Structured Sparse Models for Classification

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Alexey Castrodad

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor of Philosophy

Prof. Guillermo Sapiro

November, 2012

Acknowledgements

First, I extend my gratitude to my advisor, Prof. Guillermo Sapiro. Throughout this process, he has been an outstanding mentor, a disciplined teacher, an untiring collaborator, and a compassionate friend. I have learned so much from him as a very creative and dedicated scientist, and an admirable human being. This work would have never be the same without his persistent support, advice, inspiration, and patience.

I am also grateful to my graduate school committee, Prof. Tryphon Georgiou, Prof. Nikolaos Papanikolopoulos, and Prof. Eli Foufoula-Georgiou for their time and effort invested with me, and their very helpful suggestions and comments while working on my thesis.

I would also like to thank my research collaborators for so many insightful discussions, and also for their great company and interaction, Prof. Lawrence Carin, Dr. Edward Bosch, Dr. Robert Rand, Timothy Khuon, Zhengming Xing, Dr. Mariano Tepper, Dr. Zhongwei Tang, and Dr. John Greer. It has been a true privilege to work with them.

To my awesome lab partners, my "panas," Dr. Iman Aganj, Dr. Xue Bai, Dr. Leah Bar, Pablo "P" Cancela, Dr. Julio Duarte, Marcelo Fiori, Jordan Hashemi, Dr. Jinyoung Kim, Dr. Oleg Kuybeda, Dr. Federico Lecumberry, Dr. Mona Mahmoudi, Dr. Ignacio Ramirez, Thiago Spina, Dr. Pablo Sprechmann, Dr. Mariano Tepper and Dr. Guoshen Yu. Your company and friendship is deeply appreciated, and our lunches, coffee and blackboard times, and laughing out loud moments really made my stay very special: thank you.

I give my most sincere gratitude to Linda Jagerson, for her incredible kindness, patience, and time invested while helping me in so many ways. Also to Jill Johnson, who really helped

me with answering so many questions, and always provided me with the most accurate strategies for getting all the administrative procedures done. To the Vector program coordinators, Michael Boling, May Dean, and William Thompson for their dedication and significant effort in supporting me throughout my days in graduate school.

I am very grateful to all the people who encouraged and supported me to pursue my PhD. My master's advisor Prof. Miguel Velez-Reyes, Prof. Luis O. Jimenez, Dr. Paul Salamonowicz, Dr. Michael Egan, John Findley, and Dr. Edward Bosch. I know there are many more, and I will personally thank you. You are an example to follow.

Finally, and very importantly, I am thankful to my family, especially my parents Cosy and Raul, for their unconditional love, care, and support. To Dagmar, for her love, understanding, and patience during this educational adventure.

Dedication

To my parents Cosy and Raul, for their infinite love, and for being always so close to me, from far away.

Abstract

The main focus of this thesis is the modeling and classification of high dimensional data using structured sparsity. Sparse models, where data is assumed to be well represented as a linear combination of a few elements from a dictionary, have gained considerable attention in recent years, and its use has led to state-of-the-art results in many signal and image processing tasks. The success of sparse modeling is highly due to its ability to efficiently use the redundancy of the data and find its underlying structure. On a classification setting, we capitalize on this advantage to properly model and separate the structure of the classes.

We design and validate modeling solutions to challenging problems arising in computer vision and remote sensing. We propose both supervised and unsupervised schemes for the modeling of human actions from motion imagery under a wide variety of acquisition conditions. In the supervised case, the main goal is to classify the human actions in the video given a predefined set of actions to learn from. In the unsupervised case, the main goal is to analyze the spatio-temporal dynamics of the individuals in the scene without having any prior information on the actions themselves. We also propose a model for remotely sensed hysperspectral imagery, where the main goal is to perform automatic spectral source separation and mapping at the subpixel level. Finally, we present a sparse model for sensor fusion to exploit the common structure and enforce collaboration of hyperspectral with LiDAR data for better mapping capabilities. In all these scenarios, we demonstrate that these data can be expressed as a combination of atoms from a class-structured dictionary. These data representation becomes essentially a "mixture of classes," and by directly exploiting the sparse codes, one can attain highly accurate classification performance with relatively unsophisticated classifiers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A *sparse representation* is the expression of data as a combination of only a few (from potentially many) building blocks of information. A critical assumption of course is that of *sparsity*, and fortunately, it is a natural and observable characteristic of data from many scientific fields. For example, it is well known that the Discrete Fourier Transform or wavelets are good (off the shelf) representations of natural images because most of their energy is concentrated in very few coefficients, hence the success of lossy compression standards like JPEG. In statistics, sparsity is an important tool for model selection and for determining the most relevant factors affecting an observed phenomena, e.g., the spectral abundance of a library of materials on a scene, and also in model fitting, where a sparsity inducing regularization is in many cases preferred over the traditional minimum energy fit.

From a mathematical perspective, sparse representations of data are obtained by solving a linear system of equations, where a given basis or *dictionary* is a matrix of variables, and the associated vector of coefficients dictate how much of those variables are present in each data sample. In general, recovering the sparsest solution in such a system of equations becomes a very hard combinatorial problem. Fortunately, advances in the theory of compressive sensing has permitted the emergence of very efficient methods to handle the computational requirements on reaching the solution, either by using greedy pursuit methods, or convex relaxation

techniques. The guarantees for reaching the solution are very dependent on the properties and structure of the dictionary. Therefore, the dictionary plays an essential role when seeking sparse representations, and wisely choosing and constructing it is fundamental for designing and implementing a reliable system.

Given enough instances of the same type of signals as those observed from the source of interest, one can attempt to infer the dictionary from this training data, i.e., dictionary learning. Using these training data for learning a dictionary brings an important benefit over analytically designed ones (e.g., a Fourier basis), since it is essentially a better fitted generative model and will hopefully find the data's underlying structure. Learning from the data, while simultaneously representing it in a sparse manner is known as *sparse modeling*, a term that we will use throughout the remaining of this document.

The success of sparse modeling has been demonstrated with state-of-the-art results in many signal processing, statistics, machine learning, artificial intelligence, and computer vision applications. Its success relies in part on the proper modeling of the signal's *structure*. This structure is often a combination of different types of phenomena, for example, the composition of natural images comes from a combination of textures and edges [1]. Accounting for this inherent structure in the data is crucial, not only for solving general inverse problems like denoising or deblurring, but also for higher level tasks such as classification, where the data can be seen analogously as the combination of different groups of classes.

This work focuses in the design and understanding of structured sparse models for classification of high dimensional multi-class data. This could be performed given *a priori* knowledge of the statistics and geometrical properties of the classes, i.e., supervised classification, or by automatically inferring the proper groups in the data, i.e., unsupervised classification. In this dissertation, we report research that demonstrates the capabilities of sparse modeling to efficiently handle diverse high dimensional classification problems, and more importantly, how such rich models exploit data redundancy and allow much simpler classification rules. Some of the main questions motivating this work are:

1. *To what extent does structured sparse modeling takes advantage of sparsity and redundancy for classification applications?* As we will see in the following chapters, sparse modeling is able to implicitly reduce dimensionality without having to recur to explicit dimension reduction schemes like Principal Components Analysis, which can potentially lose critical discriminative information and cannot handle structure information. This ability to properly model the data's low dimensional structure is a key advantage for classification and source separation.

2. *How good are the sparse representation coefficients as features or patterns for classification tasks?* Feature design and extraction for visual recognition is an active research topic in the machine learning and computer vision communities. These designed features are often very sophisticated and hand crafted. We claim that the sparse codes are well suited to serve as a good feature for representation and classification, and could be seen as a particular case of feature learning [2]. In fact, we will see throughout this document that features derived directly from the sparse codes could be physically interpretable, which is desirable in many remote sensing problems.

3. *Can we better exploit the inter-class relationships between sparse codes without rigorously sophisticating the model?* Given enough class separability, structured sparse modeling is able to correctly label the classes in the problem. However, there are situations where the classes highly depend from each other. We seek at properly model inter-class relationships and apply it to the problem of action classification.

4. *How do we enforce collaboration between data samples to improve the classification performance?* We would like to incorporate and exploit the cross-relationships and homogeneity of data in order to add more stability to the modeling process. We explicitly use ideas from manifold learning and graph theory to efficiently induce spatial coherence for mapping applications, and propose extensions to deal with multiple sensor information.

Throughout the remaining of this document, we will carefully address these questions and

will support the answers with extensive experimental validation. In Chapter 2, we provide a brief overview of sparse representations and dictionary learning. Then, we present four main chapters (described below) where we describe in detail the proposed solutions of several problems arising in the remote sensing and computer vision fields. Finally, we summarize our work and provide concluding remarks in Chapter 7.

1.1 Contributions

The main contributions of this work emerge by reasonably combining existing concepts with new ideas to design computationally efficient sparse modeling pipelines.

Chapter 3: Modeling human actions

In this chapter, we propose a solution to the problem of classifying human actions from motion imagery. This study has many important applications that range from psychological studies to surveillance systems. One of the main challenges is the massive amount of data that need to be processed, which is overwhelming when compared with the limited workforce analyzing it. Our contribution poses a viable solution to a need for automatic and semi-automatic recognition systems. These not only need to be accurate, but also need to be computationally efficient. We will provide a thorough analysis of the problem and describe in detail the proposed modeling scheme. Finally, we present results with diverse and challenging datasets.

We will show that we can attain both accurate and fast action recognition through a series of sparse modeling steps (in a hierarchical manner). These steps consist in first learning each action individually, and then learning all the actions simultaneously to find underlying interclass relationships. These two steps are a much simpler modeling than those previously proposed in the literature, and yield state-of-the-art results in virtually all publicly available datasets for action classification.

Chapter 4: Group activity analysis from a single video

In this chapter, we will also focus on the analysis of human actions, and in particular the spatio-temporal grouping of activities. Unlike the setting analyzed in Chapter 3, where sufficient training data are supposed to be available beforehand, here we assume a single observation. Therefore, without the availability of these data, many of the challenges associated with motion imagery and the classification problem itself are exacerbated. Also, contrary to supervised human action classification, here we picture scenarios where no labeling is available and all data has to be extracted from the single video. Given such few data, and no *a priori* information about the nature of the actions, what we are interested is in discovering human action groups instead of action recognition. We pay particular attention on modeling the general dynamics of individual actions, and the underlying idea we propose is that the activity dictionary learned for a given individual is also valid for representing the same activity of other individuals, and not for those performing different ones, nor for him/her-self after changing activity. We demonstrate the effectiveness of the proposed framework and its robustness to cluttered backgrounds, changes of human appearance, and action variability.

Chapter 5: Subpixel spectral mapping of remotely sensed imagery

Hyperspectral imaging (HSI) systems acquire images in which each pixel contains narrowly spaced measurements of the electromagnetic spectrum, allowing spectroscopic analysis. The data acquired by these spectrometers play significant roles in biomedical, environmental, land survey, and defense applications. It contains the geometrical (spatial) information from standard electro-optical systems,and also much higher spectral resolution, essential for material identification.

In this chapter, we describe a method for subpixel modeling, mapping, and classification in hyperspectral imagery using learned block-structured discriminative dictionaries. Here, each dictionary block is adapted and optimized to represent a material in a compact and sparse manner. The spectral pixels are modeled by linear combinations of subspaces defined by the learned

dictionary atoms, allowing for linear mixture analysis. This model provides flexibility in the sources representation and selection, thus accounting for spectral variability, small-magnitude errors, and noise. A spatial-spectral coherence regularizer in the optimization allows for pixels classification to be influenced by similar neighbors. We extend the proposed approach for cases for which there is no knowledge of the materials in the scene, unsupervised classification, and provide experiments and comparisons with simulated and real data. We also present results when the data have been significantly under-sampled and then reconstructed, still retaining high-performance classification, showing the potential role of compressive sensing and sparse modeling techniques in efficient acquisition/transmission missions for hyperspectral imagery.

Chapter 6: Sensor fusion and applications to spectral mapping

In this chapter, we follow the ideas proposed in Chapter 5, and extend them to handle challenges accompanying passive imaging. With passive imaging systems, the high dependence on external sources of light causes very low signal to noise ratios in regions with shade caused by elevation differences and partial occlusions. In order to overcome this difficulty, we propose a sparse modeling scheme that allows the collaboration of information from different sensors, i.e., sensor fusion. As a direct application, we use geometric features along with spectral information to improve the classification and visualization quality of hyperspectral imagery. We merge point cloud Light Detection and Ranging (LiDAR) data and hyperspectral imagery (HSI) into a single sparse modeling pipeline for subpixel mapping and classification. The model accounts for material variability and noise by using learned dictionaries that act as spectral endmembers. Additionally, the estimated abundances are influenced by the LiDAR point cloud density. Finally, we demonstrate the advantages of the proposed algorithm with co-registered LiDAR-HSI data. Although we tested the model with remotely sensed imagery, it can be easily generalized to other data/sensor types.

Chapter 2

Background

Throughout this work, we will model a collection of data samples linearly as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \Re^{m \times n}$. Each sample $\mathbf{x} = \mathbf{D}\mathbf{a} + \mathbf{n}$, where $\mathbf{n}$ is an additive component with bounded energy ($\|\mathbf{n}\|_2^2 \leq \varepsilon$) modeling both the noise and the deviation from the model, $\mathbf{a} \in \Re^k$ are the approximation weights, and $\mathbf{D} \in \Re^{m \times k}$ is a (possibly overcomplete, $k > m$) to be learned dictionary. Assuming for the moment that $\mathbf{D}$ is fixed, a sparse representation of a sample $\mathbf{x}$ can be obtained as the solution to the following optimization problem:

$$\mathbf{a}^* = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{a}\|_0 \quad \text{s.t.} \quad \frac{1}{2}\|\mathbf{D}\mathbf{a} - \mathbf{x}\|_2^2 \leq \varepsilon, \tag{2.1}$$

where $\|\cdot\|_0$ is a pseudo-norm that counts the number of nonzero entries. This means that the data belong to the union of low dimensional subspaces defined by the dictionary $\mathbf{D}$. Under assumptions on the sparsity of the signal and the structure of the dictionary $\mathbf{D}$ [1], there exists $\lambda > 0$ such that (2.1) is equivalent to solving the unconstrained problem

$$\mathbf{a}^* = \underset{\mathbf{a}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{D}\mathbf{a} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{a}\|_1, \tag{2.2}$$

known as the Lasso [3]. Notice that the ℓ_0 pseudo norm was replaced by an ℓ_1-norm, and we prefer in our work the formulation in (2.2) over the one in (2.1) since it is more stable, convex, and easily solvable using modern optimization techniques.

2.1 Dictionary learning

The dictionary $\mathbf{D}$ can be constructed for example using wavelets basis. However, since we know instances of the signal, we learn/infer the dictionary using training data, bringing the advantage of a better data fit compared with the use of off-the-shelf dictionaries. Sparse modeling of data can be done via an alternation minimization scheme similar in nature to K-means, where we fix $\mathbf{D}$, obtain the sparse code $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \Re^{k \times n}$, then minimizing with respect to $\mathbf{D}$ while fixing $\mathbf{A}$ (both sub-problems are convex), and continue this process until reaching a (local) minimum to get

$$(\mathbf{D}^*, \mathbf{A}^*) = \underset{\mathbf{D},\mathbf{A}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{DA} - \mathbf{X}\|_F^2 + \lambda \sum_{i=1}^{n} \|\mathbf{a}_i\|_1, \tag{2.3}$$

which can be efficiently solved using algorithms like the K-SVD [4, 5].

This concludes the general formulation of sparse modeling. In the following chapters, we will adapt and improve this model in order to achieve the desired outcomes in a number of applications.

Chapter 3

Modeling Human Actions

3.1 Chapter summary

In this chapter, we model human actions from motion imagery. We propose a hierarchical structured sparse model scheme which consists of modeling individual actions at the top level, and the relationship between these actions at the second level . This scheme is conceptually simpler than most recently proposed schemes for human action recognition/classification, and achieves state-of-the-art results in very challenging scenarios.

3.2 Introduction

We are living in an era where the ratio of data acquisition over exploitation capabilities has dramatically exploded. With this comes an essential need for automatic and semi-automatic tools that could aid with the processing requirements in most technology-oriented fields. A clear example pertains to the surveillance field, where video feeds from possibly thousands of cameras need to be analyzed by a limited amount of operators on a given time lapse. As simple as it seems for us to recognize human actions, it is still not well understood how the processes in our visual system give our ability to interpret these actions, and consequently is difficult to effectively emulate these through computational approaches. In addition to the intrinsic large

variability for the same type of actions, factors like noise, camera motion and jitter, highly dynamic backgrounds, and scale variations, increase the complexity of the scene, therefore having a negative impact in the performance of the classification system. In this chapter, we focus in a practical design of such a system, that is, an algorithm for supervised classification of human actions in motion imagery.

There are a number of important aspects of human actions and motion imagery in general that make the particular task of action classification very challenging:

1. Data is very high dimensional and redundant: Each video will be subdivided into spatio-temporal patches which are then vectorized, yielding high-dimensional data samples. Redundancy occurs from the high temporal sampling rate, allowing relatively smooth frame-to-frame transitions, hence the ability to observe the same object many times (not considering shot boundaries). In addition, many (but not all) of the actions have an associated periodicity of movements. Even if there is no periodicity associated with the movements, the availability of training data implies that the action of interest will be observed redundantly, since overlapping patches characterizing a specific spatio-temporal behavior are generally very similar, and will be accounted multiple times with relatively low variation. These properties of the data allow the model to benefit from the *blessings* of high dimensionality [6], and will be key to overcoming noise and jitter effects, allowing simple data representations by using simple features, while yielding stable and highly accurate classification rates.

2. Human activities are very diverse: Two people juggling a soccer ball can do that very differently. Same for people swimming, jumping, boxing, or performing any of the activities we want to classify. Learning simple representations is critical to address such variability.

3. Different human activities share common movements: A clear example of this is the problem of distinguishing if a person is either running or jogging. Torso and arms movements may be very similar for both actions. Therefore, there are spatio-temporal structures that

are shared between actions. While one would think that a person running moves faster than a person jogging, in reality it could be the exact opposite (consider racewalking). This phenomena suggests that our natural ability to classify actions is not based only on local observations (e.g., torso and arms movements) or global observations (e.g., person's velocity) but on local *and* global observations. This is consistent with recent psychological research indicating that the perception of human actions are a combination of spatial hierarchies of the human body along with motion regularities [7]. Relationships between activities play an important role in order to compare among them, and this will be incorporated in our proposed framework via a simple deep learning structure.

4. Variability in the video data: While important applications, here addressed as well, consist of a single acquisition protocol, e.g., surveillance video; the action data we want to classify is often recorded in a large variety of scenarios, leading to different viewing angles, resolution, and general quality. This is the case for example of the YouTube data we will use as one of the testing scenarios for our proposed framework.

In this chapter, we consider these aspects of motion imagery and human actions and propose a hierarchical, two-level sparse modeling framework that exploits the high dimensionality and redundancy of the data. Differently from the recent literature, discussed in Section 3.3, we learn inter-class relationships using both global and local perspectives. As described in detail in Section 3.4, we combine ℓ_1-minimization with structured dictionary learning, and show that with proper modeling, in combination with a reconstruction and complexity based classification procedure using sparse representations, a *single feature* and a *single sampling scale* are sufficient for highly accurate activity classification on a large variety of examples.

We claim that there is a great deal of information inherent in the sparse representations that have not yet been fully explored. In [8] for example, class-decision functions were incorporated in the sparse modeling optimization to gain higher discriminative power. In the results the authors show that significant gain can be attained for recognition tasks, but always at the cost of more sophisticated modeling and optimizations. We drift away from these ideas by explicitly

exploiting the sparse coefficients in a different way such that, even though it derives from a purely generative model, takes more advantage from the structure given in the dictionary to further model class distributions with a simpler model. In Section 3.5 we evaluate the performance of the model using four publicly available datasets: the KTH Human Action Dataset, the UT-Tower Dataset, the UCF-Sports Dataset, and the YouTube Action Dataset, each posing different challenges and environmental settings, and compare our results to those reported in the literature. Our proposed framework uniformly produces state-of-the-art results for all these data, exploiting a much simpler modeling than those previously proposed in the literature. Finally, we provide concluding remarks and future research in Section 3.6.

3.3 Related Work

The recently proposed schemes for action classification in motion imagery are mostly feature-based. These techniques include three main steps. The first step deals with "interest point detection," and it consists of searching for spatial and temporal locations that are appropriate for performing feature extraction. Examples are Cuboids [9], Harris3D [10], Hessian [11], and dense sampling[1] [12, 13, 14]. This is followed by a "feature acquisition" step, where the video data at the locations specified from the first step undergo a series of transformation processes to obtain descriptive features of the particular action, many of which are derived from standard static scene and object recognition techniques. Examples are SIFT [15], the Cuboids feature [9], Histograms of Oriented Gradients (HOGs) [16], and its extension to the temporal domain, i.e., HOG3D [17], combinations of HOG and Histograms of Optical Flow (HOF) [16], Extended Speeded Up Robust Features (ESURF), Local Trinary Patterns [18], Motion Boundary Histograms (MBH) [19], and Local Motion Patterns (LMP) [20],. Finally, the third step is a "classification/labeling" process, where bag-of-features consisting of the features extracted (or vector quantized versions) from the second step are fed into a classifier, often a Support Vector Machine (SVM). Please refer to [21] and [22] for comprehensive reviews and

[1] Dense sampling is not an interest point detector *per se*. It extracts spatio-temporal multi-scale patches indiscriminately throughout the video at all locations.

pointers to feature-based as well as other proposed schemes.

In practice, it is difficult to measure what combinations of detectors and features are best for modeling human actions. In [22], the authors conducted exhaustive comparisons on the classification performance of several spatio-temporal interest point detectors and descriptors using nonlinear SVMs, using publicly available datasets. They observed that most of the studied features performed relatively well, although their individual performance was very dependent on the dataset. For example, interest point detection based feature extraction performed better than dense sampling on datasets with relatively low complexity like KTH, while dense sampling performed slightly better in more realistic/challenging datasets like UCF-Sports. In [21], the authors performed a similar evaluation using a less realistic dataset (the KTH action dataset) and concluded that the Cuboid detector combined with the Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) descriptor [23, 24] gives the best performance in terms of classification accuracy. In this work, we do not look at designing detectors or descriptors but rather give greater attention into developing a powerful model for classification using sparse modeling. We use a very simple detector and descriptor, and one single spatio-temporal scale to better show that sparse modeling is capable of taking high dimensional and redundant data and translate it into highly discriminative information. Also, given that the gain in performance of dense sampling is not significant, and it takes longer computation times, we use a simple interest point detector (by thresholding) instead of dense sampling, simply for a faster and more efficient sampling process, such that the spatio-temporal patches selected contain slightly higher velocity values relative to a larger background.

Apart from these localized, low-level descriptors, there has been research focused on the design of more global, high-level descriptors, which encode more semantic information. For example, [25] proposed to represent the video as the collected output from a set of action detectors sampled in semantic and in viewpoint space. Then, these responses are max-pooled and concatenated to obtain semantic video representations. [26] proposed to model human actions as 3D shapes on a space-time volume. Then, the solution of the Poisson equation is used to extract different types of feature descriptors to discriminate between actions. [27] proposed to

use a similar space-time volume by calculating the point correspondences between consecutive frames, and action representative features were computed by differential geometry analysis of the space-time volume's surface. Other types of high-level features are the joint-keyed trajectories or human pose, which have been used for example by [28] and [29]. [30] proposed to encode entire video clips as single vectors by using Motion Interchange Patterns (MIP), which encode sequences by comparing patches between three consecutive frames and applying a series of processes for background suppression and video stabilization.

Sparse coding along with dictionary learning has proven to be very successful in many signal and image processing tasks, especially after highly efficient optimization methods and supporting theoretical results emerged. More recently, it has been adapted to classification tasks like face recognition [31] (without dictionary learning), digit and texture classification [8, 32], hyperspectral imaging [33, 34], among numerous other applications. It has also been applied recently for motion imagery analysis for example in [35, 36, 37, 38]. In [20], the authors used learned dictionaries in three ways: individual dictionaries (one per action), a global (shared) dictionary, and a concatenated dictionary. Individual dictionaries are separately learned for each class of actions and unlabeled actions are assigned to the class whose dictionary gives the minimum reconstruction error. The concatenated dictionary is formed by concatenating all the individual dictionaries, and unlabeled actions are assigned to the class whose corresponding subdictionary contributes the most to the reconstruction. To create the (shared) dictionary, a single common and unstructured dictionary is learned using all training feature data from every class. The dictionary coding coefficients of training actions are used to train a multi-class SVM. In [36], the authors propose to learn a dictionary in a recursive manner by first extracting high response values coming from the Cuboids detector, and then using the resulting sparse codes as the descriptors (features), where PCA is optionally applied. Then, as often done for classification, the method uses a bag-of-features with K-bin histograms approach for representing the videos. To classify unlabeled videos, these histograms are fed into a nonlinear χ^2-SVM. In contrast to our work, the authors learn a basis globally, while the proposed method learns it in a per-class manner, and follows a different scheme for classification. We also learn inter-class

relationships via a two levels (deep-learning) approach.

In [37], the authors build a dictionary using vectorized log-covariance matrices of 12 hand-crafted features (mostly derived from optical flow) obtained from entire labeled videos. Then, the vectorized log-covariance matrix coming from an unlabeled video is represented with this dictionary using ℓ_1-minimization, and the video is classified by selecting the label associated with those dictionary atoms that yield minimum reconstruction error. In contrast to our work, the dictionary in [37] is hand-crafted directly from the training data and not learned. While similar in nature to the ℓ_1-minimization procedure used in our first level, the data samples in [37] are global representations of the entire video, while our method first models all local data samples (spatio-temporal patches), followed by a fast global representation on a second stage, leading to a hierarchical model that learns both efficient per-class representations (first level) as well as inter-class relationships (second level).

In [39], the authors propose a three-level algorithm that simulates processes in the human visual cortex. These three levels use feature extraction, template matching, and max-pooling to achieve both spatial and temporal invariance by increasing the scale at each level. Classification of these features is performed using a sparsity inducing SVM. Compared to our model, except for the last part of its second level, the features are hand-crafted, and is overall a more sophisticated methodology.

In [38], a convolutional Restricted Boltzmann Machine (convRBM) architecture is applied to the video data for learning spatio-temporal features by estimating frame-to-frame transformations implicitly. They combine a series of sparse coding, dictionary learning, and probabilistic spatial and temporal pooling techniques (also to yield spatio-temporal invariance), and then feed sparse codes that are max-pooled in the temporal domain (emerging from the sparse coding stage) into an RBF-SVM. Compared to our work, this method deals with expensive computations on a frame by frame basis, making the training process very time consuming. Also they train a global dictionary of all actions. In contrast, our method learns per-class/activity dictionaries independently using corresponding training data all at once (this is also beneficial when new classes appear, no need to re-train the entire dictionary). In [13], Independent Subspace

Analysis (ISA) networks are applied for learning from the data using two levels. Blocks of video data are used as input to the first ISA network following convolution and stacking techniques. Then, to achieve spatial invariance, the combined outputs from the first level are convolved with a larger image area and reduced in size using PCA, and then fed to the second level, another ISA network. The outputs from this level are vector quantized (bag-of-features approach), and a χ^2-SVM is used for classification. The method here proposed does not uses PCA to reduce the dimensionality of the data after the first level, as the dimension reduction derives more directly and naturally by using sum-pooling in a per-class manner after the first level.

Note that the hierarchical modeling of the proposed method is different from [39], [13], and [38]. These works progress from level to level by sequentially increasing spatial and/or temporal scales, thus benefiting from a multi-scale approach (spatial invariance), while our work progresses from locally oriented representations using only one scale,[2] to a globally oriented video representation deriving directly from the sparse model, and not from a bag-of-features approach or series of multi-scale pooling mechanisms. Also, the proposed scheme, as we will discuss in more detail next, produces sparse codes that contain information in a different way than the sparse codes produced with the global dictionaries in [36, 38]. This is achieved by explicit per-class learning and pooling, yielding a C-space, for C activities, representation with invariance to the per-class selection of action primitives (learned basis).

3.4 Sparse Modeling for Action Classification

3.4.1 Model overview

Assume we have a set of labeled videos, each containing 1 of C known actions (classes) with associated label $j \in [1, 2, ..., C]$.[3] The goal is to learn from these labeled videos in order to classify new incoming unlabeled ones, and achieve this via simple and computationally efficient

[2] In this work, only a single scale is used to better illustrate the model's advantages, already achieving state-of-the-art results. A multi-scale approach could certainly be beneficial.

[3] In this work, as commonly done in the literature, we assume each video has been already segmented into time segments of uniform (single) actions. Considering we will learn and detect actions based on just a handful of frames, this is not a very restrictive assumption.

paradigms. We solve this with a two-level feature-based scheme for supervised learning and classification, which follows the pipeline shown in Figure 3.1.

