86.11**	**81.02**	**69.91**

Table 4.7: Retrieval results (rank recognition percentage rates) on Kimia dataset containing rotated and scaled images.

Rank	1	2	3	4	5	6	7	8	9	10
Modified Gabor [82]	36.11	15.28	16.67	13.43	15.28	11.11	13.43	12.96	11.57	8.33
LBP [102]	41.67	21.76	17.59	13.89	9.72	14.81	10.65	8.80	9.72	7.41
DC [15]	43.52	18.52	13.89	10.19	11.57	11.11	11.57	8.80	6.48	7.41
SC [91]	99.07	98.61	95.83	95.37	92.59	84.26	82.41	74.54	75.93	67.59
IDSC+DP [93]	**100**	**99.54**	**98.15**	**99.07**	**98.61**	**97.69**	**96.30**	**95.37**	**90.74**	**84.72**
BAC [94]	**100**	97.69	88.89	90.28	84.72	70.37	63.89	56.02	48.61	32.41
RSICD	97.22	94.44	89.81	89.81	85.19	84.72	81.94	84.26	79.17	76.39

60 class datasets, respectively. Here, 192 is the size of vectorized sinogram.

Experimental results using 25 classes and 60 classes are compared in Table 4.8 and Table 4.9, respectively. The modified Gabor method gives 100% recognition rate on its first rank retrieval but degrades faster than the other methods within the first 4 rank retrievals. The average recognition rates of the RSICD are 60.17% (25-class) and 53.18% (60-class), which are higher than those of CD: 39.86% (25-class), 34.17% (60-class); LBP: 44.24% (25-class), 30.14% (60-class); and modified Gabor: 46.54% (25-class), 36.41% (60-class). This experiment shows that the RSICD is general enough that it can also perform well on a dataset that contains rotated textures.



(a) (b)

Figure 4.10: Samples images from the 25-class in-plane rotated Brodatz texture database. (a) 1st $\sim$ 12th classes: D01, D04, D06, D19, D20, D21, D22, D24, D28, D34, D52, D53; (b) 13th $\sim$ 25th classes: D56, D57, D66, D74, D76, D78, D82, D84, D102, D103, D105, D110, D111

Table 4.8: Rank recognition rates (%) on 25-class (data set 1 in [95]) in-plane rotated Brodatz database.

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Modified Gabor [82]	**100**	80.4	81.3	57.1	57.6	43.3	44.4	38.4	40.9	35.8	34.9	33.6	34.2	27.8
LBP [102]	79.3	67.8	56.9	49.8	46.9	45.8	46.2	43.1	41.6	39.8	38.9	39.1	37.1	33.1
DC [15]	88.2	69.6	59.3	55.1	47.3	44.4	41.8	38.0	35.3	34.7	32.0	28.2	27.6	22.9
RSICD	92.0	**86.9**	**85.6**	**82.0**	**76.0**	**75.6**	**71.1**	**63.6**	**57.6**	**53.1**	**52.7**	**46.7**	**45.1**	**42.2**

Table 4.9: Rank recognition rates (%) on 60-class (data set 3 in [95]) in-plane rotated Brodatz database.

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Modified Gabor [82]	**100**	71.1	72.4	40.3	42.5	34.4	34.4	27.2	30.4	27.4	24.3	21.2	22.4	19.6
LBP [102]	76.7	59.4	43.3	38.2	31.3	30.3	27.5	25.3	25.4	25.0	22.3	19.0	21.7	17.3
DC [15]	86.5	69.1	58.4	51.5	46.1	39.7	33.8	31.6	26.1	25.8	22.5	21.6	19.2	14.4
RSICD	93.1	**86.1**	**82.1**	**75.7**	**70.1**	**66.4**	**62.7**	**54.2**	**46.6**	**44.4**	**42.4**	**40.2**	**35.7**	**32.9**

4.3.4 Discussion

In this section, we discuss our experimental results in the aspects of performance, complexity and limitation.

4.3.4.1 Performance

For texture-based approaches, the Gabor method [82] extracts features using a modified Gabor filter to achieve independence of orientation and scale in the textures. The LBP [102] is a computationally simple method, but an efficient multiresolution approach based on uniform local binary patterns and nonparametric

discrimination of sample and prototype distributions for rotation invariant texture classification. Both methods are designed for texture-based images. As a result, they did not obtain good results on shape-based datasets, which can be seen from the experimental results in Sections 4.3.1 and 4.3.2.

For shape-based approaches, the SC [91] is a shape matching approach based on correspondences between points on two shapes. The SC descriptor essentially estimates the shape similarity and solves the correspondence problems. The IDSC+DP [93] uses the length of the shortest path within the shape boundary (called inner-distance) to build shape descriptors, which were shown to be robust to articulation in complicated shapes. The BAC [94] extracts features that characterize the geometric relationships between each pair of images. This method was shown to be invariant to articulations and rigid transforms.

The SC descriptor [91] relies on the correspondences between points on two shapes, while the IDSC+DP descriptor [93] is built based on the normalized inner distance. In practice, SC and IDSC+DP descriptors remain the same for similar shapes with different scales and change significantly for different shapes with different scales. Therefore, introducing scale variations (shown in Fig. 4.5(b) and Fig. 4.8(b)) in fact boosts the discriminative power of SC and IDSC+DP features, in that between-class distances are increased due to scale variations while within-class distances remain the same. Hence, SC and IDSC+DP obtained better results on the datasets containing both rotation and scale variations (see Fig. 4.6, Fig. 4.9, Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6 and Table 4.7). Furthermore, pixels in the Smithsonian leaf images appear in different grayscales. Without addi-

tional preprocessing, the shape contours as well as inner distances-based and point correspondence-based descriptors are sensitive to changes in grayscale or missing pixels (e.g., Fig. 4.7(a)(b)). These two reasons explain why SC and IDSC+DP in our experiments perform much better in the Kimia dataset with both rotation and scale variations. The BAC, too, obtained better results on the Kimia dataset. However, it is sensitive to both scale and rotation changes in Smithsonian leaf images. On the other hand, the proposed RSICD does not require any knowledge of the shape contour, and is not sensitive to grayscale changes and missing pixels in an image. In addition, the Smithsonian leaf datasets used in our experiments consist of more directional leaves than isotropic leaves, which are in favor of our assumption on directional images described in Section 4.1-A. Hence, the proposed RSICD obtained good results on the Smithsonian leaf datasets. Finally, from the experimental results, we observe that DC [15] does not give satisfactory performances in both shape-based and texture-based datasets because it uses pixel intensities as features.

4.3.4.2 Complexity

We present the relative complexity of all the methods by comparing the computation time required to obtain precision-recall curves for the rotated 18-class Smithsonian datasets in Table 4.10. This table shows both the computation time[5] and

[5]We conducted our experiments using Matlab installed in the 64-bit Windows OS on a machine with Intel(R) Core(TM) i5 CPU (2.8 GHz) and 8GB RAM.

Table 4.10: Computation time of different methods to obtain precision-recall curves shown in Fig. 4.6(a).

	Modified Gabor [82]	LBP [102]	DC [15]	SC [91]	IDSC+DP [93]	BAC [94]	RSICD
Execution time (s)	10.42	13.34	383.40	1226.34	1383.48	92.31	217.32
Unsupervised clustering	no	no	yes	no	no	no	yes

whether the unsupervised clustering is provided by each method. Note that the RSCID provides both unsupervised clustering and dictionary learning. As a result, its computation time is higher than some of the other methods. Also, both SC [91] and IDSC+DP [93] require a large amount of time for image retrieval.

4.3.4.3 Limitation

We have examined the performance of our method on various shape-based and texture-based datasets. In practice, there may be objects with background clutter. For our method to be effective, the background needs to be removed before applying our algorithm to obtain good retrieval performances. Hence, it may not provide good results on datasets where images contain objects with background clutter. The second limitation of our method is that it does not work so well for texture-based images where there are more within-class variations such as illumination changes, noise, occlusion, variant distances with 3D rotations and spatial shifts. Moreover,

for textures where there are no linear trends (e.g., isotropic textures) or inapparent linear trends, the Radon-domain sinogram may no longer be used to accurately capture the direction to give rotation-aligned features.

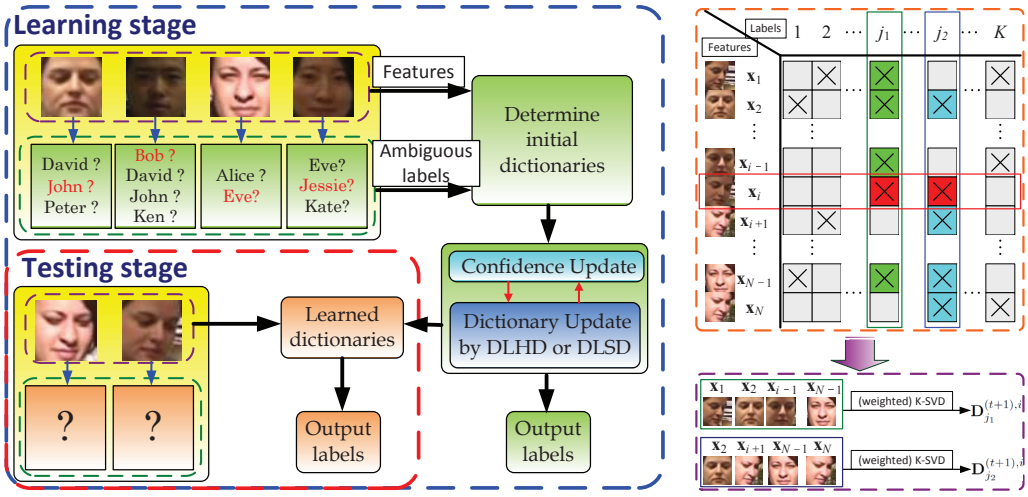
4.4 Summary

We have presented a rotation and scale invariant clustering algorithm suitable for applications such as CBIR. We extracted in-plane rotation and scale invariant features of images in the Radon domain. The initial dictionaries are learned through initial clusters that are determined using the Hamming distance between nearest-neighbor sets of each feature pair. With a view to achieve rotation and scale invariance in clustering, the proposed method learns dictionaries and clusters images in the Radon transform domain. We demonstrated the effectiveness of our approach by a series of CBIR experiments on shape-based and texture-based datasets, its robustness to missing pixels, and performance improvements compared to other Gabor-based and shape-based methods.

Chapter 5: Dictionary Learning from Ambiguously Labeled Data

In many practical image and video applications, one has access only to ambiguously labeled data. For example, given a picture with multiple faces and a caption specifying who are in the picture, the reader may not know which face goes with the names in the caption. The problem of learning identities where each example is associated with multiple labels, when only one of which is correct is often known as ambiguously learning.

Several papers have been published in the literature that address the ambiguous labeling problem. In [103], a discriminative framework was proposed based on the Expectation Maximization (EM) algorithm [104], with a maximum likelihood approach to disambiguate correct labels from incorrect ones. A semi-supervised dictionary-based learning method was proposed in [105] under the assumption that there are either labeled samples or totally unlabeled samples available for training. The method iteratively estimates the confidence of unlabeled samples belonging to each class and uses it to refine the learned dictionaries. In [4] and [5], a method was presented to determine the label using a multi-linear classifier that minimizes a convex loss function. The loss function used in [4] and [5] was shown to be a tighter convex upper bound on 0/1 loss when compared to an un-normalized 'naive'



(a) (b)

Figure 5.1: The proposed dictionary learning method. (a) Block diagram. (b) An illustration of how common label samples are collected to learn intermediate dictionaries, which are used to update the confidence for sample x_i.

method that treats each example as if it took on multiple correct labels. Several non-parametric, instance-based algorithms for partially labeled learning were proposed in [106].

Dictionary learning algorithms have been developed for supervised [107], [39], [41], [42], [43], semi-supervised [105] and unsupervised [15], [20], [17] learning. In this chapter, we consider a dictionary learning problem where each training sample is provided with a set of possible labels and only one label among them is the true one. This is a generalized semi-supervised formulation of the traditional one where there are only either labeled data or totally unlabeled data. We develop dictionary learning algorithms that process ambiguously labeled data.

Fig. 5.1(a) shows the block diagram of the proposed dictionary learning

method. Given ambiguously labeled training samples (e.g. faces), the algorithm consists of two main steps: confidence update and dictionary update. The confidence for each sample is defined as the probability distribution on its ambiguous labels. In the confidence update phase, the confidence is updated for each sample according to its residuals when the sample is projected onto different class dictionaries. Then, the dictionary is updated using a fixed confidence. In the testing stage, a novel test image is projected onto the span of the atoms in each learned dictionary. The resulting residual vectors are then used for classification.

Key contributions of this work are:

1. We propose a dictionary-based learning method when ambiguously labeled data are provided for training.

2. We present two effective approaches for updating the dictionary.

3. We show that the dictionary learning method with a soft decision rule is an EM-based dictionary learning method.

4. We propose a weighted K-SVD algorithm to account for the importance of samples according to their confidence during the learning process.

The rest of the chapter is organized as follows. Section 5.1 formulates the ambiguously labeled learning problem and presents the details of the proposed dictionary learning algorithms. We present experimental results with discussions in Section 5.2. Section 5.3 concludes the chapter with a brief summary.

5.1 Dictionary Learning from Ambiguously Labeled Data

Let $\mathcal{L} = \{(x_i, L_i), i = 1, \ldots, N\}$ be the training data. Here x_i denotes the i^{th} training sample, $L_i \subset \{1, 2, \ldots, K\}$ the corresponding multiple label set, and N the number of training samples. There are a total of K classes. The true label z_i of the i^{th} training sample is in the multi-label set L_i. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the lexicographically ordered vector representing the sample x_i. For each feature vector $\mathbf{x}_i$ and for each class j, we define a latent variable $p_{i,j}$, which represents the confidence of $\mathbf{x}_i$ belonging to the j^{th} class. By definition, we have $\sum_j p_{i,j} = 1$, and

$$\begin{aligned} p_{i,j} &= 0 \quad \text{if } j \notin L_i, i = 1, \ldots, N, \\ p_{i,j} &\in (0, 1] \quad \text{if } j \in L_i, i = 1, \ldots, N. \end{aligned} \tag{5.1}$$

Let $\mathbf{P}$ be the confidence matrix with entry $p_{i,j}$ in the i-th row and j-th column. Define $\mathbf{C}_j$ to be the collection of samples in class j represented as a matrix and $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_K]$ be the concatenation of all samples from different classes. Similarly, let $\mathbf{D}_j$ be the dictionary that is learned from the data in $\mathbf{C}_j$ and $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_K]$ be the concatenation of all dictionaries. Equipped with the above notation, the problem we study can be formally stated as follows:

For each feature vector available during training, we are given a set of labels, only one of which is correct. Given this ambiguously labeled data, how can one learn dictionaries to represent each class?

We solve the dictionary learning problem using an iterative alternating algorithm. At each iteration, two major steps are performed: confidence update and dictionary update. We demonstrate that both soft and hard decision rules produce robust dictionaries.

5.1.1 The Dictionary Learning Hard Decision approach

The dictionary learning hard decision (DLHD) approach learns dictionaries directly from class matrices[1], $\{\mathbf{C}_i\}_{i=1}^{K}$, that are determined using a hard decision for class labels for each sample x_i by selecting the classes with the maximum $p_{i,c}$ among all c's belonging to L_i. One iteration of the algorithm consists of the following steps.

Confidence Update: We use the notation $\mathbf{D}^{(t)}, \mathbf{P}^{(t)}$ to denote the dictionary matrix and confidence matrix respectively, in the t^{th} iteration. Keeping the dictionary $\mathbf{D}^{(t)}$ fixed, the confidence of a feature vector belonging to classes outside its label set is fixed to 0 and is not updated. To update the confidence of a sample belonging to classes in its label set, we first make the observation that a sample $\mathbf{x}_i$ which is well represented by the dictionary of class j, should have high confidence. In other words, the confidence of a sample $\mathbf{x}_i$ belonging to a class j should be inversely proportional to the reconstruction error that results when $\mathbf{x}_i$ is projected onto $\mathbf{D}_j$. This

[1]We refer to class matrices and clusters interchangeably.

can be done by updating the confidence matrix $\mathbf{P}^{(t)}$ as follows

$$p_{i,j}^{(t)} = \frac{\beta_j^{(t)} \exp\left(-\frac{e_{ij}^{(t)}}{2\sigma_j^{(t)}}\right)}{\sum_{k \in L_i} \beta_k^{(t)} \exp\left(-\frac{e_{ik}^{(t)}}{2\sigma_k^{(t)}}\right)}, \tag{5.2}$$

where $\beta_j^{(t)}$ and $\sigma_j^{(t)}$ are parameters (given in section 5.1.3), and

$$e_{ij}^{(t)} = \|\mathbf{x}_i - \mathbf{D}_j^{(t)}\overline{\mathbf{D}_j^{(t)}}\mathbf{x}_i\|_2^2 \tag{5.3}$$

is the reconstruction error, when $\mathbf{x}_i$ is projected onto $\mathbf{D}_j^{(t)}, \forall j \in L_i$ and $\overline{\mathbf{D}_j^{(t)}} \triangleq ((\mathbf{D}_j^{(t)})^T\mathbf{D}_j^{(t)})^{-1}(\mathbf{D}_j^{(t)})^T$ is the pseudo-inverse of $\mathbf{D}_j^{(t)}$. As shown in section 5.1.3, we derive (5.2) under the assumption that the likelihood of each sample $\mathbf{x}_i$ is a mixture of Gaussian densities, and $\beta_j^{(t)}$ is the weight associated with the density of label j.

Cluster Update:[2] Once the confidence matrix $\mathbf{P}^{(t)}$ is updated, we use it to update the class matrix $\mathbf{C}^{(t+1)}$. For each training sample $\mathbf{x}_i$, we assign it to the class j^i which gives the maximum confidence. That is,

$$j^i = \underset{k \in L_i}{\operatorname{argmax}}\, p_{i,k}^{(t)}. \tag{5.4}$$

Dictionary Update: The updated class matrices $\mathbf{C}^{(t+1)}$ are then used to train class-specific dictionaries. Given a class matrix $\mathbf{C}_j^{(t+1)}$, we seek a dictionary $\mathbf{D}_j^{(t+1)}$ that provides the sparsest representation for each example feature in this matrix,

[2]This step is necessary only for the DLHD approach.

by solving the following optimization problem

$$(\mathbf{D}_j^{(t+1)}, \mathbf{\Gamma}_j^{(t+1)}) = \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \|\mathbf{C}_j^{(t+1)} - \mathbf{D}\mathbf{\Gamma}\|_F^2,$$

$$\text{subject to } \|\boldsymbol{\gamma}_i\|_0 \leq T_0, \ \forall i, \tag{5.5}$$

where $\boldsymbol{\gamma}_i$ represents the i^{th} column of $\mathbf{\Gamma}$, $\mathbf{C}_j^{(t+1)}$ has a matrix representation whose columns are feature vectors assigned to the j-th class at iteration $(t+1)$, and T_0 is the sparsity parameter. Here, $\|\cdot\|_F$ denotes the Frobenius norm. Many approaches have been proposed in the literature for solving such optimization problem. We adapt the K-SVD algorithm [34] for solving (5.5). In the sparse-coding step, $\mathbf{D}$ is fixed and the representation vectors $\boldsymbol{\gamma}_i$s are found for the i-th column in $\mathbf{C}_j^{(t+1)}$. Then, the dictionary is updated atom-by-atom in an efficient way. The entire approach for learning dictionaries from ambiguously labeled data using hard decisions is summarized in Algorithm 9.