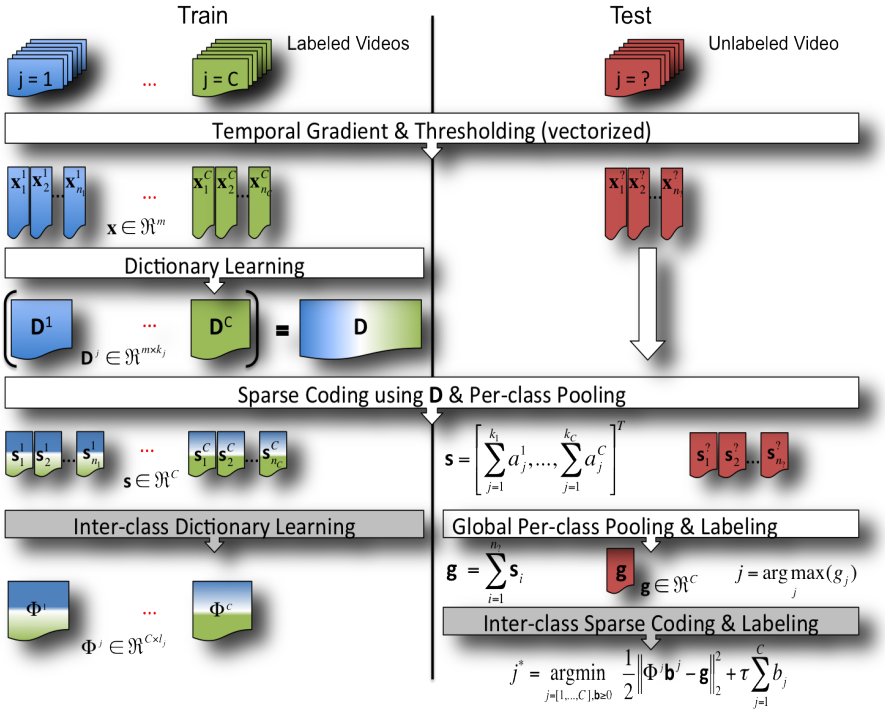


Figure 3.1: Algorithm overview. The left and right sides illustrate the learning and classification procedures, respectively. The processes in white boxes represent the first level of sparse modeling. The processes in gray boxes represent the second level. (This is a color figure.)

For learning, we begin with a set of labeled videos, and for each action separately, we extract and vectorize overlapping spatio-temporal patches consisting of the videos' temporal gradients at locations that are above a pre-defined energy threshold. In other words, we exploit spatio-temporal ($3D$) patches that have sufficient activity. During the *first level of training*, these labeled training samples (i.e., $\mathbf{x}^j$ vectors from patches belonging to videos of class j) serve as input to a dictionary learning stage (see Chapter 2, Equation 2.3). In this stage, an action-specific dictionary $\mathbf{D}^j$ of k_j atoms is learned for each of the C classes. After learning all

C dictionaries, a structured dictionary $\mathbf{D}$ consisting of the concatenation of these subdictionaries is formed. A sparse representation of these training samples (spatio-temporal $3D$ patches) using ℓ_1-minimization yields associated sparse coefficients vectors. These coefficient vectors are pooled in a per-class manner, so that they quantify the contribution from each action (i.e., the $\mathbf{s}^j$ vectors, each patch of class j producing one). Then, on a *second level of training*, these per-class pooled samples become the data used for learning a second set of action-specific dictionaries Ψ^j of l_j atoms. While the first level dictionaries $\mathbf{D}^j$ are class independent, these second level ones model the inter-relations between the classes/actions. With this, the off-line learning stage of the algorithm concludes.

To classify a video with unknown label "?," we follow the same feature extraction procedure, where test samples, $\mathbf{x}^?$'s (again consisting of spatio-temporal patches of the video's temporal gradient) are extracted and sparsely represented using the (already learned) structured dictionary $\mathbf{D}$. After sparse coding, the resulting vectors of coefficients are also pooled in a per-class manner, yielding the $\mathbf{s}^?$'s vectors. For a sometimes sufficient first level classification, a label is assigned to the video by majority voting, that is, the class with the largest contribution using all the pooled vectors is selected. For a second level classification, the same majority voted single vector is sparsely represented using the concatenation of all the dictionaries Ψ^j. The video's label j^* is selected such that the representation obtained with the $j-th$ action subdictionary Ψ^j yields the minimum sparsity *and* reconstruction trade-off.

3.4.2 Modeling Local Observations as Mixture of Actions: Level-1

Let $\mathbf{I}$ be a video, and $\mathbf{I}_t$ its temporal gradient. In order to extract informative spatio-temporal patches, we use a simple thresholding operation. More precisely, let $\mathbf{I}_t(p)$ be a $3D$ (space+time) patch of I_t with center at location $p \in \Omega$, where Ω is the video's spatial domain. Then, we extract data samples $\mathbf{x}(p) = vect(|\mathbf{I}_t(p)|)$ such that $|I_t(p)| > \delta, \forall p$, where δ is a pre-defined threshold, and $vect(\cdot)$ denotes vectorization (in other words, we consider spatio-temporal patches with above threshold temporal activity). Let all the data extracted from the videos this way be denoted by $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \Re^{m \times n}$, where each column $\mathbf{x}$ is a data sample. Here m is then the data

dimension $m = r \times c \times w$, where r, c, and w are the pre-defined number of rows, columns, and frames of the spatio-temporal patch, respectively, and n the number of extracted "high-activity" patches.

Once the action-dependent dictionaries are learned, we express each of the data samples (extracted spatio-temporal patches with significant energy) as sparse linear combinations of the different actions by forming the block-structured dictionary $\mathbf{D} = [\mathbf{D}^1, ..., \mathbf{D}^C] \in \Re_+^{m \times k}$, where $k = \sum_{j=1}^{C} k_j$. Then we get, for the entire data being processed $\mathbf{X}$,

$$\mathbf{A}^* = \underset{\mathbf{A} \succeq 0}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{D}\mathbf{A} - \mathbf{X}\|_F^2 + \lambda \sum_{i=1}^{n} \mathscr{S}(\mathbf{a}_i), \tag{3.1}$$

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \Re_+^{k \times n}$, $\mathbf{a}_i = [a_i^1, ..., a_i^{k_1}, ..., a_i^{k_C}]^T \in \Re_+^k$, and $n = \sum_{j=1}^{C} n_j$. Note that this includes all the high energy spatio-temporal patches from all the available training videos for all the classes.

Note that with this coding strategy, we are expressing the data points (patches) as a sparse linear combination of elements of the entire structured dictionary $\mathbf{D}$, not only of their corresponding class-dependent subdictionary (see also [31] for a related coding strategy for facial recognition). That is, each data sample becomes a "mixture" of the actions modeled in $\mathbf{D}$, and the component (or fraction) of the $j-th$ action mixture is given by its associated $\mathbf{a}^j$. The idea is to quantify movement sharing between actions. If none of the local movements associated with the $j-th$ action are shared, then the contribution from the other action representations will be zero, meaning that the data sample is purely pertaining of the $j-th$ action, and is quantified in $\mathscr{S}(\mathbf{a}^j)$. On the other hand, shared movements will be quantified with nonzero contributions from more than one class, meaning that the data samples representing these may lie in the space of other actions. This strategy permits to share features between actions, and to represent actions not only by their own model but also by how connected they are to the models of other actions. This cross-talking between the different action's models (classes) will be critical in the second stage of the learning model, as will be detailed below. The sparsity induced in the minimization should reduce the number of errors caused by this sharing effect. Furthermore, these mixtures can be modeled by letting $\mathbf{s} = [\mathscr{S}(\mathbf{a}^1), ..., \mathscr{S}(\mathbf{a}^C)]^T \in \Re_+^C$ be the per-class ℓ_1-norm

vector corresponding to the data sample $\mathbf{x}$, and letting $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_n] \in \Re_+^{C \times n}$ be the matrix of all per-class ℓ_1-norm samples. By doing this, the actions' contributions in the sample are quantified with invariance to the subset selection in the sub-dictionaries $\mathbf{D}^j$, and the dimensionality of the data is notably reduced to C-dimensional vectors in a reasonable way, as opposed to an arbitrary reduction using for example PCA. This reduced dimension, which again expresses the inter-class (inter-action) components of the data, low dimensional input to the next level of the learning process.

3.4.3 Modeling Global Observations: Level-2

Once we obtain the characterization of the data in terms of a linear mixture of the C actions, we begin our second level of modeling. Using the training data from each class, $\mathbf{S}^j \in \Re_+^{C \times n_j}$ (the C-dimensional s^j vectors for class j), we model inter-class relationships by learning a second set of per-class dictionaries $\Psi^j \in \Re_+^{C \times l_j}$ as:

$$\Psi^{j*} = \arg \min_{(\Psi^j, \mathbf{B}^j) \succeq 0} \frac{1}{2} \|\Psi^j \mathbf{B}^j - \mathbf{S}^j\|_F^2 + \tau \sum_{i=1}^{n_j} \mathscr{S}(\mathbf{b}^j), \tag{3.2}$$

where $\mathbf{B}^j = [\mathbf{b}_1^j, ..., \mathbf{b}_{n_j}^j] \in \Re^{l_j \times n_j}$ are the associated sparse coefficients from the samples in the $j-th$ class, and $\tau > 0$ controls the trade-off between class reconstruction and coefficients' sparsity. Notice that although the dictionaries Ψ^j are learned on a per-class basis, each models how data samples corresponding to a particular action j can have energy contributions from other actions, since they are learned from the n_j mixed coefficients $\mathbf{s}^j \in \Re_+^C$. Inter-class (actions) relationships are then learned this way.

This completes the description of the modeling as well as the learning stage of the proposed framework. We now proceed to describe how is this modeling exploited for classification.

3.4.4 Classification

In the first level of our hierarchical algorithm, we learned dictionaries using extracted spatio-temporal samples from the labeled videos. Then, each of these samples are expressed as a linear

combination of all the action dictionaries to quantify the amount of action mixtures. After class sum-pooling (ℓ_1-norm on a per-class basis) of the corresponding sparse coefficients, we learned a second set of dictionaries modeling the overall per-class contribution per sample. We now describe two decision rules for classification that derive directly from each modeling level.

Labeling After Level 1

It is expected that the information provided in $\mathbf{S}$ should be already significant for class separation. Let $\mathbf{g} = \mathbf{S1} \in \Re_+^C$, where $\mathbf{1}$ is a $n \times 1$ vector with all elements one (note that now n is the amount of spatio-temporal patches with significant energy present in a *single* video being classified). Then, we classify a video according to the mapping function $f_1(\mathbf{g}) : \Re_+^C \rightarrow \mathscr{Z}$ defined as

$$f_1(\mathbf{g}) = \{j | g_j > g_i, j \neq i, (i,j) \in [1,...,C]\}. \tag{3.3}$$

This classification, already provides competitive results, especially with actions that do not share too many spatio-temporal structures, see Section 3.5. The second layer, that due to the significant further reduction in dimensionality (to C, the number of classes), is computationally negligible, improves the classification even further.

Labeling After Level 2

There are cases where there are known shared (local) movements between actions, or cases where a video is composed of more than one action (e.g., running and then kicking a ball). As discussed before, the first layer is not yet exploiting inter-relations between the actions. Inspired in part on ideas from [40], we develop a classification scheme for the second level. Let

$$\mathscr{R}(\Psi, \mathbf{g}) = \min_{\mathbf{b} \succeq 0} \frac{1}{2} \|\Psi \mathbf{b} - \mathbf{g}\|_2^2 + \tau \mathscr{S}(\mathbf{b}), \tag{3.4}$$

then, we classify the video as

$$f_2(\mathbf{g}) = \{j | \mathscr{R}(\Psi^j, \mathbf{g}) < \mathscr{R}(\Psi^i, \mathbf{g}), j \neq i, (i,j) \in [1,...,C]\}. \tag{3.5}$$

Here, we classify by selecting the class yielding a minimum reconstruction and complexity as given by $\mathscr{R}(\Psi^j, \mathbf{g})$, corresponding to the energy associated to the $j-th$ class. Notice that in this procedure only a single vector $\mathbf{g}$ in $\Re_+^C$ needs to be sparsely represented for the whole video being classified, which is computationally very cheap of course.

3.4.5 Comparison of representations for classification

The bag-of-features approach is one of the most widely used techniques for action classification. It basically consists of applying K-means clustering to find K centroids, i.e., visual words, that are representative of all the training samples. Then, a video is represented as a histogram of visual word occurrences, by assigning one of the centroids to each of the extracted features in the video using (most often) Euclidean distance. These K centroids are found using a randomly selected subset of features coming from all the training data. While this has the advantage of not having to learn C sub-problems, it is not explicitly exploiting/modeling label information available in the given supervised setting. Therefore, it is difficult to interpret directly the class relationships in these global, high dimensional histograms (K is usually in the $3,000-4,000$ range). In addition, the visual words expressed as histograms equally weight the contribution from the data samples, regardless of how far these are from the centroids. For example, an extracted descriptor or feature from the data that does not correspond to any of the classes (e.g., background), will be assigned to one of the K centroids in the same manner as a descriptor that truly pertains to a class. Therefore, unless a robust metric is used, further increasing the computational complexity of the methods, this has the disadvantage of not properly accounting for outliers and could significantly disrupt the data distribution. In the proposed method, each of the data samples is represented as a sparse linear combination of dictionary atoms, hence represented from union of subspaces. Instead of representing an extracted feature with its closest centroid, it is represented by a *weighted* combination of atoms, thus better managing outliers. Analogue to a Mixture of Gaussians (MoG), the bag-of-features representation can be considered as a hard-thresholded MoG, where only one Gaussian distribution is allowed per sample, and its associated weight equals to one.

The learning process at the first level of the proposed model uses samples (vectorized spatio-temporal patches) from each action independently (in contrast to learning a global dictionary), and later encodes them as linear combinations of the learned dictionary atoms from all classes, where the class contribution is explicitly given in the obtained sparse codes. Since each data sample from a specific class can be represented by a different subset of dictionary atoms, the resulting sparse codes can have significant variations in the activation set. Sum-pooling in a per-class manner achieves invariance to the class subset (atom) selection. These sum-pooled vectors are used to quantify the association of the samples with each class (activity), and a significant dimensionality reduction is obtained by mapping these codes into a C-dimensional space (in contrast to performing explicit dimension reduction as in some of the techniques described above). We learn all the representations in a nonnegative fashion. This is done for two reasons. First, we use the absolute value of the temporal gradient (to allow the same representation for samples with opposite contrast), so all data values are nonnegative. Second, each data sample is normalized to have unit magnitude. After the per-class sum-pooling, this allows a mapping that is close to a probability space (the ℓ_1 norm of the sparse codes will be close to one). Therefore, the coefficients associated with each class give a good notion of the probability of each class in the extracted features.

Consider the example illustrated in Figure 3.2. Shown are the first three principal components of all the C-dimensional sum-pooled vectors corresponding to the *jog, run,* and *walk* actions from the KTH dataset (details on this standard dataset will be presented in the experimental section). As we can see, some of the data points from each class intersect with the other two classes, corresponding to shared movements, or spatio-temporal structures that may well live in any of the classes' subspaces, a per-sample effect which we call *action mixtures*. Also, the actions have a global structure and position relative to each other within the $3D$ spatial coordinates, which appears to be related to the subjects' velocity (*jog* seems to be connected to *walk* and *run*). Therefore, this local characterization obtained at the first level, where the data points are mapped into a mixture space, indeed have a global structure. Thus, the purpose of the second level is to model an incoming video by taking into account its entire data

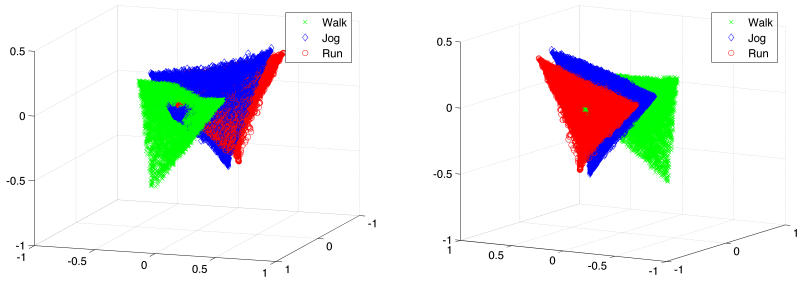


Figure 3.2: Front and rear views of the first three principal components corresponding to the per-class ℓ_1-norm of data samples (using all the training videos from the KTH dataset) after the first level of sparse coding in our algorithm. The samples in green correspond to the *walk* class, the samples in blue correspond to the *jog* class, and the samples in red correspond to the *run* class. (This is a color figure.)

distribution relative to this global structure, considering relationships between classes (actions), and expressing it sparsely using dictionary atoms that span the space of the individual actions. Such cross-action learning and exploitation is unique to the proposed model, when compared to those described above, and is achieved working on the natural low dimensional C-space, thereby being computationally very efficient.

3.5 Experimental Results

We evaluate the classification performance of the proposed method using 4 publicly available datasets: KTH, UT-Tower, UCF-Sports, and YouTube. The results presented include performance rates for each of the two levels of modeling, which we call SM-1 for the first level, and SM-2 for the second level. Separating both results will help in understanding the properties and capabilities of the algorithm in a per-level fashion. Remember that the additional computational cost of the second layer is basically zero, a simple sparse coding of a single low dimensional vector. Additionally, to illustrate the discriminative information available in the per-class sum-pooled vectors S, we include classification results of all datasets using a χ^2-kernel SVM in a

Table 3.1: Parameters for each of the datasets. The first three columns are related to feature extraction parameters. The last four columns specify sparse coding/dictionary-learning parameters.

Dataset	**Feature Extraction**			**Sparse Modeling**			
	n/clip	δ	**m**	λ	τ	k_j	l_j
KTH	30000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
UT-Tower	30000/#clips	0.10	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
UCF-Sports	30000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
YouTube	40000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$0.5/C$	768	64

one-against-the other approach, and we call this SM-SVM. In other words, the output from the first level of the proposed algorithm is the input to SM-SVM. For each classifier, we built the kernel matrix by randomly selecting 3,000 training samples. We report the mean accuracy after 1,000 runs. Finally, for comparison purposes, we include the best three performance rates reported in the literature. Often, these three are different for different datasets, indicating a lack of universality in the different algorithms reported in the literature (though often some algorithms are always close to the top, even if they do not make the top 3). Confusion matrices for SM-1 and SM-2 are also included for further analysis.

Table 3.1 shows the parameters used in SM-1 and SM-2 for each of the datasets in our experiments. The values were chosen so that good empirical results were obtained, but standard cross-validation methods can be easily applied to obtain optimal parameters. Note how we used the same basic parameters for all the very distinct datasets. The first three columns specify the amount of randomly selected spatio-temporal patches per video clip, the threshold used for interest point detection, and the size of the spatio-temporal overlapping patches, respectively. The last four columns specify the sparsity parameters and the number of dictionary atoms used for SM-1 and SM-2 modeling, respectively. Note how for simplicity we also used same dictionary size for all classes. We now present the obtained results.

3.5.1 KTH

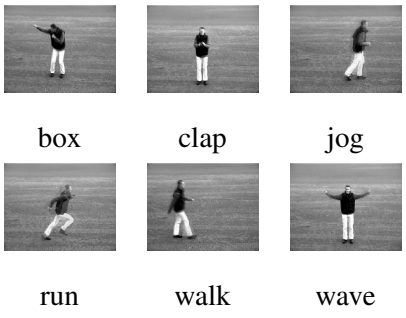


Figure 3.3: Sample frames from the KTH dataset.

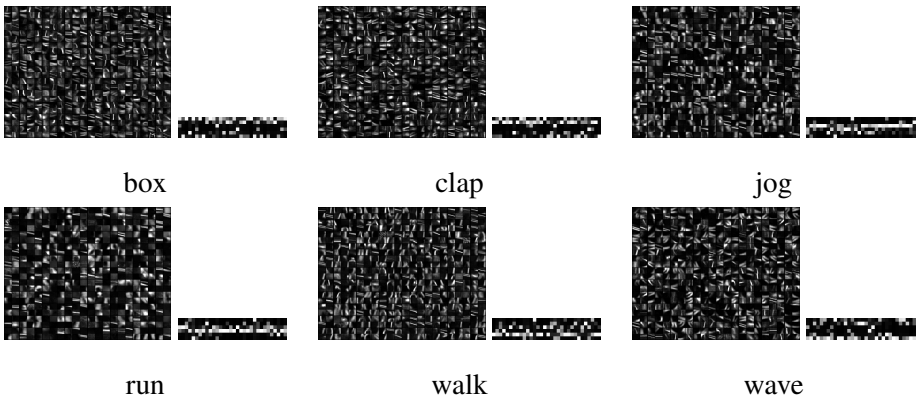


Figure 3.4: Learned action dictionaries from the KTH dataset for both levels.

The KTH dataset[4] [41] is one of the most popular benchmark action data. It consists of approximately 600 videos of 25 subjects, each performing $C = 6$ actions: *box, clap, jog, run, walk*, and *wave*. Each of these actions were recorded at 4 environment settings: outdoors, outdoors with camera motion (zoom in and out), outdoors with clothing change, and indoors. We followed the experimental settings from [41]. That is, we selected subjects $11-18$ for training and subjects $2-10$, and 22 for testing (no training performed on this set). Figure 3.3 and Figure 3.4 show sample frames from each of the actions and the learned dictionaries

[4] `http://www.nada.kth.se/cvap/actions/`

for both layers, respectively. Notice, Figure 3.4, how the second level encodes the ℓ_1 energy distributions of each class with respect to the other classes.

Table 3.2 presents the corresponding results. We obtain 97.9%, 94.4% and 96.3% with SM-SVM, SM-1 and SM-2, respectively. Confusion matrices for SM-1 and SM-2 are shown in Figure 3.5. As expected, there is some misclassification error occurring between the *jog, run,* and *walk* actions, all which share most of the spatio-temporal structures. SM-2 performs better, since it combines all the local information with the global information from **S** and **g**, respectively. The three best performing previous methods are [14] (94.2%), [42] (94.5%), and [37] (97.4%). The method described in [14] performs tracking of features using dense sampling. The method in [42] requires bag-of-features using several detectors at several levels, dimensionality reduction with PCA, and also uses neighborhood information, which is much more sophisticated than our method. The closest result to our method is 97.4%, described in [37]. Their method is similar in nature to ours, as it uses features derived from optical flow representing entire videos, further highlighting the need for global information for higher recognition. As mentioned before, there is no cross-class learning in such approach.

Table 3.2: Results for the KTH dataset.

Method	**Overall Accuracy (%)**
Wang *et al.* [14]	94.2
Kovashka *et al.* [42]	94.5
Guo *et al.* [37]	97.4
SM-SVM	**97.9**
SM-1	94.4
SM-2	96.3

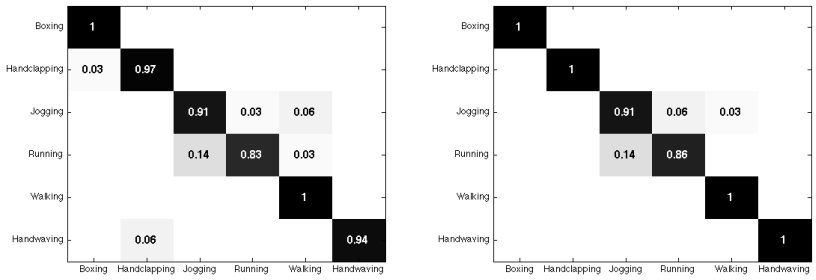


Figure 3.5: Confusion matrices from classification results on the KTH dataset using SM-1 and SM-2. The value on each cell represents the ratio between the number of samples labeled as the column's label the total number of samples corresponding to the row's label.

stand point dig walk carry run wave1 wave2 jump

Figure 3.6: Sample frames from the UT-Tower dataset.

3.5.2 UT-Tower

The UT-Tower dataset[5] [43] simulates an "aerial view" setting, with the goal of recognizing human actions from low-resolution remote sensing (people's height is approximately 20 pixels on average), and is probably from all the tested datasets the most related to standard surveillance applications. There is also camera jitter and background clutter. It consists of 108 videos of 12 subjects, each performing $C = 9$ actions using 2 environment settings. The first environment setting is an outdoors concrete square, with the following recorded actions: *point, stand, dig*, and *walk*. In the second environment setting, also outdoors, the following actions were recorded: *carry, run, wave with one arm (wave1), wave with both arms (wave2),* and *jump*. We converted all the frames to grayscale values. A set of automatically detected bounding box masks centered at each subject are provided with the data, as well as a set of automatically detected tracks for

[5] `http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html`

each subject. We used the set of bounding box masks but not the tracks. All results follow the standard for this dataset Leave One Out Cross Validation (LOOCV) procedure. Figure 3.6 shows sample frames for each action.

Table 3.3 presents the results. We obtained 98.1%, 97.2%, and 100% for SM-SVM, SM-1, and SM-2, respectively. The only confusion in SM-1 occurs between the *point* and *stand* classes and between the *wave1* and *wave2* classes (see Figure 3.7), since there are evident action similarities between these pairs, and the low resolution in the videos provides a low amount of samples for training. The methods proposed in [44] and [12] both obtained 93.9%. In [44], the authors use a Hidden Markov Model (HMM) based technique with bag-of-features from projected histograms of extracted foreground. The method in [12] uses two stages of random forests from features learned based on Hough transforms. The third best result was obtained with the method in [37] as reported in [45]. Again, our method outperforms the other methods with a simpler approach.

Table 3.3: Results for the UT-Tower dataset.

Method	**Overall Accuracy (%)**
Guo *et al.* [37, 45]	97.2
Vezzani *et al.* [44]	93.9
Gall *et al.* [12]	93.9
SM-SVM	98.1
SM-1	97.2
SM-2	**100**

3.5.3 UCF-Sports

The UCF-Sports dataset[6] [46] consists of 150 videos acquired from sports broadcast networks. It has $C = 10$ action classes: *dive, golf swing, kick, weight-lift, horse ride, run, skateboard,*

[6] `http://server.cs.ucf.edu/~vision/data.html\#UCFSportsActionDataset`

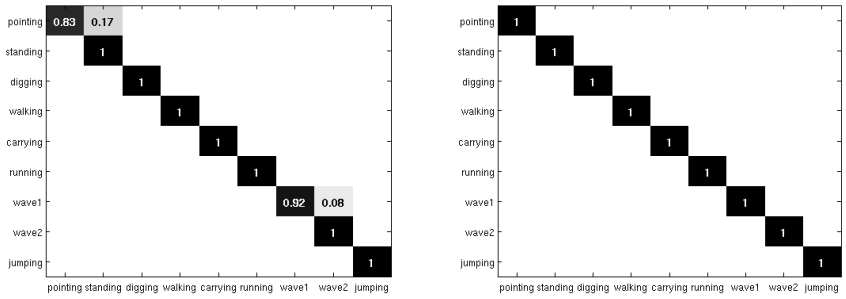


Figure 3.7: Confusion matrices from classification results on the UT-Tower dataset using SM-1 and SM-2.

Figure 3.8: Sample frames from the UCF-Sports dataset.

swing (on a pommel horse and on the floor), swing (on a high bar), and *walk*. This dataset has camera motion and jitter, highly cluttered and dynamic backgrounds, compression artifacts, and variable illumination settings at variable spatial resolution, and 10 fps. We followed the experimental procedure from [22], which uses LOOCV. Also as in [22], we extended the dataset by adding a flipped version of each video with respect to its vertical axis, with the purpose of increasing the amount of training data (while the results of our algorithm are basically the same without such flipping, we here preformed it to be compatible with the experimental settings in the literature). These flipped versions were only used during the training phase. All videos are converted to gray level for processing. We also used the spatial tracks provided with the dataset for the actions of interest.

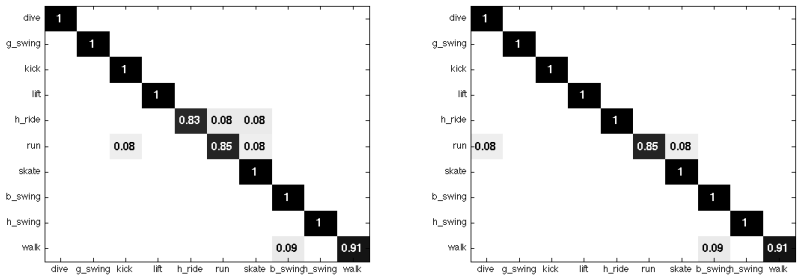


Figure 3.9: Confusion matrices from classification results on the UCF-Sports dataset using SM-1 and SM-2.

Classification results are presented in Table 3.4, and we show the SM-1 and SM-2 confusion matrices in Figure 3.9. We obtained 94.7%, 96.0%, and 97.3% overall classification rates with SM-SVM, SM-1, and SM-2, respectively. In this case, all three SM methods achieve higher classification accuracies than those previously reported in [42], [13], and [14]. We observe misclassification errors in the *run* and *horse ride* classes for the SM-1, and are alleviated by SM-2.

Table 3.4: Results for the UCF-Sports dataset.

Method	**Overall Accuracy (%)**
Le *et al.* [13]	86.5
Wang *et al.* [14]	88.2
Kovashka *et al.* [42]	87.5
SM-SVM	94.7
SM-1	96.0
SM-2	**97.3**

Figure 3.10: Sample frames from the YouTube dataset.

3.5.4 YouTube

The YouTube Dataset[7] [47] consists of $1,168$ sports and home videos from YouTube with $C = 11$ types of actions: *basketball shooting, cycle, dive, golf swing, horse back ride, soccer juggle, swing, tennis swing, trampoline jump, volleyball spike*, and *walk with a dog*. Each of the action sets is subdivided into 25 groups sharing similar environment conditions. Similar to the UCF-Sports dataset, this is a more challenging dataset with camera motion and jitter, highly cluttered and dynamic backgrounds, compression artifacts, and variable illumination settings. The spatial resolution is 320×240 at variable $15 - 30$ fps. We followed the experimental procedure from [47], that is, a group-based LOOCV, where training per action is based on 24 out of 25 of the groups, and the remaining group is used for classification. We also converted all frames to grayscale values. Figure 3.10 shows sample frames from each action.

Table 3.5 shows the overall classification results of our proposed method and comparisons with the state of the art methods, and Figure 3.11 shows the confusion matrices corresponding to SM-1 and SM-2. We obtain overall classification rates of 83.8%, 86.3%, and 89.5% from SM-SVM, SM-1, and SM-2, respectively.

The accuracy attained by SM-SVM is in the same ballpark as the best reported results using

[7] `http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html`

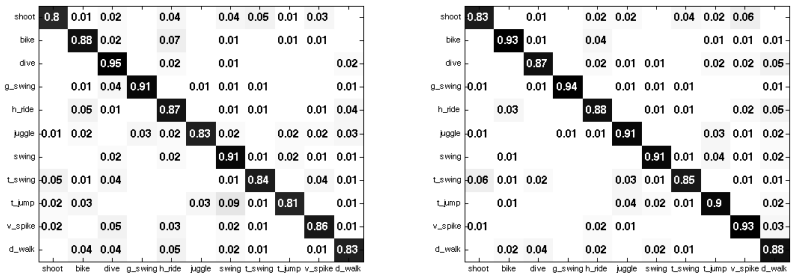


Figure 3.11: Confusion matrices from classification results on the YouTube dataset using SM-1 and SM-2.

dense trajectories, which again incorporates dense sampling at multiple spatio-temporal scales using more sophisticated features, in addition to tracking. Again, the global *and* local nature of SM-2 greatly helps to achieve the highest accuracy, as it decreased the scattered instances of misclassification obtained by SM-1 by implicitly imposing sparsity in a grouping fashion.

Table 3.5: Results for the YouTube dataset.

Method	**Overall Accuracy (%)**
Le *et al.* [13]	75.8
Wang *et al.* [14]	84.2
Ikizler-Cinbis *et al.* [48]	75.2
SM-SVM	83.8
SM-1	86.3
SM-2	**89.5**

3.5.5 Computation Time

The computational efficiency of the model comes from performing simple temporal gradient and thresholding operations in the feature extraction step, and simple decision rules from classification that come directly from the sparse coding of the data. The experiments were conducted on an Intel Core 2 Duo (2.53 GHz) with 4 GB of memory using MATLAB. Using the training videos in [41] from the KTH dataset, it took 4,064 seconds overall, that is, feature extraction, learning, and classification. The most time consuming part was the sparse coding, with 2045 seconds, followed by dictionary learning, with 993 seconds in total. The overall feature extraction procedure took 441 seconds. Testing on a single $120 \times 160 \times 100$ video, it took a total of 17 seconds, where 3.39 seconds correspond to feature extraction, and 11.90 seconds correspond to classification, thus taking approximately 15 seconds of computation overall, or 6.7 frames per second.

3.5.6 Summary

Summarizing these results, we reported an increase in the classification accuracy of 0.5% in KTH, 2.8% in UT-Tower, 9.1% in UCF-Sports, and 5.3% in YouTube. While the prior state-of-the-art results where basically obtained with a variety of algorithms, our proposed framework uniformly outperforms all of them without per-dataset parameter tuning, and often with a significantly simpler modeling and classification technique. These results clearly show that the dimension reduction attained from $\mathbf{A}$ to $\mathbf{S}$ and the local to global mapping do not degrade the discriminative information, but on the contrary, they enhance it.

To further stress the generality of our algorithm, we have not tuned parameters for any of the datasets. Some parameters though could be adapted to the particular data, e.g., the patch size should be adapted to the spatial and temporal resolution of the videos if taken from the same camera.

Following the simplicity of the framework here presented, one might be tempted to go even

simpler. For example, we could consider replacing the learned dictionaries by simpler vector-quantization. We have investigated that and obtained that for example, for the UCF-Sports dataset, the results are significantly worse, attaining a classification accuracy of 18%.

Finally, we have observed that the natural nonnegativity constraint often improves the results, although sometimes the improvement is minor, and as a consequence, we opted to leave it as part of the framework.

3.6 Conclusion

We presented a two-level hierarchical sparse model for the modeling and classification of human actions. We showed how modeling local and global observations using concepts of sparsity and dictionary learning significantly improves classification capabilities. We also showed the generality of the algorithm to tackle problems from multiple diverse publicly available datasets: KTH, UT-Tower, UCF-Sports, and YouTube, with a relatively small set of parameters (uniformly set for all the datasets), a single and simple feature, and a single spatio-temporal scale.

Although simple in nature, the model gives us insight into new ways of extracting highly discriminative information directly from the combination of local and global sparse coding, without the need of explicitly incorporating discriminative terms in the optimization problem and without the need to manually design advanced features. In fact, the results from our experiments demonstrate that the sparse coefficients that emerge from a multi-class structured dictionary are sufficient for such discrimination, and that even with a simple feature extraction/description procedure, the model is able to capture fundamental inter-class distributions.

The model's scalability could become a challenge when the number of classes is very large, since it will significantly increase the size of the dictionary. In such case, it would be useful to integrate newly emerging algorithms for fast sparse approximations such as those proposed by [49] and [50], hence rendering the model more efficient. We are also interested in incorporating locality to the model, which could provide additional insight for analyzing more sophisticated

human interactions. In addition, using a combination of features (e.g., multiscale) as the learning premise would help in dealing with much more complex data acquisition effects such as multi-camera shots and rapid scale variations such as those present in the Hollywood-2 human actions dataset [51].

Chapter 4

Group activity analysis from a single video

4.1 Chapter summary

In this chapter, we present a framework for unsupervised group activity analysis from a single video. Our working hypothesis is that human actions lie on a union of low-dimensional subspaces, and thus can be efficiently modeled as sparse linear combinations of atoms from a learned dictionary representing the action's primitives. Contrary to prior art, and with the primary goal of spatio-temporal action grouping, in this work only one single video segment is available for both unsupervised learning and analysis without any prior training information. After extracting simple features at a single spatio-temporal scale, we learn a dictionary for each individual in the video during each short time lapse. These dictionaries allow us to compare the individuals' actions by producing an affinity matrix which contains sufficient discriminative information about the actions in the scene leading to grouping with simple and efficient tools. With diverse publicly available real videos, we demonstrate the effectiveness of the proposed framework and its robustness to cluttered backgrounds, changes of human appearance, and action variability.

4.2 Introduction

The need for automatic and semi-automatic processing tools for video analysis is constantly increasing. This is mostly due to the acquisition of large volumes of data that need to be analyzed by a much limited human intervention. In recent years, significant research efforts have been dedicated to tackle this problem. In this work, we focus on the analysis of human actions, and in particular the spatio-temporal grouping of activities.

Our understanding of human actions and interactions makes us capable of identifying and characterizing these on relative short time intervals and in an almost effortless fashion. Ideally, we would like to teach a computer system to do exactly this. However, there are challenges that exacerbate the problem, many of which come from the electro-optical system acquiring the data (e.g., noise, jitter, scale variations, illumination changes, and motion blur), but mostly from the inherent complexity and variability of human actions (e.g., shared body movements between actions, periodicity/aperiodicity of body movements, global properties such as velocity, and local properties such as joint dynamics).

With the availability of large amounts of training data, the above challenges are alleviated to some extent. This is at the foundation of many classification methods that rely on the redundancy of these large datasets, and on the generalization properties of modern machine learning techniques, to properly model human actions. In supervised human action classification, a template model for each class is learned from large labeled datasets. Then, unlabeled actions are classified accordingly to the class model that best represents them. In this work, we focus on a very different problem, that is, no labeling is available and all data has to be extracted from a single video.[1] A natural question to ask here is *what can we do when only a single unlabeled video is available?* Given such few data, and no *a priori* information about the nature of the actions, what we are interested in this work is in human action grouping instead of action recognition.

Consider for example a camera observing a group of people waiting and moving in line in an

[1] Even if the video is long, we divide it into short-time intervals to alleviate the action mixing problem. During each short time interval, we have limited data available, and labels are never provided.

airport security checkpoint. We would like to automatically identify the individuals performing anomalous (out of the norm) actions. We do not necessarily know what is the normal action nor the anomalous one, but are interested in knowing when a "different from the group" action is occurring on a given time lapse, and in being able to locate the corresponding individual (in space and time). Situations like this not only occur in surveillance applications, but also in psychological studies (i.e., determining outlier autistic behavior in a children's classroom, or identifying group leaders and followers), and in the sports and entertainment industry (e.g., identifying the offensive and defensive teams, or identifying the lead singer in a concert).

We focus on modeling the general dynamics of individual actions in a *single* scene, with no *a priori* knowledge about the actual identity of these actions nor about the dynamics themselves. We propose an intuitive unsupervised action analysis framework based on sparse modeling for space-time analysis of motion imagery. The underlying idea we propose is that the activity dictionary learned for a given individual is also valid for representing the same activity of other individuals, and not for those performing different ones, nor for him/her-self after changing activity. We make the following main contributions:

- **Unsupervised action analysis:** We extend the modeling of human actions in a relatively unexplored area of interest. That is, we analyze unknown actions from a group of individuals during consecutive short-time intervals, allowing for action-based video summarization from a single video source.

- **Solid performance using a simple feature:** We use a simple feature descriptor based on absolute temporal gradients, which, in our setting, outperforms more sophisticated alternatives.

- **Sparse modeling provides sufficient discriminative information:** We demonstrate that the proposed sparse modeling framework efficiently separates different actions and is robust to visual appearance even when using a single basic feature for characterization and simple classification rules.

- **Works on diverse data:** We provide a simple working framework for studying the dynamics of group activities, automatically detecting common actions, changes of a person's action, and different activities within a group of individuals, and test it on diverse data related to multiple applications.

The remainder of the chapter is structured as follows. In Section 4.3, we provide an overview of recently proposed methods for unsupervised action classification. Then, in Section 4.4, we give a detailed description of the proposed modeling and classification framework. We demonstrate the pertinence of our framework in Section 4.5 with action grouping experiments in diverse (both in duration and content) videos. Finally, we provide concluding remarks and directions for future work in Section 4.6.

4.3 Background and model overview

In this section, we review recent techniques for human action classification which are related to the present work. We focus on feature extraction and modeling, and will cover unsupervised scenarios (see Chapter 3 for a review on supervised schemes).

As seen in the previous chapter, it is clear that the choice of detectors and descriptors and their respective performances highly depend on the testing scenarios, including acquisition properties, dataset physical settings, and the modeling techniques. Most of these features work well in the context for which they were proposed, and changing the context might adversely affect their performance. Let us emphasize that feature design is not our main goal in this chapter. We next describe the feature extraction scheme used throughout this work, which, although very simple, works very well in all our scenario, hence highlighting the advantages of the very simple proposed model.

The proposed feature. In order to properly capture the general spatio-temporal characteristics of actions, it is always desirable to have a large number of training samples. We aim at characterizing actions from scarce data and, under these conditions, we are able to properly

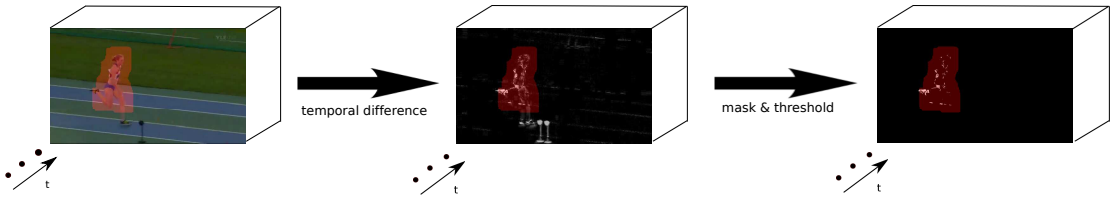


Figure 4.1: Scheme of the interest point extraction method. For extracting the interest points for a given individual, we keep the points (1) whose temporal gradient exceeds a given threshold and (2) that lie inside the individual's (dilated) segmentation mask. The contrast of the temporal gradient is enhanced for improved visualization.

model the actions using a simple feature (the overall scheme is illustrated in Fig. 4.1). We start by tracking and segmenting [52] the individuals whose actions we analyze. This segmentation masks allow us to focus mostly on the individuals while disregarding (most of) the background. We then set a simple interest point detector based on the absolute temporal gradient. For each individual, the points where the absolute temporal gradient is large enough (i.e., it exceeds a pre-defined threshold) become interest points for training and modeling. The feature is also very simple: it consists of a 3D (space and time) absolute temporal gradient patch around each interest point. As we will illustrate in Section 4.5.1, this combination works better than some more sophisticated alternatives in the literature.

4.3.1 Unsupervised models

Apart from the recent literature on human action classification (see Chapter 3), these models have been extended to account for human interactions and group activities. [53] introduced a model to classify nearby human activities by enforcing homogeneity on both the identity and the scene context on a frame by frame basis. [54] proposed to detect and localize individual, structured, and collective human activities (segmented as foreground) by using Kronecker (power) operations on learned activity graphs, and then classify these based on permutation-based graph matching. [55] proposed a model to label group (social) activities using audio and video by

learning latent variable spaces of user defined, class-conditional, and background attributes. [56] proposed to track and estimate collective human activities by modeling label information at several levels of a hierarchy of activities going from individual to collective, and encoding their respective correlations. Our work is similar to these in the sense that we seek to analyze group activities by exploiting the correlations of individual's actions. However, all of the above mentioned schemes require a large amount of labeled training data, which are not available for single video analysis. For this reason, we now turn the attention to unsupervised approaches.

In multi-class supervised classification, labeled training samples from different classes are required, and for anomalous events detection, "normal" training samples are needed. The majority of the publicly available data benchmarks for human action classification usually contain only one person and one type of action per video, and are usually accompained by tracking bounding boxes [57]. This is different from our testing scenario, where only a single video (segment) is available, containing more than one person, without other prior annotations.

Several works addressing unsupervised human action classification have been proposed. [58] used probabilistic Latent Semantic Analysis (pLSA) and Latent Semantic Analysis (LSA) to first learn different classes of actions present in a collection of unlabeled videos through bag-of-words, and then apply the learned model to perform action categorization in new videos. This method is improved by using spatio-temporal correlograms to encode long range temporal information into the local features [59]. [60] proposed a bag-of-words approach using Term Frequency-Inverse Document Frequency features and a data-stream clustering procedure. A spatio-temporal link analysis technique combined with spectral clustering to learn and classify the classes was proposed by [47]. All of these methods employ a bag-of-words approach with sophisticated features and require training data to learn the action representations, while we only work on one single video segment, and a much simpler feature. Our work also departs from correlation-based video segmentation. These methods usually correlate a sample video clip with a long video to find the similar segments in the target video [61], while our work treats all the actions in one video equally and automatically find the groups of the same action. The work presented here shares a similar (but broader) goal as that from Zelnik-Manor and Irani

[62]. Their unsupervised action grouping technique works for a single video containing one individual, comparing the histograms of 3D gradients computed throughout each short-time intervals. We consider a more general setting in which the video length ranges between one second to several minutes, and contains more than one individual, with individuals performing one or more actions and not necessarily the same action all the time.

Bearing these differences in mind, we now proceed to describe the proposed model in detail.

4.4 Unsupervised modeling of human actions

In this work, we assume there are $P \geq 1$ individuals performing simultaneous actions in the video. We first use an algorithm based on graph-cuts [52] to coarsely track and segment the individuals. These tracked segmentations will be used as masks from which features for each individual will be extracted. We later show that these coarse masking procedure is sufficient for reliably grouping actions with our method.

We first extract spatio-temporal patches from the absolute temporal gradient image, around points which exceeds a pre-defined temporal gradient threshold η. These m-dimensional spatio-temporal patches from the j-th person are the data used to train the corresponding dictionary $\mathbf{D}^j, j = 1,2,\cdots,P$. Let us denote by n_j the number of extracted patches from the $j-th$ individual. More formally, we aim at learning a dictionary $\mathbf{D}^j \in \mathbb{R}^{m\times k_j}$ such that a training set of patches $\mathbf{X}^j = [\mathbf{x}_1,\ldots,\mathbf{x}_{n_j}] \in \mathbb{R}^{m\times n_j}$ can be well represented by linearly combining a few of the basis vectors formed by the columns of $\mathbf{D}^j$, that is $\mathbf{X}^j \approx \mathbf{D}^j\mathbf{A}^j$. Each column of the matrix $\mathbf{A}^j \in \mathbb{R}^{k_j\times n_j}$ is the sparse code corresponding to the patch from $\mathbf{X}^j$. In this work we impose an additional nonnegativity constraint on the entries of $\mathbf{D}^j$ and $\mathbf{A}^j$. This problem can then be casted as the optimization

$$\min_{(\mathbf{D}^j,\mathbf{A}^j)\succeq 0} \frac{1}{2}\|\mathbf{X}^j - \mathbf{D}^j\mathbf{A}^j\|_F^2 + \lambda\|\mathbf{A}^j\|_{1,1}, \tag{4.1}$$

where $\succeq$ denotes the element-wise inequality, λ is a positive constant controlling the trade-off between reconstruction error and sparsity (numerous techniques exist for setting this constant

(see for example [3]), $\|\bullet\|_{1,1}$ denotes the ℓ_1 norm of a matrix, that is, the sum of its coefficients, and $\|\bullet\|_F$ denotes the Frobenius norm. Since Equation (4.1) is convex with respect to the variables $\mathbf{A}^j$ when $\mathbf{D}^j$ is fixed and vice versa, it is commonly solved by alternatively fixing one and minimizing over the other.[2]

We will show next how to use these learned dictionaries for comparing simultaneously performed actions on a single time interval (Section 4.4.1), and to detect action changes of the individuals along the temporal direction (Section 4.4.2), with a special case when $P = 2$ in Section 4.4.3. Finally, a spatio-temporal joint grouping for a long video is presented in Section 4.4.4. The algorithm's pipeline is outlined in Fig. 4.2.

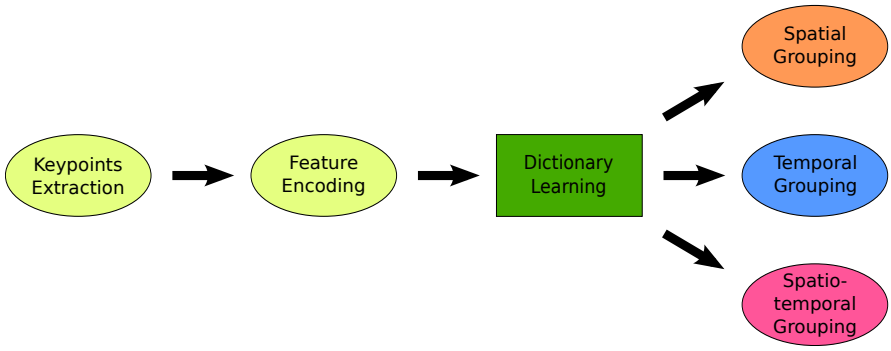


Figure 4.2: Algorithmic pipeline of the proposed method. The first three stages (keypoints extraction, feature extraction and dictionary learning) are common to all presented analysis tools, while specific techniques at the pipeline's end help answer different action-grouping questions. Although we propose tools for solving all the different stages in this pipeline, the core contribution of this work is in the modeling of actions via dictionary learning (the corresponding stage is denoted by a rectangle). This allows to use very simple techniques in the previous stages and much flexibility in the subsequent ones.

[2] Recent developments, e.g., [63, 49, 5, 64], have shown how to perform dictionary learning and sparse coding very fast, rendering the proposed framework very efficient.

4.4.1 Comparing simultaneously performed actions

On a standard supervised classification scenario, subdictionaries $\mathbf{D}^j$ are learned for each human action, and are concatenated together to form a global dictionary. Then, new unlabeled data from human actions are represented by this global dictionary and are classified into the class where the corresponding subdictionary plays the most significant role in the reconstruction (see Chapter 3). In our case, we do not have labeled data for learning and classification. During reconstruction, each person will obviously tend to prefer its own subdictionary from the global dictionary (since the subdictionary is learned from the very same data), consequently inducing poor discrimination power. To handle this difficulty, a "leave-one-out" strategy is thus proposed: each individual action j is represented in a global dictionary that excludes its corresponding subdictionary $\mathbf{D}^j$.

Let us assume that for each person $j \in [1,P]$, we have learned a dictionary $\mathbf{D}^j \in \mathbb{R}^{m\times k_j}$ using the patches $\mathbf{X}^j$. We concatenate the $P-1$ dictionaries $\{\mathbf{D}^i\}_{i=1\ldots P, i\neq j}$ (that is, without including $\mathbf{D}^j$), to build the dictionary $\overline{\mathbf{D}^j} \in \mathbb{R}^{m\times k}$, $k = \sum_{i=1,i\neq j}^{P} k_i$. To test the similarity between the action performed by the j-th person and those performed by the rest of the group, we solve

$$\min_{\overline{\mathbf{A}^j}\succeq 0} \frac{1}{2}\|\mathbf{X}^j - \overline{\mathbf{D}^j}\,\overline{\mathbf{A}^j}\|_F^2 + \lambda\|\overline{\mathbf{A}^j}\|_{1,1}\ . \tag{4.2}$$

The computed sparse-codes matrix $\overline{\mathbf{A}^j}$ is the concatenation of the sparse codes blocks $\{\overline{\mathbf{A}^{j,i}}\}_{i=1\ldots P, i\neq j}$ such that

$$\begin{aligned}\overline{\mathbf{D}^j}\,\overline{\mathbf{A}^j} &= \left[\mathbf{D}^1,\ldots,\mathbf{D}^{j-1},\mathbf{D}^{j+1},\ldots,\mathbf{D}^P\right]\left[\overline{\mathbf{A}^{j,1}}^T\ldots\overline{\mathbf{A}^{j,j-1}}^T,\overline{\mathbf{A}^{j,j+1}}^T\ldots\overline{\mathbf{A}^{j,P}}^T\right]^T\\ &= \mathbf{D}^1\overline{\mathbf{A}^{j,1}} + \cdots + \mathbf{D}^{j-1}\overline{\mathbf{A}^{j,j-1}} + \mathbf{D}^{j+1}\overline{\mathbf{A}^{j,j+1}}\cdots + \mathbf{D}^P\overline{\mathbf{A}^{j,P}}.\end{aligned}$$

We use $\|\overline{\mathbf{A}^{j,i}}\|_{1,1}$ to encode the level of similarity between the action corresponding to the j-th person and the action corresponding to the i-th person, $\forall i \neq j$. Let us motivate this choice with the following example. If two persons, j and i are performing similar actions and person i' is performing a different action, when trying to represent $\mathbf{X}^j$ with the dictionary $\overline{\mathbf{D}^j}$, a larger ℓ_1 energy (activation) is expected from the block $\overline{\mathbf{A}^{j,i}}$ (corresponding to $\mathbf{D}^i$) than from that of $\overline{\mathbf{A}^{j,i'}}$

(corresponding to $\mathbf{D}^{i'}$). We then define the action-similarity matrix $\mathbf{S} \in \mathbb{R}^{P \times P}$, whose entries s_{ij} are defined as

$$s_{ij} = \begin{cases} \min\left(\frac{\|\overline{\mathbf{A}^{i,j}}\|_{1,1}}{\|\overline{\mathbf{A}^{i}}\|_{1,1}} \, , \, \frac{\|\overline{\mathbf{A}^{j,i}}\|_{1,1}}{\|\overline{\mathbf{A}^{j}}\|_{1,1}} \right) & \text{if } i \neq j \, , \\ 1 & \text{otherwise} \, . \end{cases} \tag{4.3}$$

The minimum is used to enforce reciprocal action similarity, and the normalization ensures that comparisons between all individual actions are fair.

We then consider the matrix $\mathbf{S}$ as the affinity matrix of a nonoriented weighted graph G. Although numerous techniques can be used to partition G, in this work we use a simple approach that proved successful in our experiments (recall that the expected number of persons P is small in a group, in contrast to a crowd, so clustering techniques, which rely on statistical properties of the graph, are neither needed nor appropriate). We simply remove the edges of G that correspond to entries s_{ij} such that $s_{ij} < \tau$, for a given threshold τ. For a properly chosen threshold, this edge removal will cause G to split into several connected components, and we consider each one as a group of persons performing the same action. In an ideal scenario where all actions are equal, the similarity scores $\|\overline{\mathbf{A}^{j,i}}\|_{1,1}$ ($i = 1, \ldots, P, i \neq j$) will also be similar. Since in Equation (4.3) we normalize them, setting the threshold to $1/(P-1)$ seems to be a natural choice. However, in practice, the distribution of these coefficients is not strictly uniform. For example, in a video with four skeletons dancing in a synchronous fashion (see Fig. 4.3), the similarity scores in the resulting affinity matrix still show slight variations. We thus set $\tau = \frac{r}{P-1}$, where $r \in [0,1]$ is a relaxation constant (in our experiments we found that $r = 0.9$ was sufficient to cope with this nonuniformity effect).

4.4.2 Temporal analysis: Who changed action?

In the previous section, we presented the modeling and grouping scheme for a fixed time interval (a given video segment). The matrix $\mathbf{S}$ provides sufficient information to determine if there are different actions occurring during an interval, and to determine which individual/s are

performing them.[3] Suppose that on a given interval $t-1$, all P individuals are performing the same action, then, on the next time interval t, the first $P-1$ individuals change action while the P-th individual remains doing the same. From the affinity matrix $\mathbf{S}$ there is no way of determining if the first $P-1$ persons changed while the P-th person remained doing the same or vice-versa. An additional step is thus necessary in order to follow the individuals' action evolution in the group. A possible solution for this problem at a small additional computational cost is as follows.

Let the minimized energy

$$\mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_t^j) = \min_{\mathbf{A}^j \succeq 0} \frac{1}{2}\|\mathbf{X}_t^j - \mathbf{D}_t^j \mathbf{A}^j\|_F^2 + \lambda \|\mathbf{A}^j\|_{1,1} \tag{4.4}$$

be the j-th individual's $\ell_{2,1}$ representation error with his/her own dictionary at time t. Then, we measure the evolution of the reconstruction error per individual as

$$E_{t-1,t}^j = |(\mathscr{R}^*(\mathbf{X}_{t-1}^j, \mathbf{D}_t^j) + \mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_{t-1}^j) - \mathscr{R}^*(\mathbf{X}_{t-1}^j, \mathbf{D}_{t-1}^j) - \mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_t^j))|, \tag{4.5}$$

and

$$\mathbf{E}_{t-1,t} = \frac{1}{C}[E_{t-1,t}^1, ..., E_{t-1,t}^P], \tag{4.6}$$

where $C = \sum_{j=1}^P E_{t-1,t}^j$ is a normalization constant. $\mathbf{E}_{t-1,t}$ captures the action changes in a per person manner, a value of $E_{t-1,t}^j/C$ close to 1 implies that the representation for the j-th individual has changed drastically, while 0 implies the individual's action remained exactly the same. In the scenario where nobody changes action, all $E_{t-1,t}^j, \forall j \in [1,P]$ will be similar. We can apply a similar threshold $\mu = r/P$ to detect the actions' time changes (note that we now have P persons instead of $P-1$).

4.4.3 Special case: $P = 2$

If there are only two individuals in the video ($P = 2$), the grouping strategies from sections 4.4.1 and 4.4.2 become ambiguous. The similarity matrix $\mathbf{S}$ would have all entries equal to 1 (always

[3] The same framework can be applied if we have a single individual and just want to know if all the activities he/she is performing in $P > 2$ time intervals are the same or not, each time interval taking the place of an "individual."

one group), and there would be no clear interpretation of the values in $\mathbf{E}_{t-1,t}$. Therefore, for this particular case where $P = 2$, we define the one time interval measure (at time t) as

$$E_t^{i,j} = \max\left(\frac{|\mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_t^j) - \mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_t^i)|}{|\mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_t^i)|}, \frac{|\mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_t^i) - \mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_t^j)|}{|\mathscr{R}^*(\mathbf{X}_t^j, \mathbf{D}_t^j)|}\right), \tag{4.7}$$

where i and j denote each subject. Also, we let the evolution of the reconstruction error from $t-1$ to t of each individual $i \in [1,2]$ be

$$E_{t-1,t}^i = \max\left(\frac{|\mathscr{R}^*(\mathbf{X}_{t-1}^i, \mathbf{D}_t^i) - \mathscr{R}^*(\mathbf{X}_{t-1}^i, \mathbf{D}_{t-1}^i)|}{|\mathscr{R}^*(\mathbf{X}_{t-1}^i, \mathbf{D}_{t-1}^i)|}, \frac{|\mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_{t-1}^i) - \mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_t^i)|}{|\mathscr{R}^*(\mathbf{X}_t^i, \mathbf{D}_t^i)|}\right). \tag{4.8}$$

Finally, we use the threshold $\mu = r/2$, where a value greater than μ implies that the subject is performing a different action. Note that these formulations do not replace the algorithms in sections 4.4.1 and 4.4.2, which are more general and robust to treat the cases where $P > 2$.

4.4.4 Joint spatio-temporal grouping

The above discussion holds for a short-time video containing a few individuals. Clustering techniques, which rely on statistical properties of the data, are neither needed nor appropriate to handle only a few data points. Simpler techniques were thus needed.