5.1.2 The Dictionary Learning Soft Decision approach

The dictionary learning soft decision (DLSD) approach learns dictionaries that are used to update the confidence for each sample $\mathbf{x}_i$, based on the weighted distribution of other samples that share the same candidate label belonging to L_i. The weighted distribution of other samples sharing a given candidate label c is computed through the normalization of all $p_{l,c}$'s with $l \neq i$. In what follows, we describe the different steps of the algorithm.

Confidence Update: In this step, given the intermediate dictionary $\mathbf{D}^{(t),i}$ learned

Algorithm 9: Iteratively learning dictionaries using hard decision and updating confidence.

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$ and initial dictionaries

$$\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}].$$

Output: Dictionary $\mathbf{D}^* = [\mathbf{D}_1^*|\mathbf{D}_2^*|\dots|\mathbf{D}_K^*]$.

Algorithm:

1. Repeat the following steps to refine the confidence until the maximum iteration number $\mathrm{T_c}$ is reached:

   1.1 **Confidence Update**: For each feature vector $\mathbf{x}_i$, calculate the residuals $e_{ij}^{(t)}$ using (5.3). Use $e_{ij}^{(t)}$ to update confidence $p_{i,j}^{(t)}$ using (5.2).

   1.2 **Cluster Update**: Assign each feature vector $\mathbf{x}_i$ to $\mathbf{C}_{j^i}^{(t+1)}$ according to (5.4).

   1.3 **Dictionary Update**: When the class assignment for all $\mathbf{x}_i$'s is completed, build dictionary $\mathbf{D}_j^{(t+1)}$ from $\mathbf{C}_j^{(t+1)}, \forall j \in \{1, 2, \dots, K\}$ using the K-SVD algorithm and obtain $\mathbf{D}^{(t+1)} = [\mathbf{D}_1^{(t+1)}|\mathbf{D}_2^{(t+1)}|\dots|\mathbf{D}_K^{(t+1)}]$.

2. Return $\mathbf{D}^* = \mathbf{D}^{(\mathrm{T_c})}$, where $\mathrm{T_c}$ is the iteration number at which the learning algorithm converges.

from the previous iteration for each sample $\mathbf{x}_i$, we calculate the residuals $e_{ij_l}^{(t),i}$ using $\mathbf{D}_{j_l}^{(t),i}$ for all j_l in L_i as

$$e_{ij_l}^{(t),i} = \|\mathbf{x}_i - \mathbf{D}_{j_l}^{(t),i}\overline{\mathbf{D}_{j_l}^{(t),i}}\mathbf{x}_i\|_2^2. \tag{5.6}$$

We then use (5.2) to update the confidence $p_{i,j_l}^{(t)}$, with $e_{ij}^{(t)}$ replaced by $e_{ij_l}^{(t),i}$.

Dictionary Update: In this step, the confidence matrix $\mathbf{P}^{(t)}$ is given. For each $\mathbf{x}_i$, we build the intermediate dictionaries for all labels in $L_i = \{j_1, j_2, \dots j_{|L_i|}\}$. In particular, we learn $\mathbf{D}^{(t+1),i} = [\mathbf{D}_{j_1}^{(t+1),i}|\mathbf{D}_{j_2}^{(t+1),i}|\dots|\mathbf{D}_{j_{|L_i|}}^{(t+1),i}]$, where each $\mathbf{D}_{j_l}^{(t+1),i}$ is built using soft decision from samples $\mathbf{x}_k \neq \mathbf{x}_i$ with $p_{k,j_l}^{(t+1)} > 0$. Fig. 5.1(b) shows an example of how these common ambiguous label samples are collected to learn the intermediate dictionaries $\mathbf{D}_{j_l}^{(t+1),i}$. The cell marked with '$\times$' at the (i, j) entry indicates a non-zero $p_{i,j}^{(t)}$. All the other empty cells indicate zero confidence. As shown in this example, only samples corresponding to green and blue cells are used to learn $\mathbf{D}_{j_1}^{(t+1),i}$ and $\mathbf{D}_{j_2}^{(t+1),i}$, respectively. To learn the intermediate dictionaries for $\mathbf{x}_i$, exclusion of $\mathbf{x}_i$ (corresponding to red cells) is necessary to enhance discriminative learning. Let $\{\mathbf{x}_{i_m}\}_{m=1}^{N(i,j_l)}$ be the collection of these samples. Its matrix form is denoted by $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \dots\ \mathbf{y}_{N(i,j_l)}]$, where $\mathbf{y}_m$, $m \in \{1, \dots, N(i, j_l)\}$, is a column vectorized form of some collected sample $\mathbf{x}_{i_m}$. Let $\mathbf{w} = [w_1\ w_2 \dots\ w_{N(i,j_l)}] = [p_{i_1,j_l}^{(t)}\ p_{i_2,j_l}^{(t)} \dots\ p_{i_{N(i,j_l)},j_l}^{(t)}]$, where the weight w_m reflects the relative amount of contribution from $\mathbf{x}_{i_m}$ when learning the dictionary. The objective of the weighted K-SVD algorithm can then

be formulated as

$$
\begin{aligned}
[\mathbf{D}_{j_l}^{(t+1),i}\ \mathbf{\Gamma}_{j_l}^{(t+1),i}] &= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \sum_{m=1}^{N(i,j_l)} w_m \|\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma}_m\|_2^2, \\
&\text{subject to } \|\boldsymbol{\gamma}_m\|_0 \leq T_0, \forall m, \\
&= \underset{\mathbf{D},\mathbf{\Gamma}}{\operatorname{argmin}} \|(\mathbf{Y} - \mathbf{D}\mathbf{\Gamma})\mathbf{W}\|_F^2, \\
&\text{subject to } \|\boldsymbol{\gamma}_m\|_0 \leq T_0, \forall m, \qquad (5.7)
\end{aligned}
$$

where $\mathbf{W}$ is a square weighting matrix with its diagonal filled with $\{\sqrt{w_m}\}_{m=1}^{N(i,j_l)}$, and zeros elsewhere. One can solve the above weighted optimization problem by modifying the K-SVD algorithm as follows:

- *Sparse Coding Stage*: For $m = 1, 2, \ldots, N(i, j_l)$, compute $\boldsymbol{\gamma}_m$ for $\mathbf{y}_m$ by solving

$$\min_{\boldsymbol{\gamma}} \|(\mathbf{y}_m - \mathbf{D}\boldsymbol{\gamma})\sqrt{w_m}\|_2^2, \text{ subject to } \|\boldsymbol{\gamma}\|_0 \leq T_0.$$

- *Codebook Update Stage:* This step remains the same as the original K-SVD algorithm except that the overall error representation matrix $\mathbf{E}_k$ is changed to $\mathbf{E}_k = (\mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \boldsymbol{\gamma}_T^j)\mathbf{W}$, where $\mathbf{d}_j$ is the j-th column of $\mathbf{D}$ and $\boldsymbol{\gamma}_T^j$ is the j-th row of $\mathbf{\Gamma}$ found in the previous sparse coding stage.

After $\mathrm{T_c}$ soft decision iterations, we assign the label with the maximum confidence. The labeled class matrices are used to learn the final dictionary $\mathbf{D}^* = \mathbf{D}^{(\mathrm{T_c})} = [\mathbf{D}_1^{(\mathrm{T_c})}|\mathbf{D}_2^{(\mathrm{T_c})}|\ldots|\mathbf{D}_K^{(\mathrm{T_c})}]$ via the K-SVD algorithm. This step is the same as 1.2 and 1.3 in Algorithm 9 with t set equal to $\mathrm{T_c}$. The entire DLSD approach is summarized in Algorithm 10.

Algorithm 10: Iteratively learning dictionaries using soft decision and updating confidence.

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$.

Output: Dictionary $\mathbf{D}^* = [\mathbf{D}_1^* | \mathbf{D}_2^* | \dots | \mathbf{D}_K^*]$.

Algorithm:

1. Repeat the following iterations to refine confidence until the maximum iteration number $\mathrm{T_c}$ is reached:

   1.1 **Confidence Update**: Use (5.6) to calculate the residuals $e_{ij_l}^{(t),i}$, $\forall j_l \in L_i$. Then, update the confidence $p_{i,j_l}^{(t)}$ by (5.2) to obtain the confidence matrix $\mathbf{P}^{(t+1)}$.

   1.2 **Dictionary Update**: Based on $\mathbf{P}^{(t)}$, do the following for each $\mathbf{x}_i$ with $L_i = \{j_1, j_2, \dots j_{|L_i|}\}$: Construct the weighting matrix $\mathbf{W}$.

   Use (5.7) to build $\mathbf{D}_{j_l}^{(t+1),i}$ from which the dictionary

$$\mathbf{D}^{(t+1),i} = [\mathbf{D}_{j_1}^{(t+1),i} | \mathbf{D}_{j_2}^{(t+1),i} | \dots | \mathbf{D}_{j_{|L_i|}}^{(t+1),i}]$$

   is obtained.

2. When $t = \mathrm{T_c}$, follow 1.2 and 1.3 in Algorithm 9 to build the final dictionary $\mathbf{D}^* = \mathbf{D}_c^{(\mathrm{T_c})}$.

5.1.3 DLSD is an EM-based approach

The proposed DLSD is indeed an EM [108], [109], [110] dictionary learning approach. In particular, to find $\mathbf{D}^{(t+1),i}$ given $\mathbf{x}_i$ and $\mathbf{D}^{(t),i}$, in the E-step we first compute the following conditional expectation

$$E\left[\log p(\{\mathbf{x}_l\}_{l=1,l\neq i}^{N}, \{Z_l\}_{l=1,l\neq i}^{N}|\mathbf{D}^i)|\mathbf{x}_i, \mathbf{D}^{(t),i}\right], \tag{5.8}$$

where Z_l is the random variable that corresponds to the true label z_l of the observed sample $\mathbf{x}_l$. We assume the likelihood of sample $\mathbf{x}_l$ given $\mathbf{D}^i$ is a mixture of Gaussian densities expressed by $p(\mathbf{x}_l|\mathbf{D}^i) = \sum_{j=1}^{K} \alpha_j p_j(\mathbf{x}_l|\mathbf{D}_j^i)$, where $\alpha_j' s$ are normalized weights associated with the density of label $j' s$ with $\sum_{j=1}^{K} \alpha_j = 1$, and $p_j(\mathbf{x}_l|\mathbf{D}_j^i) = \frac{1}{\sqrt{2\pi\sigma_j}} \exp\left(-\frac{\|\mathbf{x}_l - \mathbf{D}_j^i \boldsymbol{\gamma}_l\|_2^2}{2\sigma_j}\right)$ for some σ_j. Moreover, $\boldsymbol{\gamma}_l$ is a coefficient vector for representing $\mathbf{x}_l$ using $\mathbf{D}_j^i$. For independent $\mathbf{x}_l' s$, it can be shown that (5.8) equals

$$\sum_{j=1}^{K} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \left(\log(\alpha_j) + \log(p_j(\mathbf{x}_l|\mathbf{D}_j^i))\right), \tag{5.9}$$

where

$$p_{l,j}^{(t)} \triangleq p(Z_l = j|\mathbf{x}_l, \mathbf{D}^{(t),i}) = \frac{\alpha_j p_j(\mathbf{x}_l|\mathbf{D}_j^{(t),i})}{\sum_{k=1}^{K} \alpha_k p_k(\mathbf{x}_l|\mathbf{D}_k^{(t),i})}. \tag{5.10}$$

In the M-step, we maximize (5.9) by finding $\boldsymbol{\alpha}^{(t+1)} \triangleq [\alpha_1^{(t+1)}, ..., \alpha_K^{(t+1)}]$ and $\mathbf{D}^{(t+1),i} = [\mathbf{D}_1^{(t+1),i}|...|\mathbf{D}_K^{(t+1),i}]$ such that

$$\begin{aligned} \boldsymbol{\alpha}^{(t+1)} &= \underset{\boldsymbol{\alpha}=[\alpha_1,\alpha_2,...,\alpha_K]}{\operatorname{argmax}} \sum_{j=1}^{K} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(\alpha_j), \\ &= \underset{\alpha_j}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(\alpha_j), \forall j \in \{1, ..., K\}, \end{aligned} \tag{5.11}$$

$$
\begin{aligned}
\mathbf{D}^{(t+1),i} &= \underset{\mathbf{D}=[\mathbf{D}_1^i|\mathbf{D}_2^i|...|\mathbf{D}_K^i]}{\operatorname{argmax}} \sum_{j=1}^{K}\sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(p_j(\mathbf{x}_l|\mathbf{D}_j^i)), \\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \log(p_j(\mathbf{x}_l|\mathbf{D}_j^i)), \\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmax}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \left(-\frac{\log(\sigma_j)}{2} - \frac{\|\mathbf{x}_l - \mathbf{D}_j^i\boldsymbol{\gamma}_l\|_2^2}{2\sigma_j}\right), \\
&= \underset{\mathbf{D}_j^i}{\operatorname{argmin}} \sum_{l=1,l\neq i}^{N} p_{l,j}^{(t)} \|\mathbf{x}_l - \mathbf{D}_j^i\boldsymbol{\gamma}_l\|_2^2, \ \forall j \in \{1,...,K\}, \\
&= \underset{\mathbf{D}_{j_l}^i}{\operatorname{argmin}} \sum_{m=1}^{N(i,j_l)} w_m \|\mathbf{y}_m - \mathbf{D}_{j_l}^i\boldsymbol{\gamma}_m\|_2^2, \forall j_l \in L_i.
\end{aligned}
\tag{5.12}
$$

The optimization problem in (5.12) can be solved by the weighted K-SVD algorithm in (5.7). $\sigma_{j_l}^{(t+1)}$ can be approximated by the average residual over $\{\mathbf{y}_m\}_{m=1}^{N(i,j_l)}$. That is, $\sigma_{j_l}^{(t+1)} = \frac{1}{\eta(i,j_l)}\sum_{m=1}^{N(i,j_l)} w_m \|\mathbf{y}_m - \mathbf{D}_{j_l}^{(t+1),i}\boldsymbol{\gamma}_m\|_2^2, \forall j_l \in L_i$, where $\eta(i,j_l) = \sum_{m=1}^{N(i,j_l)} w_m$. Moreover, as α_{j_l} sums to one over j_l, (5.11) leads to $\alpha_{j_l}^{(t+1)} = \frac{\sum_{i_m=1}^{N(i,j_l)} p_{i_m,j_l}^{(t)}}{N(i,j_l)}$. We then compute $\beta_{j_l}^{(t+1)} = \frac{\alpha_{j_l}^{(t+1)}}{\sqrt{2\pi\sigma_{j_l}^{(t+1)}}}$, and update $p_{i,j_l}^{(t+1)}$ by (5.2).

5.1.4 Determining initial dictionaries

The performance of both DLSD and DLHD will depend on the initial dictionaries as they determine how well the final dictionaries are learned through successive alternating iterations. As a result, initializing our method with proper dictionaries is critical. In this section, we propose an algorithm that uses both ambiguous labels and features to determine the initial dictionaries.

For the i-th sample, we initialize the corresponding row of $\mathbf{P}$ uniformly for all

$j \in L_i$. Hence,

$$\mathbf{P}^{(0)} \triangleq \left[p_{i,j}^{(0)}\right], \text{ where } p_{i,j}^{(0)} = \frac{1}{|L_i|}, \text{ if } j \in L_i, i = 1, \ldots, N.$$

At iteration $t = 0$, we build dictionaries for the sample $\mathbf{x}_i$, denoted by $\mathbf{D}^{(0),i} = [\mathbf{D}_{j_1}^{(0),i}|\mathbf{D}_{j_2}^{(0),i}|\ldots|\mathbf{D}_{j_{|L_i|}}^{(0),i}]$, where the intermediate dictionary $\mathbf{D}_{j_k}^{(0),i}$ is learned from samples other than $\mathbf{x}_i$ with ambiguous label $j_k \in L_i$. These samples are collected in the same way as described in section 5.1.2. Next, $\mathbf{x}_i$ is assigned to class $\hat{j}^i$ such that it gives the lowest residual. In other words,

$$\hat{j}^i = \underset{j_k \in L_i}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{D}_{j_k}^{(0),i}\overline{\mathbf{D}_{j_k}^{(0),i}}\mathbf{x}_i\|_2^2. \tag{5.13}$$

Initial clusters are obtained after the class assignment for all samples is completed. Each initial dictionary is then learned from the corresponding cluster using the K-SVD algorithm [34]. We summarize this initialization approach in Algorithm 11.

Note that our method is very different from the approach that learns dictionaries from partially labeled data [105]. The work in [105] learns class discriminative dictionaries while our work learns class reconstructive dictionaries. In addition, from the formulation in [105] we see there are either labeled samples or totally unlabeled samples available for training. In contrast, in our partially labeled formulation, all samples are ambiguously labeled according to three controlled parameters. In fact, formulations in [105] and [15] (for totally unlabeled samples) are special cases of the ambiguously labeled formulation presented in our work.

Algorithm 11: Using initial confidence to learn initial dictionaries.

Input: Training samples $\mathcal{L} = \{(x_i, L_i)\}$ and the initial confidence, $\mathbf{P}^{(0)}$.

Output: Initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}]$.

Algorithm:

1. Initialization: $i \leftarrow 1$; $\mathbf{C}_j^{(0)} \leftarrow \{\}, \forall j \in \{1, 2, \dots, K\}$.
2. Repeat the following for every $\mathbf{x}_i$:
   2.1 Construct $\mathbf{D}^{(0),i} = [\mathbf{D}_{j_1}^{(0),i}|\mathbf{D}_{j_2}^{(0),i}|\dots|\mathbf{D}_{j_{|L_i|}}^{(0),i}]$, where $\mathbf{D}_{j_k}^{(0),i}$ is built from $\mathbf{x}_l$'s such that $l \neq i$.
   2.2 Augment $\mathbf{C}_{\hat{j}^i}^{(0)}$ with $\mathbf{x}_i$, where $\hat{j}^i$ is obtained from (5.13).
3. Establish initial dictionaries $\mathbf{D}^{(0)} = [\mathbf{D}_1^{(0)}|\mathbf{D}_2^{(0)}|\dots|\mathbf{D}_K^{(0)}]$, where $\mathbf{D}_j^{(0)}$ is learned from $\mathbf{C}_j^{(0)}$ using the K-SVD algorithm.

5.2 Experiments

To evaluate the performance of the proposed dictionary method, we performed two sets of experiments defined in [4] [5]: inductive experiments and transductive experiments. We report the average test error rates (for inductive experiments) and the average labeling error rates (for transductive experiments), which were computed over 5 trials.

In an inductive experiment, samples are split in half into a training set and a test set. Each sample in the training set is ambiguously labeled according to controlled parameters, while each sample in the test set is unlabeled. In each trial, using the learned dictionaries from the training set, the test error rate is calculated as the ratio of the number of test samples that are erroneously labeled, to the total number of test samples. In a transductive experiment, all samples with ambiguous

labels are used to train the dictionaries. In each trial, the labeling error rate is calculated as the ratio of the number of training samples that are erroneously labeled, to the total number of training samples.

Following the notations in [5], the controlled parameters are: p (proportion of ambiguously labeled samples), q (the number of extra labels for each ambiguously labeled sample) and ϵ (the degree of ambiguity - the maximum probability of an extra label co-occurring with a true label, over all labels and inputs [5]). We selected the following three datasets for performance evaluations: Labeled Faces in the Wild (LFW) [111], the CMU PIE dataset [112] and the TV series 'LOST' dataset [5].

(a)

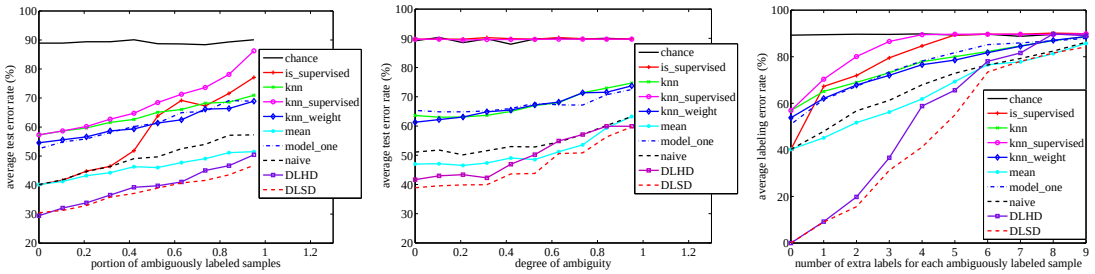
(b)

Figure 5.2: (a) FIW(10b) 10-class dataset. (b) CMU PIE 18-class dataset - left: first 9 classes, right: second 9 classes. In each dataset, face images belonging to the same class are shown in a row.

5.2.1 Labeled Faces in the Wild dataset

The LFW database [111] was originally designed to address pair matching problems. Cropped and resized images of the LFW database were provided by the authors of [5]. In our experiment, we use one of the resulting subsets, FIW(10b), a balanced subset which contains the first 50 images for each of the top 10 most frequent subjects [5]. Fig. 5.2(a) shows this dataset, where faces of the same subject are shown in one row. We resized each image to 55×45 pixels, and took the histogram equalized column-vector (2475×1) as input features. Figures 5.3(a) and (b) show average test error rates (for inductive experiments) of the proposed dictionary method (DLHD and DLSD) versus p and ϵ, respectively. For comparison, in the same figure we show the average test error rates of other existing baseline methods[3] reported in [4], [5]. Both dictionary methods are comparable to the Convex Learning from Partial Labels (CLPL) method (denoted as 'mean') [5]. Fig. 5.3(c) shows the average labeling error rates (for transductive experiments) versus q curves. The DLHD method outperforms the other compared methods when the number of extra labels is less than or equal to 5. The DLSD approach gives slightly better performance than the DLHD approach.

[3]As definitions of these baselines can be found in [4], [5], these definitions are not described again here due to space limitation.



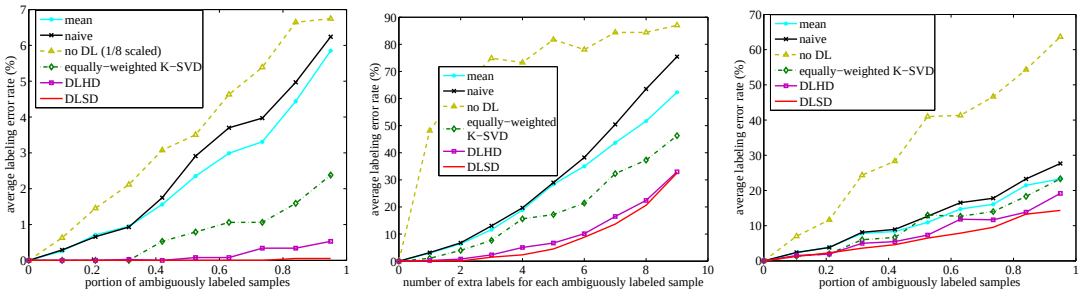
(a) (b) (c)

Figure 5.3: Performance of the proposed dictionary methods and other baselines [4], [5] on the LFW dataset. (a) Average test error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95], q = 2$, inductive). (b) Average test error rates versus the degree of ambiguity for each ambiguously labeled sample ($p = 1, q = 1, \epsilon \in [1/(L-1), 1]$, inductive). (c) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, ..., 9]$, transductive). The proposed dictionary methods are comparable to the CLPL method ('mean').

5.2.2 CMU PIE dataset

The PIE dataset was designed for illumination challenges. The dataset contains 21 images under varying illumination conditions of 68 subjects. We took the first 18 subjects for our experiments and the resulting dataset is shown in Fig. 5.2(b), where each row presents images of the same subject under various illumination conditions. All images are resized to 48×40 and projected onto a 181-dimension subspace that is spanned by the 5th to the 185th eigenvectors obtained through the principle component analysis (PCA). Figures 5.4(a) and (b) show the average labeling error rates versus p and q in transductive experiments. We compare the proposed method with the CLPL method (denoted as ('mean') and 'naive' methods) [4], [5][4]. Clearly, when either p or q is zero in transductive experiments, there exist no ambiguous labels and hence the labeling errors are zero. In Fig. 5.4(a), all compared methods provides good labeling performances. When 95% of samples are ambiguously labeled, the lowest average error labeling rate, 0.05%, is achieved by the DLSD approach. As shown in Fig. 5.4(b), both DLHD and DLSD outperform other compared methods for all numbers of extra labels.

[4]We obtained the code for CLPL ('mean') and 'naive' methods from `http://www.timotheecour.com/`. Both the 'naive' method and the normalized 'naive' method [103] give very similar results [5].



(a) (b) (c)

Figure 5.4: Performance of the proposed dictionary methods, two baseline methods (no dictionary learning - 'no DL', and standard K-SVD - 'equally-weighted K-SVD'), CLPL ('mean') and 'naive' methods [4], [5] on transductive experiments. (a) and (c) Average labeling error rates versus the proportion of ambiguously labeled samples ($p \in [0, 0.95], q = 2$) on the PIE and LOST datasets, respectively. (b) Average labeling error rates versus the number of extra labels for each ambiguously labeled sample ($p = 1, q \in [0, 1, ..., 9]$) on the PIE dataset.

5.2.3 TV series 'LOST' dataset

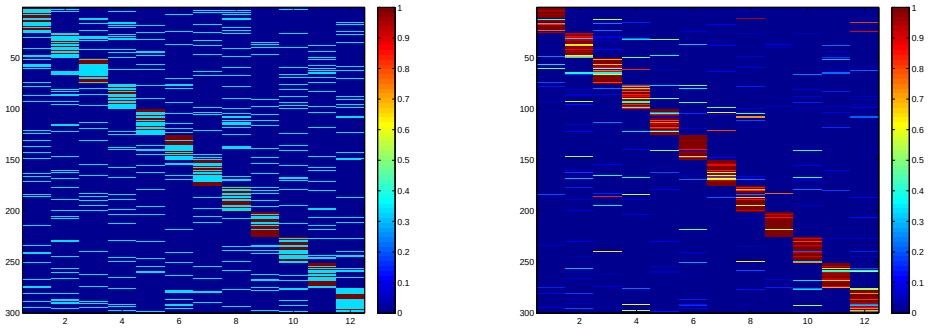
We obtained the cropped face images of TV series 'LOST' that were provided on-line by the authors of [5]. The original dataset contains 1122 registered face images across 14 subjects, and each subject contains from 18 up to 204 face images. In our experiment, we chose 12 subjects with at least 25 faces images per subject and for each chosen subject, we collected his/her first 25 face images. We resized each image to 30×30 pixels, and took the histogram equalized column-vector (900×1) as input features. Fig. 5.4(c) show the average labeling error rates versus p curves in transductive experiments. It is observed that when 95% of samples are ambiguously labeled, DLSD achieves the lowest error labeling rate, of 14.33%.

5.2.4 Discussions

To explain the performance gain of our dictionary learning approach, in all three plots of Fig. 5.4, we show the plots of two additional baseline methods: 'no dictionary learning (DL)' and 'equally-weighted K-SVD'. The 'no DL' method utilizes features and ambiguous labels only, without learning dictionaries. This baseline collects for each class c, all its possible samples (i.e, $\mathbf{x}_i$'s with $p_{i,c}^{(t)} > 0$) at each iteration t, and uses them directly as a set of basis atoms. The 'equally-weighted K-SVD' method contrasts the DLSD method by simply using equal weights among possible samples of each label for dictionary learning. In other words, it ignores the weighting matrix $\mathbf{W}$ in (5.7) and learns dictionaries by the standard K-SVD algorithm. Reconstruction errors for both baseline methods are computed

using the same L-2 norm as in (5.6) to update the confidence. These figures show that the 'no DL' method was not able to obtain satisfactory results. The 'equally-weighted K-SVD' method did not perform as well as DLHD and DLSD. In particular, the performance degradation of the 'equally-weighted K-SVD' method highlights the importance of $\mathbf{W}$ computed from the DLSD method. Comparing DLHD and DLSD, we observe that DLHD performs not as well as the DLSD in that the hard-threshold confidence in DLHD is locally constrained, and hence it may not give the global optimal $\mathbf{W}$ for dictionary learning. In addition, while the state-of-the-art CLPL ('mean') method may be sensitive to face images with certain within-class variation due to illumination changes (e.g., in Fig. 5.2(b), (c)) and noise, the learned dictionary atoms in the proposed method are able to account for these variations to some degree. Therefore, the performance of our dictionary-based approach is better than those of the CLPL ('mean') and other compared baseline methods.

Moreover, in order to examine the updates of the confidence matrices, in Fig. 5.5, we further show the initial (at $t = 0$) and updated (using DLSD at $t = 20$) confidence matrices corresponding to this experiment, where samples and labels are indexed vertically and horizontally, respectively. Without any prior knowledge, ambiguously labeled samples have equally probable initial confidences. At $t = 20$, we observe that the updated confidences for most samples tend to converge as they become impulse-shape where the confidence value is 1 for one label, and zero for the other labels.



(a) (b)

Figure 5.5: Initial and updated confidence matrices on the TV series 'LOST' (12-class) dataset. (a) Initial confidence, $\mathbf{P}^{(0)}$. (b) $\mathbf{P}^{(20)}$ (using DLSD at $t = 20$). While ambiguously labeled samples have equally probable initial confidences, the updated confidences at $t = 20$ become impulse-shape (i.e., confidence value is 1 for one label, and zero for other labels) for most samples.

5.3 Summary

Dictionary learning methods have been shown to be state-of-the-art in many supervised, unsupervised and semi-supervised classification problems. We have extended it to the case of ambiguously labeled learning, where each example is supplied with multiple labels, only one of which is correct. The proposed method iteratively estimates the confidence of samples belonging to each of the classes and uses it to refine the learned dictionaries. Experiments using three publicly available datasets demonstrate the improved accuracy of the proposed method compared to state-of-the-art ambiguously labeled learning techniques.

Chapter 6: Salient Views and Geometric Dictionaries for Object Recognition

The concept of characteristic views was first proposed in [113], [114] for object recognition. This concept was defined in such a way that two views belonging to the same Characteristic View Class (CVC) are topologically equivalent, and they can be related by a 3D transformation. The transformation consists of geometric rotation, translation and perspective projection [115]. [115] proposes a framework to partition the viewing space and to find the set of characteristic views for planar-faced solid objects. This work was later extended in [116], which essentially computes the characteristic views of objects with curved-surface.

There are a number of approaches for describing what is contained in a view [21], [22]. For view-based representations, human perceivers are influenced by factors such as familiarity with the object being viewed, the similarity of a given view to known views of visually-similar objects and the pose of the object [21]. Three-quarter views with all visible front, top and side, are often used as candidate views[1].

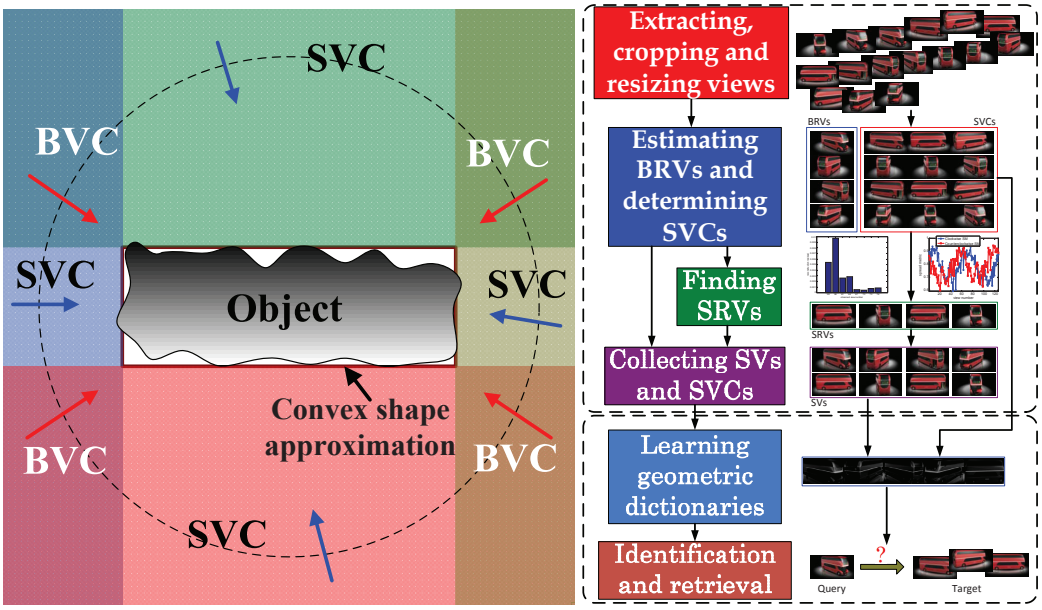
[1]In the viewing space there are in fact infinite number of viewpoints. Candidate views are views seen from a (possibly large but) finite subset of viewpoints [23]. Given a view descriptor, the objective in [23] is to find the maximum of this descriptor among the candidate views.

As noted in [23], three-quarter views are essentially the views that most humans prefer when looking at an object. These views are also known as the *canonical views* [22].

In [117], saliency was defined as the amount of energy not captured by the basis set in an eigenspace representation. A greedy algorithm was proposed for subset selection where the saliency of every ensemble view is first computed and then the view with the highest saliency is added to the subset. The subset is then updated using the eigenspace representation updating algorithm [118], [119] so that the task of salient view selection can be realized in a dynamic environment.

We propose a sparse representations based approach for selecting the salient views of an object [22], [23]. Given an object, we assume its shape can be approximated by a simple convex polygon with multiple number of sides. A side view class (SVC) is defined as the set of all views of the corresponding side of the shape, while a boundary view class (BVC) refers to views where two or more sides can be seen simultaneously. Fig. 6.1(a) illustrates distinct regions of SVCs and BVCs given an approximate convex polygon shape for an object. The shape in this polygon consists of four sides, which give four SVCs and four BVCs under orthographic projection. These eight classes are exactly the eight CVCs of the approximate convex polygon shape. Using the object's approximate convex polygon and its sides, we categorize salient views into two categories: *boundary representative views* (BRVs) which have more visible sides and object surfaces, and therefore are more attractive from a human perception point of view; and *side representative views* (SRVs) which best describe the underlying SVCs. In Fig. 6.1(a), BRVs and SRVs are views seen from

directions marked with red and blue arrows, respectively. Fig. 6.1(b) shows the block diagram of the proposed two-stage approach for finding the salient views. Views are extracted from a video sequence, cropped and properly resized. In the first stage (in blue) the boundary scores are computed using a sparsity-based spread metric to estimate BRVs and determine SVCs. In the second stage (in green), for each side a set of SRVs that best represent a corresponding side are chosen by minimizing a representation error.



(a) (b)

Figure 6.1: (a) Convex polygon shape approximation and the associated SVC/BVC regions. (b) Block diagram of the proposed salient view selecting approach and its application to object recognition using geometric dictionaries.

The BRVs and SRVs are the salient views (SVs). Important applications of SVs include object recognition and retrieval. In these applications, objects are retrieved or classified from different perspective views. To show the effectiveness of

our method, we introduce the notion of geometric dictionaries (in light blue) that are built using the SVs and SVCs. These geometric dictionaries can then be used for 3D object recognition and retrieval applications (in light brown).

Key contributions of this chapter are:

1. We propose a sparse representation-based approach for selecting the salient views of an object.

2. Our method is based on characteristic views. It selects representative views of visible sides and object surfaces.

3. We introduce the notion of geometric dictionaries based on the salient views for object recognition and retrieval.

4. We demonstrate the effectiveness of our approach on four publicly available 3D object datasets.

This chapter is organized as follows. In section 6.1, we present our method for estimating the BRVs of an object. In Section 6.2, we describe our approach for determining the SRVs. In section 6.3, we detail the geometric dictionary learning method using SVs, and its application to object recognition and retrieval. Experimental results and discussions on recognition using geometric dictionaries as well as sparse-to-full reconstruction, are presented in section 6.4. Section 6.5 concludes the chapter with a brief summary and future work.

6.1 Estimating boundary representative views

It has been shown [115] that for a convex planar-faced solid object, planes obtained by expanding the object's faces partition the viewing space. These partitions are referred to as *type-A planes*. These planes are used to partition the viewing space into regions called characteristic view domains (CVDs). Whenever two views belong to the same CVD, every viewable point in one view is also viewable in the other view, and vice versa. Using this idea on the assumed approximate convex polygon shape of a given object, we use a metric called spread metric, to compute the boundary scores. The views that give the maximum boundary scores are selected as BRVs. In what follows, we describe the spread metric and the selection of BRVs.

Spread Metric: As the viewing space of an object contains infinite number of viewpoints, there are infinite number of views of the object. Hence, we restrict our objective to search for salient views among a finite number of views. Let $\mathbf{S}_m$, where $m \in \{1, 2, ..., N\}$, be a finite subset of the m-th SVC. $\mathbf{S}_m$ consists of a finite number of views that are approximately topologically equivalent as they can be related by 3D transformations. We can further subdivide $\mathbf{S}_m$ into exclusive subgroups $\mathbf{s}_i$'s such that $\mathbf{S}_m = \mathbf{s}_1 \bigcup \mathbf{s}_2 \bigcup ... \bigcup \mathbf{s}_{k_m}$ and $\mathbf{s}_i$ contains views that are fairly close to each other as they are viewed from locations with small rotation or translation differences. Let $\mathbf{z}$ denote a candidate view. We use a spread metric denoted by SM $= 1 -$ SCI, to represent the saliency of $\mathbf{z}$ relative to $\mathbf{S}_m$, where SCI stands for

Sparsity Concentration Index defined in [9], [120]. SCI is a measure of sparsity of the coefficient representation of a vector under some basis. Low values of SCI (i.e., high SM) indicate that the given view is fairly informative relative to the existing group. The SM of $\mathbf{z}$ relative to $\mathbf{S}_m$ is computed by

$$\mathrm{SM}_m(\mathbf{z}) = \frac{k_m \left(1 - \max\limits_{i \in \{1,2,\ldots,k_m\}} \frac{\|\delta_{m,i}(\mathbf{x}_m)\|_1}{\|\mathbf{x}_m\|_1}\right)}{k_m - 1}, \tag{6.1}$$

where $\mathbf{x}_m$ is the representation of the candidate view $\mathbf{z}$ under $\mathbf{S}_m$ and $\|\mathbf{x}\|_1 = \sum_i |x_i|$. That is,

$$\mathbf{x}_m = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{z} = \mathbf{S}_m \mathbf{x}. \tag{6.2}$$

In (6.1), $\delta_{m,i}(\mathbf{x}_m)$ is a vector whose only nonzero entries are the entries of $\mathbf{x}_m$ that belong to the i-th subgroup of $\mathbf{S}_m$. It can be shown that $\mathrm{SM}_m(\mathbf{z}) \in [0, 1]$. The larger the $\mathrm{SM}_m(\mathbf{z})$ is, the larger the saliency possessed by $\mathbf{z}$ relative to $\mathbf{S}_m$. Large $\mathrm{SM}_m(\mathbf{z})$ is a strong indication that $\mathbf{z}$ belongs to a subset different from $\mathbf{S}_m$.