Now, for long video sequences, action analysis with the previously described simple tools becomes troublesome. Luckily, clustering methods are ideal for this scenario. Hence, we use them to do joint spatio-temporal action grouping (here, by spatial we mean across different individuals). We consider each individual in a given time interval as a separate entity (an individual in two different time intervals is thus considered as two individuals). Dictionaries are learned for each spatio-temporal individual and an affinity matrix is built by comparing them in a pairwise manner using equations (4.7) or (4.8) (notice that in this setting, the two equations become equivalent). We simply apply to this affinity matrix a non-parametric spectral clustering method that automatically decides the number of groups [65].

4.5 Experimental results

In all the reported experiments, we used $n = \min(\min_j(n_j), 15,000)$ overlapping temporal gradient patches of size $m = 15 \times 15 \times 7$ for learning a dictionary per individual. The tracked segmentation mask for each individual [52] is dilated to ensure that the individual is better covered (sometimes the segmentation is not accurate for the limbs under fast motions). Only the features belonging to the tracked individuals are used for action modeling and dictionary learning. The dictionary size was fixed to $k_j = 32$ for all j, which is very small compared to the patch dimension (undercomplete). The duration of the tested video segments (short-time intervals) is one second for action grouping per time interval, and two seconds for temporal analysis. Longer videos (from 5 seconds to 320 seconds) were used for joint spatio-temporal grouping. All the tested videos are publicly available and summarized in Table 4.1.[4]

We provide a rough running-time estimate, in order to show the efficiency of the proposed framework. Our non-optimized Matlab code for feature extraction takes less than 10 seconds to process 30 frames of a standard VGA video (640×480 pixels), containing five individuals, and about 3 seconds for 30 frames of a video with lower-resolution (480×320 pixels) containing three individuals. As for the sparse modeling, we used the SPAMS toolbox,[5] taking approximately 1 second to perform dictionary learning and sparse coding of 15,000 samples. Notice that code optimization and parallelization would significantly boost the performance, potentially obtaining a real-time process.

4.5.1 Action grouping per time interval

We now test the classification performance of the framework described in Section 4.4.1 for grouping actions per time interval. The cartoon-skeletons dancing in the Skeleton video, as shown in Fig. 4.3, were segmented and tracked manually to illustrate that individual actions have intrinsic variability, even when the actions performed are the same, justifying the relaxation

[4] We only present a few frames for each video in the figures, please see the supplementary material and mentioned links for the complete videos.

[5] `http://spams-devel.gforge.inria.fr/`

Table 4.1: Description summary of the videos used in the experiments.

Video	fps	Figures	Description
Skeleton[a]	30	4.3	Four skeletons dancing in a similar manner.
Long jump[b]	25	4.4, 4.14	Three athletes in a long jump competition.
Gym[c]	30	4.5	Three persons in a gym class.
Kids[d]	30	4.6	Three kids dancing in a TV show.
Crossing[e,f]	30	4.8	Two pedestrians crossing the street and the other one waiting.
Jogging[g,f]	30	4.9	Six persons jogging in a park.
Dancing[h,f]	30	4.10	Five persons rehearsing a dance act.
Singing-dancing[i]	30	4.7	A singer and four dancers performing in a theater.
Tango[j]	30	4.11	Three couples dancing Tango.
Mimicking[k]	30	4.13	Gene Kelly in a self-mimicking dancing act.
Fitness[l]	30	4.12, 4.15	Three persons in a fitness class.
Outdoor[m]	30	4.16	Action classification video used by [62]

[a] http://www.youtube.com/watch?v=h03QBNVwX8Q

[b] http://www.youtube.com/watch?v=bia-x_linh4.

[c] http://www.openu.ac.il/home/hassner/data/ASLAN/ASLAN.html

[d] http://www.youtube.com/watch?v=N0wPQpB4eMk

[e] http://www.eecs.umich.edu/vision/activity-dataset.html

[f] Bounding boxes surrounding each person are provided, and they were used instead of the tracking/segmentation approach.

[g] http://www.eecs.umich.edu/vision/activity-dataset.html

[h] http://www.eecs.umich.edu/vision/activity-dataset.html

[i] http://www.youtube.com/watch?v=R9msiIqkI34

[j] http://www.youtube.com/watch?v=IkFjg7m-jzs

[k] http://www.youtube.com/watch?v=_DC6heLMqJs

[l] http://www.youtube.com/watch?v=BrgPzpOGBcw

[m] http://www.wisdom.weizmann.ac.il/mathusers/vision/VideoAnalysis/Demos/EventDetection/OutdoorClusterFull.mpg

coefficient r. Notice that this effect is not a by-product of the tracking/segmentation procedure, since it is done manually in this example.

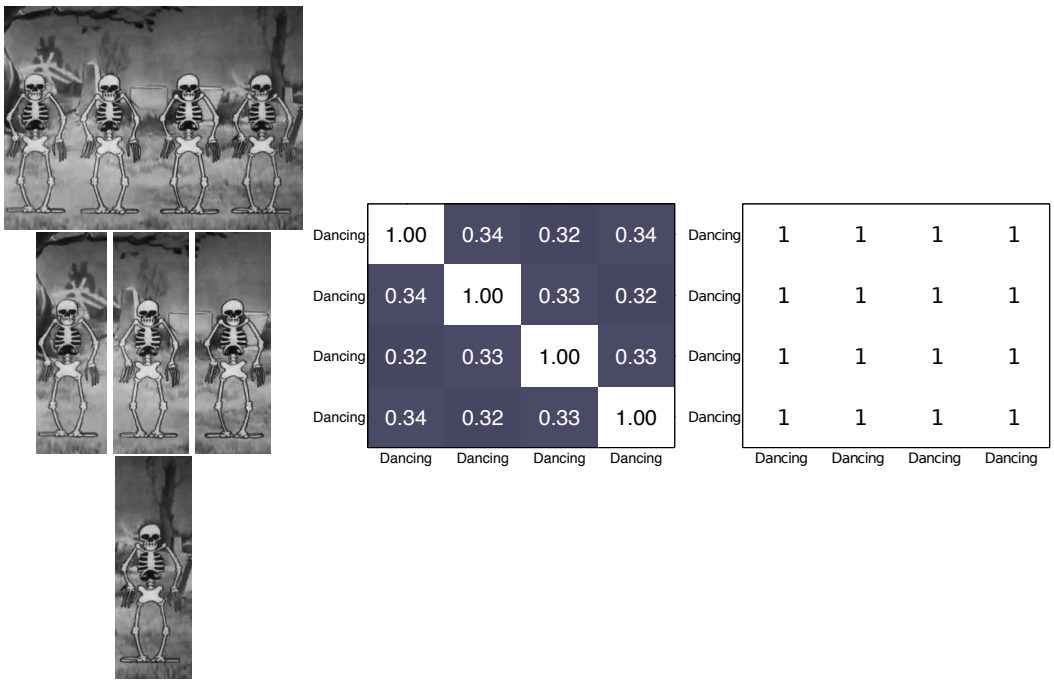


Figure 4.3: Sample frames from the Skeleton video. **Left**: Four skeletons dancing in the same manner were manually segmented and tracked during a single one second interval. **Center**: Affinity matrix before binarization. The slight differences in the entries are due to the intrinsic variability of the action itself. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/3 = 0.3$. The values in the entries show slight variation but all larger than 0.3, so they are all binarized to 1, and thus the four skeletons are grouped together.

We then analyzed the Long jump video, shown in Fig. 4.4, where three persons are performing two actions: running and long-jumping. We tested every possible configuration of three video individuals (with corresponding time segments cropped from the original video), showing that we are doing action classification and not person classification. These results are summarized in Table 4.2, where we obtained one single grouping error (see the third configuration).

Table 4.2: Three people running (R) and then long-jumping (LJ). The grouping decision is represented with letters A and B, with only one grouping error on the third configuration (cells are colored to facilitate the comparison between the results and the ground truth, matching colors means correct result). See Fig. 4.4 for sample frames from this video.

	Persons	**Action grouping**							
Ground Truth	I	R	LJ	LJ	R	R	R	LJ	LJ
	II	R	LJ	R	LJ	R	LJ	R	LJ
	III	R	LJ	R	R	LJ	LJ	LJ	R
Result	I	A	A	A	A	A	A	A	A
	II	A	A	A	B	A	B	B	A
	III	A	A	A	A	B	B	A	B

The test for the Gym video, shown in Fig. 4.5, consists of three persons performing two actions: punching and dancing. The results are shown in Table 4.3. In this test, we again obtained only one grouping error.

To further validate that we are doing action classification (not just classifying the person itself), and that we can treat more imbalanced cases (most individuals performing an action), we also conducted a 'cloning' test, using the Gym video (Fig. 4.5). In this scenario, we added a fourth person by artificially replicating one of the original three, but performing (in a different video segment) an action different than the one of the original time interval. The results are shown in Table 4.4. A correct grouping result was attained, confirming that the proposed method only perceives actions, and is robust to differences in human appearance.

On a more imbalanced scenario, we analyzed the Singing-dancing video, where five individuals are dancing while the remaining one is singing, see Fig. 4.7. From the affinity matrix, we observe that the row and column corresponding to the second (singing) person have smaller values, binarized to zero after applying the thresholding operation. The threshold in this case is

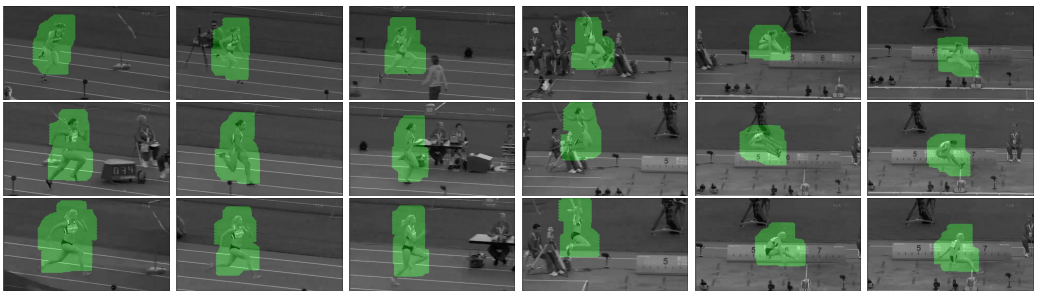


Figure 4.4: Sample frames from the Long jump video. On each row, an athlete is running and then long-jumping. Colored segmentation masks are displayed. (This is a color figure.)

$\tau = 0.9/4 = 0.225$, slightly larger than the values 0.21 in entries $(1,4)$ and $(4,1)$, which should be binarized to 1. Since the persons are grouped as connected components, we still obtain the correct groups. More imbalanced tests for the Jogging, Dancing and Crossing videos (shown in figures 4.8, 4.9 and 4.10) also give correct grouping results.

Two additional tests were performed using the Tango video, where there are three couples dancing Tango (Fig 4.11). Instead of treating each individual separately, we considered each couple as a single entity and applied the proposed method. The returned affinity matrix shows that the proposed method correctly groups the three couples as one group (if they are performing the same activity) or two groups (if they are performing different activities).

Let us now turn our attention to the employed features to point out the effectiveness of our simple approach. We conducted experiments with 24 configurations of the video segments from the Long jump, the Gym, and the Kids videos, see figs. 4.4, 4.5, and 4.6 (8 configurations from each of them, similar to the configurations in tables 4.2 and 4.3). We compared our simple feature (temporal gradient detector) against several feature detectors (the cuboid [9], and Harris 2D [66] (also see the local motion patterns (LMP) by [20]) and descriptors (our 3D temporal gradient patches, HOG3D [17], and the cuboid [9]). The Harris 2D detector detects spatially distinctive corners in each image frame, while the cuboid detector relies on applying separable linear filters, which produce high responses at local image intensities containing periodic

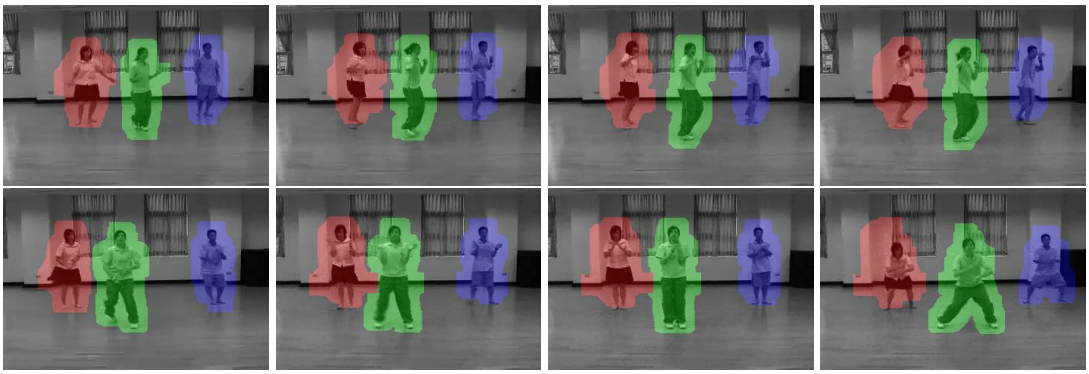


Figure 4.5: Sample frames from the Gym video. On the first row, the three individuals are punching; on the second row, they are dancing. The segmentation masks for the different individuals appear in different colors. (This is a color figure.)

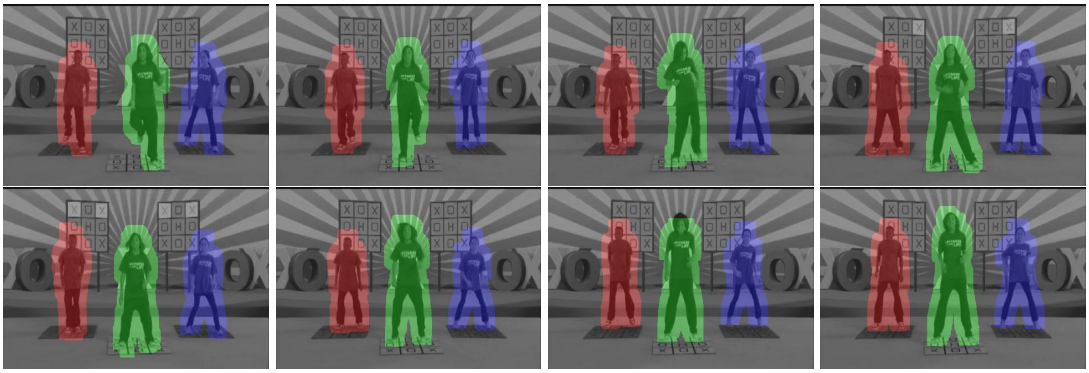


Figure 4.6: Sample frames from the Kids video. Fist row: dancing. Second row: jumping. The segmentation masks for the different individuals appear in different colors. (This is a color figure.)

frequency components. It also responds to any region with spatially distinct characteristics undergoing a complex motion [9]. It is important to mention that both detectors produce fewer feature points than the temporal gradient detector. As for the descriptors, they all produce vectors with comparable dimensionalities: $m = 1,575$ for the temporal gradient patch, $m = 1,440$ for the cuboid descriptor, and $m = 1,000$ for HOG3D. The results are presented in Table 4.5,

Table 4.3: Analysis of the Gym video. Three persons are punching (P) or dancing (D) (cells are colored to facilitate the comparison). The grouping decision is shown with values A or B, with only one grouping error. See Fig. 4.5 for some typical frames.

	Persons	**Action Grouping**							
Ground Truth	I	P	D	P	D	D	D	P	P
	II	P	D	D	P	D	P	D	P
	III	P	D	D	D	P	P	P	D
Result	I	A	A	A	A	A	A	A	A
	II	A	A	B	B	A	A	B	A
	III	A	A	B	A	B	A	A	B

where the best grouping performance is obtained with the proposed detector/descriptor based on the temporal gradient. According to the evaluation by [22], HOG3D performs well in combination with dense sampling, which can capture some context information. But this is not appropriate in our unsupervised grouping framework for single videos. The cuboid detector gives good results in combination with the temporal gradient descriptor and HOG3D, but the cuboid descriptor seems to under perform. The Harris 2D detector only produces 2D feature points, which do not necessarily undergo significant motion over time. Even though we employ temporal gradient here, we are not claiming that the other features are intrinsically or generally bad, since they work extremely well on supervised scenarios. Nevertheless, for the problem at hand, where data is scarce, our simple feature performs better.

4.5.2 Temporal analysis experiments

To test the proposed strategy for dealing with temporal action changes, we processed several video configurations with two consecutive time intervals. The main goal is to identify the individuals who changed actions.

Table 4.4: Analysis of the Gym video. Three persons are punching (P) or dancing (D), and a 'clone' is added. The second column denotes the person's index. The fourth person, that is the clone, is the same as one of the first three, but is doing something different. For example, I-D means that the person I was cloned. The (perfect) grouping decision is shown inside the table (cells are colored to facilitate the comparison). See Fig. 4.5 for some typical frames.

Persons		**Action Grouping**						
Ground Truth	I	P	P	P	P	D	D	D
	II	P	P	P	P	D	D	D
	III	D	P	P	P	D	D	D
	'Clone'	II-D	I-D	II-D	III-D	I-P	II-P	III-P
Result	I	A	A	A	A	A	A	A
	II	A	A	A	A	A	A	A
	III	B	A	A	A	A	A	A
	'Clone'	B	B	B	B	B	B	B

Table 4.6 summarizes the results by applying the method described in Section 4.4.2 to the Long jump video (Fig. 4.4). Correct results were obtained when analyzing a video subset of the involved persons changing their actions (experiments 3 and 4). In the case where all the persons change action simultaneously or keep doing the same action (experiments 1 and 2), we observe incorrect results. The proposed framework for representing actions is not the source of this issue. It is a consequence of the normalization in Equation (4.6), which compares individuals who are either all changing their action or all continuing their previous action and hence provides no discriminative power in this case.

Although it is not easy to extract a general rule for every possible scenario, the vector $[E^i_{t-1,t}, E^i_{t,t+1}, E^i_{t+1,t+2}, ...]$ (see Equation (4.6, p. 47)) provides useful information about how an individual's actions evolves over time. An example is shown in Fig. 4.12, where we build this

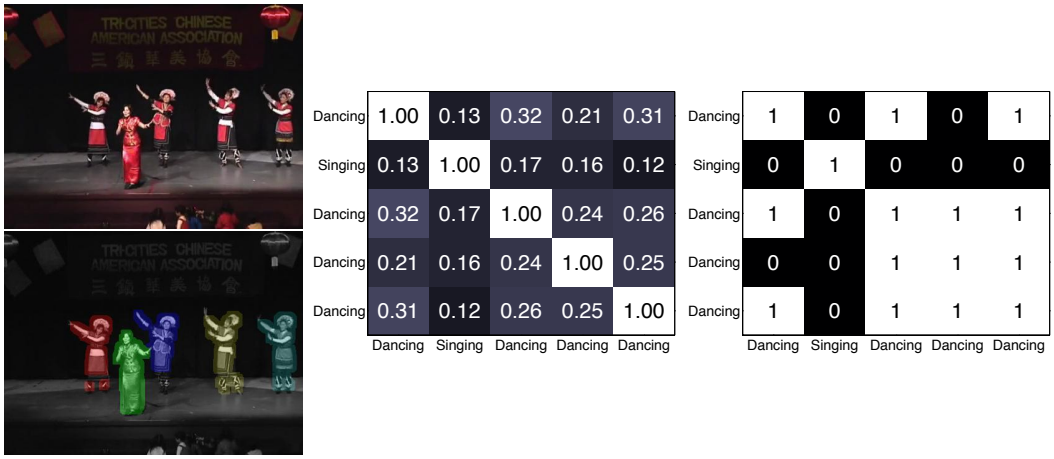


Figure 4.7: Sample frames from the Singing-dancing video. **Left**: Five persons, a singer and four dancers, were tracked/segmented during a one second interval (the masks are displayed in colors on the bottom left, the singer appearing in green). **Center**: Affinity matrix. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/4 = 0.225$. Note that the values in the entries of the second row and the second column (except the diagonal entries) are small, hence binarized to zero. This implies that the second person is doing a different action than the group. The binarization on entries (1,4) and (4,1) fails to be 1, not affecting the grouping, since the persons are grouped as connected components. Two groups are correctly detected, the four dancers and the singer. (This is a color figure.)

vector for one individual in the Gym video (see Fig.4.15), using seven consecutive one-second time intervals (from t to $t+6$). During this time lapse, the individual changes his action one time. More complex rules can be derived from this readily available information. The action-change rule provided in Section 4.4.2 will nonetheless be already useful in many cases.

Finally, we present an example using the Mimicking video for the special case (P=2) described in Section 4.4.3. It consists of two seconds interval ($t-1$ and t) from a comedy show, where two dancers (i and j) are mimicking each other (see Fig. 4.13). Using equations (4.7) and (4.8), and a threshold $\mu = 1/2$, we obtain $E^{i,j}_{t-1} = 2.29$, $E^{i,j}_{t} = 1.07$, $E^{i}_{t-1,t} = 1.62$, and $E^{j}_{t-1,t} = 1.64$. These results correctly imply that the dancers were performing different actions

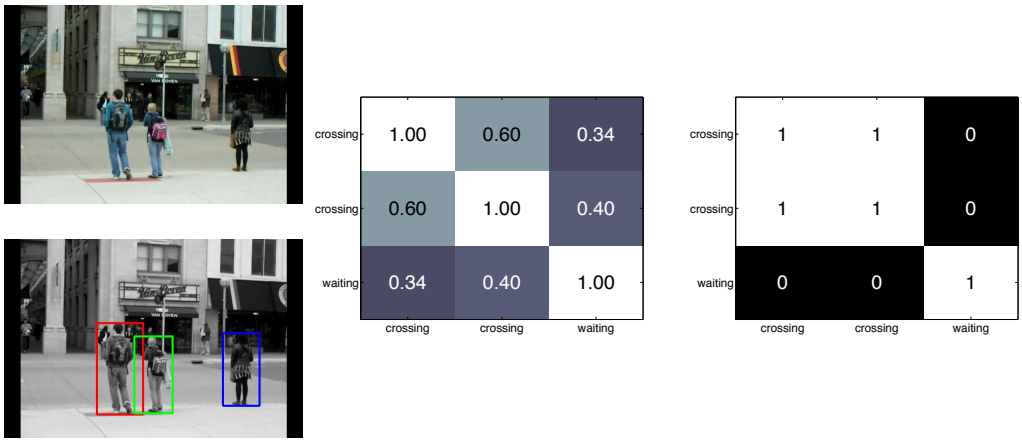


Figure 4.8: **Left**: A sample frame of the Crossing video. Two pedestrians at left are crossing the street while the one at right is waiting. The provided bounding boxes were used instead of running the tracking/segmentation algorithm. **Center**: Affinity matrix. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/2 = 0.45$. Two groups are correctly detected. (This is a color figure.)

on each of the two seconds, and that both dancers went through an action change from $t-1$ to t.

4.5.3 Joint spatio-temporal grouping

We further analyzed three videos, i.e., the Long jump, Fitness, and Outdoor videos, in which human actions are jointly grouped in time and space (recall that by space we mean across individuals), applying the nonparametric spectral clustering algorithm by [65] to the pairwise affinity matrix described in Section 4.4.4.

The first experiment, using 5 seconds from the Long jump video is shown in Fig. 4.14. The 3 individuals are first running then long-jumping. We thus consider that there are $15 = 5 \cdot 3$ individuals in the video. The clustering algorithm on this 15×15 affinity matrix gives 2 correct clusters, even though the three athletes have different appearance.

The second experiment was conducted on a 40 seconds sequence from the Fitness video,

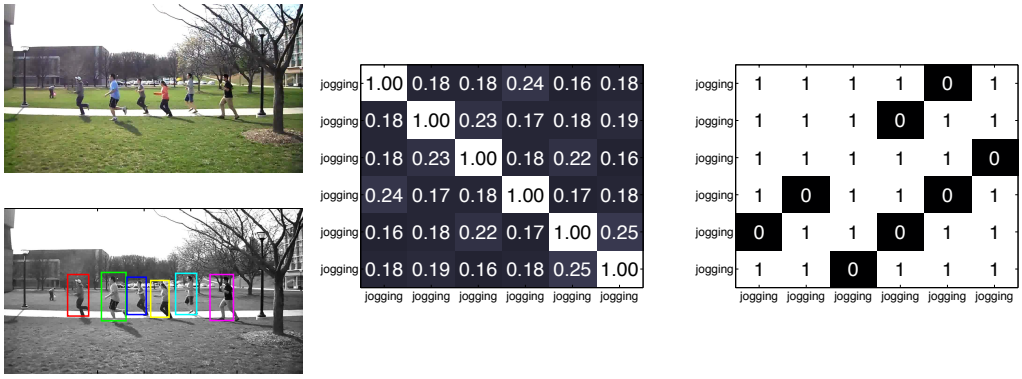


Figure 4.9: **Left**: A sample frame of the Jogging video. The provided bounding boxes were used instead of running the tracking/segmentation algorithm. **Center**: Affinity matrix. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/5 = 0.18$. Note that some entries are a little smaller than the threshold 0.18, thus binarized to be 0. But the grouping result is still correct since persons are grouped as connected components. One single group is correctly detected. (This is a color figure.)

in which three individuals are doing gym exercises (Fig. 4.15). Notice that, even though their actions are synchronized, we do not provide this information *a priori* to the algorithm. The clustering algorithm returned 5 clusters from the 120×120 affinity matrix, and 4 should have been found. There is an over splitting for the individual in the middle from frame 300 to frame 690, meaning that in this case either auto-similarity was captured in excess or the action of this person is actually different (such granularity in the action can actually be observed by carefully watching the video). There are also some clustering incorrect results in the transition period between two actions due to temporal mixing effects (our intervals are arbitrarily fixed and may incorrectly divide the actions during the switch). Note also that the ground truth was manually built from visual observation and is also fixed to having hard action transitions. Considering overlapping or shorter segments will alleviate this issue if additional temporal accuracy is needed.

In the third experiment we processed the Outdoor video (Fig. 4.16). The video contains only

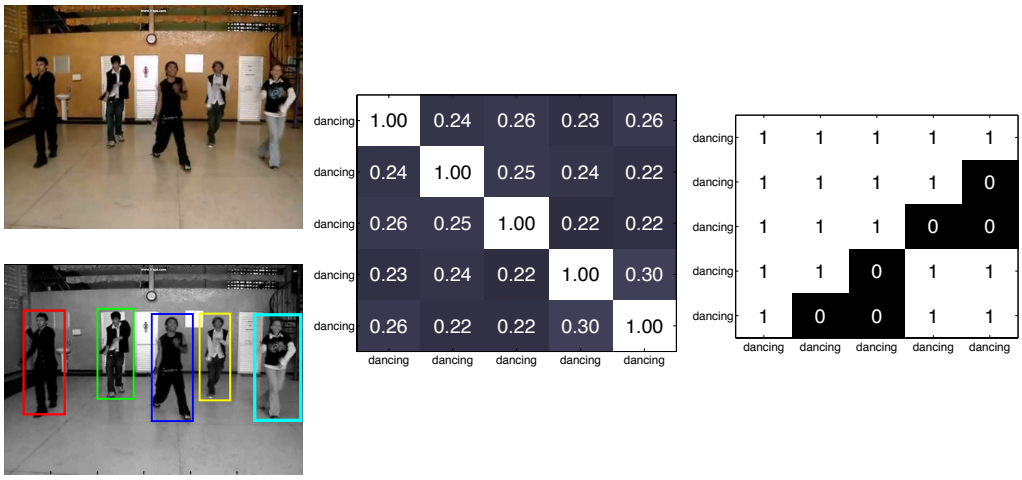


Figure 4.10: **Left**: A sample frame of the Dancing video. The provided bounding boxes were used instead of running the tracking/segmentation algorithm. **Center**: Affinity matrix. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/4 = 0.225$. Note that some entries are a little smaller than the threshold 0.225, thus binarized to be 0. But the grouping result is still correct since persons are grouped as connected components. One single group is correctly detected. (This is a color figure.)

one individual per frame, which changes appearance (different individuals appear over time with different clothing). This video exhibits very slow motions and in order to capture enough action information in the temporal direction, we first subsampled the video by a factor of 2 in the temporal direction before applying the proposed method. The clustering is consistent with visual observation. We observe some incorrect labels, again due to the fixed transition period between two actions, that is, one time interval can contain two actions and the action type is not well defined in this situation for the corresponding time segment.

4.6 Conclusion

We presented an unsupervised sparse modeling framework for action-based scene analysis from a single video. We model each of the individual actions independently, via sparse modeling

Table 4.5: The grouping classification accuracy on 24 example configurations from the Long jump, the Gym, and the Kids videos by using several detectors/descriptor combinations.

Detector	Descriptor		
	Temporal gradient	HOG3D[a]	Cuboid[b]
Temporal gradient	**87.5%**	75.0%	—[d]
Cuboid[b]	79.1%	79.1%	66.7%
Harris 2D[c]	75.0%	75.0%	—[d]

[a] [17], implementation available at `http://lear.inrialpes.fr/~klaeser/software_3d_video_descriptor`.

[b] [9], implementation available at `http://vision.ucsd.edu/~pdollar/files/code/cuboids/`.

[c] Harris points are detected in each frame. Patches around these keypoints are used to construct the spatio-temporal descriptors. This is similar to the local motion patterns (LMP) proposed in [20].

[d] A separate implementation of the cuboid descriptor is not available.