Finding boundary representative views: In this section, we describe our method for finding the BRVs. We consider only the 3D views of an object with respect to the Y axis rotation ($0^\circ \sim 360^\circ$) under the orthographic projection. Initially, no knowledge on $\mathbf{S}_m$ is given. The spread metric of a candidate view is instead computed relative to a set of views within a sliding window (i.e. a set of views with consecutive view indices) on the path of rotation. We refer to the spread metric that is a function of the sliding window as the boundary score. The BRVs are the views with the maximum boundary scores. Without loss of generality, let $\{\mathbf{z}_j\}_{j=0}^{M-1}$ be the original full 3D views of a given object in the clockwise positive direction

(i.e., as j increases, it goes in the clockwise direction[2]) where M is the number of full views (FVs). After $\mathbf{z}_{M-1}$, the sequence rounds back to $\mathbf{z}_0$ as these are rotated views with respect to the Y axis. Now, for any given $\mathbf{z}_j$, we calculate its boundary score as follows:

$$\widetilde{\mathrm{SM}}_{W_{j(\beta,\gamma)}}(\mathbf{z}_{j(\alpha)}) = \frac{\gamma\left(1 - \max\limits_{i\in\{1,2,\ldots,\gamma\}} \frac{\|\delta_{W,i}(\mathbf{x}_{W,j(\alpha)})\|_1}{\|\mathbf{x}_{W,j(\alpha)}\|_1}\right)}{\gamma - 1}, \tag{6.3}$$

where $j(\alpha) \triangleq \mathrm{mod}(j+\alpha, L)$, and

$$W_{j(\beta,\gamma)} \triangleq \left(\mathbf{z}_{j(-\beta-\gamma+1)}\ \mathbf{z}_{j(-\beta-\gamma+2)}\ \ldots\ \mathbf{z}_{j(-\beta)}\right). \tag{6.4}$$

In (6.3), $\mathbf{x}_{W,j(\alpha)}$ is the representation of the candidate view $\mathbf{z}_{j(\alpha)}$ under $W_{j(\beta,\gamma)}$. That is,

$$\mathbf{x}_{W,j(\alpha)} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ s.t. } \ \mathbf{z}_{j(\alpha)} = W_{j(\beta,\gamma)}\mathbf{x}. \tag{6.5}$$

Similar to $\delta_{m,i}(\mathbf{x}_{m,j})$ in (6.1), here $\delta_{W,i}(\mathbf{x}_{W,j(\alpha)})$ is a masked version of $\mathbf{x}_{W,j(\alpha)}$ such that its only nonzero entry is the one that corresponds to the i-th column vector of $W_{j(\beta,\gamma)}$. In this setting, for a given $\mathbf{z}_j$, we calculate the SM of the view ahead of it by α units of indices, with respect to the set formed from the $(\beta+\gamma-1)$-th view up to the β-th view behind $\mathbf{z}_j$. That is, this set is formed according to a β-index logged window with size γ.

Figure 6.2 illustrates how we compute the boundary score. Consider two SVCs: m-th SVC (in color purple) and $(m+1)$-th SVC (in color yellow), and one BVC in between. Since in the beginning no information on SVCs is provided, the choice of basis is unknown and no spread metric can be calculated. Instead, we use a sliding

[2]The same analysis follows for the counterclockwise position assumption as well.

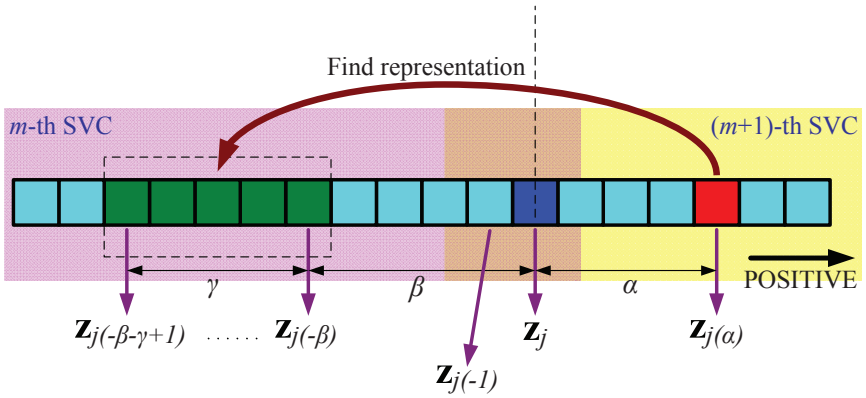


Figure 6.2: An illustration of finding the boundary score.

window with a predetermined size γ to select views and form $W_{j(\beta,\gamma)}$ (consisting of views in color green). To find the boundary score at $\mathbf{z}_j$, we calculate the spread metric of $\mathbf{z}_{j(\alpha)}$ which leads $\mathbf{z}_j$ by α units of views, with respect to $W_{j(\beta,\gamma)}$ which lags $\mathbf{z}_j$ by β units of views. Note that α and β should be properly tuned according to not only the complexity of object but also the view sampling interval. If α and β are too small, the spread metric is not obvious as $\mathbf{z}_{j(\alpha)}$ is close to a member of $W_{j(\beta,\gamma)}$. On the other hand, whenever α and β are too large, so are the spread metric since $\mathbf{z}_{j(\alpha)}$ is close to none of $W_{j(\beta,\gamma)}$. In both these cases the spread metric can no longer be a discriminative measure for BRVs. With properly chosen α and β, one could expect the boundary score at $\mathbf{z}_j$ when $\mathbf{z}_j$ is in the BVC (i.e., overlapped region) to be higher than those when $\mathbf{z}_{j(\alpha)}$ and members in $W_{j(\beta,\gamma)}$ are in the same SVC.

6.2 Side representative view(s) selection

Representative views can either be interpreted as a sparse representation (i.e., coefficients) under some basis, or can be used as sparse observation where sparse

coefficients under some basis can be found. In this section, with representative views regarded as sparse observations, we propose a procedure for finding an object-dependent basis set. We assume that camera parameters are not known.

We assume that distinct SVCs are independent of each other. Without loss of generality, we consider the first SVC, $[\mathbf{z}_0\ \mathbf{z}_1\ ...\ \mathbf{z}_{k_1-1}]$. Its singular value decomposition (SVD) is $[\mathbf{z}_0\ \mathbf{z}_1\ ...\ \mathbf{z}_{k_1-1}] = \mathbf{V}\Sigma\mathbf{U}^T$, where $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ ...\ \mathbf{v}_L]$ is an L-by-L matrix (L is total number of pixels of an image); $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ ...\ \mathbf{u}_{k_1}]$ is a k_1-by-k_1 matrix; and $\sigma_1, ...\sigma_{k_1}$ are the eigenvalues (i.e., first k_1 diagonal entries of Σ). It can be shown that

$$\begin{aligned}\mathbf{y}_1 \triangleq \begin{pmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_{k_1-1} \end{pmatrix} &= \begin{pmatrix} | & | & ... & | \\ \mathbf{c}_1 & \mathbf{c}_2 & ... & \mathbf{c}_{k_1} \\ | & | & ... & | \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{k_1} \end{pmatrix} \\ &= \begin{pmatrix} — & \mathbf{Q}_1 & — \\ — & \vdots & — \\ — & \mathbf{Q}_{k_1} & — \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{k_1} \end{pmatrix} = \mathbf{R}\mathbf{w},\end{aligned} \tag{6.6}$$

where $\mathbf{c}_i$ ($i \in \{1, ..., k_1\}$) is the column-vectorized form of matrix $\mathbf{v}_i\mathbf{u}_i^T$, and each $\mathbf{Q}_j$ ($j \in \{1, ..., k_1\}$) is a L-by-k_1 matrix. Note that $\mathbf{Q}_j$ is not a 1-by-k_1 row vector. In (6.6), the matrix $\mathbf{R}$ is an object-dependent basis set, and $\mathbf{w}$ contains eigenvalues $\sigma_1, ...\sigma_{k_1}$ as coefficients.

Our objective is to select l_1 out of k_1 views as representative views that best represent the SVC, where $l_1 < k_1$. There are $\binom{k_1}{l_1}$ possible ways to select them. Consider one way in which the selected views are $\mathbf{z}_{s_1}, ..., \mathbf{z}_{s_{l_1}}$, which form a column vector $\mathbf{y}_s$. Next, we pick the corresponding matrices $\mathbf{Q}_{s_1}, ..., \mathbf{Q}_{s_{l_1}}$, to form a sub-basis

$\mathbf{R}_s$:

$$\mathbf{y}_s \triangleq \begin{pmatrix} \mathbf{z}_{s_1} \\ \vdots \\ \mathbf{z}_{s_{l_1}} \end{pmatrix}; \qquad \mathbf{R}_s \triangleq \begin{pmatrix} \mathbf{Q}_{s_1} \\ \vdots \\ \mathbf{Q}_{s_{l_1}} \end{pmatrix}. \tag{6.7}$$

We solve the following equation using the ℓ_1 norm:

$$\hat{\mathbf{x}}_{(\mathbf{y}_s)} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y}_s = \mathbf{R}_s\mathbf{x}. \tag{6.8}$$

Since $l_1 < k_1$, less constraints are involved in solving (6.8) than in (6.6), and one would expect $\hat{\mathbf{x}}_{(\mathbf{y}_s)}$ to be sparser than $\mathbf{w}$. Among all possible $\binom{k_1}{l_1}$ ways, the one which gives the least sparse-to-full reconstruction residual is chosen. In other words, we seek

$$\hat{\mathbf{y}_s} = \arg\min_{\mathbf{y}_s} \|\mathbf{y}_1 - \mathbf{R}\hat{\mathbf{x}}_{(\mathbf{y}_s)}\|_2. \tag{6.9}$$

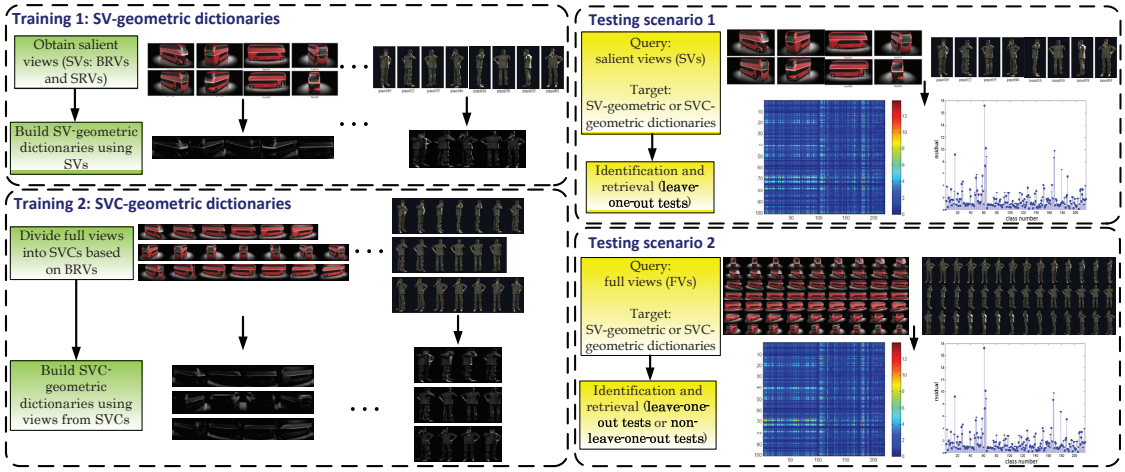
The corresponding best reconstruction is closest to $\mathbf{y}_1$, and can be thought of as the one directly reconstructed using sparse observations from these l_1 representative views. It has sparse representation $\hat{\mathbf{x}}_{(\hat{\mathbf{y}_s})}$ under the basis $\mathbf{R}$ defined in (6.6).

6.3 Geometric Dictionaries

Important applications of SVs include object recognition and retrieval where one wants to recognize or retrieve images having the same object while taken from different perspectives [8], [3], [27]. We introduce the notion of geometric dictionaries for this application.

Geometric dictionaries are dictionaries that geometrically represent a 3D object based on views taken from the object's full geometric perspectives, and mean-

while, remove the 3D redundancy. The geometric dictionaries hence can be built either from SVs (i.e., BRVs and SRVs), or from views belonging to SVCs. Here, we refer to the dictionaries built from SVs and the dictionaries built from SVCs by SV-geometric dictionaries and SVC-geometric dictionaries, respectively.



(a) (b)

Figure 6.3: Illustration of different training and testing scenarios for recognition/retrieval. (a) Two training scenarios: SV-geometric and SVC-geometric dictionaries. (b) Two testing scenarios: SVs vs. SV/SVC-geometric dictionaries and FVs vs. SV/SVC-geometric dictionaries.

6.3.1 SV-geometric dictionaries

Let the SVs of the ith object be denoted by $\{\mathbf{a}_l^i\}_{l=1}^{n_i}$. Then, the following optimization problem can be solved to obtain the corresponding SV-geometric dictionary

$\mathbf{B}_i$:

$$(\mathbf{B}_i, \mathbf{\Lambda}_i) = \arg\min_{\mathbf{B},\mathbf{\Lambda}} \|\mathbf{A}_i - \mathbf{B}\mathbf{\Lambda}\|_F^2, \text{ s.t. } \|\boldsymbol{\lambda}_{i,l}\|_0 \leq T_0,$$
$$\forall l \in \{1, ..., n_i\}, \forall i \in \{1, ..., P\}, \tag{6.10}$$

where P is the total number of objects (i.e. classes) in the target gallery, $\boldsymbol{\lambda}_{i,l}$ represents the l^{th} column of $\mathbf{\Lambda}_i$, $\mathbf{A}_i$ is the matrix whose columns are $\mathbf{a}_l^i$s and T_0 is the sparsity parameter. Here, $\|\mathbf{A}\|_F$ is the Frobenius norm of matrix $\mathbf{A}$ defined by $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{ij}^2}$, and the norm $\|\boldsymbol{\lambda}\|_0$ counts the number of non-zero elements in $\boldsymbol{\lambda}$.

We use the K-SVD algorithm for learning the geometric dictionaries. Please refer to Section 4.2.2.1 for more details on the K-SVD algorithm.

6.3.2 SVC-geometric dictionaries

Let $\mathbb{C}_{i,j}$ be the j-th SVC of the i-th object, and $\mathbf{C}_{i,j}$ be the corresponding matrix that contains views (each in a column-vectorized form) as its columns. Using the K-SVD algorithm, we learn a sub-dictionary $\mathbf{D}_{i,j}$ that best represents $\mathbf{Y}_{i,j}$ by solving the following optimization problem

$$(\mathbf{D}_{i,j}, \mathbf{\Gamma}_{i,j}) = \arg\min_{\mathbf{D},\mathbf{\Gamma}} \|\mathbf{C}_{i,j} - \mathbf{D}\mathbf{\Gamma}\|_F^2, \text{s.t.} \|\boldsymbol{\gamma}_l\|_0 \leq T_0,$$
$$\forall l \in \{1, ..., k_j^{(i)}\}, \forall j \in \{1, ..., m_i\}, \forall i \in \{1, ..., P\}, \tag{6.11}$$

where m_i is the number of SVCs, $k_j^{(i)}$ is the number of views belonging to $\mathbb{C}_{i,j}$, $d_j^{(i)}$ is the number of dictionary atoms, and $\mathbf{\Gamma}_{i,j}$ is a $d_j^{(i)} \times k_j^{(i)}$ coefficient matrix that contains $\boldsymbol{\gamma}_l$s as its columns. We concatenate $\mathbf{D}_{i,j}$s to form a SVC-geometric

dictionary $\mathbf{D}_i$. In other words, $\mathbf{D}_i = [\mathbf{D}_{i,1}\ \mathbf{D}_{i,2}\ ...\ \mathbf{D}_{i,m_i}]$.

6.3.3 View-based object identification and image retrieval

In this section, we show how the geometric dictionaries can be used for view-based object recognition and retrieval.

Recognition: Given a query $\mathbf{h}$, in a particular view, we project it onto the span of the atoms in each geometric dictionary. Let $\mathbf{E}_i$ be the ith target class's geometric dictionary (either SV-geometric dictionary $\mathbf{B}_i$ or SVC-geometric dictionary $\mathbf{D}_i$, depending on applications). The approximation and residual vectors can then be calculated as

$$\mathbf{h}^i = \mathbf{E}_i \mathbf{E}_i^{\dagger} \mathbf{h}, \tag{6.12}$$

and

$$\mathbf{r}^i(\mathbf{h}) = \mathbf{h} - \mathbf{h}^i = (\mathbf{I} - \mathbf{E}_i(\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T)\mathbf{h}, \tag{6.13}$$

respectively, where $\mathbf{E}_i^{\dagger} \triangleq (\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T$ is the pseudoinverse of $\mathbf{E}_i$, and $\mathbf{I}$ is the identity matrix. As $\mathbf{E}_i$ leads to the best representation for the ith target object, it is assumed that $\|\mathbf{r}^i(\mathbf{h})\|_2$ will be small if $\mathbf{h}$ belongs to the ith class and larger for the other classes. Therefore, if

$$i^* = \arg \min_{1 \le i \le P} \|\mathbf{r}^i(\mathbf{h})\|_2, \tag{6.14}$$

then $\mathbf{h}$ is identified as belonging to the i^*th class in the target gallery as the corresponding geometric dictionary gives the minimum reconstruction error.

Retrieval: For image retrieval, we search for the relevance of $\mathbf{h}$ among the views belonging to the i^*th target class by a G-nearest-neighbor criterion, where G is

the number of retrieved images for **h**. The resulting geometric dictionary-based recognition or retrieval algorithm is denoted as GDR.

Fig. 6.3 (a) illustrates two training scenarios for building SV-geometric dictionaries and SVC-geometric dictionaries, respectively. In the testing phase, the query views can either be SVs or FVs. These two testing scenarios are illustrated by Fig. 6.3 (b). We refer to our Salient View selection based on Sparse Representation approach as SVSR. Algorithm 12 summarizes the overall procedure of the proposed SVSR with GDR for object recognition and retrieval using salient views and geometric dictionaries.

6.4 Experimental results

In this section, we demonstrate the performance of our method in finding salient views as well as object recognition and retrieval on 3D video sequences. All 3D video sequences used in our experiments are sequences of still images taken at regular intervals of $0^\circ \sim 360^\circ$ and $0^\circ \sim 180^\circ$ (with respect to the Y axis) for objects and faces, respectively.

6.4.1 Salient Views

We selected three available sequences of 3D videos for our experiments on salient view selection: the BUS sequence[3], the HEAD sequence[4] and the JONES

[3]http://vimeo.com/3066167

[4]http://vimeo.com/15198240

Algorithm 12: The proposed SVSR with GDR.

Input: Full 3D views of the target gallery, and query views.

Algorithm:

1. For each view of the ith target object, use (6.3) to compute the boundary score. Choose views with the highest boundary scores as BRVs.

2. Use BRVs to divide the FVs into SVCs. For each SVC, use (6.9) to find its class representative views. Then obtain SRVs for all SVCs.

Training:

3. Collect SVs (i.e., BRVs and SRVs) and SVCs. Use (6.10) and (6.11) to build SV-geometric dictionary $\mathbf{B}_i$ and SVC-geometric dictionary $\mathbf{D}_i$, respectively.

4. Repeat **1**, **2** and **3** for all objects (classes) in the target gallery.

Testing:

5. Recognition and retrieval - For each query view, determine the closest target class by (6.14), from which the relevances can be found by the nearest neighbor criterion.

Output:

1. SVs, SVCs and geometric dictionaries of the target gallery.

2. The closest target class and the relevance to each the query views.

sequence[5]. A given video is converted into a set of images, each of which is one view of the object at some particular rotation angle with respect to the Y axis, ranging from 0° to 360°. Images are cropped and resized in the preprocessing stage. Figure 6.4 shows these sequences of images. There are 126 views (2.85° increment per view), 32 views ($\sim$ 11.25° increment per view), and 51 views ($\sim$ 7.05° increment per view) for BUS, HEAD and JONES sequences, respectively. In these figures, the sequence of images going from the left to the right in each row, and then from the top row to the bottom row, corresponds to the (camera) clockwise direction. We calculate the spread metrics with $W_{\beta,\gamma}$ sliding in both clockwise (positive) direction, and counterclockwise (negative) direction.

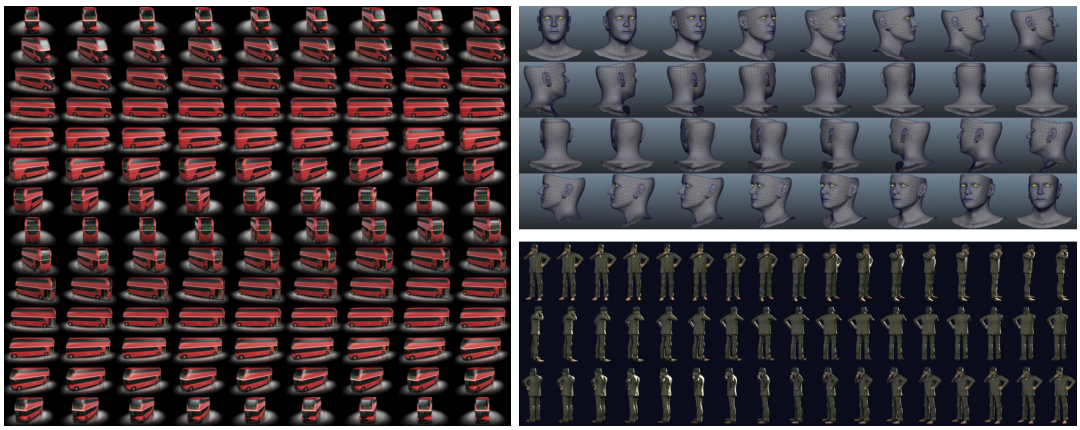
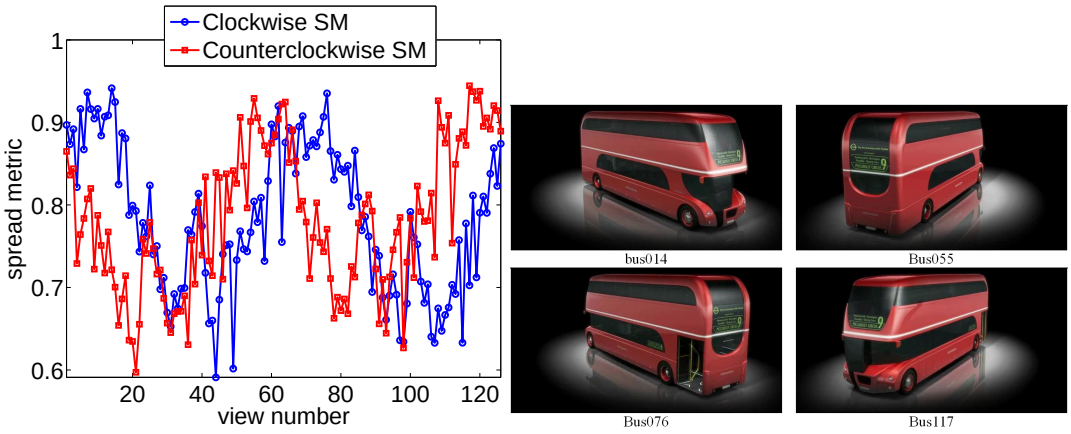


Figure 6.4: Sequences of 3D views. Left: the BUS sequence (126 views); right top: the HEAD sequence (32 views); right bottom: the JONES sequence (51 views).

By assuming that the approximate convex polygon shape has four perceptible sides for the object in each of these sequences, we pick four peaks from spread metric scores. In addition, we use the fact that any two peaks shall be separated

[5]http://www.youtube.com/watch?v=vq1UeTW6uKE

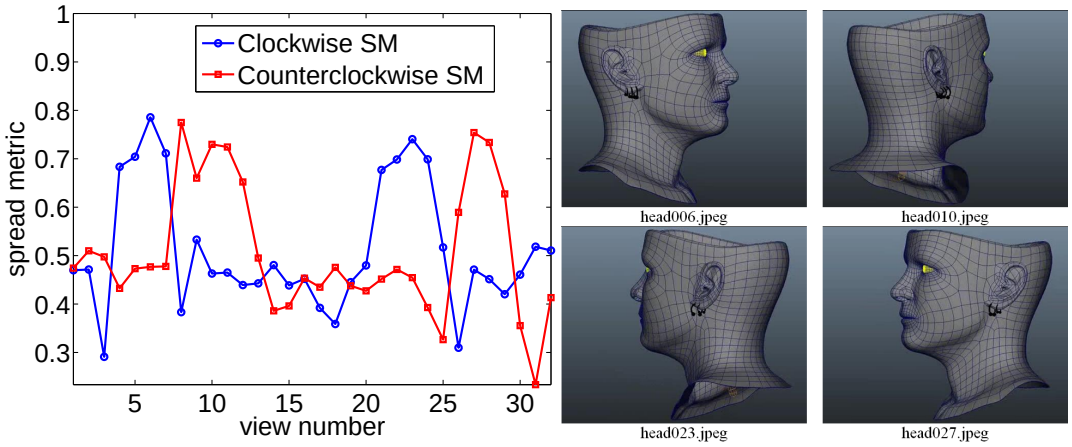


(a) (b)

Figure 6.5: Finding BRVs for the BUS sequence: (a) Clockwise SM and counterclockwise SM. (b) Estimated BRVs.

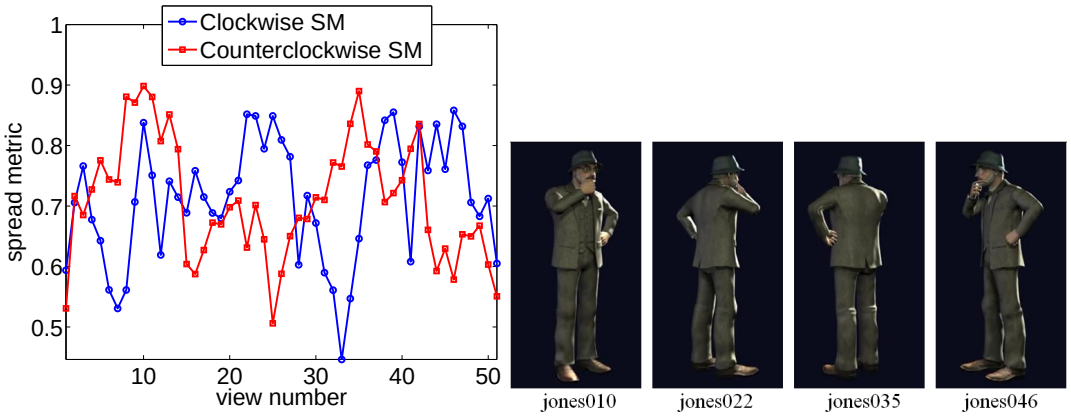
by a certain gap, otherwise peaks may be located within the same BVC (the gap is 22.5° for the BUS sequence, and 30° for HEAD and JONES sequences). Figures 6.5, 6.6 and 6.7 show the results. For the BUS sequence, Figure 6.5(a) suggests that the views with number 014, 055, 076 and 117 are selected as BRVs as shown in Figure 6.5(b). Likewise, Figures 6.6(a) and 6.7(a) suggest views with number 006, 010, 023 and 027, and views with number 010, 022, 035 and 046 as BRVs, shown in Figure 6.6(b) and 6.7(b). It is expected that these BRVs are those with more sides and visible surfaces as suggested in [22], [23], and hence human perceivers are more sensitive to them.

Fig. 6.8 shows the four SVCs which are separated using the estimated BRVs. Taking into account the overall computational load, we evenly down-sample views in each class, such that each class has no greater than nine views. In Figure 6.8, we use green lines to mark distinct SVCs. It can be seen that for most cases, views



(a) (b)

Figure 6.6: Finding BRVs for the HEAD sequence: (a) Clockwise SM and counterclockwise SM. (b) Estimated BRVs.



(a) (b)

Figure 6.7: Finding BRVs for the JONES sequence: (a) Clockwise SM and counterclockwise SM. (b) Estimated BRVs.

belonging to the same SVC come with more similar poses than those of views that are from distinct SVCs. Figure 6.9 shows the resulting SRVs. For each SVC, we pick only one view with the minimum sparse-to-full reconstruction error (i.e., $l_1 = 1$). The results of the BUS sequence are shown in Figure 6.9(a), where views with numbers 034, 070, 096 and 126 are obtained with the minimum residuals calculated by (6.9) and are representatives of SVCs shown in the first row up to the fourth row at the left top of Figure 6.8, respectively. Similarly, for the HEAD sequence, views in Figure 6.9(b) with numbers 009, 014, 027 and 031 are obtained as SRVs of the left bottom 4 rows in Figure 6.8, whereas views in Figure 6.9(c) with numbers 016, 030, 039 and 003 are SRVs of those 4 rows of SVCs shown at the right of Figure 6.8, for the JONES sequence.

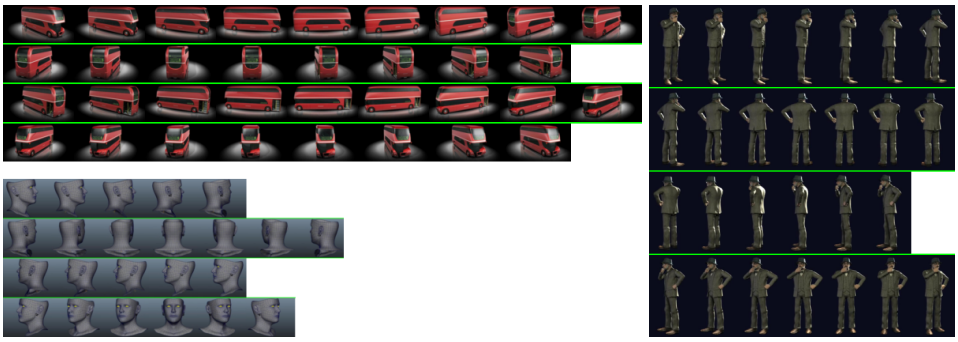


Figure 6.8: Estimated 4 SVCs with down-sampled views. Left top: the BUS sequence; left bottom: the HEAD sequence; right: the JONES sequence.

Intuitively, one would expect a SRV to be the side view that capture the most energy compared to other within-class views, and thus have minimum sparse-to-full reconstruction residuals according to (6.8) and (6.9). It is not hard to see this phenomenon by comparing representative views in Figure 6.9 with their associated

classes in Figure 6.8. For all these sequences, the SRVs are generally pretty close to side views: frontal view, left-side view, right-side view and back view. Finally, the salient views are selected from both BRVs and SRVs.

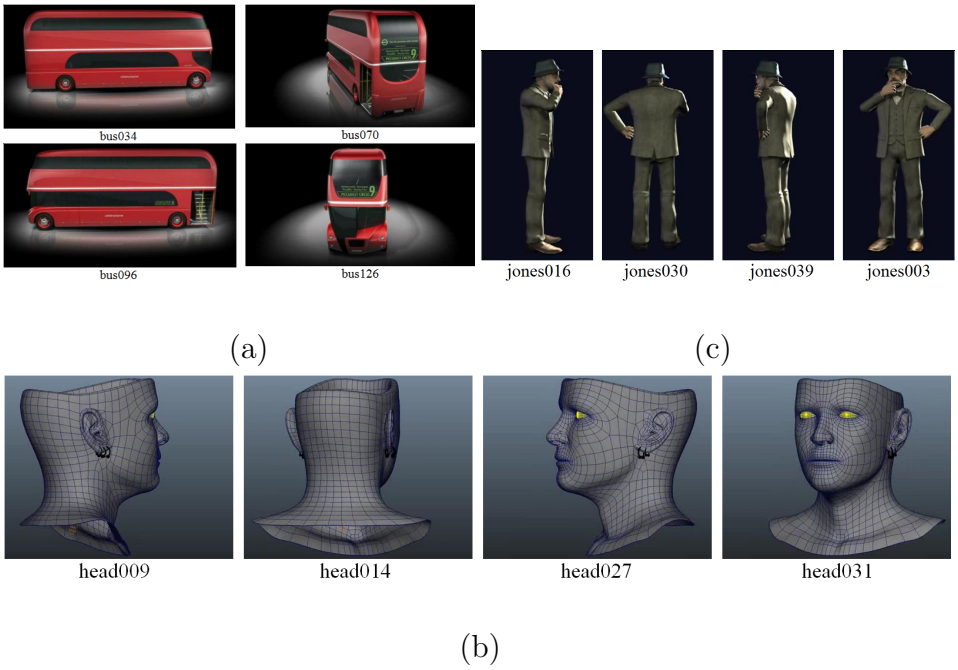


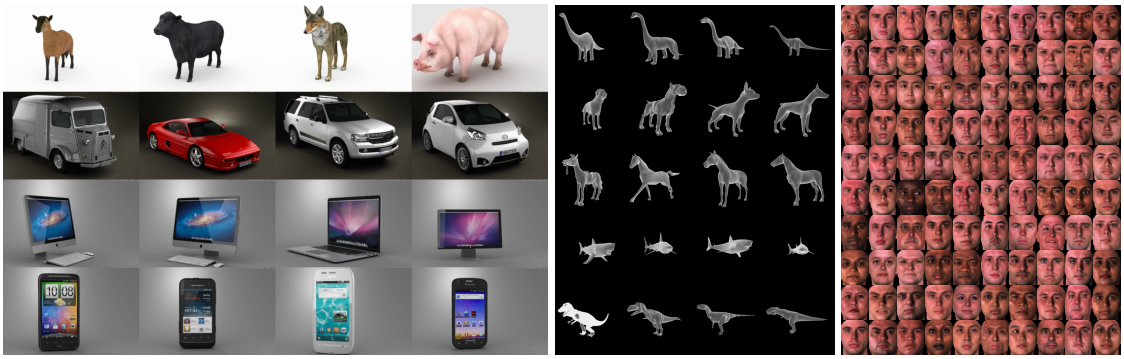
Figure 6.9: SRVs of (a) the BUS sequence (b) the HEAD sequence (c) the JONES sequence.

6.4.2 Object Recognition and Retrieval using Geometric Dictionaries

In this section, we demonstrate the performance of our method in object recognition and retrieval on four datasets: Humster3D videos [121], Princeton 3D models [122], Vetter's 3DFS database [73] and Human ID database [80]. For each view of the four datasets, we took its grayscale image as the input feature.

We compare the proposed SVSR with two other state-of-the-art approaches proposed in [117] and [61], and one baseline approach. In [117], Winkeler *et al.*

proposed a greedy algorithm for subset selection. The saliency was defined as the amount of energy not captured by the basis set for an eigenspace representation. In their approach, the saliency of every ensemble view is computed and the one with the highest saliency is added to the subset. In [61], Shroff *et al.* proposed a video summarization algorithm to select exemplar frames. Their algorithm optimizes a linear combination of *diversity* and *square error*, where *diversity* represents the scatter of exemplars to their mean, while the *square error* represents the summation of all class scatters.



(a) (b) (c)

Figure 6.10: Example images from 3D datasets. (a) Humster3D videos. First row: animals; second row: vehicles; third row: LCDs; fourth row: i-phones (b) Princeton 3D models. First row: apatosauruses; second row: dogs; third row: horses; fourth row: sharks; fifth row: trexes (c) Vetter's 3DFS database (100 subjects).

We refer to the methods in [117] and [61] by SS (for Subset Selection) and VS (for Video Summarization), respectively. For fair comparisons, the SS, VS and SVSR methods are all followed by the GDR algorithm for building geometric dictionaries. On the other hand, the baseline method is the one that randomly selects salient

views, followed by a nearest neighbor (NN) classifier without using dictionaries. This baseline-NN is provided in contrast with the SS-GDR, VS-GDR, and SVSR-GDR methods. For each model, we selected 8 SVs using SVSR (4 BRVs and 4 SRVs, i.e. $n_i = 8$ and $m_i = 4$), VS, SS and baseline algorithms. Unless otherwise stated, the number of dictionary atoms is set equal to 8 for the SV-geometric dictionaries, and 20 for the SVC-geometric dictionaries. Moreover, as the baseline-NN selects salient views randomly, we reported its average performance over 20 trials.

We evaluate the methods in terms of identification and retrieval performances. The retrieval performance includes the precision-recall curves and average retrieval performance [90], [27]. Please refer to Section 4.3 for definitions of precision, recall and average retrieval performance.

(a)

(b)

(c)

Figure 6.11: Example down-sampled FVs from 3D datasets. (a) Humster3D videos. (b) Princeton 3D models. (c) Vetter's 3DFS database.

Humster3D videos: Humster3D videos [121] contain a wide range of videos of 3D models including vehicles (1068), furniture (375), electronics (104), animals & plants (30) and life & leisure (28). We selected a subset containing 16 videos in the

following 4 categories for our experiments: animals (4), vehicles (4), LCDs (4) and i-phones (4). Each video contains 100 views and each view was resized to 24×42 pixels. Fig. 6.10(a) shows example images from these 16 videos, and Fig. 6.11(a) shows a series of down-sampled FVs of the first (top-left) video shown in Fig. 6.10(a).

Table 6.1 shows the object/category rank-1 recognition rates under various combinations of query views and geometric dictionaries. The query views can be either FVs or SVs, and the target gallery is given by either the SV-geometric dictionaries or the SVC-geometric dictionaries. In addition, we conducted 'leave-one-out' tests, where the geometric dictionary associated with the true class of the query object is excluded from the target gallery; and 'no leave-one-out' tests, where the geometric dictionary associated with the true class of the query object is included in the target gallery. As SV-geometric dictionaries are built using few SVs, both 'leave-one-out' tests and 'no leave-one-out' tests were conducted for the gallery of SV-geometric dictionaries. On the other hand, since SVC-geometric dictionaries are built using almost all FVs (SVs excluded), 'no leave-one-out' tests become trivial, and hence only 'leave-one-out' tests were conducted for the gallery of SVC-geometric dictionaries.

As shown in Table 6.1, SVSR-GDR obtained the highest rank-1 recognition rate. It also obtained the highest category (among animals, vehicles, LCDs and i-phones) recognition rates for most tests. Compared to SS-GDR and VS-GDR, the baseline-NN was able to obtain better performances because the between-class distances (and between-category distances) possessed in the gallery are large enough. Fig. 6.12 (a) shows the corresponding precision-recall curves. The proposed SVSR-

GDR achieved the highest precision rates. Note that for each query view, the precision-recall curves were plotted using only the first 100 (class population size) retrieved views from the target gallery. All retrieved images are either in the same class as the query view, or in a different class. This explains the horizontal behavior of the precision-recall curves[6]. Both precision and recall rates in this case reflect the percentage of query views that were correctly retrieved by gallery views belonging to the true classes. Fig. 6.12 (b) shows the average retrieval performance given that 8 gallery images were retrieved for each query image. The overall average retrieval rates (among 16 classes) of baseline-NN, SS-GDR, VS-GDR and SVSR-GDR are 6.29, 5.61, 6.75 and 7.04, respectively. The SVSR-GDR obtained the best the overall average retrieval performance.