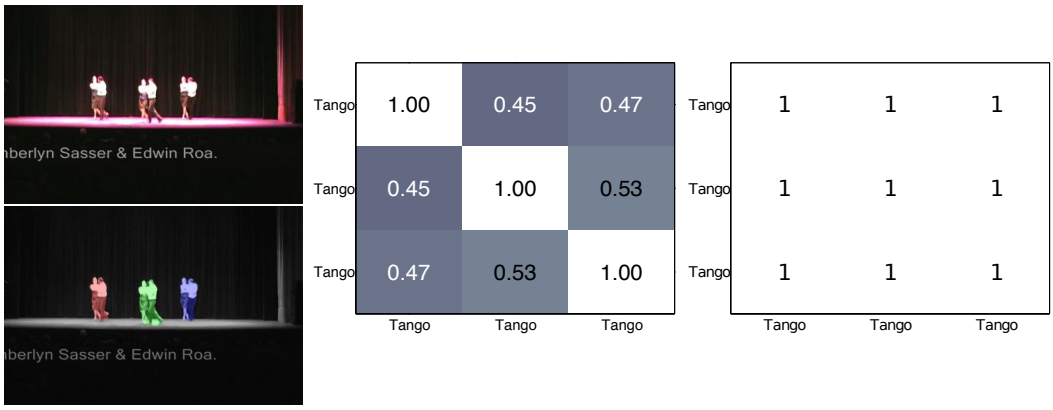
techniques, and build a group affinity matrix. Applying relatively simple rules based on representation changes, the proposed method can efficiently and accurately tell whether there are different actions occurring on the same short-time interval, and across different intervals, including detecting possible action changes by any of group members. In addition, we extended the method to handle longer motion imagery sequences by applying standard spectral clustering techniques to a larger spatio-temporal affinity matrix. We tested the performance of the framework with diverse publicly available datasets, demonstrating its potential effectiveness for diverse applications.

We also showed that by using a single and simple feature in such a scarce data scenario outperforms standard and more sophisticated ones. This indicates that further research on good features for unsupervised action classification is much needed.

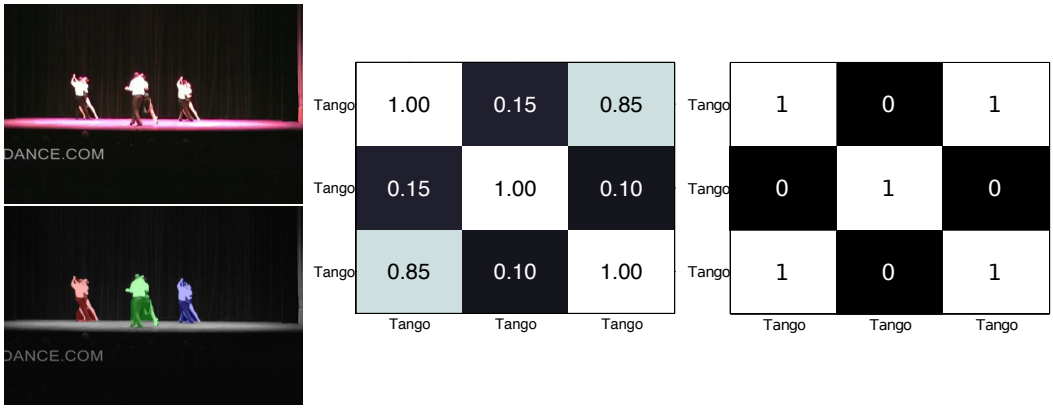
Table 4.6: Temporal analysis of the Long jump video. Three persons in a race track on consecutive time intervals. 'R' and 'LJ' denote running and long-jumping, respectively. A value above $\mu = 0.3$ in the action evolution vector $\mathbf{E}_{t-1,t}$ means that person's action has changed.

Person	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
	Interval		$\mathbf{E}_{t-1,t}$	Interval		$\mathbf{E}_{t-1,t}$	Interval		$\mathbf{E}_{t-1,t}$	Interval		$\mathbf{E}_{t-1,t}$
	$t-1$	t		$t-1$	t		$t-1$	t		$t-1$	t	
A	R	R	**0.378**	R	LJ	**0.309**	R	R	0.133	R	R	0.202
B	R	R	**0.336**	R	LJ	**0.353**	R	LJ	**0.437**	R	R	0.142
C	R	R	0.286	R	LJ	**0.338**	R	LJ	**0.430**	R	LJ	**0.656**

We are currently working on extending the model to handle interactions, that is, meta-actions performed by several persons. Also, going from purely local features to semi-local ones by modeling their interdependences might provide a way to capture more complex action dynamics. Finally, action similarity is an intrinsically multiscale issue (see example in Figure 14, where the middle lady is performing the same coarse action but in a different fashion), therefore calling for the incorporation of such concepts in action clustering and detection.



(a) A single group is correctly identified.



(b) Two different activities are correctly identified (the couples on the left and right are grouped together, while the couple in the middle is isolated)

Figure 4.11: **Left**: Sample frames from the Tango video, where three couples are dancing Tango during a one second interval were tracked/segmented (masks are displayed in different colors on the bottom left). Each couple was treated as a single entity. **Center**: affinity matrix. **Right**: Binarized affinity matrix after applying the threshold $\tau = 0.9/2 = 0.45$. (This is a color figure.)

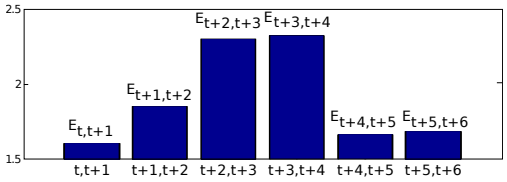


Figure 4.12: The vector $[E^i_{t,t+1}, E^i_{t+1,t+2}, ...E^i_{t+5,t+6}]$ (see Equation (4.6, p. 47)) over seven consecutive time intervals for one individual in the Gym video (see Fig.4.15), during which he/she performs two different actions. We can see that the values of $E^i_{t,t+1}$, $E^i_{t+1,t+2}$, $E^i_{t+4,t+5}$, and $E^i_{t+5,t+6}$ are small, reflecting no change of action in that interval. The other values ($E^i_{t+2,t+4}$ and $E^i_{t+3,t+4}$) are relatively big because the individual is changing actions. The transition between actions is not instantaneous, lasting for about two seconds.

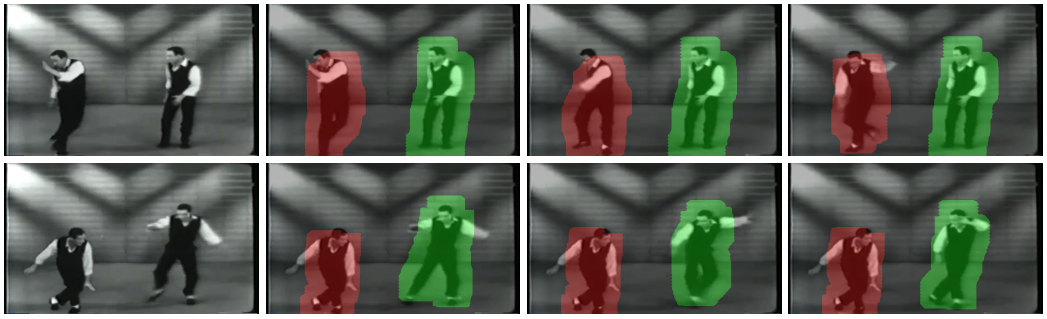


Figure 4.13: On the first column, two frames from the Mimicking video. On the remaining columns, the tracking/segmentation masks are displayed in colors. The two dancers are correctly detected as performing different actions. (This is a color figure.)

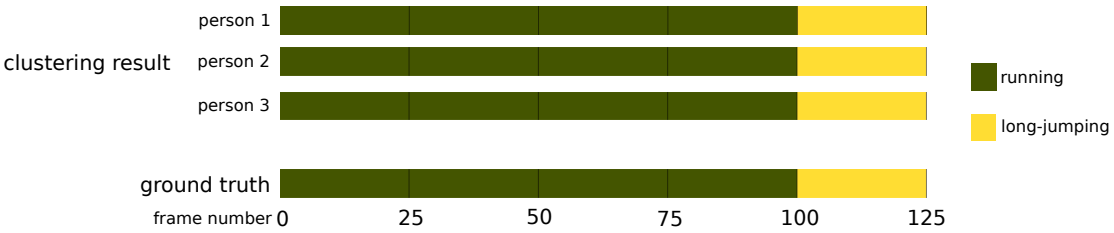


Figure 4.14: 5 seconds from the Long jump video, where three athletes are running and long-jumping, see sample frames in Fig. 4.4. Our method correctly identifies two actions.

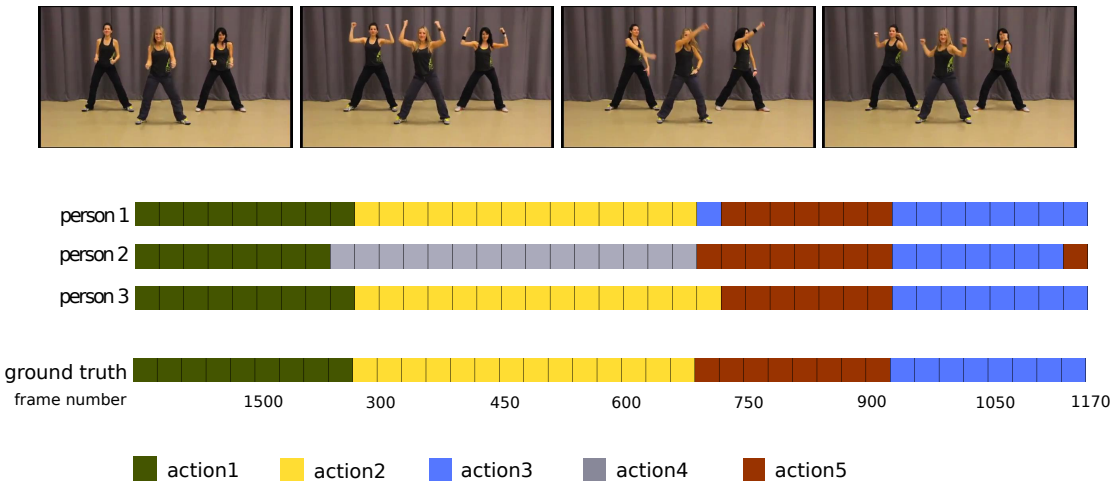


Figure 4.15: 40 seconds analysis of the Fitness video, where five clusters are detected instead of four. Between frames 300 and 690, all three persons are doing the same corase action and this over-splitting can be explained by granular action variability, where the person in the middle presents auto-similarity (she is somewhat more energetic than the others).

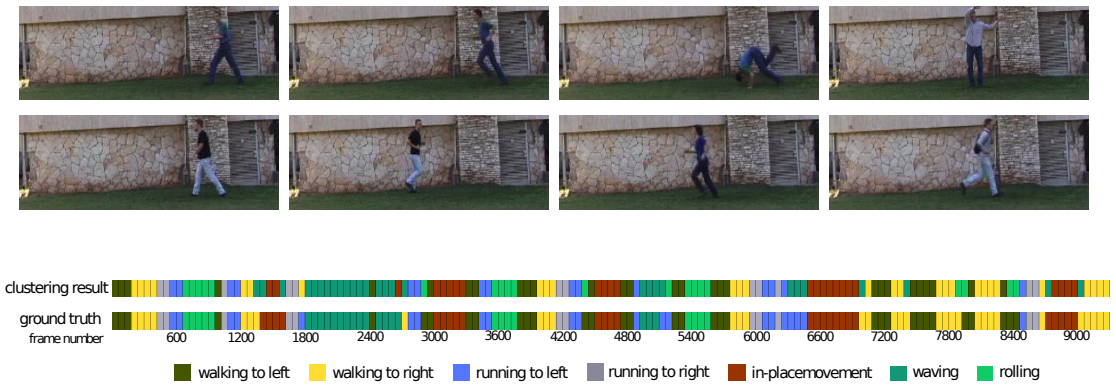


Figure 4.16: 320 seconds from the Outdoor video. In accordance with visual observation, seven clusters are identified. There are a few clustering errors in the transition periods between the actions due to the discrete temporal nature of the particular analysis here exemplified.

Chapter 5

Subpixel spectral mapping of remotely sensed imagery

5.1 Chapter summary

In this chapter, we present a method for modeling, mapping, and classification in hyperspectral imagery using learned block-structured discriminative dictionaries. The spectral pixels are modeled by linear combinations of subspaces defined by the learned dictionary atoms, allowing for linear mixture analysis. This model provides flexibility in the sources representation and selection, thus accounting for spectral variability, small-magnitude errors, and noise. A spatial-spectral coherence regularizer in the optimization allows for pixels classification to be influenced by similar neighbors. We extend the proposed approach for cases for which there is no knowledge of the materials in the scene, unsupervised classification, and provide experiments and comparisons with simulated and real data. We also present results when the data have been significantly under-sampled and then reconstructed, still retaining high-performance classification, showing the potential role of compressive sensing and sparse modeling techniques in efficient acquisition/transmission missions for hyperspectral imagery.

5.2 Introduction

Hyperspectral imaging (HSI) systems acquire images in which each pixel contains narrowly spaced measurements of the electromagnetic spectrum, typically within the visible and long wave infrared region ($400 - 14000$ nm), allowing spectroscopic analysis. The data acquired by these spectrometers play significant roles in biomedical, environmental, land-survey, and defense applications. It contains the geometrical (spatial) information from standard electro-optical systems, and also much higher spectral resolution, essential for material identification.

There are numerous intrinsic challenges associated with effective ground mapping and characterization applications when using overhead HSI:

1. Noise of the collected measurements and sensor artifacts: Noise directly affects detection accuracy and sensitivity to low presence of materials.

2. Complicated schemes of energy interaction between the targeted area and the spectrometer: This causes the total count of photons at the sensor's photoelectric array to include energy from contributing factors from the atmosphere such as aerosols and water vapor.

3. Spectral variability of materials, and spectral mixing: This challenge occurs at the surface level, where spatial resolution and light reflected of nonuniform surfaces generate intrinsic spectral variability associated with each material, and spectral mixing, where each pixel is often composed of a combination of materials. In this case it is difficult to match the data with controlled (laboratory) measured spectra.

4. High dimensionality: HSI data are composed from a large number of spectral bands, posing challenges in visualization, interpretation, and transmission tasks. Nevertheless, these narrowly spaced bands are highly-correlated and redundant, which allows one to carefully exploit "blessings" of high dimensionality.

5.3 Discriminative sparse representations in hyperspectral imagery

Assuming there are C possible materials in the scene, where C_j is the $j-th$ class representing the $j-th$ material. Let the training set for C_j be $\mathbf{X}^j = [\mathbf{x}_1^j, ..., \mathbf{x}_{n_j}^j]$, a matrix where the column $\mathbf{x}_i^j \in \Re^m$ is the $i-th$ training sample corresponding to the $j-th$ class. At the training phase of the algorithm, we learn a dictionary (for each class) by solving the following sparse modeling problem:

$$(\mathbf{D}^j, \mathbf{A}^j) := \arg\min_{\mathbf{D}^j, \mathbf{A}^j} \frac{1}{2}\|\mathbf{X}^j - \mathbf{D}^j\mathbf{A}^j\|_F^2 + \lambda\|\mathbf{A}\|_p, \tag{5.1}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm, $\mathbf{D}^j \in \Re^{m\times k_j}$ is the learned $j-th$ class dictionary, and $\mathbf{A}^j = [\mathbf{a}_1, ..., \mathbf{a}_{n_j}] \in \Re^{k_j \times n_j}$ is the associated matrix of sparse coefficients.

Once these dictionaries are learned, we seek to assign a class label j to each pixel (or block of pixels stacked in column format) to be classified. As proposed in [67], we apply a sparse coding step to the samples $\mathbf{x}$ using each of the learned dictionaries, and simply select the label j corresponding to $\mathbf{D}^j$ that gives the minimum value of

$$\mathscr{R}(\mathbf{x}, \mathbf{D}^j) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}^j\mathbf{a}\|_2^2 + \lambda\|\mathbf{a}\|_1, \quad \forall j, \tag{5.2}$$

$\alpha \in \Re^M$. In other words, our classifier is simply the mapping

$$f(\mathbf{x}) = \{j | \mathscr{R}(\mathbf{x}, \mathbf{D}^j) < \mathscr{R}(\mathbf{x}, \mathbf{D}^i), i \in [1, ..., C], i \neq j\}. \tag{5.3}$$

This means that pixels efficiently represented by the collection of subspaces defined by a common dictionary $\mathbf{D}^j$ are classified together. This measure for supervised classification accounts both for reconstruction (fitting) error and sparsity. The sparsity term especially helps in the presence of noise and/or other artifacts. This naturally comes from the fact that the labeling will tend to prefer the class where the data can be represented in the sparsest way possible, even in cases where the reconstruction error for the tested signal is the same for more than one class. See also [68] for a related penalty when considering the data itself instead of class-dictionaries.

Employing this scheme resulted in very effective and accurate spectral classification. For example, we tested it on HSI data scene APHill. This *APHill* scene was acquired with the

HyMAP sensor, with a total of 432,640 pixels. Each pixel is a 106 dimensional vector after removing the high water absorption and damaged channels. The "known" material labels for APHill, and their corresponding training and validation samples are: **C1:** coniferous trees (967, 228); **C2:** deciduous trees (2346, 234); **C3:** grass (1338, 320); **C4:** lake1 (202, 38); **C5:** lake2 (112, 122); **C6:** crop (1026, 58); **C7:** road (197, 50); **C8:** concrete (74, 25); and **C9:** gravel (87, 38).

In the experiment, samples from the image itself were used to train the classifier. Training and validation classification accuracies for each of the 9 classes are summarized in Table 5.1.

Table 5.1: Per class classification accuracies for the dictionaries learned from the APHill image (without subsampling for this example). First row: classification for training samples. Second row: classification for validation samples.

C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
0.997	0.990	0.996	1	1	0.998	1	1	1
0.951	1	1	1	1	1	0.72	1	0.973

Pixels with incorrect label assignments most often occurred for the coniferous/deciduous/grass (**C1/C2/C3**), and road/concrete/gravel (**C7/C8/C9**) classes. This should not be surprising. First, grass and trees share common spectral features (e.g., high amplitude at the green visible and near infra-red regions). Also, it is common to encounter mixing between those two materials (trees surrounded by grass). Similarly, for the case of concrete and road, spatial resolution plays an important role (sidewalks around roads), but also the fact that concrete and road are spectrally very similar.

5.4 Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery

A more realistic approach to HSI classification is to allow for a pixel to have one or more labels, each corresponding to a material class (in contrast to assigning a single label to each pixel, not accounting for spectral mixing). We extended the classification scheme described in 5.3 to address this more general scenario.

The spectral mixing problem can be seen as a particular case of source separation, where signatures from pure materials are combined to produce the pixel's spectral response. Spectral mixing is caused by a variety of physical factors, many of them often occurring in a nonlinear fashion, and are often difficult to physically model or require many *in situ* parameters associated with the characteristics of the target and the environmental conditions at the time of acquisition. Thus for simplicity of development, and as commonly done in the literature [69], we focus on a linear mixing model (LMM), that is, each pixel is represented as a linear combination of sources or *endmembers*. The coefficients associated with each of these endmembers are called the *fractional abundances*. These fractional abundances indicate the contribution of the associated endmember to the analyzed pixel.

One of the most used models for HSI source separation is the Constrained Least Squares Model (CLS), where the nonnegative coefficients associated with each pixel are constrained to sum to one; see [69] and references therein for more details on the CLS and other proposed models. It is also desirable that the abundance vectors be sparse, meaning that the material at each pixel is explained with as few as possible pure sources. The sum to one constraint of the nonnegative coefficients (i.e., that the coefficient vector are on a simplex), in the CLS model is known to induce a sparse solution, hence a fixed ℓ_1 norm on every coefficient vector. More recently, the Least Squares ℓ_1 (LSℓ_1) model was proposed for this spectral unmixing problem [68, 70]. In this model, the sum to one constraint was relaxed, meaning an ℓ_1-norm constraint on the abundance coefficients needs to be minimized, instead of summing strictly one (no learned dictionary is exploited in these works).

An extension to the problem of spectral mixing can be naturally formulated using the concepts from Section 5.3, adapting them to the LMM. Here $\mathbf{D}^j$ represents the $j-th$ material and $\mathbf{A}^j$ its corresponding abundances. Compared to the standard LMM, where the endmembers are real spectral signatures, here the endmembers are represented as a set of subspaces, thus are learned atoms and their combinations and not actual pure materials. This gives the flexibility to account for material variability caused by factors like noise and nonhomogeneous substances. In addition, the pixels pertaining to a certain class have the flexibility to (sparsely) select the corresponding material atoms that best fit them, and thus more degrees of freedom for a better reconstruction/representation, still with a compact representation. Representing endmembers with more than one representative vector has been used for example in [71], suggesting that endmembers (especially in vegetation) should be represented by a set of spectra, where the abundances were calculated for each element of this set.

The main idea is to train a dictionary for each class, and then form a block-dictionary $\mathbf{D} = [\mathbf{D}^1, ..., \mathbf{D}^C] \in \Re^{m \times k}$, where $k = \sum_{j=1}^{C} k_j$. In this way, the sparse coding on each pixel comes from a sparse "mixed" union of subspaces. We use the constrained sparse coding step of 5.2. We found this choice preferable over the strict sum to one constraint. It avoids the need to introduce a zero vector as an endmember (zero-shade endmember), while still allowing shade and dark pixels to be accounted for. Although a sum to less or equal to one constraint alleviates this issue [72], our choice is less restrictive, giving potentially sparser and more accurate results since it allows more freedom to attain a better class fit.

After the subdictionaries $\mathbf{D}^j$ forming $\mathbf{D}$ have been learned, one per class j, the proposed method for HSI classification and abundance mapping solves the following optimization problem:

$$\min_{(\mathbf{a}_i \in \Re^k) \succeq 0} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \lambda_S \sum_{i=1}^{n} \mathscr{S}(\mathbf{a}_i) + \lambda_G \sum_{i=1}^{n} \mathscr{G}(\mathbf{a}_i, \mathbf{w}_i), \tag{5.4}$$

where the first two terms account for reconstruction and sparsity, respectively, and the third term accounts for a grouping and coherence constraint, which is explained next. With this framework, the ℓ_1 energy of the $\mathbf{a}_i$ coefficients corresponding to the block $\mathbf{D}^j$ in $\mathbf{D}$ indicates the amount of material from the $j-th$ class in the mixture for the pixel i. Similarly, we can use the

reconstruction limited to atoms and coefficients of a given class to determine the contribution of that class to the pixel i.

5.4.1 Imposing spatial coherence

Up to this point, each pixel was treated independently from each other ($\lambda_G = 0$ in the equation above). To exploit the geometric structure in the image, one can make the estimation of the abundance coefficients $\mathbf{a}_i$ for a given pixel to be influenced by neighboring pixels, introducing spatial and spectral coherence in the modeling and coding process. This coherence will depend both on the pixels' spectral shape *and* the coefficient vector similarities. This can be implemented by defining a function $\mathscr{G}$ that behaves as a grouping (coupling) term on the coefficients,

$$\mathscr{G}(\mathbf{a}_i, \mathbf{w}_i) = \|\mathbf{M}(\mathbf{a}_i - \sum_{l \in \eta} w_{il}\mathbf{a}_l)\|_2^2, \tag{5.5}$$

where η denotes the neighborhood of the $i-th$ pixel. We define a weighting function $w_{il} = \frac{1}{Z_i}\exp^{\frac{-\|\mathbf{x}_i - \mathbf{x}_l\|_2^2}{\sigma^2}}$, where Z_i is a pixel-dependent normalization constant, such that $\sum_{l\in\eta} w_{il} = 1$, and σ^2 is a density parameter controlling the width of the Gaussian (here set to be the average of the data's pairwise Euclidean distance, either local for each pixel or global for the whole data). This weighting function is close to 1 if the pixels are very similar and 0 if orthogonal. Its purpose is to compare the $i-th$ pixel with a weighted linear combination of its neighbors. There is no guarantee that pixels with strong similarities will select the same active set (atoms) from the $j-th$ class subdictionary $\mathbf{D}^j$, even in cases where they have similar mixtures. We do want these coefficient vectors to be coupled by similar ℓ_1-norm in each block. For this purpose, $\mathbf{M} \in \Re^{C\times k}$ is defined as $\mathbf{M} = \begin{bmatrix} \mathbf{1}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}^2 & & \\ \vdots & & \ddots & \\ \mathbf{0} & \cdots & & \mathbf{1}^C \end{bmatrix}$, where $\mathbf{1}^j \in \Re^{1\times k_j}$ is a vector of ones corresponding to the number k_j of atoms per subdictionary. The purpose of $\mathbf{M}$ is to compare the similarity, in the ℓ_1 norm, of the coefficients from each subdictionary, promoting similar per-block ℓ_1 norm for similar pixels.

The neighborhood η is not necessarily restricted to spatial neighbors, but also nonlocal (all across the image) neighbors with strong similarities, encouraging similar or related areas across the whole data to cooperate in the classification. Let the weights be represented in the matrix $\mathbf{W} \in \Re^{n \times n}$. This weight matrix is formed by the sum of local and nonlocal weights, $\mathbf{W} = \mathbf{W}_{nl} + \mathbf{W}_s$, where $\mathbf{W}_{nl}$ is a weight matrix associated with similar abundance vectors, and $\mathbf{W}_s$ is the weight matrix associated with the spatial neighbors; $\mathbf{W}$ is similar to the matrix used in the nonlocal means algorithm [73] and the spectral/spatial kernel matrix approach of [74]. Incorporating this grouping term gives robustness to noise and stability in the coefficient (abundances) estimates, capturing non-linearities and complicated structures in the feature space. Note that these weights can be calculated prior to optimization (standard techniques to accelerate non-local means can be used if desired). The incorporation of nonlocal information/coherence and both spatial and spectral neighborhoods and similarities was found to improve the classification results when compared with only local and spatial constraints, which are a particular case of the model. This is particularly reflected for example when there is (significant) missing information (see experimental section) or large variability in the class.

The optimization problem in (5.4) can no longer be solved independently for each pixel due to the coupling in the coefficient vectors. However, we efficiently solve this by considering ($\mathbf{I}$ is an identity matrix)

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \lambda_G \mathbf{A}\mathbf{W} \end{bmatrix}, \quad \tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \lambda_G \mathbf{I} \end{bmatrix}.$$

We solve the coupling using a standard Gauss-Seidel type of iteration (or primal decomposition), where we iteratively solve the problem by first calculating the sparse coefficients with no coupling, storing a copy, and then re-calculating the subsequent coding iteration using this copy.

This concludes the model for supervised HSI classification and unmixing. We next present experimental results supporting this proposed framework.

5.5 Supervised Source Separation Experiments

We consider a series of experiments to test the performance of the proposed algorithm. We use three HSI scenes described below. We compare our proposed framework with standard methods for HSI classification, such as an Euclidean Distance classifier (ED) and a Spectral Angle Mapper classifier (SAM), see below for the exact definitions.[1] In addition, we include an ℓ_1-based optimization scheme that uses all available training samples as dictionary (see next for a detailed description of where the training samples come from), hence a dictionary of data samples of much larger cardinality than our compact learned dictionary. Our algorithm is termed the Dictionary Modeling (DM) when no spatial coherence is imposed, and Dictionary Modeling with Spatial Coherence (DMS) when the spatial coherence is imposed. For all the following experiments, we set the algorithm parameters to be $\lambda_S = 0.5/\sqrt{b}$, and $\lambda_G = 0.01$ (only for DMS). The weight matrix $\mathbf{W}$ was built using a 3×3 spatial region surrounding the pixel, and 4 non-local neighbors. A maximum number of iterations for the dictionary learning phase was set to 150, and 5000 for the sparse coding phase. The results were found to be stable to the particular selection of these parameters, which can in general be done via standard cross-validation. We selected the sparse coding technique called Least Angle Regression and Shrinkage (LARS) [75], and used the fast implementation in the Sparse Modeling Software (SPAMS), publicly available at `http://www.di.ens.fr/willow/SPAMS/`. For DM, DMS, and ℓ_1 methods, and mostly for computational/implementation convenience, all pixels were normalized to have unit norm prior to processing.

5.5.1 HSI Data Sets

We process 3 hyperspectral scenes for these experiments, Figure 5.1:

[1] These methods have been selected due to their popularity and simplicity. Other methods will lead to decisions such as the needed dimension of the working space, which might obscure the underlying virtue of the tested techniques.

1. AVIRIS Indian Pines: The Indian Pines is a small portion of the Northwest Tippecanoe County in Indiana, acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) system in 1992. It consists of a $145 \times 145 \times 220$ datacube reduced to 188 bands after removing water absorption and noisy bands. The data are publicly available at `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`. The scene consists mainly of 16 classes, mostly vegetation and agricultural crops, and a full scene classification ground-truth is available.

2. HyDICE Urban: The Urban scene was acquired with the Hyperspectral Digital Collection Experiment (HyDICE) sensor over Copperas Cove, Texas. It consists of a $307 \times 307 \times 210$ datacube reduced to 162 channels after removing the water absorption and noisy bands. 8 classes were manually selected. The data are publicly available at `http://www.agc.army.mil/hypercube/`.

3. HyMAP APHill: The APHill scene was taken with the Hyperspectral Mapper (HyMAP) over Virginia (with permission from the US Army Engineer Research and Development Center, Topographic Engineering Center, Fort Belvoir, VA). It consists of a $645 \times 400 \times 128$ datacube, reduced to 106 channels after removing noisy and water absorption bands. Nine classes were manually selected for the experiments.