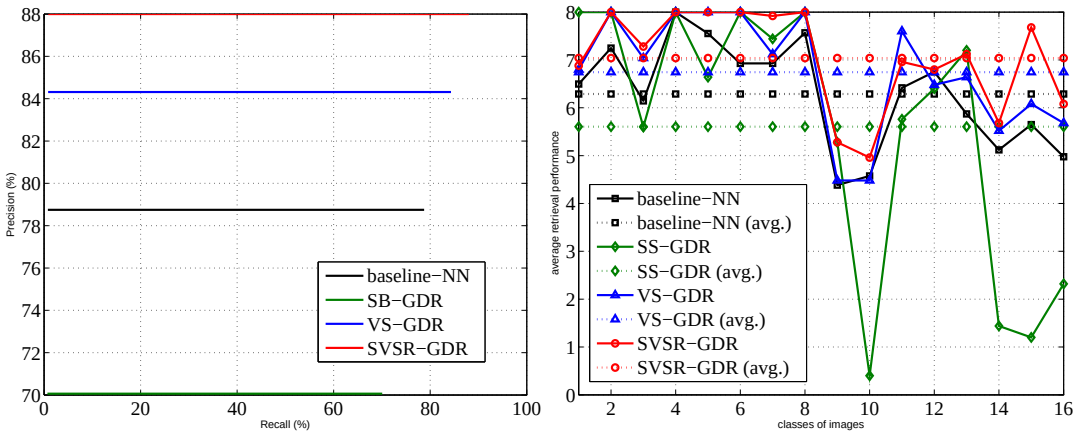
Princeton 3D models: Princeton 3D models (version 1) [122] contain a database of 1814 3D polygonal models collected from the internet. We selected a subset containing 20 models across the following 5 animal categories for experiments: apatosaurus (m273, m274, m275, m276), dog (m88, m89, m91, m92), horse (m103, m106, m107, m108), shark (m76, m77, m78, m80), and trex (m267, m269, m271, m272). We extracted 90 views from each model and each view was resized to 30×30 pixels. Fig. 6.10(b) shows example images from these 20 models, and Fig. 6.11(b) shows a series of down-sampled FVs of the first (top-left) model shown in Fig. 6.10(b).

Table 6.2 shows object/category rank-1 recognition rates. While the proposed

[6]The same reason explains the horizontal behavior of precision-recall curves in Fig. 6.13(a) and Fig. 6.14(a).

Experiments \ Algorithms	baseline-NN	SS-GDR	VS-GDR	SVSR-GDR
1a. Object identification using FVs as query and SV-geometric dictionaries as target (no leave-one-out)	83.19	79.20	90.30	**91.15**
1b. Category identification using FVs as query and SV-geometric dictionaries as target (no leave-one-out)	99.84	96.90	**100**	**100**
2. Category identification using FVs as query and SV-geometric dictionaries as target (leave-one-out)	**93.25**	75.60	88.75	90.35
3. Category identification using FVs as query and SVC-geometric dictionaries as target (leave-one-out)	93.96	89.80	90.85	**92.50**
4. Category identification using SVs as query and SV-geometric dictionaries as target (leave-one-out)	**92.03**	76.56	89.06	91.41
5. Category identification using SVs as query and SVC-geometric dictionaries as target (leave-one-out)	92.11	85.94	92.19	**93.75**
Average	92.40	84	91.86	**93.19**

Table 6.1: Rank-1 recognition rates (%) on the Humster3D videos.



(a) (b)

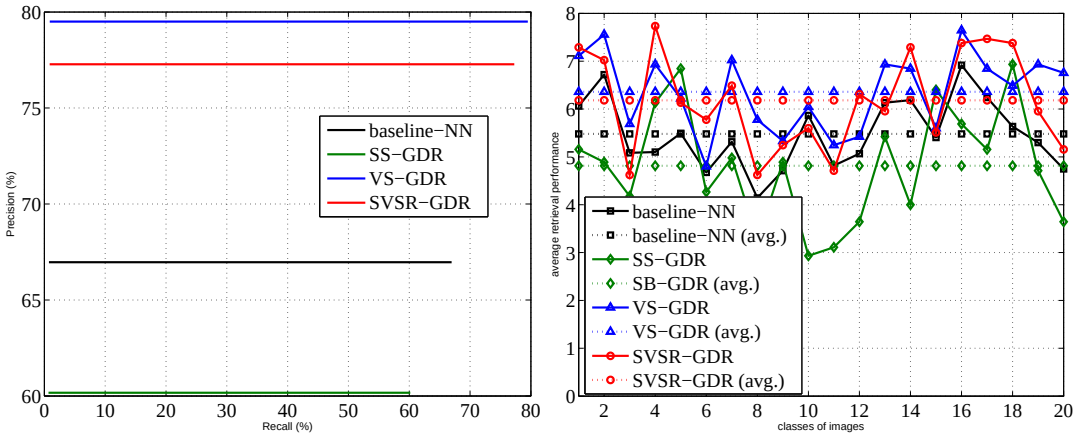
Figure 6.12: Image retrieval results on Humster3D videos. (a) Precision-recall curves and (b) the average retrieval performance. The proposed SVSR-GDR achieves the best precision rates and the overall average retrieval performance.

SVSR-GDR obtained the second highest recognition rate, it ranks the highest in a majority (3 out of 5) of category recognition tests. Moreover, the baseline-NN obtained the lowest average recognition rate. This can be explained by the fact that the between-class distances (and between-category distances) among the gallery classes are no longer large. In fact, compared to the Humster3D videos, more class outliers may exist in the target gallery from this dataset. Fig. 6.13 (a) shows the corresponding precision-recall curves. The proposed SVSR-GDR achieved the second highest precision rates. Fig. 6.13 (b) shows the average retrieval performance given that eight gallery images were retrieved for each query image. The overall average retrieval rates (among 20 classes) of baseline-NN, SS-GDR, VS-GDR and SVSR-GDR are 5.48, 4.81, 6.36 and 6.18, respectively.

Vetter's 3D face database: Vetter's 3D face database [73] contains 100 face

Experiments \ Algorithms	baseline-NN	SS-GDR	VS-GDR	SVSR-GDR
1a. Object identification using FVs as query and SV-geometric dictionaries as target (no leave-one-out)	67	59.22	**77.38**	74.86
1b. Category identification using FVs as query and SV-geometric dictionaries as target (no leave-one-out)	85.60	77.93	89.91	**91.42**
2. Category identification using FVs as query and SV-geometric dictionaries as target (leave-one-out)	57.50	54.86	63.11	**63.99**
3. Category identification using FVs as query and SVC-geometric dictionaries as target (leave-one-out)	61.64	76.30	76.71	**77.41**
4. Category identification using SVs as query and SV-geometric dictionaries as target (leave-one-out)	64	**75**	73.75	68.13
5. Category identification using SVs as query and SVC-geometric dictionaries as target (leave-one-out)	70.56	**84.38**	82.50	82.50
Average	67.72	71.28	**77.23**	76.39

Table 6.2: Rank-1 recognition rates (%) on the Princeton 3D models.



(a) (b)

Figure 6.13: Image retrieval results on Princeton 3D models. (a) Precision-recall curves and (b) the average retrieval performance. The proposed SVSR-GDR ranks the second (close to VS-GDR), in both precision-recall and average retrieval performances.

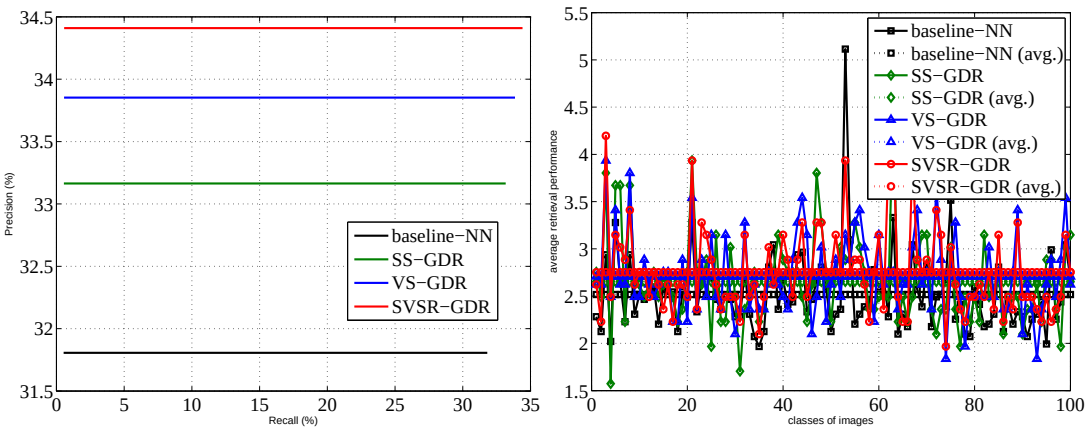
models. We extracted 60 views (rotated from $0° \sim 180°$ with respect to the Y axis) from each model and resized each view to 30×30 pixels. Fig. 6.10(c) shows example images from all 100 models, and Fig. 6.11(c) shows a series of down-sampled FVs of the first (top-left) model shown in Fig. 6.10(c).

As each model belongs to an independent subject class, there is no need to divide the models into categories. Therefore, no category recognition tests were conducted. Table 6.3 shows face recognition rates. The proposed SVSR-GDR obtained the highest recognition rate. Fig. 6.14 (a) and (b) show the corresponding precision-recall curves and the average retrieval performance (given 8 retrieved gallery images). The proposed SVSR-GDR obtained the best precision rates and average retrieval performance. The overall average retrieval rates (among 100 classes) of baseline-NN,

SS-GDR, VS-GDR and SVSR-GDR are 2.52, 2.65, 2.71, 2.75, respectively.

Experiments \ Algorithms	baseline-NN	SS-GDR	VS-GDR	SVSR-GDR
Face identification using FVs as query and BRV-geometric dictionaries as target (no leave-one-out)	31.47	33.16	33.87	**34.41**

Table 6.3: Rank-1 recognition rates (%) on Vetter's 3D face database.



(a) (b)

Figure 6.14: Image retrieval results on Vetter's 3D face database. (a) Precision-recall curves and (b) the average retrieval performance. The proposed SVSR-GDR obtained the best precision rates and the overall average retrieval performance.

Human ID database: In this subsection, we select the Human ID database [80] to demonstrate the effectiveness of the proposed SVSR-GDR on video-based face recognition for real people. Please refer to Section 3.3.3 for this dataset.

In our experiment, we choose videos of a subset of 60 out of 284 subjects. For each of these selected subjects, there are videos of moving facial mug shots, facial

speech and dynamic facial expressions, shown in the first three rows of Fig. 3.5, respectively. Similar to the Vetter's database used for our experiments in the previous section, the facial mug shot video contains poses from the left side pose to the right side pose (from $0^\circ \sim 180^\circ$ with respect to the Y axis), incremented in a step of 22.5°. In addition, while the moving facial mug shot videos contain subjects' neutral faces in different poses, the speech and dynamic facial expression videos capture variant facial expressions (mainly disgust and laughter) of subjects in a single frontal pose.

The face region of each frame extracted from the selected videos was properly cropped and resized to 30×24 pixels as a view. We used SVs from the moving facial mug shot videos to construct SV-geometric dictionaries, and evaluated these dictionaries using query FVs from the same subject's moving facial mug shot videos, facial speech videos, and dynamic facial expression videos. Table 6.4 shows rank-1 face recognition rates among 60 classes. As shown, the proposed SVSR-GDR obtained the highest (average) recognition rates. Comparing different video types, we observe that faces of subjects in the speech and expression videos appear in a single frontal pose, which can be accounted for by the geometric dictionaries as the moving facial mug shot videos also contain frontal face images. However, low recognition rates were obtained on these videos. This can be explained by the fact that these videos contain facial variations that are novel to the original facial mug shot videos, and hence are more challenging for recognition.

Experiments \ Algorithms	baseline-NN	SS-GDR	VS-GDR	SVSR-GDR
FVs of moving facial mug shot videos as query and SV-geometric dictionaries of moving facial mug shot videos as target	67.63	68.91	77.87	**82.45**
FVs of facial speech videos as query and SV-geometric dictionaries of moving facial mug shot videos as target	20.52	15.00	43.33	**53.33**
FVs of dynamic facial expression videos as query and SV-geometric dictionaries of moving facial mug shot videos as target	28.71	20.00	**45.00**	**45.00**
Average	38.95	34.64	55.40	**60.26**

Table 6.4: Rank-1 recognition rates (%) on the Human ID database.

6.4.3 Sparse-to-full reconstruction from salient views

In this section, we demonstrate the performance of sparse-to-full reconstruction using SVs to show the reconstruction power of all compared algorithms. Our experiments were conducted on Vetter's 3DFS database [73].

Given a SV, the method of image-based visual hull (IBVH) [123], [124] was used to build correspondences between the SV and each synthesized view. The IBVH method performs pixel-to-pixel mapping based on spatial correspondences according to the geometry, without explicitly rendering from the 3D model to reconstruct the synthesized view. All pixels that can be seen from synthesized view are mapped from the corresponding pixels of the SV. When the desired view is located between two SVs, it can be reconstructed using the two synthesized views from the SVs, according to the relative perspectives between the desired view and the two SVs. In particular, the number of pixel columns from either of the synthesized views is determined by the ratio of the perspective between the desired view and one synthesized view, to the perspective between the desired view and the other synthesized view. To show this idea, let the desired view be denoted by $\mathbf{t}_{\theta_d}$, and two salient views be denoted by $\mathbf{t}_{\theta_1}$ and $\mathbf{t}_{\theta_2}$, where $\theta_1 \leq \theta_d \leq \theta_2$ are the view perspectives with respect to the Y axis. Let $\hat{\mathbf{t}}_{\theta_1}$ and $\hat{\mathbf{t}}_{\theta_2}$ denote the two synthesized views from $\mathbf{t}_{\theta_1}$ and $\mathbf{t}_{\theta_2}$, respectively. Let C be the number of columns of $\mathbf{t}_{\theta_d}$ in its 2D matrix form. Then, at θ_d, the reconstructed view, $\tilde{\mathbf{t}}_{\theta_d}$ is synthesized in such a way that its right $\lfloor C\frac{\theta_d-\theta_1}{\theta_2-\theta_1} \rfloor$ columns are mapped from the same right $\lfloor C\frac{\theta_d-\theta_1}{\theta_2-\theta_1} \rfloor$ columns of $\tilde{\mathbf{t}}_{\theta_2}$, while its left $\lceil C\frac{\theta_2-\theta_d}{\theta_2-\theta_1} \rceil$ columns are mapped from the same left $\lceil C\frac{\theta_2-\theta_d}{\theta_2-\theta_1} \rceil$ columns of $\tilde{\mathbf{t}}_{\theta_1}$. On the other

hand, if the desired view is not located between two SVs, then all columns of its reconstructed view are directly contributed from the synthesized view of the closet SV.

Fig. 6.15 illustrates an example of the reconstructed view at 0° using two IBVH-based synthesized views from the SVs at -45° and 30°. The number of columns from the left of the reconstructed view contributed using the same columns of the synthesized view at -45°, and the number of columns from the right of the reconstructed view contributed using the same columns of the synthesized view at 30°, have a ratio of 30 to 45, which are perspectives between the desired view to the SVs at 30° and -45°, respectively. The reconstructed view at 0° has a shorter distance to the desired view than the two synthesized views, either of which is contributed from only one SV.

Table 6.5 shows average reconstruction errors using two SVs produced from different algorithms on the Vetter's 3D face database. Each view is resized to 112×95 pixels. For our SVSR method, two BRVs with the highest boundary scores computed using (6.3) are selected as SVs. The reconstruction error is computed using the ℓ_2-norm distance between the desired view and the reconstructed view in the normalized grayscale. The 'baseline1' refers to the random selection of two SVs, while 'baseline2' refers to the selection of SVs at fixed -45° and 45°, two candidate perspectives to recognize people.

Fig. 6.16(a) and (b) show the average reconstruction errors versus subject indices ($1 \sim 100$) and perspectives ($-90^\circ \sim 90^\circ$), respectively. As shown in Fig. 6.16(a), the proposed SVSR obtained the lowest average reconstruction errors.

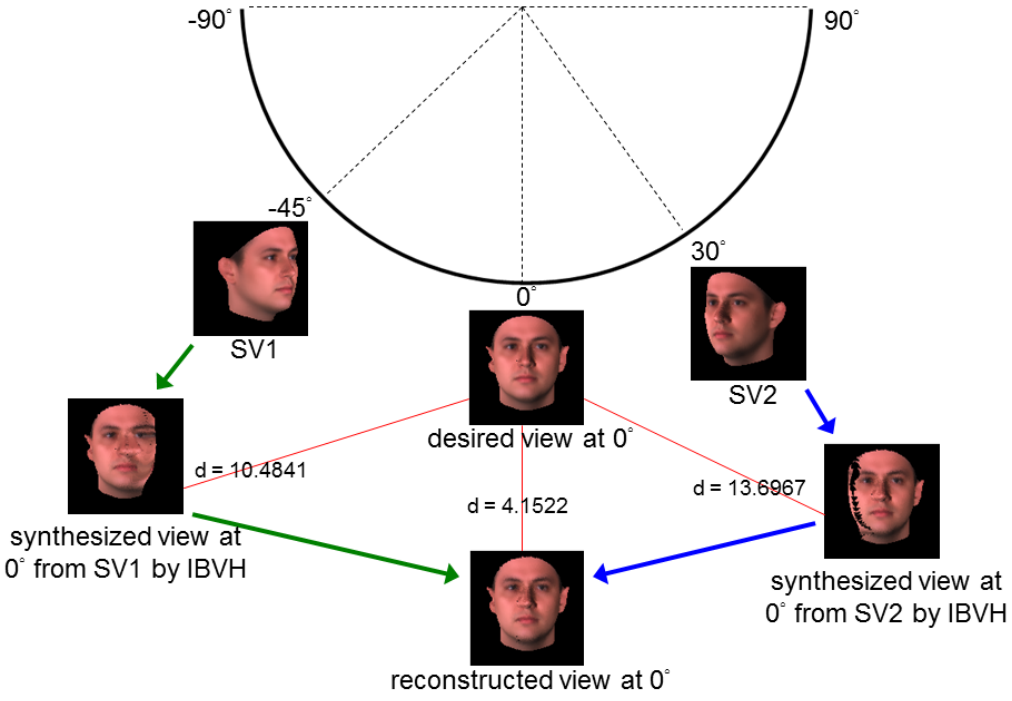


Figure 6.15: IBVH-based reconstruction. The reconstructed view at 0° using two IBVH-based synthesized views from salient views at −45° and 30°, has a shorter distance to the desired view than the two synthesized views, either of which is contributed from only one SV.

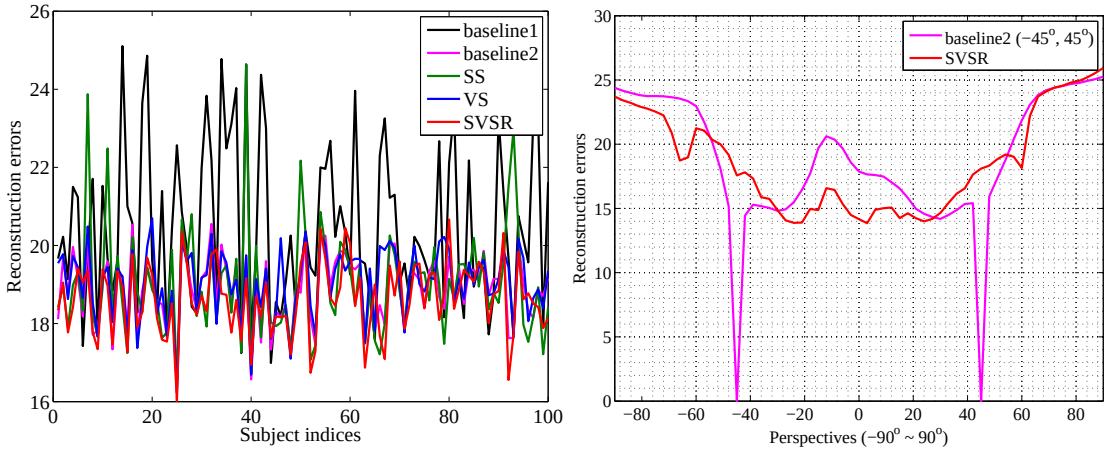
In Fig. 6.16(b), both SVs (BRVs) of our SVSR are concentrated among $[-28°, -20°]$, $[-4°, 4°]$ and $[20°, 28°]$. This means views within these ranges of perspectives have higher reconstruction power. On the other hand, the baseline2 method has zeros at $-45°$ and $45°$ since the SVs are always fixed at these two perspectives.

Experiments \ Algorithms	baseline1	baseline2	SS	VS	SVSR
Average reconstruction errors	20.4584	18.8793	19.0119	19.0483	**18.6331**

Table 6.5: Average reconstruction errors on Vetter's 3D face database. The reconstruction error is computed as the ℓ_2-norm distance between the desired view and the reconstructed view in the normalized grayscale.

6.4.4 Discussion

Among all compared methods, we observed VS-GDR and the proposed SVSR-GDR obtained close performances. This can be explained by the fact that both VS and SVSR aim to find object representative views. The slight difference is that the VS minimizes the cost as a linear combination of *diversity* and *square error*, while the proposed SVSR finds representative views that either contain more sides (BRVs) or minimize the reconstruction error (SRVs). The SS, on the other hand, defines the saliency as the information relative to a representation. It turns out that the SS finds discriminative views. While discriminative views are not necessarily representative, the SS is not optimal for object recognition and retrieval



(a) (b)

Figure 6.16: Reconstruction errors on Vetter's 3D face database. (a) Average reconstruction errors versus subject indices and (b) Average reconstruction errors versus perspectives. The proposed SVSR obtained the lowest average reconstruction errors. Both SVs (BRVs) are concentrated among $[-28°, -20°]$, $[-4°, 4°]$ and $[20°, 28°]$. On the other hand, the 'baseline2' method has zeros at $-45°$ and $45°$ since the SVs are always fixed at these two perspectives.

applications. Furthermore, the baseline-NN works well only when the within-class variation is small and between-class (or between-category) distances are large enough in the target gallery. It is sensitive to a few outliers in the target gallery that either increases the within-class scatter and/or decreases the between-class distances.

Our experimental results in section 6.4.3 are useful for applications whose objective is to find a fixed number of representative views such that for every view $\mathbf{x}$ of the given object, the intersection between the cone formed by projecting the silhouette image into the 3D space through the camera center of the view $\mathbf{x}$, and the shape of the object, is contained in the intersection between the visual hull formed by the corresponding cones (i.e. the intersection among the cones) projected from camera centers of these representative views, and the shape of the object. A BRV is a view with as many as visible sides (i.e. faces) of the convex polygon shape approximation on the object's shape, and hence is a view whose characteristic view domain (CVD) [115] covers as many as viewpoints as possible. Therefore, the visual hull formed by cones of all viewpoints in the CVD of the BRV contains the intersection between the object shape and the corresponding cones of views that are as many as possible. In other words, BRVs are candidates of the representative views in the objective. As shown in Table 6.5 and Fig. 6.16, our SVSR gives BRVs that obtained the lowest reconstruction errors among other compared methods.

6.5 Summary

We presented a two-stage approach based on sparse representation to find the salient views of an object. The first stage computes the spread metric and boundary scores to estimate boundary representative views. Using these estimated representative views, full views are roughly partitioned into different side view classes. In the second stage, side representative views are determined that have minimum class sparse-to-full reconstruction residuals. We constructed geometric dictionaries using the salient views and side view classes for applications in 3D object recognition and retrieval. By a series of experiments on four publicly available 3D datasets, we demonstrated the effectiveness of our approach over two existing state-of-the-art algorithms and baseline methods through the performances on object recognition, retrieval and sparse-to-full reconstruction.

Chapter 7: Conclusions and directions of future work

We presented a video dictionary learning framework for unconstrained video-to-video person recognition. We partitioned the images extracted from a given video. Each partition captures different pose and illumination conditions and is encoded in different video sub-dictionaries. Each sub-dictionary encodes a face in a particular viewing condition. The sub-dictionaries are combined to form video specific dictionaries for recognition. Using video dictionaries, we then proposed another effective joint sparsity-based approach that takes into account correlations as well as coupling information between frames of a video while enforcing joint sparsity within each frame's observation. To enhance discriminative recognition, we learned kernel dictionaries to handle the nonlinearities in video data. We used human's upper body features and motion identity cues to improve the recognition accuracy. Furthermore, we extended the existing unconstrained video-to-video face recognition algorithms to the one that explicitly addresses the challenge of matching probe and gallery videos in different poses. Our approaches include a sparse representation-based alignment method that generates pose aligned features through pre-designed reference sets under a sparsity constraint, and a dictionary rotation method that directly rotates gallery video dictionary atoms in both their harmonic basis and 3D

geometry to match the poses of the probe videos. Various experiments on publicly available data sets show that the proposed methods can perform significantly better than many state-of-the-art algorithms in the literature.

Next, we proposed a rotation and scale invariant clustering algorithm suitable for applications such as CBIR. We extracted rotation and scale invariant features of images in the Radon domain. The initial dictionaries are learned through initial clusters which are determined using the Hamming distance between nearest-neighbor sets of each feature pair. With a view to achieving rotation and scale invariance in clustering, the proposed method learns dictionaries and clusters images in the Radon domain. We demonstrated by experiments on shape-based and texture-based datasets the effectiveness of the proposed method for CBIR applications, its robustness to missing pixels, and performance improvements compared to other Gabor-based and shape-based methods. Robustness to within-class variations of texture images is one of the important research directions. We will continue with feature extraction from the Radon-domain sinogram, via filtering or transformation techniques. We will also consider using local features. One possible way is to divide the texture (or its sinogram) into several patches, from each of which the local feature is extracted. Then we combine all extracted features using an efficient fusion technique to improve the recognition performance. In addition, we will modify the Radon-domain sinogram based approach for our algorithm to adapt to textures without apparent linear structures and for isotropic textures. Furthermore, as our method being based on heuristic K-SVD lacks theoretical guarantee on convergence, we will also provide analytical evidences on convergence.

Next, we extended dictionary learning to the case of ambiguously labeled learning, a general kind of semi-supervised learning where each example is supplied with multiple labels, only one of which is correct. Our method aims to solve the problem of ambiguously labeled multiclass-classification using an iterative algorithm. It iteratively estimates the confidence of samples belonging to each of the classes and uses it to refine the learned dictionaries. The confidence is updated for each sample according to its residuals computed from previously learned dictionaries, while dictionaries are updated through the most recent confidence. The dictionaries are updated using either soft (EM-based) or hard decision rules. Experiments using three publicly available datasets demonstrate the improved accuracy of the proposed method compared to state-of-the-art ambiguously labeled learning techniques.

Finally, we presented a two-stage approach based on sparse representation to find salient views of an object. The first stage computes the spread metric and boundary scores to estimate boundary representative views. Using these estimated representative views, full views are roughly partitioned into different side view classes. In the second stage, side representative views are determined that have minimum class sparse-to-full reconstruction residuals. To evaluate our approach, we constructed geometric dictionaries using our salient views and side view classes for applications in 3D object identification and image retrieval. By a series of experiments on four publicly available 3D datasets, we demonstrated the effectiveness of our approach over two existing state-of-the-art algorithms and baseline methods using experiments on object recognition, retrieval and sparse-to-full reconstruction. We will continue exploring the IBVH-based reconstruction power of the boundary

representative views. We will extend our work on view selection among the full 3D views taken at all perspectives (rotations w.r.t. all three axes) in various distances. We will also evaluate the robustness of our approach to noise and occlusions.

Chapter A: Proof: each of the initial partitions obtained by $1 \sim 6$ in Algorithm 1 of Chapter 2, must contain an exemplar

A.1 2-cluster case:

Let the set $S = \{x_j | j = 1, 2, ..., N\} = S_1 \bigcup S_2$. Assume $d(x_p, x_q) > d(x_i, x_j)$, $\forall (i, j) \neq (p, q)$. Clearly, it is reasonable to assign x_p and x_q in different clusters. Let x_r and x_s be exemplars of the two clusters, respectively. Without loss of generality, let x_p be x_r's nearest neighbor, and x_q be x_s's nearest neighbor (i.e., $d(x_r, x_p) < d(x_r, x_q)$ and $d(x_s, x_q) < d(x_s, x_p)$).

Claim: x_r is x_p's nearest neighbor, and x_s is x_q's nearest neighbor. In other words, x_r is the exemplar of the cluster which contains x_p; and x_s is the exemplar of the cluster which contains x_q (i.e., $d(x_p, x_r) < d(x_p, x_s)$ and $d(x_q, x_s) < d(x_q, x_r)$). In the following we prove by contradiction:

1) If $d(x_p, x_r) > d(x_p, x_s)$ and $d(x_q, x_s) < d(x_q, x_r)$, then both x_q and x_p are in the same cluster whose exemplar is x_s. We reach a contradiction to the fact that x_q and x_p are in different clusters.

2) If $d(x_p, x_r) < d(x_p, x_s)$ and $d(x_q, x_s) > d(x_q, x_r)$, then both x_q and x_p are in the same cluster whose exemplar is x_r. We reach a contradiction to the fact that x_q and x_p are in different clusters.

3) If $d(x_p, x_r) > d(x_p, x_s)$ and $d(x_q, x_s) > d(x_q, x_r)$, then from the assumption we have $d(x_p, x_r) > d(x_q, x_s)$ and $d(x_q, x_s) > d(x_p, x_r)$, which is again a contradiction.

Therefore, $d(x_p, x_r) < d(x_p, x_s)$ and $d(x_q, x_s) < d(x_q, x_r)$. That is, x_r is x_p's nearest neighbor, and x_s is x_q's nearest neighbor.

A.2 K-cluster case:

Consider K-cluster case for general K. Let set $S = \{x_j | j = 1, 2, ..., N\} = \bigcup_{i=1}^{K} S_i$. Assume

$$d(x_{u(1)}, x_{u(2)}, ..., x_{u(K)}) > d(x_{i(1)}, ..., x_{i(K)}), \quad \forall [i(1) \ ... \ i(K)]^T \neq [u(1) \ ... \ u(K)]^T. \tag{A.1}$$

Clearly, it is reasonable to assign $x_{u(1)}, x_{u(2)}, ..., x_{u(K)}$ into different K clusters. Let $x_{v(1)}$, $x_{v(2)}$, ..., $x_{v(K)}$ be the true exemplars of these K clusters, respectively. Without loss of generality, let $x_{u(k)}$ be $x_{v(k)}$'s nearest neighbor, $\forall k \in \{1, ..., K\}$. Equivalently,

$$d(x_{v(k)}, x_{u(k)}) < d(x_{v(k)}, x_j), \quad \forall j \in \{u(1), ..., u(k-1), u(k+1), ..., u(K)\}. \tag{A.2}$$

Claim: $x_{v(k)}$ is $x_{u(k)}$'s nearest neighbor, $\forall k \in \{1, ..., K\}$. In other words,

$$d(x_{u(k)}, x_{v(k)}) < d(x_{u(k)}, x_j), \quad \forall j \in \{v(1), ..., v(k-1), v(k+1), ..., v(K)\}, \forall k \in \{1, ..., K\}. \tag{A.3}$$

This is equivalent to that $x_{v(k)}$ is the exemplar of the cluster which contains $x_{u(k)}, \forall k \in \{1, ..., K\}$.

Assume $x_{\tilde{v}(1)}$, $x_{\tilde{v}(2)}$, ..., $x_{\tilde{v}(K)}$ are exemplars of clusters which contain $x_{u(1)}$, $x_{u(2)}$, ..., and $x_{u(K)}$, respectively. We intend to show $\tilde{v}(k) = v(k), \forall k \in \{1, 2, ..., K\}$.

Given any $k \in \{1, 2, ..., K\}$, if $\tilde{v}(k) \neq v(k)$, we can find i with $i \neq k$ such that $\tilde{v}(i) = v(k)$. Under this assumption, $x_{u(i)}$ belongs to the cluster whose exemplar is $\tilde{v}(i) = v(k)$. On the other hand, since $x_{u(k)}$ is $x_{v(k)}$'s nearest neighbor, we have $d(x_{v(k)}, x_{u(k)}) < d(x_{v(k)}, x_{u(i)})$. Therefore $x_{u(k)}$ also belongs to the cluster which $x_{u(i)}$ belongs to. We have reached a contradiction to the initial assumption that $x_{u(1)}, x_{u(2)}, ..., x_{u(K)}$ belong to different K clusters, respectively. Hence, $\tilde{v}(k) = v(k)$.

We can follow the same procedure of proof by contradiction for all k, to show that $\tilde{v}(k) = v(k), \forall k \in \{1, 2, ..., K\}$. Therefore, $x_{v(k)}$ is $x_{u(k)}$'s nearest neighbor, $\forall k \in \{1, ..., K\}$.

Chapter B: More on the harmonic basis rotation

We present more details on the harmonic basis rotation, as well as the derivations for equations (3.5), (3.6), and our underlying assumption that $\mathcal{R}_\delta(\cdot)$ is a linear function to arrive at (3.6).

It has been shown in [58], [59] that the d-dimensional column-vectorized image $\mathbf{y}_\theta$ can be represented by a linear combination of nine harmonic basis images plus an error vector. Hence we have,

$$\mathbf{y}_\theta = \mathbf{B}_\theta \hat{\boldsymbol{\gamma}} + \mathbf{e} = \mathbf{A}_\theta \boldsymbol{\alpha} + \mathbf{e}_1, \tag{B.1}$$

where $\mathbf{A}_\theta$ is a d-by-9 matrix with each column representing a particular basis image; $\boldsymbol{\alpha}$ is a coefficient vector; $\mathbf{e}$ and $\mathbf{e}_1$ are error vectors. When $\theta = \theta_0$ (i.e. frontal pose), it is assumed the probability density functions of rows of $\mathbf{A}_{\theta_0}$ are Gaussian distributed with sample mean vectors and covariance matrices that can be estimated from basis images in the bootstrap set (e.g. Vetter's 3D face database under frontal pose) [58], [59]. The basis harmonic matrix $\mathbf{A}_{\theta_0}$ hence can be recovered through computing the maximum *a posteriori* (MAP) estimates. Using the estimate of $\mathbf{A}_{\theta_0}$, the corresponding coefficient vector α and error term $\mathbf{e}_1$ of a given novel face image can further be estimated [59].

It can be shown that rotating an input image $\mathbf{y}_\theta$ by δ according to the 3D face

model is approximated through the completion of two steps: (1) Perform δ-rotation on the harmonic basis $\mathbf{A}_\theta$ of $\mathbf{y}_\theta$ [58]. (2) Apply spatial translation and interpolation according to the 3D δ-rotation matrix. In step (1), a matrix $\mathbf{L}_\delta \triangleq \mathbf{L}_{\delta_a}\mathbf{L}_{\delta_e}\mathbf{L}_{\delta_z}$ is multiplied with the harmonic basis $\mathbf{A}_\theta$ in (B.1), where matrices $\mathbf{L}_{\delta_a}$, $\mathbf{L}_{\delta_e}$, and $\mathbf{L}_{\delta_z}$ are used to change the harmonic basis in accordance with the azimuth, elevation and z axis rotations, respectively[1]. We denote the resulting intermediate image vector by $\tilde{\mathbf{y}}_{\theta+\delta}$. In step (2), a spatial translation and interpolation operator $\mathcal{R}_\delta(\cdot)$ determined by the 3D rotation matrix, is applied on $\tilde{\mathbf{y}}_{\theta+\delta}$ such that each pixel in the final output image $\mathbf{y}_{\theta+\delta}$ is either directly mapped using a real pixel in $\tilde{\mathbf{y}}_{\theta+\delta}$, or through interpolation among all mapped real pixels in its neighborhood.