Table 5.2 summarizes the class and training/testing samples information for each of these datacubes.

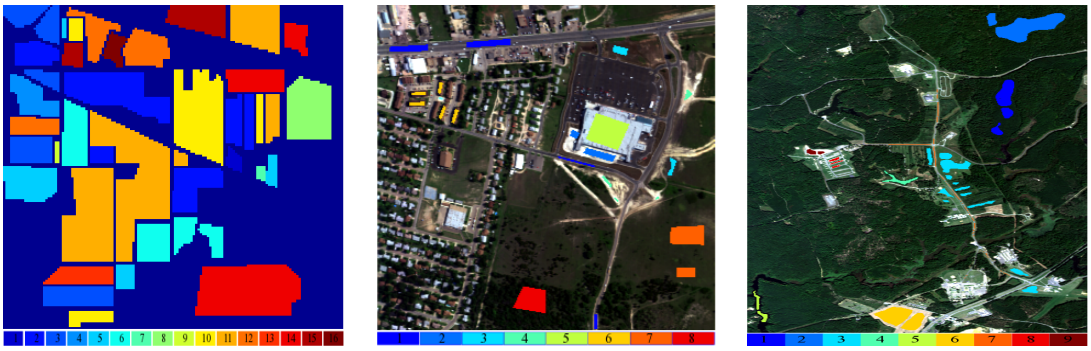
5.5.2 Experiment 1: Supervised Multi-label Mapping

In the first experiment, we look at the overall training accuracy for the three datasets. We compare our proposed algorithm with two classical approaches, minimum Euclidean Distance (ED) and Spectral Angle Mapper (SAM). We also compare this approach with an ℓ_1 minimization scheme without learned dictionaries. We define these classifiers as

1. Euclidean Distance:

Table 5.2: Class labels and number of samples per class for the Indian Pines, APHill, and Urban HSI cubes.

Class		Samples	
Label	**Name**	**Train**	**Test**
Indian Pines			
1	Alfalfa	27	27
2	Cornnotill	717	717
3	Corn-min	417	417
4	Corn	117	117
5	Grass/Pasture	248	249
6	Grass/Trees	373	374
7	Grass/pasture-mowed	13	13
8	Hay-windrowed	244	245
9	Oats	10	10
10	Soybeans-notill	484	484
11	Soybeans-min	1234	1234
12	Soybean-clean	307	307
13	Wheat	106	106
14	Woods	647	647
15	Bldg-Grass-Tree-Drives	190	190
16	Stone-steel towers	47	48
APHill			
1	Coniferous	597	598
2	Deciduous	1290	1290
3	Grass	829	829
4	Lake1	120	120
5	Lake2	117	117
6	Crop	792	792
7	Road	123	123
8	Concrete	49	50
9	Gravel	62	63
Urban			
1	Road	254	255
2	Concrete	106	107
3	Dark soil	76	76
4	Bright soil	39	40
5	Gray rooftop	417	417
6	Brown rooftop	95	96
7	Grass	338	339



(a) (b) (c)

Figure 5.1: HSI cubes used in this work. (a) Ground-truth of AVIRIS Indian Pines (b) HyDICE Urban: patch colors corresponds to each of the 8 known classes. (c) HyMAP APHill: patch colors corresponds to each of the 9 known classes. (This is a color figure.)

$f_{\text{ED}}(\mathbf{x}) = \{j | \|\mathbf{x} - \hat{\mathbf{x}}^j\|_2 \leq \|\mathbf{x} - \hat{\mathbf{x}}^i\|_2, i \neq j, \forall i, j \in [1, 2, ..., C]\}$, where $\hat{\mathbf{x}^j}$ is the averaged spectra of all training samples from the $j-th$ class.

2. Spectral Angle Mapper: The SAM is defined as $\text{SAM}(\mathbf{x}, \mathbf{y}) = \cos^{-1}(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2})$, hence the mapping is $f_{\text{SAM}}(\mathbf{y}) = \{j | \text{SAM}(\mathbf{y}, \hat{\mathbf{x}}^j) \leq \text{SAM}(\mathbf{y}, \hat{\mathbf{x}}^i), i \neq j, \forall i, j \in [1, 2, ..., C]\}$.

3. ℓ_1 minimization: Here we use an approximate sparse solution to $\mathbf{a}^* = \text{argmin}_{\mathbf{a} \succeq 0} \|\mathbf{a}\|_1$ s.t. $\|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 \leq \varepsilon$, for some small error $\varepsilon = 0.05$ (as with all the comparisons reported in this paper, the parameters leading to the best results were selected). Here $\mathbf{D}$ is a matrix with all the training samples available from all classes, each block $\mathbf{D}^j$ corresponding to the samples from class j (concatenated to create $\mathbf{D}$). We assign a label following the mapping $f_{\ell_1}(\mathbf{x}) = \{j | \mathscr{R}(\mathbf{a}^j) \leq \mathscr{R}(\mathbf{a}^i), i \neq j, \forall i, j \in [1, 2, ..., C]\}$, where $\mathbf{a}^j$ is the coding portion corresponding to the sub-dictionary $\mathbf{X}^j$ for class j and $\mathscr{R}$ is the reconstruction error when coding the data sample (pixel) only with samples selected by $\mathbf{a}^j$ (equivalent to reconstruction after setting to zero all elements of $\mathbf{a}^*$ but those in $\mathbf{a}^j$). Similar results are obtained with $\mathscr{S}$ (the ℓ_1 norm) instead of $\mathscr{R}$.

We divided the set of known samples from each class into training and testing sets by random selection for 25 draws. We selected the number of atoms per sub-dictionary to be $k_j = \min(50, n_j)$, where n_j is the total number of training samples for the $j-th$ class. The results are summarized in Table 5.3, showing the average and standard deviation of the overall classification accuracy for the 25 runs. We make the following observations from this experiments:

1. The results show that using the compact learned dictionary in the proposed framework outperforms the other more classical approaches. Compared to using the data itself as dictionary, a significantly reduced computational cost is obtained as well.

2. The spatial coherence term significantly improved the classification accuracy in the Indian Pines dataset, mainly due to the large uniform areas. Also, the ED and SAM methods performed poorly on these data, using averaged spectra was not sufficient for a good class representation. In contrast, the proposed approach, while simple in nature, efficiently selects the atoms that best represent the rich classes.

3. Although the classes seem to be more "separable" in the APHill and Urban datasets, leading to relatively good results using ED and SAM, the proposed approach provides close to 100% accuracy, which could play an essential role in certain mapping applications.

We also compare the classification performance of our proposed method with linear and nonlinear Support Vector Machine (SVM) classifiers on Indian Pines. On a first comparison, we follow the experimental settings of [76] (only train and test on classes 2, 3, 6, 8, 12, 13, and 14), using multi-class linear and nonlinear (Radial Basis Function) SVMs (optimized for best performance using cross-validation). As in [76], we used the libSVM software [77]. Since these SVMs do not explicitly enforce spatial information, we did not incorporate the four spatial neighbors in DMS for this particular comparison. Averaging 25 runs, the linear SVM obtained an overall classification accuracy of 90.6%, and the nonlinear SVM obtained 95.7%. We obtained 88.2% and 96.5% using DM and the (simplified) DMS, respectively. On a second experiment, we compare the overall classification accuracies of the spatio-spectral kernel

SVM method recently proposed in [78] with the full DMS. The authors in [78] report the best overall classification accuracy of 72.3% using 1% of all labeled samples for training. We obtained 73.4% using DMS. We also tested using 5% of all labeled samples for training, where the authors in [78] report the best overall classification accuracy of approximately 83.5% (from their provided graph), while we obtained 84.4% with the proposed DMS. These best results reported in [78] use an 11×11 spatial window at 2 different scales, while DMS simply includes the four spatially connected neighbors (north, south, west, and east pixels). On a more direct comparison, the result reported in [78] using a 3×3 spatial window is 75.1%.[2]

We also processed all datasets with several algorithms from the MultiSpec software package available at `https://engineering.purdue.edu/~biehl/MultiSpec/documentation.html`. We obtained classification accuracies using ED, SAM, Fisher Linear Discriminant (FDL), Constrained Energy Minimization (CEM) matched filter, and the Extraction and Classification of Homogeneous Objects (ECHO) classifier, which uses the FDL, and a 2×2 spatial window. We compared with the proposed DM and DMS, and the results are summarized in Table 5.4. In all cases, DMS attained the highest classification accuracy. On the Urban results, there is no performance gain of DMS over DM, implying that spectral information was sufficient to correctly classify the materials.

Following these comparisons with standard techniques we observe the quality of the proposed HSI modeling framework, leading to state-of-the-art classification results.

5.5.3 Experiment 2: Mapping of Reconstructed Data From Significant Undersampling

Recently, a non-parametric (Bayesian) approach to sparse modeling and compressed sensing was proposed in [79], where most of the data are eliminated uniformly at random (in the HSI

[2] In this experimental setting reported in [78], 220 AVIRIS spectral channels were used for processing. We also used these channels.

Table 5.3: Results for the first supervised classification experiment. Shown are the mean (first row) and standard deviation (second row) of 25 runs for three HSI datasets. Best results are in bold.

Image	**Accuracy (mean and standard deviation)**				
	ED	**SAM**	ℓ_1	**DM**	**DMS**
Indian Pines	0.3788	0.4437	0.8639	0.878	**0.9352**
	0.0082	0.006	0.0045	0.0048	0.0038
APHill	0.9006	0.9151	0.9758	0.9894	**0.9966**
	0.0034	0.0031	0.0023	0.0011	0.0009
Urban	0.9013	0.972	0.9894	0.999	**0.9992**
	0.0047	0.0034	0.0032	0.0009	0.0007

Table 5.4: Supervised classification accuracy comparison using ED, SAM, FDL, CEM matched filter, and ECHO classifiers from MultiSpec. DM and DMS results are based on the same experimental settings. Best results are in bold.

Image	**Accuracy**						
	ED	**SAM**	**FLD**	**CEM**	**ECHO**	**DM**	**DMS**
Indian Pines	0.454	0.566	0.774	0.711	0.780	0.879	**0.933**
APHill	0.924	0.919	0.993	0.966	0.994	0.993	**0.997**
Urban	0.891	0.945	0.991	0.893	0.991	**0.993**	**0.993**

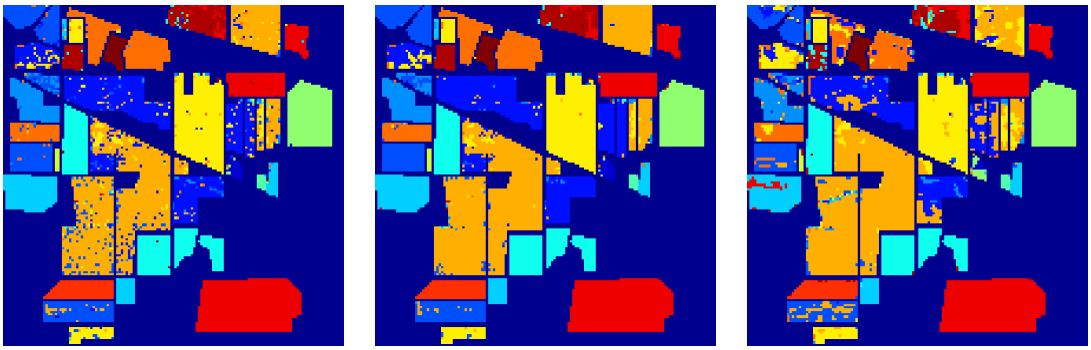
cube in our case), and then reconstructed using a dictionary that is learned only from the available data. The method automatically estimates the dictionary size, and makes no explicit assumption on the noise variance. In addition, it can deal with non-uniform noise sources in the different bands, a problem often encountered in HSI. As with the work here described, it is assumed in this method that each 3D block in the HSI cube may be represented as a linear combination of dictionary elements of the same dimension, plus noise, and the dictionary elements are learned *in situ* based on the observed data (no *a priori* training). The number of dictionary elements needed for representation of any particular such block is typically small relative to the block dimensions, and all the image blocks are processed jointly (collaboratively) to infer the underlying dictionary. With a Bayesian perspective, priors are imposed on the coding coefficients, dictionary elements, and hyperparameters. The Bayesian inference is efficiently performed via efficient modern optimization techniques.

While the method can reconstruct the data with a high Peak Signal to Noise Ratio (PSNR), we investigate how well the spectral information is preserved, that is, we test how material classification is affected after randomly "throwing away" most of the data and then interpolating. This experiment further supports the validity of the proposed HSI model and motivates the investigation of new sensing paradigms, where parts of the data are not sensed or read-out for faster performance. In addition, and as we will see below, these tests show that the proposed classifications framework can efficiently address unexpected data loss.

In Table 5.5 we provide an idea on how this subsampling and reconstruction affects the spectral angles between the original and reconstructed pixels. Most of the high angles are due to two reasons: first, small spatial areas could not be appropriately reconstructed, since the approach uses information from neighboring pixels (spatial window) for interpolation. Secondly, areas with low SNR fail to be efficiently reconstructed. As an example, the lakes in the APHill scene have the highest spectral angles, mainly because most of the energy from the sun was absorbed by the water. These problems can be addressed by an adaptive sampling strategy, instead of the random one here employed for testing the classification accuracy.

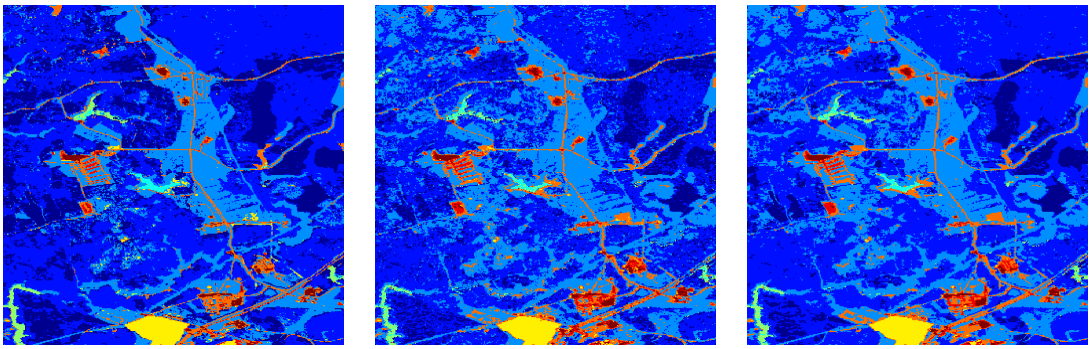
Table 5.5: PSNR and spectral angles between the original and reconstructed datasets. The first column shows the parameters selected for the reconstruction, the spatial window size and the amount of available data used for reconstruction. The spectral angles are in degrees.

Image	**Reconstruction PSNR and Spectral Angle**				
Indian Pines	**PSNR**	**min**	**max**	**avg.**	**median**
$3\times3,2\%$	35.1827	0.6558	27.9661	2.6519	2.2808
$3\times3,5\%$	42.0383	0.3353	21.9318	1.2400	1.0551
$3\times3,10\%$	48.0072	0.2864	8.1674	0.7328	0.6849
$3\times3,20\%$	50.0940	0.2694	8.2588	0.6186	0.5964
$4\times4,2\%$	36.5539	0.5219	25.9245	2.1256	1.7854
$4\times4,5\%$	42.7197	0.3121	18.8338	1.1327	0.9553
$4\times4,20\%$	50.0670	0.2856	8.3672	0.6197	0.5952
$5\times5,2\%$	36.4205	0.4386	27.2215	2.0617	1.6809
APHill					
$3\times3,2\%$	33.2542	0.5492	56.9784	2.7071	2.0083
$3\times3,5\%$	38.2726	0.2923	74.2222	1.2897	0.9920
$3\times3,10\%$	43.8655	0.2559	22.6844	0.9632	0.7928
$3\times3,20\%$	46.1789	0.3563	14.1513	1.1429	0.9838
$4\times4,2\%$	37.8549	0.4632	74.1300	2.4906	1.8053
$4\times4,5\%$	40.0584	0.3383	65.6407	1.6301	1.2753
$4\times4,20\%$	46.2491	0.3196	19.6792	1.3074	1.0892
$5\times5,2\%$	33.5847	0.4393	72.7613	2.4333	1.7413
Urban					
$3\times3,2\%$	30.3053	2.5799	61.4310	7.8448	7.1313
$3\times3,5\%$	38.8684	2.4565	32.9800	6.3067	6.2368
$3\times3,10\%$	43.0309	2.4738	26.7932	6.0401	6.0933
$3\times3,20\%$	46.2861	0.4358	21.9183	1.5765	1.2608
$4\times4,2\%$	32.4952	0.6723	59.0952	4.8124	3.7554
$4\times4,5\%$	40.6783	0.6273	37.3109	2.4521	1.9213
$4\times4,20\%$	45.6692	0.4387	23.1958	1.6735	1.3312
$5\times5,2\%$	32.8143	0.7574	59.6115	4.5828	3.5597



(a) (b) (c)

Figure 5.2: Effect of the proposed coherence term on Indian Pines. From left to right: (a) classification with no spatial coherence, (b) classification with spatial coherence, and (c) reconstructed data (3×3, 20%) using a dictionary learned from the original data and spatial/spectral coherence. (This is a color figure.)



(a) (b) (c)

Figure 5.3: Effect of spatial coherence and significant sub-sampling in APHill mapping. (a) Original, (b) mapping after reconstructing 98% of the data ($3 \times 3, 2\%$) with no spatial coherence, and (c) mapping after reconstructing 98% of the data ($3 \times 3, 2\%$) with spatial/spectral coherence. (This is a color figure.)

Table 5.6 summarizes the classification accuracies obtained for the HSI datasets under several under-sampling conditions. We compare our proposed algorithm with the same ones used in Experiment 1. We make the following observations:

1. The proposed DM/DMS algorithms outperform the other methods in most cases. However, there are some cases with the Indian Pines images where the ℓ_1 minimization method with data as dictionary performs slightly better, particularly in extreme cases were only 2% of the data are used. This difference in performance was mainly on classes with a small number of training samples (e.g., Alfalfa, Oats, and Grass/pasture-mowed classes). The learned dictionary was not able to appropriately model the class (this can be solved letting the dictionary be the data itself when not enough training samples are available). Nevertheless, this problem was solved with the DMS method, which effectively uses both spatial and spectral information from other pixels for a "grouping" effect in the abundance estimates, which is clearly shown in figures 5.2 and 5.3. Such use of spatial and spectral information if critical for highly under-sampled data.

2. Classification accuracies are very similar to those obtained using the original datasets without sub-sampling, especially with the $3 \times 3, 20\%$, and $4 \times 4, 20\%$ realizations, thus showing that it is possible to get highly accurate classification performance with only a fifth of the data. The decrease in classification accuracy for highly under-sampled data comes from errors involving highly similar classes, or insufficient training samples. The proposed approach is not always able to separate those similar classes when there is not sufficient data (understanding if there is sufficient information at all for such separation is an interesting subject of research). For instance, focusing on datasets reconstructed from only 2% of the data and a 3×3 spatial window, we observed that in the APHill scene, most of the misclassified pixels correspond to the "Lake1" and "Lake2" classes, with a 54.17% and 73.50% accuracy, respectively. From these classes, only 5 pixels were labeled from a class different than the "Lake1" or "Lake2" class. Similar errors occurred with the "Coniferous" and "Deciduous" classes, and the "Concrete" and "Road" classes.

In the Urban scene, the two classes with the lowest performance for the DM method were "Road" and "Concrete", with 80% and 84.1121% respectively, with only 14 samples with misclassification errors from a different class.

3. We observe, for example for APHill, that sometimes a slight classification improvement is obtained when less samples are available (e.g., compare for DM the results for 5% with those for 20% available data). This minor difference is well between the expected numerical accuracy, and can be the result of this or of additional smoothing that is naturally obtained when reconstructing from very few samples.

4. While maintaining performance with under-sampled data is not so difficult (or critically relevant) for low-performing techniques, such as EM for Indian Pines, it is challenging when such performance is originally very high, as with DMS for APHill or Urban (with almost perfect classification for the full data). As demonstrated in this table, DMS manages to maintain such high accuracy.

The results shown in Table 5.6 were obtained using training samples derived from the reconstructed images used for validation. In a more realistic approach, we provide classification results for cases in which the learning process was done *a priori* using data from the original dataset (no sub-sampling), and the validation is done on the reconstructed dataset. The idea is to show that a pre-learned dictionary (which is a compact representation of the classes) could be used in future acquisitions. These results are summarized in Table 5.7 for the HSI scenes after eliminating 80% of the data.

Finally, we have also tested completely eliminating 10% of the bands, simulating for example a sensor defect, where the optical arrays for several wavelengths are damaged. In addition to having missing bands, a large percentage of the rest of the data were removed at random as before. The results are summarized in Table 5.8. This shows that we can miss both entire bands and significant amounts of data at random, and still obtain very reasonable classification results, out-performing all the other tested methods.

Table 5.6: Overall classification accuracies for the datasets reconstructed from highly undersampled data. The first column shows the spatial window and data percentage used for reconstruction.

Image	**Accuracy**				
	ED	**SAM**	ℓ_1	**DM**	**DMS**
Indian Pines					
$3\times3, 2\%$	0.3823	0.3086	0.5070	0.4426	**0.7666**
$3\times3, 5\%$	0.3742	0.4177	0.7477	0.7381	**0.8781**
$3\times3, 10\%$	0.3797	0.4332	0.8216	0.8141	**0.9001**
$3\times3, 20\%$	0.3796	0.4359	0.8339	0.8253	**0.9007**
$4\times4, 2\%$	0.3819	0.3767	0.6901	0.6650	**0.8785**
$4\times4, 5\%$	0.3821	0.4272	0.8006	0.8021	**0.9014**
$4\times4, 20\%$	0.3807	0.4380	0.8467	0.8287	**0.9115**
$5\times5, 2\%$	0.3801	0.3813	0.7477	0.7724	**0.8986**
APHill					
$3\times3, 2\%$	0.8920	0.8544	0.8976	0.9111	**0.9822**
$3\times3, 5\%$	0.8885	0.9121	0.9681	0.9736	**0.9917**
$3\times3, 10\%$	0.8933	0.9119	0.9751	0.9772	**0.9917**
$3\times3, 20\%$	0.8943	0.9149	0.9782	0.9792	**0.9915**
$4\times4, 2\%$	0.8991	0.8890	0.9455	0.9546	**0.9897**
$4\times4, 5\%$	0.8933	0.9169	0.9819	0.9804	**0.9950**
$4\times4, 20\%$	0.8943	0.9232	0.9862	0.9874	**0.9932**
$5\times5, 2\%$	0.9028	0.9064	0.9628	0.9739	**0.9930**
Urban					
$3\times3, 2\%$	0.9085	0.8836	0.9334	0.9496	**0.9944**
$3\times3, 5\%$	0.8911	0.9452	0.9907	0.9963	**0.9981**
$3\times3, 10\%$	0.8930	0.9645	0.9919	0.9969	**0.9994**
$3\times3, 20\%$	0.9042	0.9701	0.9894	0.9981	**1.0000**
$4\times4, 2\%$	0.9123	0.9011	0.9577	0.9807	**0.9969**
$4\times4, 5\%$	0.9048	0.9564	0.9944	0.9988	**0.9994**
$4\times4, 20\%$	0.9048	0.9708	0.9888	0.9988	**1.0000**
$5\times5, 2\%$	0.9135	0.8955	0.9664	0.9907	**0.9975**

Table 5.7: Overall classification accuracy for reconstructed data with only 20% of the pixels used for reconstruction. The class dictionaries were learned *a priori* using the original dataset.

Image	**Accuracy**				
	ED	**SAM**	ℓ_1	**DM**	**DMS**
Indian Pines	0.3720	0.4432	0.7811	0.7967	**0.8434**
APHill	0.8353	0.9269	0.9397	0.9538	**0.9691**
Urban	0.8936	0.8955	0.9589	0.9670	**0.9844**

Table 5.8: Classification results for the Urban dataset when entire bands are missing in addition to the missing data at random as before. The data are reconstructed from this highly undersampled data before classification. The first column shows the spatial window, data percentage, and percentage of bands used for reconstruction.

Image	**Accuracy**				
	ED	**SAM**	ℓ_1	**DM**	**DMS**
Original	0.9079	0.9850	0.9850	0.9981	**1.0000**
$2\times2, 2\%, 90\%$	0.8609	0.8571	0.8797	0.9117	**0.9568**
$2\times2, 5\%, 90\%$	0.8966	0.9586	0.9718	0.9925	**0.9962**
$4\times4, 2\%, 90\%$	0.8910	0.8835	0.9305	0.9718	**0.9699**
$4\times4, 5\%, 90\%$	0.8947	0.9380	0.9474	0.9774	**0.9906**

5.5.4 Intermezzo

We make the following conclusions about the supervised experiments shown above:

1. The proposed method is superior in terms of classification performance when compared with standard classification approaches.

2. Accurate material classification can be obtained even when the majority of the data are missing. This is a clear example of how the data redundancy in high dimensional HSI can actually be beneficial for several tasks, including less expensive acquisition and faster transmission (at the cost of more sophisticated post-processing).

We now extend this to the unsupervised case, where there is no data for pre-training of the dictionary.

5.6 Unsupervised source separation and abundance mapping

In previous sections we showed a methodology for mapping HSI when we know *a priori* the classes of interest (supervised mapping). The first stage consisted of learning per-class subdictionaries, and the second stage consisted of estimating the corresponding abundances. In this section we address the case for which there is no *a priori* information about the sources present in the scene, which could be seen as a blind source separation problem (unsupervised). A significant amount of research has been dedicated to automatically determining these endmembers. Many of the algorithms are simplex-projection based, where the vertices of the volume enclosing the data are considered as the endmembers [80]. The classical endmember estimation algorithms assume that there are pure pixels in the image. Examples of this are the N-Finder [81], Vertex Component Analysis (VCA) [82], and Pixel Purity Index (PPI) [83]. Algorithms that do not make this assumption include Dependent Component Analysis (DECA) [84], Minimum Volume Simplex Analysis (MVSA) [85], and more recently, the Simplex Identification by Split Augmented Lagrangian (SISAL) method [86], which breaks the unmixing problem into smaller convex (and simpler) problems.

A second class of algorithms addressing this problem is based on nonnegative matrix factorization (NMF), attempting to find the matrices $\mathbf{D}$ and $\mathbf{A}$ such that $\mathbf{X} \approx \mathbf{DA}$. Constrained versions of NMF for HSI unmixing have been extensively proposed, e.g., [87, 88, 89]. In addition, works that incorporate a sparsity constraint in NMF have been proposed in [90, 91], also in combination with a spatial constraint [92]. It is important to mention that in [92], the spatial constraint enforces similarities in the abundance vectors, while in our approach (DMS) we enforce local and non-local similarities that are weighted by spectral information. Minimum Volume Constrained (MVCNMF) [93], is a method for which the NMF is constrained to have minimum volume (thus limiting the non-uniqueness of the NMF because of the inward force). Finally, Minimum Dispersion NMF [94] was proposed as a method to deal with very flat spectra, where the constraint is to minimize the sum of the variances of the endmembers. In contrast with our proposed framework, these approaches force the materials to be represented by a single spectra, a very limiting model as explained before. In our method we let each data sample select the best possible sparse linear combination of atoms from a learned endmember subdictionary. As seen in the previous sections for supervised classification, representing a class/material by a single spectra is a limiting factor in terms of class reconstruction and accurate mapping. This motivates us to extend the above proposed dictionary based approach to the unsupervised case.

Our framework could be seen as a generalization of a sparsity and spatially constrained NMF, where the sources contain more than one atom per material. In order to automatically learn the sources online, just for the image being analyzed, and since the problem is non-convex, we need an initialization procedure such that the learned subdictionary for a given class maintains a relative separation from the subdictionaries learned for the other classes. We could naively set the number of atoms k_j per block and randomly select samples from the data, though doing this offers no guarantee of a valid subdictionary separation (and therefore separation between the sources). In this work, the subdictionaries are initialized by a fully constrained NMF (nonnegativity and sum to one of the abundance coefficients, CNMF). This sets the initial number of atoms per subdictionary to be $k_j = 1$ for all the classes j. Then, we will increase the number of atoms per subdictionary block as we progress in adapting the dictionary to the

data. The idea is to eventually find the best class representation by increasing the dimensionality of the space where the class lives (represented by the subdictionary). It is intuitive that increasing this dimensionality will characterize the material more appropriately. Nevertheless, as the number of atoms per subdictionary increases, the cross-correlation (or coherence) of the dictionary, and in particular between the different subdictionaries, will increase. This has a negative impact in classification, especially with classes that are very close to each other. In addition, it has been shown in the sparse modeling literature that dictionary incoherence plays a crucial role in obtaining the correct sparse representation [1]. This is, the dictionary needs to be as incoherent as possible to get a proper sparse representation of the signal, noting that if the coherence is maximum, any subset selection from the dictionary would yield similar solutions, hence losing uniqueness and discriminative power. To overcome this, we impose an additional constraint for keeping the different blocks of the dictionary as "pure" as possible by explicitly imposing a degree of incoherence. This is detailed next.