Based on this assumption, step-by-step derivations for equations (3.5) and (3.6) are shown as follows.

$$\begin{aligned}
\mathbf{y}_{\theta+\delta} &\approx \mathcal{R}_\delta(\tilde{\mathbf{y}}_{\theta+\delta}) \\
&= \mathcal{R}_\delta(\mathbf{A}_\theta\mathbf{L}_\delta\boldsymbol{\alpha} + \mathbf{e}_1) \\
&= \mathcal{R}_\delta(\check{\mathbf{B}}_{\theta+\delta}\hat{\boldsymbol{\gamma}} + \mathbf{e}) \\
&= \mathcal{R}_\delta(\check{\mathbf{B}}_{\theta+\delta})\hat{\boldsymbol{\gamma}} + \mathcal{R}_\delta(\mathbf{e}) \\
&= \mathbf{B}_{\theta+\delta}\hat{\boldsymbol{\gamma}} + \mathcal{R}_\delta(\mathbf{e}), \qquad \text{(B.2)}
\end{aligned}$$

where $\check{\mathbf{B}}_{\theta+\delta}$ is $\mathbf{B}_\theta$ with δ-rotated harmonic basis. It is obtained by multiplying the

[1]Here δ_a and δ_e correspond to the azimuth angle $-\theta$ and elevation angle $-\beta$ defined in [58], respectively. $\mathbf{L}_{\delta_a}$ and $\mathbf{L}_{\delta_e}$ are equivalent to matrices given by (3) and (4) in [58]. $\mathbf{L}_{\delta_a}$ can be derived in a similar way.

harmonic basis of each column of $\mathbf{B}_\theta$ with $\mathbf{L}_\delta$. Using (4.4) and (B.2), we have

$$\begin{aligned}
\|\mathbf{y}_{\theta_0,\mathrm{SRA}} - \mathbf{y}_{\theta,\mathrm{SRA}}\|_2 &= \|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - \mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}\|_2 \\
&= \|\mathbf{B}_{\theta_0}\hat{\boldsymbol{\gamma}}_0 - (\mathbf{B}_{\theta+\delta}\hat{\boldsymbol{\gamma}}|_{\delta=-\theta})\|_2 \\
&\approx \|\mathbf{y}_{\theta_0} - \mathbf{e}_0 - (\mathbf{y}_{\theta+\delta} - \mathcal{R}_\delta(\mathbf{e})|_{\delta=-\theta})\|_2 \\
&= \|\mathbf{y}_{\theta_0} - \mathbf{e}_0 - \mathbf{y}_{\theta_0} + \mathcal{R}_{-\theta}(\mathbf{e})\|_2 \\
&= \|\mathcal{R}_{-\theta}(\mathbf{e}) - \mathbf{e}_0\|_2 \\
&\leq \|\mathcal{R}_{-\theta}(\mathbf{e})\|_2 + \|\mathbf{e}_0\|_2. \qquad \text{(B.3)}
\end{aligned}$$

The third equality in (B.2) holds if $\mathcal{R}_\delta(\cdot)$ is a linear function. To show the linearity of $\mathcal{R}_\delta(\cdot)$, we consider $\mathbf{x}' = [x'_1 \ \ldots \ x'_d]^T = \mathcal{R}_\delta(\mathbf{x})$, where $\mathbf{x} = [x_1 \ \ldots \ x_d]^T = \check{\mathbf{B}}_{\theta+\delta}\hat{\boldsymbol{\gamma}} + \mathbf{e}$. Let I_1 be the set of indices in $\mathbf{x}'$ where the pixels are directly mapped from real pixels of $\mathbf{x}'$, and I_2 be the set of indices in $\mathbf{x}'$ where the pixels are linearly interpolated using neighboring mapped pixels of $\mathbf{x}'$. For indices i's belonging to I_1, we have

$$\begin{aligned}
x'_i &= x_{f(i)} \\
&= \sum_m \check{b}_{f(i),m}\hat{\boldsymbol{\gamma}}_m + e_{f(i)} \\
&= \sum_m b'_{i,m}\hat{\boldsymbol{\gamma}}_m + e'_i, \ \forall i \in I_1, \qquad \text{(B.4)}
\end{aligned}$$

where $f(\cdot)$ is a index mapping function determined by $\mathcal{R}_\delta(\cdot)$; $b_{i,m}$ is the (i,m)th entry of $\check{\mathbf{B}}_{\theta+\delta}$; $\hat{\boldsymbol{\gamma}}_m$ is the mth entry of $\hat{\boldsymbol{\gamma}}$; e_i is the ith entry of $\mathbf{e}$; and $b'_{i,m}$, e'_i are entries of the corresponding rotated version (by δ) of columns in $\check{\mathbf{B}}_{\theta+\delta}$ and $\mathbf{e}$, respectively. In particular, the 3D rotation matrix R_δ with rotation $\delta = \{\delta_a, \delta_e, \delta_z\}$ can be written as the product of individual matrices: $\mathbf{R}_\delta = \mathbf{R}_x(\delta_a)\mathbf{R}_y(\delta_e)\mathbf{R}_z(\delta_z)$.

Under the assumption of orthographic projection, the $f(\cdot)$ in (B.4) is defined as

$$f(i) = g\left(\underset{\mathbf{u}:\ g(\mathbf{R}_\delta \mathbf{u})=i}{\operatorname{argmax}} \left(\mathbf{R}_\delta \mathbf{u}\right)^T \left(0\ 0\ 1\right)^T\right), \tag{B.5}$$

where $\mathbf{u} = (u_x\ u_y\ u_z)^T$ denotes the vector of 3D coordinates of a pixel and $g(\cdot)$ is the column-vectorization function that maps the corresponding $\mathbf{u}$ of the pixel to an index of the one-dimensional column vector. While the $\mathcal{R}_\delta(\cdot)$ is not one-to-one, when two or more pixels $\mathbf{u}$'s (before rotation) are mapped to the same index i (after rotation), (B.5) resolves this confliction by selecting the pixel whose z coordinate after rotation is the maximum. This is due to the fact that this pixel stays at the topmost and hence occludes other competing pixels.

Since the $\mathcal{R}_\delta(\cdot)$ is not onto, there exist indices in $\hat{\mathbf{x}}$ where no real indices in $\mathbf{x}$ are mapped from. In this case, interpolation among the neighboring mapped pixels can be used to compute for these indices. Therefore, for indices i's belonging to I_2, we have

$$\begin{aligned}
x'_i &= \sum_{j\in n(i)} \omega_j x'_j \\
&= \sum_{j\in n(i)} \omega_j x_{f(j)} \\
&= \sum_{j\in n(i)} \omega_j \sum_m \left(\check{b}_{f(j),m}\hat{\boldsymbol{\gamma}}_m + e_{f(i)}\right) \\
&= \sum_m \left(\sum_{j\in n(i)} \omega_j \check{b}_{f(j),m}\right)\hat{\boldsymbol{\gamma}}_m + \sum_{j\in n(i)} \omega_j e_{f(j)} \\
&= \sum_m \left(\sum_{j\in n(i)} \omega_j b'_{j,m}\right)\hat{\boldsymbol{\gamma}}_m + \sum_{j\in n(i)} \omega_j e'_j \\
&= \sum_m b'_{i,m}\hat{\boldsymbol{\gamma}}_m + e'_i,\ \forall i \in I_2,
\end{aligned} \tag{B.6}$$

where $b'_{i,m}$ and e'_i are computed through an assumed linear interpolation using mapped pixels in the neighborhood $n(i)$ of the ith pixel. The pixel-wise linearity under the design according to (B.4) and (B.6) for all d pixels shows $\mathcal{R}_\delta(\cdot)$ is linear.

Bibliography

[1] P. K. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2273–2286, Nov. 2011.

[2] A. J. O'Toole, P. J. Phillips, S. Weimer, D. A. Roark, J. Ayyad, R. Barwick, and J. Dunlop, "Recognizing people from dynamic and static faces and bodies: Dissecting identity with a fusion approach," *Vision Research*, vol. 51, no. 1, pp. 74–83, 2011.

[3] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition from video," *European Conference on Computer Vision (ECCV)*, October 2012.

[4] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 919–926, June 2009.

[5] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 1225–1261, 2011.

[6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, Dec. 2003.

[7] P. J. Phillips, "Matching pursuit filters applied to face identification," *IEEE Trans. on Image Processing*, vol. 7, no. 8, pp. 1150–1164, Aug. 1998.

[8] V. M. Patel, T. Wu, S. Biswas, P. J. Philips, and R. Chellappa, "Dictionary-based face recognition under variable lighting and pose," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 3, pp. 954–965, June 2012.

[9] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210 –227, Feb. 2009.

[10] M. Tistarelli, S. Z. Li, and R. Chellappa, *Handbook of Remote Biometrics: For Surveillance and Security*. Springer, 2009.

[11] O. Arandjelovic and R. Cipolla, "Face recognition from video using the generic shape-illumination manifold," *European Conference on Computer Vision*, vol. 3954, pp. 27–40, 2006.

[12] P. K. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical analysis on stiefel and grassmann manifolds with applications in computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[13] Y. Hu, A. S. Mian, and R. Owens, "Sparse approximated nearest points for image set classification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 27–40, 2011.

[14] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen, "Image sets alignment for video-based face recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2626–2633, June 2012.

[15] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2042–2045, March 2010.

[16] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3501–3508, June 2010.

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2797, June 2009.

[18] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2008.

[19] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *Preprint*, 2011.

[20] Y.-C. Chen, C. S. Sastry, V. M. Patel, P. J. Phillips, and R. Chellappa, "Rotation invariant simultaneous clustering and dictionary learning," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[21] M. J. Tarr and D. J. Kriegman, "What defines a view?" *Vision Research*, vol. 41, pp. 1981–2004, 2001.

[22] V. Blanz, M. J. Tarr, and H. H. Bülthoff, "What object attributes determine canonical views?" *Perception*, vol. 28, pp. 575–599, 1999.

[23] O. Polonsky, G. Patané, B. Silvia, C. Gotsman, and M. Spagnuolo, "What's in an image? towards the computation of the "Best" view of an object," *The Visual Computer*, vol. 21(8-10), pp. 840–847, 2005.

[24] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips, "Video-based face recognition via joint sparse representation," *IEEE International Conference on Automatic Face and Gesture Recognition*, April 2013.

[25] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, "Dictionary-based face and people recognition from unconstrained video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, under review.

[26] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips, "Adaptive representations for video-based face recognition across pose," *IEEE International Conference on Computer Vision (ICCV)*, under review.

[27] Y.-C. Chen, C. S. Sastry, V. M. Patel, P. J. Phillips, and R. Chellappa, "In-plane rotation and scale invariant clustering using dictionaries," *IEEE Trans. on Image Processing*, vol. 22, no. 6, pp. 2166–2180, June 2013.

[28] Y.-C. Chen, V. M. Patel, J. K. Pillai, R. Chellappa, and P. J. Phillips, "Dictionary learning from ambiguously labeled data," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[29] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips, "Salient view selection based on sparse representation," *IEEE International Conference on Image Processing (ICIP)*, October 2012.

[30] ——, "Salient views and geometric dictionaries for object recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, under review.

[31] A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics*. Springer, 2006.

[32] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Visual tracking and recognition using probabilistic appearance manifolds," *Computer Vision and Image Understanding*, vol. 99, pp. 303–331, 2005.

[33] S. Z. Li, *Handbook of Face Recognition*. Springer, 2011.

[34] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[35] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, June 2010.

[36] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.

[37] K. Etemad and R. Chellappa, "Separability-based multiscale basis selection and feature extraction for signal and image classification," *IEEE Trans. on Image Processing*, vol. 7, no. 10, pp. 1453–1465, Oct. 1998.

[38] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries," *Tech. Report, University of Minnesota*, Dec. 2007.

[39] K. Huang and S. Aviyente, "Sparse representation for signal classification," *Neural Information Processing Systems Conference (NIPS)*, vol. 19, pp. 609–616, 2007.

[40] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2698, 2010.

[41] M. Ranzato, F. Haung, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[42] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[43] ——, "Supervised dictionary learning," *Advances in Neural Information Processing Systems*, Dec. 2008.

[44] L. Bar and G. Sapiro, "Hierarchical invariant sparse modeling for image analysis," *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 2397–2400, Sept. 2011.

[45] V. M. Patel and R. Chellappa, "Sparse representations, compressive sensing and dictionaries for pattern recognition," *Proc. Asian Conference on Pattern Recognition (ACPR)*, 2011.

[46] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, vol. 68, pp. 49–67, Feb. 2006.

[47] L. Meier, S. V. D. Geer, and P. Bhlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society: Series B*, vol. 70, pp. 53–71, Feb. 2008.

[48] A. Kale, A. K. Roychowdhury, and R. Chellappa, "Fusion of gait and face for human identification," *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2004.

[49] X. Zhou and B. Bhanu, "Integrating face and gait for human recognition at a distance in video," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, pp. 1119–1137, Oct. 2007.

[50] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The HumanID Gait Challenge Problem: Data Sets, Performance, and Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, Feb. 2005.

[51] P. J. Phillips, P. J. Flynn, J. R. Beveridge, W. T. Scruggs, A. J. O'Toole, D. Bolme, K. W. Bowyer, B. A. Draper, G. H. Givens, Y. M. Lui, H. Sahibzada, J. A. Scallan III, and S. Weimer, "Overview of the multiple biometrics grand challenge," *International Conference on Biometrics*, 2009.

[52] National Institute of Standards and Technology, "Multiple Biomertic Grand Challenge (MBGC)," *http://www.nist.gov/itl/iad/ig/mbgc.cfm*.

[53] ——, "Face and Ocular Challenge Series (FOCS)," *http://www.nist.gov/itl/iad/ig/focs.cfm*.

[54] R. Chellappa, J. Ni, and V. M. Patel, "Remote identification of faces: problems, prospects, and progress," *Pattern Recognition Letters*, vol. 33, no. 15, pp. 1849–1859, Oct. 2012.

[55] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

[56] A. Lanitis, C. Taylor, and T. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 743 –756, July 1997.

[57] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. on Image Processing*, vol. 13, no. 11, pp. 1491–1506, Nov. 2004.

[58] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, Apr. 2000.

[59] G. Aggarwal, A. Veeraraghavan, and R. Chellappa, "3d facial pose tracking in uncalibrated videos," *International Conference on Pattern Recognition and Machine Intelligence*, 2005.

[60] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understanding, Special Issue on Face Recognition*, vol. 91, pp. 214–245, July 2003.

[61] N. Shroff, P. Turaga, and R. Chellappa, "Video précis: Highlighting diverse aspects of videos," *IEEE Trans. on Multimedia*, vol. 12, no. 8, pp. 853–868, Dec. 2010.

[62] M. Aharon, M. Elad, and B. A., "K-SVD: An algorithm for desigining overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, Nov. 2006.

[63] H. Nguyen, V. M. Patel, N. Nasrabadi, and R. Chellappa, "Kernel dictionary learning," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012.

[64] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1 problems in compressive sensing," *SIAM Journal on Scientific Computing*, vol. 33, pp. 250–278, 2011.

[65] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. on Image Processing*, vol. 20, pp. 681–695, March 2011.

[66] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparsity-based robust multimodal biometrics recognition," *European Conference on Computer Vision Workshop on Information Fusion in Computer Vision for Concept Recognition*, 2012.

[67] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[68] R. Wang and X. Chen, "Manifold discriminant analysis," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 429–436, 2009.

[69] M. K. Kim, O. Arandjelovic, and R. Cipolla, "Discrimitive learning and recognition of image set classes using canonical correlations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1005–1018, June 2007.

[70] R. Wang, S. Shan, X. Chen, and W. Gao, "Manifold-manifold distance with application to face recognition based on image set," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[71] H. Cevikalp and B. Triggs, "Face recognition based on image sets," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2567–2573, 2010.

[72] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *A.I. Memo*, no. 1531, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[73] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, September 2003.

[74] Z. Yue, W. Zhao, and R. Chellappa, "Pose-encoded spherical harmonics for face recognition and synthesis using a single image," *EURASIP Journal on Advances in Signal Processing*, pp. 65:1–65:18, January 2008.

[75] L. Zhang and D. Samaras, "Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 351–363, March 2006.

[76] A. Gee and R. Cipolla, "Determining the gaze of faces in images," *Image and Vision Computing*, vol. 12, no. 10, pp. 639–647, 1994.

[77] T. Horprasert, Y. Yacoob, and L. Davis, "Computing 3-D head orientation from a monocular image sequence," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 242–247, 1996.

[78] J.-G. Wang and E. Sung, "EM enhancement of 3D head pose estimated by point at infinity," *Image and Vision Computing*, vol. 25, no. 12, pp. 1864–1874, 2007.

[79] K. Okada and C. v. d. Malsburg, "Face recognition and pose estimation with parametric linear subspaces," *Applied Pattern Recognition, Studies in Computational Intelligence*, vol. 91, pp. 49–74, 2008.

[80] A. J. O'Toole, J. Harms, S. L. Snow, D. R. Hurst, M. R. Pappas, J. H. Ayyad, and H. Abdi, "A video database of moving faces and people," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 812–816, May 2005.

[81] G. Haley and B. Manjunath, "Rotation-invariant texture classification using a complete space-frequency model," *IEEE Trans. on Image Processing*, vol. 8, no. 2, pp. 255–269, Feb. 1999.

[82] C. S. Sastry, M. Ravindranath, A. K. Pujari, and B. Deekshatulu, "A modified Gabor function for content based image retrieval," *Pattern Recognition Letters*, pp. 293–300, 2007.

[83] M. Do and M. Vetterli, "Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models," *IEEE Trans. on Multimedia*, vol. 4, no. 4, pp. 517–527, Dec. 2002.

[84] C.-M. Pun and M.-C. Lee, "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 590–603, May 2003.

[85] H. Wersing, J. Eggert, and E. Körner, "Sparse coding with invariance constraints," *Proc. Int. Conf. Artificial Neural Networks*, pp. 385–392, 2003.

[86] Y. Tomokusa, M. Nakashizuka, and Y. Iiguni, "Sparse image representations with shift and rotation invariance constraints," *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 256–259, Jan. 2009.

[87] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, "Shift and 2D rotation invariant sparse coding for multivariate signals," *IEEE Trans. on Signal Processing*, vol. 60, no. 4, pp. 1597–1611, April 2012.

[88] L. Bar and G. Sapiro, "Hierarchical dictionary learning for invariant classification," *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 3578–3581, March 2010.

[89] P. Ungureanu, E. David, and L. Goras, "On rotation invariant texture classification using two-grid coupled CNNs," *Seminar on Neural Network Applications in Electrical Engineering*, pp. 33–36, Sept. 2006.

[90] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[91] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, April 2002.

[92] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 550–571, 2004.

[93] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[94] S. Biswas, G. Aggarwal, and R. Chellappa, "Robust estimation of albedo for illumination-invariant matching and shape recovery," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 884–899, March 2009.

[95] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 1004–1008, 2005.

[96] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[97] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comp*, vol. 31, no. 2, pp. 890–912, 2008.

[98] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[99] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *1993 Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, 1993.

[100] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[101] P. Brodatz, "Texture: A photographic album for artists and designers," *New York: Dover*, 1966.

[102] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[103] R. Jin and Z. Ghahramani, "Learning with multiple labels," *Proceedings of Neural Information Processing Systems (NIPS)*, pp. 897–904, 2002.

[104] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[105] A. Shrivastava, J. K. Pillai, V. M. Patel, and R. Chellappa, "Learning discriminative dictionaries with partially labeled data," *International Conference on Image Processing (ICIP)*, 2012.

[106] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," *Intell. Data Anal.*, vol. 10, no. 5, pp. 419–439, Sep. 2006.

[107] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791 –804, April 2012.

[108] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *ICSI and U.C. Berkeley, TR-97-021*, April, 1998.

[109] F. Dellaert, "The expectation maximization algorithm," *Georgia Institute of Technology, GIT-GVU-02-20*, February, 2002.

[110] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, October, 2007.

[111] G. Huang, V. Jian, and E. Learned-Miller, "Unsupervised joint alignment of complex images," *International Conference on Computer Vision (ICCV)*, 2007.

[112] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[113] I. Chakravarty and H. Freeman, "Characteristic views as a basis for three-dimensional object recognition," *Proceedings of SPIE*, vol. 336, pp. 37–45, 1982.

[114] H. Freeman and I. Chakravarty, "The use of characteristic views in the recognition of three dimensional objects," *Pattern Recognition in Practice*, 1980.

[115] R. Wang and H. Freeman, "Object recognition based on characteristic view classes," *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 8–12, 1990.

[116] S. Chen and H. Freeman, "Characteristic-view modeling of curved-surface solids," *International Journal of Pattern Recognition and Artificial Intelligence - IJPRAI*, vol. 10, pp. 537–560, 1996.

[117] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," *IEEE CVPR*, vol. 2, pp. 511–516, 1999.

[118] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang, "An eigenspace update algorithm for image analysis," *Graphical Models and Image Processing*, vol. 59, no. 5, pp. 321–332, 1997.

[119] B. S. Manjunath, S. Chandrasekaran, and Y. F. Wang, "An eigenspace update algorithm for image analysis," *Proceedings of International Symposium on Computer Vision - ISCV*, pp. 551–556, 1995.

[120] J. Pillai, V. Patel, R. Chellappa, and N. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1877 –1893, Sept. 2011.

[121] 3D model store Humster3D, "http://humster3d.com/."

[122] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," *Shape Modeling International, Genova, Italy*, June 2004.

[123] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," *SIGGRAPH*, pp. 369–374, July 2000.

[124] Z. Yue and R. Chellappa, "Synthesis of silouettes and visual hull reconstruction for articulated humans," *IEEE Trans. on Multimedia*, vol. 10, no. 8, pp. 1565–1577, December 2008.