5.6.1 Imposing Spectral Incoherence

The (sub-)dictionary incoherence plays the role of keeping the estimated subdictionary (block) for a given class as orthogonal as possible with each subdictionary corresponding to the other classes. In order to perform classification, the subdictionaries representing similar classes such as "Grass" and "Trees," need to maintain a certain degree of separation, since there are common features in the spectra that are shared among the classes (e.g., high amplitude in the visible/near-infrared regions). As the number of atoms per dictionary block increases, the more susceptible the dictionary becomes to these "mixing effects." We want to avoid as much as possible a class using atoms from the "wrong dictionary," and unless explicitly constrained/encouraged not to, similar classes will learn some similar atoms, potentially misleading in the dictionary selection. To address this, we define a subdictionary incoherence term following [95], where the authors have reported performance improvements when incorporating this term in recognition tasks,

$$\mathscr{I}(\mathbf{D}^j) = \sum_{i \neq j}^{C} \|\mathbf{D}^{iT}\mathbf{D}^j\|_F^2. \tag{5.6}$$

With this we encourage the subdictionary $\mathbf{D}^j$ to be separated from the rest of the subdictionaries. Since the atoms will not be completely orthogonal (e.g., due to the overcompletness of $\mathbf{D}$), we penalize using (5.6). To update each of the dictionary atoms $\mathbf{d}_i^j$, we use the following update rule:

$$\begin{aligned}
\mathbf{d}_i^{j,t} &\leftarrow P\{\mathbf{d}_i^{j,t-1}\} \\
&= \max(0, \mathbf{d}_i^{j,t-1} + \mu \frac{\partial}{\partial \mathbf{d}_i^j} \mathscr{R}_{\ell_2} + \lambda_I \mathscr{I}(\mathbf{D})) \\
&= \max(0, \mathbf{d}_i^{j,t-1} + 2\mu((\mathbf{D}^j\mathbf{A}^j - \mathbf{X}^j)\mathbf{A}_i^{jT} \\
&\quad + \lambda_I(\overline{\mathbf{D}}\overline{\mathbf{D}}^T)\mathbf{d}_i^{j,t-1}),
\end{aligned} \tag{5.7}$$

where $\overline{\mathbf{D}}$ is the concatenation of the subdictionary blocks not including the class being updated, $\mathbf{A}_i^j$ is the $i-th$ row of the coefficients matrix $\mathbf{A}^j$, and λ_I is the penalty parameter related to the subdictionaries incoherence. Again, each atom is normalized to have unit norm. Note that we update the atoms using the analyzed data $\mathbf{X}^j$ (unsupervised case). In the supervised case, $\mathbf{X}^j$ included the samples representing the $j-th$ class, and this was known *a priori*. Since we do not have training samples, we set during the iterations $\mathbf{X}^j$ as the set of samples with maximum ℓ_1 norm in the corresponding class coefficients. For example, we initialize our algorithm using CNMF. Then, we split the data by assigning to $\mathbf{X}^j$ the data samples whose largest abundance coefficients correspond to the $j-th$ class. We then proceed to update the dictionaries using this assignment following (5.7), and once these are updated, we re-assign the data following the same criteria. This is in the style of classical K-means, but with subdictionaries as "centroids," a different update rule (5.7), and an ℓ_1 criteria for assignment. For simplicity considerations, in this work the subdictionaries have an equal amount of atoms.

5.6.2 Stopping Criteria

In our scheme, we initialize the sparse modeling process using CNMF, and we progress the class representation by adding an atom (initialized at random) to each subdictionary. We desire the best possible class representation, so we stop adding atoms when the reconstruction error

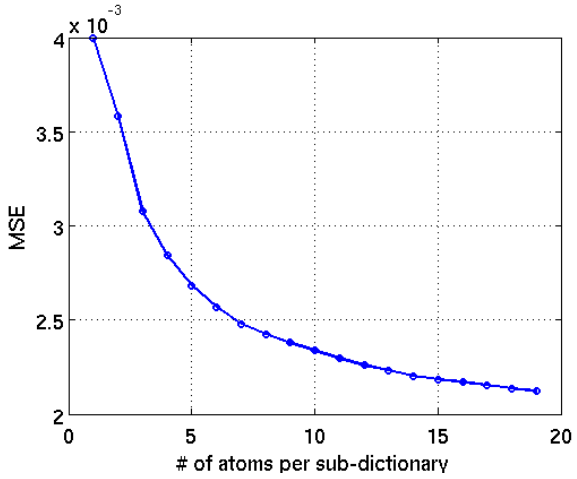


Figure 5.4: Reconstruction Mean Square Error (MSE) as a function of the number of atoms per sub-dictionary for a single run in the Urban dataset.

stops decreasing (or decreases below a threshold),

$$(\{\frac{1}{n}\|\mathbf{X}-\mathbf{DA}\|_F^2\}^{k_j}-\{\frac{1}{n}\|\mathbf{X}-\mathbf{DA}\|_F^2\}^{k_j+1})\leq\delta, \tag{5.8}$$

where δ is a threshold specified by the user or learned via cross-validation. Figure 5.4 shows an example of this decrease in reconstruction error for a single run of the algorithm on the Urban dataset. In this example, we experimentally observed that the error reduces to about $1/2$ very fast, and after about 6 atoms the reconstruction error stops changing much (the curve's slope reduces).[3] The algorithm for unsupervised mapping is summarized in Figure 5.5.

5.7 Unsupervised Source Separation Experiments

In this section we test and compare our proposed algorithm with simulated and real HSI datacubes.

[3] Recent works have started to address the automatic selection of critical parameters such as the number of atoms in the dictionary [96, 79]. Adapting these techniques to the model here proposed is the subject of future research.

Input: Hyperspectral scene $\mathbf{X}$, sparsity parameter λ_S, coherence parameter λ_G, subdictionary incoherence parameter λ_I, and stopping threshold δ.

Output: Sparse matrix of fractional abundances $\mathbf{A}$ for $\mathbf{x}_i, \quad i = 1, ...n$, and learned endmember dictionary $\mathbf{D}$.

Initialization

- Perform a Constrained Nonnegative Matrix Factorization (CNMF), $k_j = 1$ for all the classes j,

$$(\mathbf{A}^*, \mathbf{D}^*) = \text{argmin}_{\mathbf{A} \succeq 0, \mathbf{1}^T\mathbf{A} = \mathbf{1}^T, \mathbf{D} \succeq 0} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2.$$

Simultaneous Learning

While (5.8) is not satisfied:

- Set $k_j \leftarrow k_j + 1$ (for all the classes) and initialize new atoms $\mathbf{d}^j$ randomly.
- Set $\mathbf{X}^j$ as all data samples with largest abundance ℓ_1 norm corresponding to $\mathbf{D}^j$.
- Dictionary Update Stage:

$$\min_{\mathbf{D}^j \succeq 0} \sum_{i=1}^{n_j} \|\mathbf{x}_i^j - \mathbf{D}^j \mathbf{a}_i^j\|_2^2 + \lambda_S \sum_{i=1}^{n_j} \mathscr{S}(\mathbf{a}_i^j) + \lambda_G \sum_{i=1}^{n_j} \mathscr{G}(\mathbf{a}_i^j, \mathbf{w}_i) + \lambda_I \mathscr{I}(\mathbf{D}^j)$$

- Abundance Mapping Stage:

$$\min_{\mathbf{a}_i \succeq 0} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{D}\mathbf{a}_i\|_2^2 + \lambda_S \sum_{i=1}^{n} \mathscr{S}(\mathbf{a}_i) + \lambda_G \sum_{i=1}^{n} \mathscr{G}(\mathbf{a}_i, \mathbf{w}_i).$$

Figure 5.5: Algorithm for subpixel unsupervised classification in HSI.

5.7.1 Experiments with Simulated HSI Data

We perform a series of experiments to compare the performance of the proposed algorithm with recently developed unmixing schemes:

1. A fully constrained NMF (CNMF, see references above). This is a standard fully constrained Nonnegative Matrix Factorization, where the sum to one constraint in the abundance coefficients is strictly enforced. This will also serve as our initialization algorithm as explained above.

2. Vertex Component Analysis (VCA) [82]. This is a minimum volume simplex approach. This algorithm assumes that there are pure pixels in the image. This serves as the initialization algorithm for SISAL below.

3. Simplex Identification via Split Augmented Lagrangian (SISAL) [86]. It does not assume pixel purity in the data.

Both the VCA and SISAL algorithms come from the authors website. The code is publicly available at `http://www.lx.it.pt/~bioucas/code.htm`.

The simulated data are generated using the USGS spectral library available at `http://speclab.cr.usgs.gov/spectral-lib.html`, corresponding to the AVIRIS sensor, which consists of 500 mineral spectral signatures with 224 spectral bands. The "true" abundance values are generated following a Dirichlet probability density function (pdf) with density parameter of 0.1. This distribution guarantees nonnegativity and sum to one in the abundance coefficients. The dataset is generated with a total of $10,000$ observations, each pixel having a maximum purity of 0.8. No projections or dimension reduction techniques were applied to the data prior to running the experiments. The simulations test the algorithms using the following variants:

1. Noise was added to the dataset. The signal to noise ratio (SNR) is calculated as $\text{SNR} = 10 \cdot \log(\frac{\|\mathbf{X}\|_F^2}{\|\mathbf{N}\|_F^2})$, where $\mathbf{X}$ is the noiseless data. We include noise levels of 40, 30, 20, and 15 dB.

2. The number of sources was selected to be 3, 6, and 10.

Our measure of performance for this experiment is the mean squared error (MSE) between the original (ground truth) and computed abundance values, $\text{MSE}(\hat{\mathbf{A}}) = \frac{1}{n \cdot C}\|\mathbf{A} - \hat{\mathbf{A}}\|_F^2$. For all the experiments presented in this paper, parameters were optimized to obtain the best possible results for each tested algorithm. The parameters for the VCA algorithm were chosen to be the default values in the public domain code released by the authors. For SISAL, we chose the non-negativity enforcing parameter $\tau = 10$, which gave the best results in our simulations from the choices $\tau = [0.01, 0.1, 1, 10]$. For our algorithm we set the sparsity constraint as $\lambda_S = 0.1/\sqrt{b}$, selected best from the possible values $\lambda_S = [0.01/\sqrt{b}, 0.1/\sqrt{b}, 0.5/\sqrt{b}, 1/\sqrt{b}]$, $\lambda_G = 0$, and $\lambda_I = 0.005$, selected best from the possible values $\lambda_I = [0.001, 0.005, 0.01, 0.1]$, and a stopping parameter $\delta = 1 \times 10^{-4}$. Since the data generated enforce the sum of abundance coefficients to one, each estimated abundance vector in DM was normalized by its sum. Each experiment was run 50 times, each time generating a datacube by drawing at random spectral signatures from the spectral library. The results are summarized in Figure 5.6. We have tested only the proposed DM, and not DMS, since for these simulated data there is no spatial information.

We make the following observations:

1. VCA's performance is affected mainly due to the assumption regarding the presence of pure endmembers, specifically for low number of sources (3 and 6), and high SNRs. As the number of sources increases and the SNR decreases, VCA shows reduced sensitivity to noise as compared to SISAL, and seems more stable than SISAL, particularly when the number of sources is high.

2. The proposed DM method performs better under lower SNRs and as the number of sources is increased, when compared to SISAL and VCA. This is the case even when the algorithm is forced to have a fixed sparsity constraint. In addition, the data in this simulation assumes there is no spectral variability in the sources, while such variability will further favor our proposed richer modeling framework. In all cases, DM performs

better than CNMF. The average number of atoms per sub-dictionary used in these simulations resulted in 2.98 atoms.

3. We also experimented with an increase in the pixels' purity level to 0.95 to make VCA a more competitive approach (Figure 5.6, part d). As expected, VCA performs better with this setting, especially at high SNR. As the SNR decreases, the DM method attains a lower MSE.

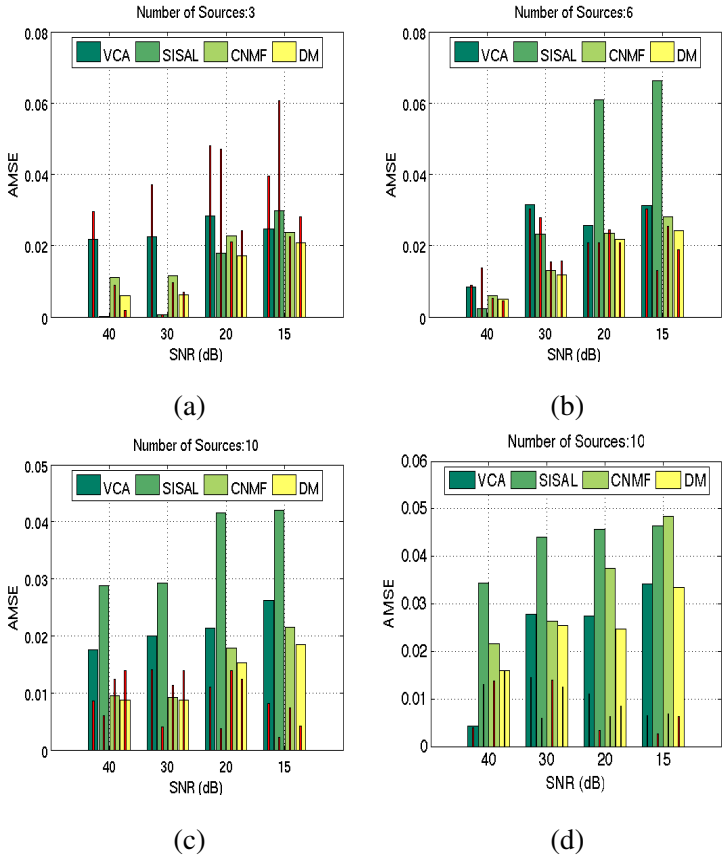


Figure 5.6: Tested algorithms' performance in terms of abundance MSE for different values of SNR (dB). (a) 3 sources, (b) 6 sources, (c) 10 sources, all with purity level of 0.8, and (d) 10 sources with purity level of 0.95. The green/yellow (thick) bars correspond to the average MSE, and the red (thin) bars to the error standard deviation. (This is a color figure.)

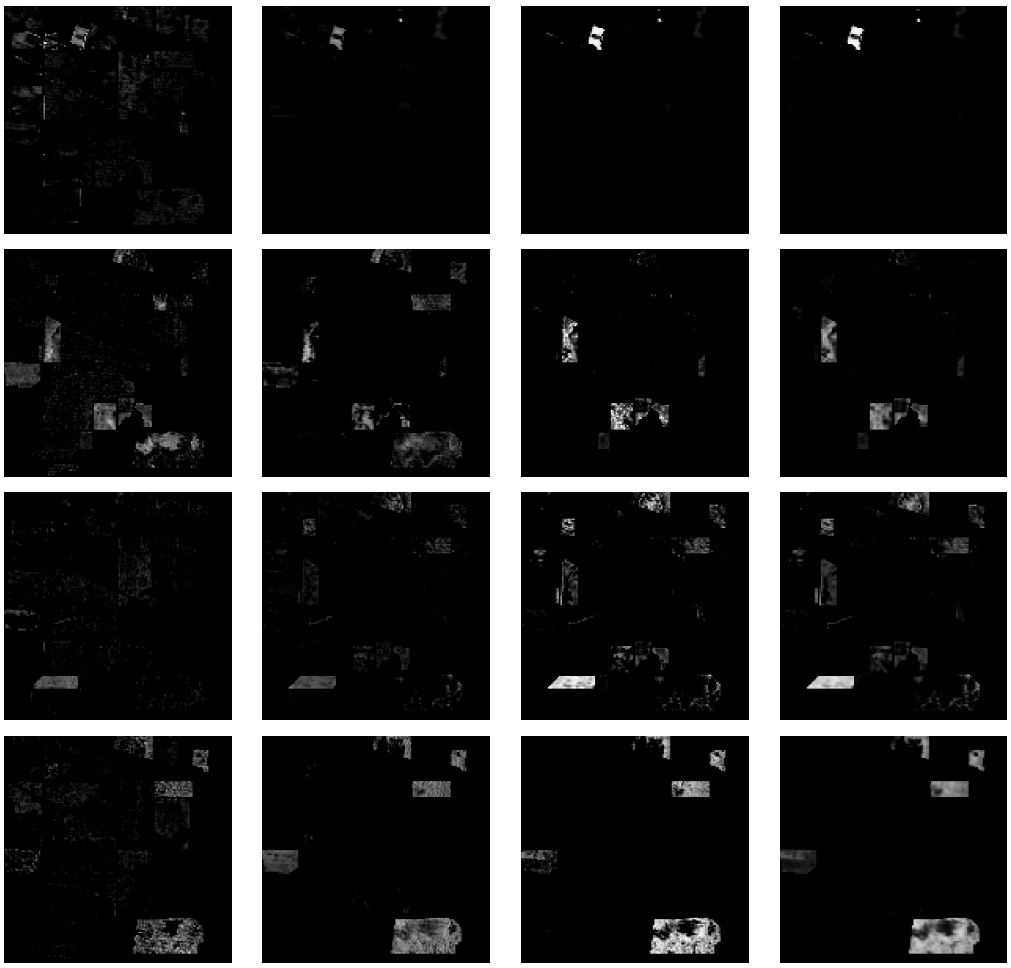
5.7.2 Experiments with Real HSI Data

We applied our proposed unsupervised algorithms DM and DMS to the three real HSI datacubes. For this experiment we show several abundance maps for each of the datacubes. We also included the results obtained by SISAL and CNMF (SISAL's parameter $\tau = 0$ gave the best result for all images). While both CNMF and SISAL are very recent and powerful algorithms developed for HSI unmixing, still, our proposed approach show advantages over these methods. For DM and DMS, we selected $\lambda_S = 0.5/\sqrt{b}$, $\lambda_G = 0.01$ (same values as those selected for the supervised experiments), $\lambda_I = 0.005$, and $\delta = 1 \times 10^{-6}$. For the Indian Pines dataset, we show the results for the estimated "Stone-steel towers," "Grass/Tress," "Wheat," and "Woods" classes. For the APHill dataset, we selected the estimated abundance values for "Road," "Coniferous" trees, and "Crop" classes. Finally, the three abundance maps shown for the Urban data are "Road," "Bright soil," and "Gray rooftop" classes. With the selected stopping criteria for DM and DMS, the algorithms stopped at 18 and 15 atoms, respectively, for the Indian Pines image; at 15 and 8 for the APHill image; and at 7 and 7 for the Urban image. Results are presented in figures (5.7), (5.8), and (5.9). These figures illustrate abundance maps obtained from the computed unmixing coefficients. Each pixel has a computed vector of coefficients (or abundances), and the presented mappings correspond to the values of the coefficients for a particular endmember. A dark pixel in the mapping means that the material is not present or has low presence, while a very bright pixel means that the material is very abundant. For SISAL and CNMF, the mappings correspond to the coefficient associated with each material (one spectra represents an endmember). For the proposed DM and DMS, the mapping corresponds to the ℓ_1 norm of the vector associated to the endmember sub-dictionary/class (ℓ_1 of the computed α^j). All the figures are scaled from 0 to 1.[4]

We make the following observations from these experiments:

1. For the Indian Pines image, all the algorithms provided similar results in correctly identifying the stone tower, but also estimated high abundance values in other areas, clearly

[4] In these experiments, the ℓ_1 norm of the abundance coefficients from the sub-dictionaries in DM and DMS resulted in values smaller or equal to 1.



(a) (b) (c) (d)

Figure 5.7: Abundance maps corresponding to three classes from the Indian Pines dataset. The first row corresponds to the ”Stone-steel towers” class. The second row corresponds to the “Grass/Trees” class, the third row corresponds to the “Wheat“ class, and the fourth row corresponds to the “Woods” class. Dark pixels correspond to regions where the class is not present, bright pixels correspond to regions where the class is present. All the abundance maps are scaled from 0 to 1. (a) SISAL (b) NMF (c) DM (d) DMS.

seen in the SISAL abundance map. Both DM and DMS produced a much "brighter" spatial region corresponding to that class. Looking at the "Grass/Trees" class, the DMS algorithm was the best at identifying the locations of the class. SISAL produced the "cleanest" result for the "Wheat" class abundance estimate, while CNMF, DM, and DMS incorrectly estimated "Wheat" at other spatial regions. Nevertheless, the brightest pixels in the DM and DMS algorithms correspond to the correct class. Similarly, brighter values were obtained by DM and DMS for the "Woods" class. Incorrect abundance estimates occurred at regions corresponding to the "Bldg-Grass-Tree-Drives" and "Grass/Pasture" classes, where there are intrinsic similarities between these classes.

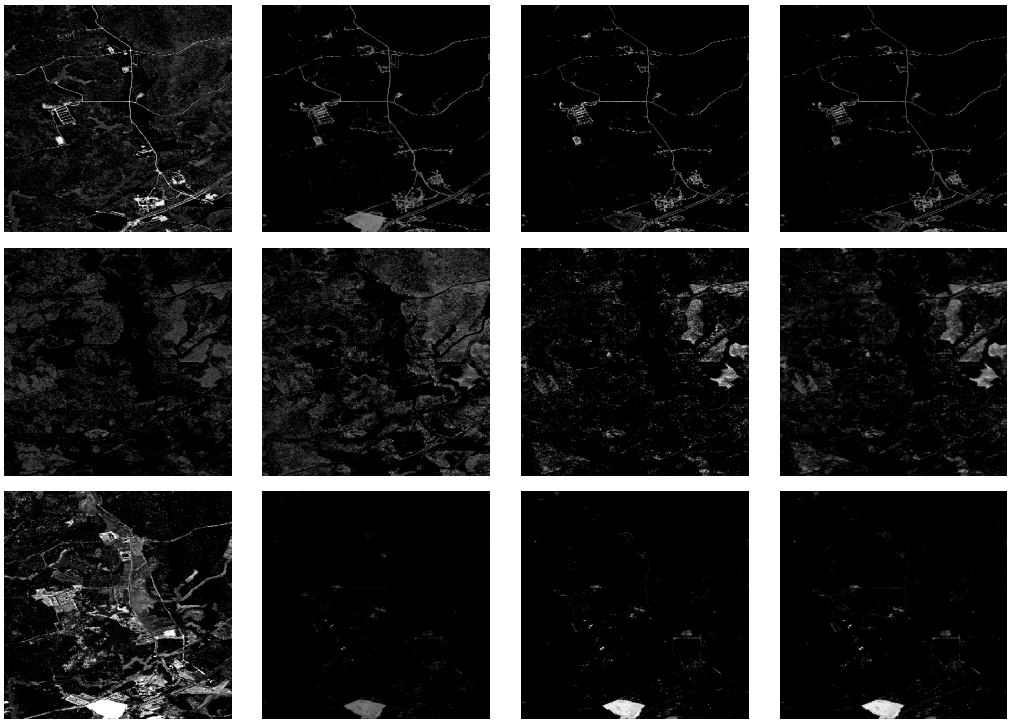
2. Considering the APHill results, we observe that both DM and DMS gave the cleanest results. For example, the road estimation of SISAL includes pixels pertaining to the lakes, some other wet areas, and vegetation. Observe that there are bright values corresponding to the "Crop" region in the "Road" abundance map, an effect that is also seen in the "Crop" abundance map. Also, the estimation of "Coniferous" trees appears much more concentrated in the appropriate regions for the DM and DMS abundance maps. SISAL and CNMF abundance values have lower intensities, and are more scattered over the vegetation regions. For the "Crop" abundance maps, SISAL estimates abundance values outside the region, mostly grass. CNMF, DM, and DMS estimates are similar, with higher abundance values in the "Crop" region for DM and DMS, and some other scattered areas, mostly grass. Again, fractional abundances corresponding to the "Crop" area are also seen in the "Road" abundance map for CNMF.

3. For the Urban image, we observe mixing effects occurring in the "Road" estimation of the SISAL algorithm, particularly with "Dark soil." Notice that the abundance values are increasingly brighter in the DM and DMS estimates as compared with both SISAL and CNMF. For the "Bright soil" estimates, there are high-valued pixels in the "Concrete" and "Road" regions in the CNMF estimates. Again, these effects are minimized in the DM and DMS estimates. For the "Brown rooftop" class, there is a high correlation in the

abundance estimates with "Road" and "Concrete" in SISAL, and with "Gray rooftop" in CNMF. This effect was attenuated by the DM and DMS algorithms. Again, this shows a clear advantage of a better class representation when using a learned structured dictionary instead of a single spectra.

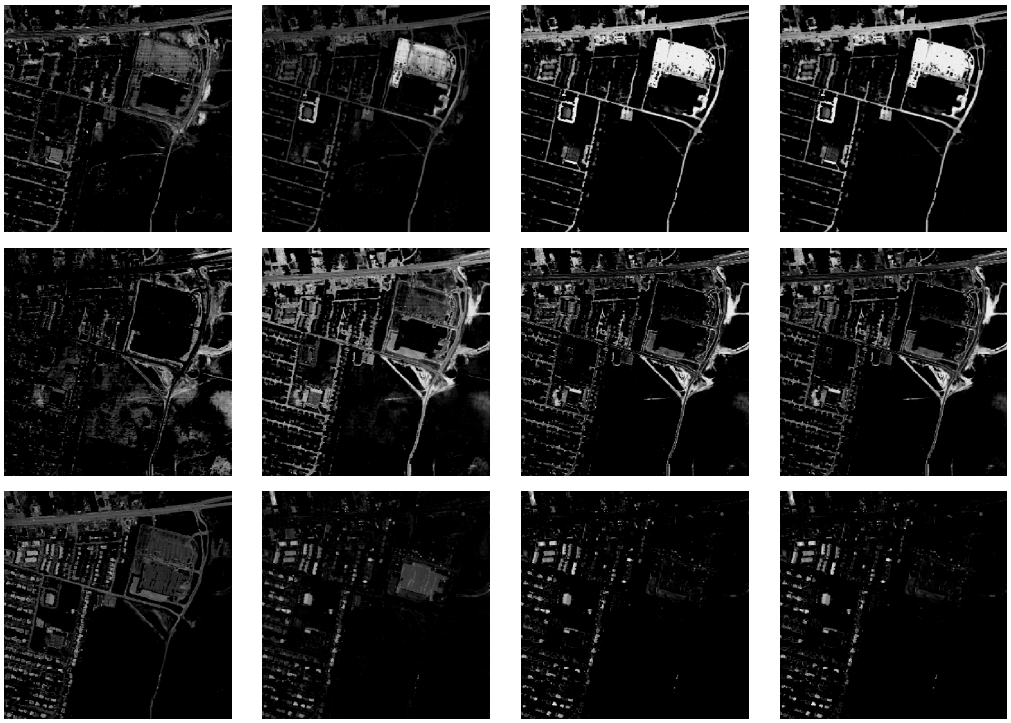
4. It is worthwhile mentioning that DM and DMS results are very similar to each other. Slight differences are observed due to the "smoothing" effects produced by the spatial/spectral neighbors associated with the DMS's grouping term. Although not explicitly observed in these results, there are advantages of this term in providing with additional robustness, as previously detailed in the supervised classification experiments.

Given that we have training and validation data from each of the tested HSI datasets, we also conducted a numerical (quantitative) experiment to test the classification accuracy of our proposed unsupervised method. Since we know the locations of the "known" materials, we mapped the estimated sources from the unsupervised case with the corresponding classes from the supervised case. Tables 5.9 and 5.10 summarize these results for APHill and Urban, respectively. There is a significant gain in classification accuracy when compared with the CNMF method (our initial condition), in particular for the "Road" class, for which most of the pixels pertaining to that class were confused with "Concrete" in the CNMF method. Similarly, most of the pixels corresponding to the "Grass" class are mislabeled as "Trees" using CNMF (see Figure 5.10). The only observed deterioration is for the confusion of some "Concrete" pixels with "Road" and "Gravel," three very similar classes. For the Urban data, we observe very low classification accuracies for the "Road" and "Concrete" classes in the CNMF results. This is because the "Bright soil" class was dominant at those pixels. This effect was mitigated with the proposed DM and DMS algorithms. We also observe a slightly lower accuracy for the "Trees" class. The confusion with the "Grass" class produced this effect. These are examples of the importance of a dictionary-based class representation and classification method, making the proposed DM/DMS a richer and more appropriate model to deal with high-dimensional HSI.



(a) (b) (c) (d)

Figure 5.8: Abundance maps corresponding to three classes from the APHill dataset. The first row corresponds to the "Road" class, the second row corresponds to the "Coniferous" class, and the third row corresponds to the "Crop" class. Dark pixels correspond to regions where the class is not present, bright pixels correspond to regions where the class is present. All the abundance maps are scaled from 0 to 1. (a) SISAL (b) NMF (c) DM (d) DMS.



(a) (b) (c) (d)

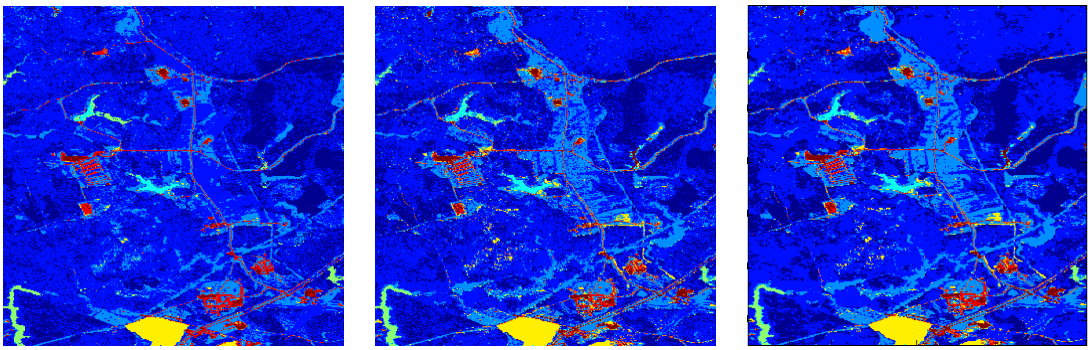
Figure 5.9: Abundance maps corresponding to three classes from the Urban dataset. The first row corresponds to the "Road" class, the second row corresponds to the "Bright soil" class, and the third row corresponds to the "Brown rooftop" class. Dark pixels correspond to regions where the class is not present, bright pixels correspond to regions where the class is present. All the abundance maps are scaled from 0 to 1. (a) SISAL (b) NMF (c) DM (d) DMS.

Table 5.9: Unsupervised per-class overall classification accuracy for the APHill dataset based on the training and validation data used in the supervised case. See Table 5.2 for the class labels.

Method	**Class label**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
CNMF	92.5523	63.6047	17.0688	**100.0000**	**100.0000**	**100.0000**	1.2146	**97.9798**	92.0000
DM	92.8870	78.9535	67.1894	**100.0000**	**100.0000**	**100.0000**	55.0607	62.6263	98.4000
DMS	**99.7490**	**91.7054**	**73.8239**	**100.0000**	**100.0000**	**100.0000**	**63.5628**	76.7677	**99.2000**

Table 5.10: Unsupervised per-class overall classification accuracy for the Urban dataset based on the training and validation data used in the supervised case. See Table 5.2 for the class labels.

Method	**Class label**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CNMF	3.9448	1.4563	**100.0000**	**100.0000**	95.8801	**89.6739**	**100.0000**	**93.7238**
DM	96.2451	85.3659	**100.0000**	**100.0000**	99.7528	88.5870	**100.0000**	86.6521
DMS	**96.2525**	**90.7767**	**100.0000**	**100.0000**	**100.0000**	**89.6739**	**100.0000**	87.6356



(a) (b) (c)

Figure 5.10: Performance of the unsupervised algorithms in terms of classification accuracy. Each color represents an estimated material matching with the training and validation samples' coordinates (known *a priori*). See Table 5.2 and Figure 5.1 for the color code and class labels. (a) CNMF, (b) DM, and (c) DMS. (This is a color figure.)

5.8 Conclusion

We have proposed supervised and unsupervised HSI composition mapping algorithms using learned sparse models. We reported the results on three hyperspectral datasets, and also showed the potential for a Bayesian compressed-sensing technique to help in solving acquisition, transmission, and storage issues related to HSI. With this framework, noise, class variability, and data redundancy are efficiently addressed by a structured sparse modeling classification technique that incorporates spatial/spectral regularization and class-dictionary incoherence.

While we currently increase the number of atoms uniformly for all sub-dictionaries in the unsupervised scenario, it will be interesting to do this class-dependent, letting different classes to learn different dictionary sizes. We plan to further exploit the proposed framework by considering adaptive sampling techniques, where sub-sampling is avoided in areas of limited spatial coverage or high spectral uncertainty. We are also further exploring the capabilities of the proposed framework to deal with the scenario where entire spectral bands or regions are missing, the preliminary results here reported are very encouraging. Lastly, we are investigating the possibility to classify directly from the sub-sampled data, without the need for pre-reconstruction, which is very important for real time applications [97].

Chapter 6

Sensor fusion and applications to spectral mapping

6.1 Chapter summary

Several studies suggest that the use of geometric features along with spectral information improves the classification and visualization quality of hyperspectral imagery. These studies normally make use of spatial neighborhoods of hyperspectral pixels for extracting these geometric features. In this chapter, we propose to merge point cloud Light Detection and Ranging (LiDAR) data and hyperspectral imagery (HSI) into a single sparse modeling pipeline for subpixel mapping and classification. The model accounts for material variability and noise by using learned dictionaries that act as spectral endmembers. Additionally, the estimated abundances are influenced by the LiDAR point cloud density, particularly helpful in spectral mixtures involving partial occlusions and illumination changes caused by elevation differences. We demonstrate the advantages of the proposed algorithm with co-registered LiDAR-HSI data.

6.2 Introduction

Consider a region in a scene where there are trees partially occluding a road. These elevation differences cause single pixels to have energy reflected from both the tree leaves and the road, and are also affected by shade. This problem motivates the use of additional information sources that could potentially mitigate these effects. LiDAR point cloud data provides precise range information on a three dimensional space. In particular, LiDAR acquires one or multiple elevation measurements per single (discretized) ground planar coordinates. When these point cloud data are co-registered with the hyperspectral scene, it gives insight into identifying structural change including partial occlusions within a spectral pixel. This advantage has motivated several works to use LiDAR and HSI for improved classification. For example, in [98], the authors used depth information from LiDAR as part of the parametrization required for a bio-optical model to perform underwater benthic mapping. In [99], the authors studied the possible correlations of the surface roughness and minerals' spectral content. In [100], LiDAR information was used to better localize small targets by first performing a background/foreground segmentation on the elevation map, and then using regions of interest based on height for improved small target detection. It has also been applied for obtaining higher discrimination between savanna tree species by the use of hand crafted decision trees [101, 102, 103]. With the exception of [98], these works require careful hand tuning of decision operations, and require a sequential processing of LiDAR and HSI. Our framework drifts away from these approaches because the model *simultaneously* uses information from both data sources to estimate the pixels labels, and exploits the data redundancy available from HSI and LiDAR to address the modeling and processing challenges of high-dimensional visualization and classification. The HSI cube is expressed as a sparse linear combination of learned sources (dictionary atoms), giving meaningful material abundance estimates, without explicit dimension reduction or subspace projection preprocessing steps. We impose spatial coherence in the sparse modeling-based classification. This efficiently fuses spectral (HSI) and structural (LiDAR) information by incorporating local and nonlocal connectivities between local regions in the scene, leading to a grouping criteria that

induces a robust and stable abundance mapping.

6.3 Modeling HSI

We employ the same modeling strategy for HSI presented in Chapter 5, and we repeat it here for convenience. Assuming there are C materials, this model aims to learn a block-structured dictionary of materials, where the $j-th$ block is representative of the $j-th$ material, $j \in [1,...,C]$. Learning each material subdictionary can be summarized as to solve the following bi-convex optimization problem

$$\begin{aligned}(\mathbf{D}^{j*},\mathbf{A}^{j*}) &= \arg\min_{(\mathbf{D}^j,\mathbf{A}^j)\succeq 0}\{\frac{1}{2}\|\mathbf{D}^j\mathbf{A}^j-\mathbf{X}^j\|_F^2 \\ &\quad +\lambda\sum_{i=1}^{n_j}\|\mathbf{a}_i^j\|_1\} \\ &= \arg\min_{(\mathbf{D}^j,\mathbf{A}^j)\succeq 0}\mathscr{H}(\mathbf{X}^j;\mathbf{D}^j,\mathbf{A}^j) \end{aligned} \tag{6.1}$$

where $\mathbf{X}^j$ are the n_j pixels pertaining to the $j-th$ class, and $\lambda \geq 0$ is a parameter that controls the trade-off between reconstruction quality and sparsity. After learning the material dictionaries in a separate fashion, the structured dictionary $\mathbf{D} = [\mathbf{D}^1,...,\mathbf{D}^C]$ is assembled and used for solving for the corresponding abundance coefficients coming from a linear combination of atoms from $\mathbf{D}$.

6.4 HSI-LiDAR fusion

Up to this point, each pixel is treated independently from each other. To exploit the structural information available from LiDAR data in the scene, one can enforce the estimation of the abundance coefficients $\mathbf{A}$ to be influenced by the geometry of the point cloud data, introducing spatial and spectral coherence in the modeling and coding process. This coherence will depend both on the pixels' spectral shape *and* the geometry of LiDAR data. Let $\mathscr{F}$ be a grouping

(coupling) term on the coefficients,

$$\mathscr{F}(\mathbf{M}, \mathbf{w}_i; \mathbf{a}_i) = \|(\mathbf{M}\mathbf{a}_i - \sum_{l \in \eta} w_{il}\mathbf{M}\mathbf{a}_l)\|_1, \tag{6.2}$$

where η denotes a predefined neighborhood associated to the $i-th$ pixel. $\mathscr{F}$ will highly depend on the weighting function w_{il}. An example of such a function is

$$w_{il} = \frac{1}{Z_i} e^{-(\frac{\|\mathbf{x}_i - \mathbf{x}_l\|_2^2}{\sigma_s^2})} + \frac{1}{X_i} e^{-(\frac{\|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_l\|_2^2}{\sigma_r^2})}, \tag{6.3}$$

where Z_i and C_i are pixel-dependent normalization constants, such that $\sum_{l \in \eta} w_{il} = 1$, $\hat{\mathbf{r}}$ is a spatial window (patch) around each of the concatenated LiDAR range and intensity samples, obtained from the minimum elevation and average intensity per discrete spatial coordinate, and σ_s^2, σ_r^2 are density parameters for the spectral and range content, respectively, controlling the width of the weighting function (here set to be the average of the data's pairwise Euclidean distance, either local for each pixel or global for the whole data). $\mathbf{M}$ is the sum operator as previously defined in Chapter 5. This weighting function is close to 1 if the both the hyperspectral pixels and the LiDAR local coordinates are homogeneous, and 0 otherwise.

Finally, we can summarize our proposed approach as to solving the following optimization problem

$$\mathbf{A}^* = \underset{\mathbf{A} \geq 0}{\operatorname{argmin}} \quad \{\mathscr{H}(\mathbf{D}, \mathbf{X}; \mathbf{A}) + \beta \sum_{i=1}^{n} \mathscr{F}(\mathbf{M}, \mathbf{w}_i; \mathbf{a}_i)\}, \tag{6.4}$$

where $\beta \geq 0$ is a parameter controlling the amount of interaction between LiDAR and HSI. Notice that $\mathscr{F}(\mathbf{M}, \mathbf{w}_i; \mathbf{A}_i)$ introduces a variable coupling. We efficiently solve this using the Split Bregman method[104]. First, we reformulate Equation 6.4 as

$$\begin{aligned} \min \mathscr{H}(\mathbf{X}, \mathbf{D}; \mathbf{A}) + \sum_{i=1}^{n} \|\mathbf{v}_i\|_1 + \sum_{i=1}^{n} \|\mathbf{u}_i\|_1 \\ \text{s.t.} \quad \mathbf{v}_i = \mathbf{a}_i, \mathbf{u}_i = \mathbf{M}\mathbf{a}_i - \sum_{l \in \eta} w_{i,l}\mathbf{M}\mathbf{a}_l. \end{aligned} \tag{6.5}$$

Second, the constraints are enforced by applying an augmented Lagrangian formulation:

$$\begin{aligned}\min \mathscr{H}(\mathbf{X},\mathbf{D};\mathbf{A}) + \sum_{i=1}^{n} \|\mathbf{v}_i\|_1 + \sum_{i=1}^{n} \|\mathbf{u}_i\|_1 + \lambda \sum_{i=1}^{n} < \mathbf{b}_i, \mathbf{a}_i - \mathbf{v}_i > \\ + \beta \sum_{i=1}^{n} < \mathbf{c}, \mathbf{M}\mathbf{a}_i - \sum_{l\in\eta} w_{i,l}\mathbf{M}\mathbf{a}_l - \mathbf{u}_i > + \sum_{i=1}^{n} \frac{\lambda}{2} \|\mathbf{A}_i - \mathbf{v}_i\|_2^2 \\ + \sum_{i=1}^{n} \frac{\beta}{2} \|\mathbf{M}\mathbf{a}_i - \sum_{l\in\eta} w_{i,l}\mathbf{M}\mathbf{a}_l - \mathbf{u}_i\|_2^2, \end{aligned} \tag{6.6}$$

where $\mathbf{g} = \sum_{l\in\eta} w_l \mathbf{M}\mathbf{a}_l$, and we maximize for the dual variables $\mathbf{b}$ and $\mathbf{c}$, and minimize for $\mathbf{a}, \mathbf{v}$, and $\mathbf{u}$. Finally, the proposed abundance mapping algorithm is reduced to solving the following subproblems independently:

$$\begin{aligned}
\mathbf{a}^{t+1} &= \underset{\mathbf{a}\succeq 0}{\operatorname{argmin}} \{ \frac{1}{2}\|\mathbf{D}\mathbf{a} - \mathbf{x}^t\|_2^2 + \frac{\lambda}{2}\|\mathbf{a}^t - \mathbf{v}^t + \mathbf{b}^t\|_2^2 \\
&\quad + \frac{\beta}{2}\|\mathbf{M}\mathbf{a}^t - \mathbf{g}^t + \mathbf{c}^t\|_2^2 \}, && (6.7) \\
\mathbf{v}^{t+1} &= \underset{\mathbf{v}\succeq 0}{\operatorname{argmin}} \{ \|\mathbf{v}^t\|_1 + \frac{\lambda}{2}\|\mathbf{a}^{t+1} - \mathbf{v}^t + \mathbf{b}^t\|_2^2 \}, && (6.8) \\
\mathbf{u}^{t+1} &= \underset{\mathbf{u}\succeq 0}{\operatorname{argmin}} \{ \|\mathbf{u}^t\|_1 + \frac{\beta}{2}\|\mathbf{M}\mathbf{a}^{t+1} - \mathbf{g}^{t+1} + \mathbf{c}^t\|_2^2 \}, && (6.9) \\
\mathbf{b}^{t+1} &= \mathbf{b}^t - \mathbf{v}^{t+1} + \mathbf{a}^{t+1}, && (6.10) \\
\mathbf{c}^{t+1} &= \mathbf{c}^t - \mathbf{g}^{t+1} + \mathbf{M}\mathbf{a}^{t+1} - \mathbf{u}^{t+1}. && (6.11)
\end{aligned}$$

These subproblems are solved until convergence in the ℓ_2-norm of $\mathbf{a}$, which takes about 50 iterations in our experiments. Note that the subproblems can be solved simply via inversion (Equation (6.7)), shrinking (equations (6.8) and (6.9)), and explicitly (equations (6.10) and (6.11)), see [104] for more details. [1] This concludes the subpixel modeling procedure. Full-pixel labeling derives directly by selecting the $i-th$ pixel's label corresponding to the maximum element of $\mathbf{M}\mathbf{a}_i$. We now proceed with experiments supporting our model.

[1] The nonnegativity constraint is enforced by projecting into nonnegative numbers.

6.5 Experiments

In this section, we validate our model by applying it to co-registered HSI and LiDAR data. This dataset consists of an airborne data collection over Gulfport, Mississippi, in November, 2010. The scene is composed of low density urban and coastal regions. The HSI data was acquired with a CASI-1500 sensor, with a spectral range of 375-1050 nm in 72 bands. The LiDAR data was acquired with an Optech ALTM Gemini sensor, operating at a wavelength of 1064 nm. These data are co-registered at 1 m spatial resolution, with a total of 324×500 pixels. We analyze two scenarios: supervised and unsupervised mapping. On both scenarios, we compare the mapping results with and without LiDAR information, hence highlighting the benefits of the proposed joint modeling scheme. We selected $\lambda = \frac{0.5}{\sqrt{b}}$, and $\alpha = 0.9$. The neighborhood η for each pixel patch of 3×3,[2] was composed using 4 spatially connected overlapping patches and the 4 most similar patches across the entire image spatial domain. In Figure 6.1, we show false color composites from the scene for the LiDAR and HSI data, respectively.

(a) (b) (c)

Figure 6.1: Fusion of HSI and LiDAR data from the Gulfport scene. (a) Depth-intensity-average map from LiDAR. (b) False color RGB from hyperspectral scene. (c) False color RGB from HSI-LiDAR fused scene. (This is a color figure.)

On a supervised setting, we used *a priori* averaged spectra from 11 materials; specifically, the sample means of 11 regions of interest extracted from the scene. These materials are labeled: *C1: canvas, C2: fabric #1, C3: fabric #2, C4: trees, C5: healthy grass, C6: grounds, C7:*

[2] Spatial patches were used for the LiDAR depth and intensity data. Single pixels were used for HSI data.

asphalt, C8: red roof, C9: brown roof, C10: tan roof, and *C11: sand*. These spectra served as the dictionary **D**. We processed the data by using the proposed mapping algorithm for $\beta = 0$ and $\beta = \frac{0.1}{\sqrt{b}}$, that is, with and without fusion. Subfigures 6.2(a) and 6.2(b) show these full-pixel mappings. Notice how the estimates are smoother in Subfigure 6.2(b), for instance, a more homogeneous region around the red building on the lower left of the image. Also, there are *grounds* pixels that are incorrectly labeled as *concrete*, and are correctly labeled by activating the fusion term in the proposed model.

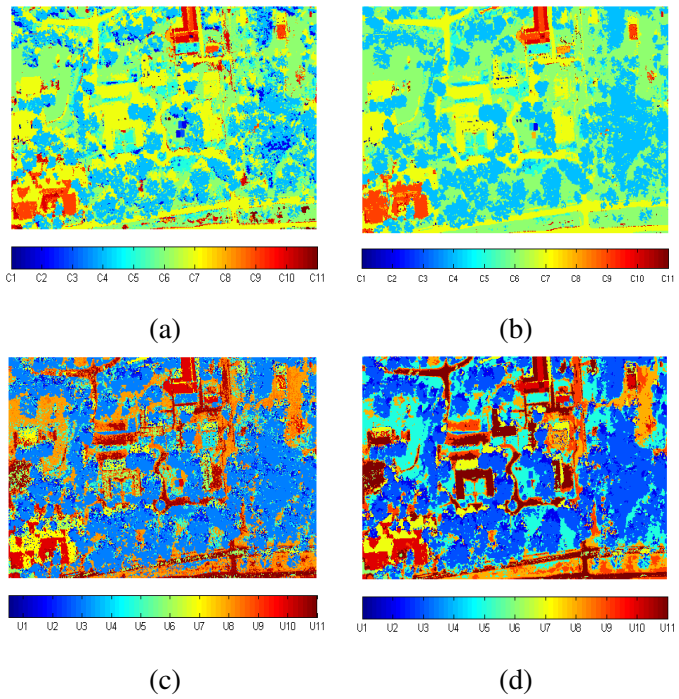


Figure 6.2: (a) Supervised spectral mapping, no fusion. (b) Supervised spectral spectral mapping with fusion. (c) Unsupervised spectral mapping, no fusion. (d) Unsupervised spectral mapping with fusion. (This is a color figure.)

On an unsupervised setting, we followed the endmember learning procedure from Chapter 5. Basically, **D** is initialized using a single estimated spectra for each of the C materials using a nonnegative matrix factorization, where the abundance coefficients are constrained to sum to

one, and continues adding atoms to each subdictionary $\mathbf{D}^j$ until the change in reconstruction error reaches 1×10^{-4}. We applied the proposed fusion algorithm after learning $\mathbf{D}$ with $C = 11$ materials (labeled as U's in Figure 6.2). In Figure 6.3, we show a spectral sample reconstructed using the proposed model corresponding to a small tree under the shade from a taller building. This sample is compared with the original HSI sample, and the average (supervised) spectra from the *trees* class. The fused spectra shows a higher amplitude in the channels corresponding to green and red wavelength. This is due to the collaboration effect of the proposed model, enforcing homogeneous regions from LiDAR and HSI to have similar abundance values. Subfigures 6.2(c) and 6.2(d) illustrate the full-pixel mappings with and without fusion. Again, we observe a smoother mapping in Subfigure 6.2(d). Finally, in Subfigure 6.1(c), we illustrate the false color composite after applying the proposed model (in an unsupervised manner). Notice how the effect of shading caused by the sun in the HSI scene is significantly alleviated in the new representation.

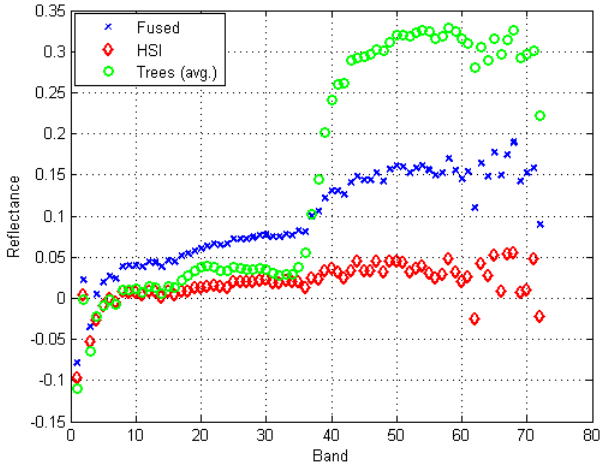


Figure 6.3: Sample spectra illustrating the influence of LiDAR data into spectral estimation, pixel (162,160). (This is a color figure.)

6.6 Conclusion

We presented a sparse modeling algorithm for source separation and classification using hyperspectral imagery and LiDAR. The range information from LiDAR data provides rich structural information, and is used to enhance the classification performance in HSI. An affinity function that combines the spectral information along with the spatial information in LiDAR is incorporated in the model to promote collaboration between the two data sources. The proposed unsupervised algorithm learns a structured dictionary representing the spectral sources/endmembers, and expresses each pixel as a sparse linear combination of the dictionary atoms. These coefficients provide information for spectral abundance mapping and classification. We performed experiments using real HSI/LiDAR data illustrating the advantages of multimodal information for remote sensing applications. In particular, we showed how using this model alleviates the effects of partial occlusions caused by elevation differences and shading.

Chapter 7

Conclusions

In this thesis, we designed and implemented structured sparse models to solve several high dimensional data classification tasks. We summarize our contributions as follows:

Fast, accurate action recognition: We showed that the problem of action classification can be interpreted as a "mixture" of partial body movements. Even when using a single and simple feature, i.e., the temporal gradient of the video, the proposed structured sparse model provides sufficient discriminative power, and a simple classification rule emerging directly from the sparse codes is sufficient for highly accurate classification. While most of the previously proposed action recognition schemes can take up to days for learning the actions of interest from a large database, our proposed method performs very fast. We also showed how to exploit inter-class relationships directly from the sparse codes, and by re-modeling these on a deep learning layer, we increased the discriminative power at a very low computation cost.

Automatic single video analysis: We developed a framework to perform activity based analysis from a single instance of motion imagery. We showed that activity dictionaries learned from

each individual in the scene gives us sufficient information for analyzing the spatio-temporal dynamics in the video, including individual action changes, and group activities.

Robust subpixel mapping: We developed a unified framework for modeling, source separation, and mapping of remotely sensed hyperspectral imagery, where we improved upon standard unmixing algorithms to account for intra-class spectral variability and noise. We also showed that the sparse codes emerging from this model are physically meaningful, and also serve as an accurate tool for full-pixel mapping and classification. In addition, we showed that highly accurate spectral mappings can be attained even when only acquiring a small fraction of the spectral measurements. This demonstrates the ability of the method to properly model and exploit the inherent structure and redundancy of the data, opening doors for new sensing modalities with much faster transmission rates.

Sensor fusion: We proposed a sparse modeling scheme for sensor fusion. This scheme consists in explicitly enforcing the structural homogeneity from the multimodal data, creating a collaborative encoding scheme. This procedure demonstrated to be very effective at simultaneously combining information from active (LiDAR) and passive (hyperspectral) imaging systems, hence drastically enhancing the visualization quality and mapping capabilities.

References

[1] A. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[2] M.A. Ranzato, Y.L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *NIPS*, 2007.

[3] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[4] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[6] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*, 2000.

[7] R. Blake and M. Shiffrar. Perception of human motion. *Annual Review of Psychology*, 58(1):47–73, 2007.

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, pages 1033–1040, 2008.

[9] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2nd Joint IEEE International Workshop on*, pages 65–72, 2005.

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[11] G. Willems, T. Tuytelaars, and L. van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages II: 650–663, 2008.

[12] J. Gall, A. Yao, N. Razavi, L. van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition (accepted for publication). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011.

[13] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng. Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[14] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, Apr. 2011.

[15] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, 2007.

[16] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[17] A. Klser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[18] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, pages 492–497, 2009.

[19] N. Dalal and B. Triggs. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[20] T. Guha and R.K. Ward. Learning sparse representations for human action recognition. *IEEE Trans Pattern Anal Mach Intell*, 34(8):1576–1588, 2012.

[21] L. Shao and R. Mattivi. Feature detector and descriptor evaluation in human action recognition. In *CIVR*, pages 477–484, 2010.

[22] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[23] M. Pietikinen and T. Ojala. Texture analysis in industrial applications. In J.L.C. Sanz, editor, *Image Technology - Advances in Image Processing, Multimedia and Machine Vision*, pages 337–359. Springer-Verlag, 1996.

[24] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):915–928, June 2007.

[25] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[26] L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.

[27] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, volume 1, pages 984–989, 2005.

[28] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[29] A. Basharat and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, pages 1–8, 2007.

[30] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[31] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2008.

[32] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508, 2010.

[33] A. Castrodad, Z. Xing, J.B. Greer, E. Bosch, L. Carin, and G. Sapiro. Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(11):4263–4281, 2011.

[34] A.S. Charles, B.A. Olshausen, and C.J. Rozell. Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 2011.

[35] C. Cadieu and B. A. Olshausen. Learning transformational invariants from natural movies. In *NIPS*, pages 209–216, 2008.

[36] T. Dean, R. Washington, and G. Corrado. Recursive sparse, spatiotemporal coding. In *ISM*, pages 645–650, 2009.

[37] K. Guo, P. Ishwar, and J. Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. In *AVSS*, pages 188–195, 2010.

[38] G.W. Taylor, R. Fergus, Y.L. Le Cun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, pages VI: 140–153, 2010.

[39] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8, 2007.

[40] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP*, 2010.

[41] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, pages 32–36, 2004.

[42] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, pages 2046–2053, 2010.

[43] C.C. Chen, M. S. Ryoo, and J. K. Aggarwal. UT-Tower Dataset: Aerial View Activity Classification Challenge. http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html, 2010.

[44] R. Vezzani, B. Davide, and R. Cucchiara. HMM based action recognition with projection histogram features. In *ICPR*, pages 286–293, 2010.

[45] M.S. Ryoo, C.C. Chen, J.K. Aggarwal, and A. R. Chowdhury. An overview of contest on semantic description of human activities (sdha) 2010. In *ICPR-Contests*, pages 270–285, 2010.

[46] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[47] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[48] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *ECCV*, pages 494–507, 2010.

[49] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, pages 399–406, 2010.

[50] Z.J. Xiang, H. Xu, and P.J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS*, pages 900–908. 2011.

[51] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009.

[52] N. Papadakis and A. Bugeau. Tracking with occlusions via graph cuts. *IEEE Trans Pattern Anal Mach Intell*, 33(1):144–157, 2011.

[53] S. Khamis, V.I. Morariu, and L.S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *ECCV*, 2012.

[54] S. Todorovic. Human activities as stochastic kronecker graphs. In *ECCV*, 2012.

[55] Y. Fu, T. Hospedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *ECCV*, 2012.

[56] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012.

[57] T. Hassner and L. Wolf. The action similarity labeling challenge. *IEEE Trans Pattern Anal Mach Intell*, 34(3):615–621, 2012.

[58] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int J Comput Vis*, 79(3):299–318, 2008.

[59] S. Savarese, A. DelPozo, J.C. Niebles, and L. Fei-Fei. Spatial-Temporal correlatons for unsupervised action classification. In *WVMC*, pages 1–8, 2008.

[60] A. Willem, V. Madasu, and W. Boles. Adaptive unsupervised learning of human actions. In *ICDP*, pages 1–6, 2009.

[61] E. Shechtman and M. Irani. Space-time behavior based correlation - OR - how to tell if two underlying motion fields are similar without computing them? *IEEE Trans Pattern Anal Mach Intell*, 29(11):2045–2056, 2007.

[62] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *IEEE Trans Pattern Anal Mach Intell*, 28(9):1530–1535, 2006.

[63] H. Xu Z. J. Xiang and P. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS*, 2011.

[64] A. Bronstein, P. Sprechmann, and G. Sapiro. Learning efficient structured sparse models. In *ICML*, 2012.

[65] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *NIPS*, pages 1601–1608, 2004.

[66] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[67] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP*, 2010.

[68] Z. Guo and S. Osher. Template matching via l1 minimization and its application to hyperspectral target detection. Technical Report 09-103, UCLA, www.math.ucla.edu/applied/cam/, 2009.

[69] N. Keshava and J.F. Mustard. Spectral unmixing. *Signal Processing Magazine, IEEE*, 19(1):44–57, 2002.

[70] Z. Guo, T. Wittman, and S. Osher. L1 unmixing and its applications to hyperspectral image enhancement. Technical Report 09-30, UCLA, www.math.ucla.edu/applied/cam/, 2009.

[71] C.A. Bateson, G.P. Asner, and C.A. Wessman. Endmember bundles: a new approach to incorporating endmember variability into spectral mixture analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(2):1083–1094, 2000.

[72] M. Velez-Reyes and S. Rosario. Solving adundance estimation in hyperspectral unmixing as a least distance problem. In *IGARSS*, volume 5, pages 3276–3278, 2004.

[73] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65, 2005.

[74] G. Camps-Valls, T. Bandos, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45:2044–3054, 2007.

[75] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[76] J. Munoz-Marf, L. Bruzzone, and G. Camps-Vails. A support vector domain description approach to supervised classification of remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(8):2683 –2692, aug. 2007.

[77] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[78] G. Camps-Valls, N. Shervashidze, and K. M. Borgwardt. Spatio-spectral remote sensing image classification with graph kernels. *Geoscience and Remote Sensing Letters, IEEE*, 7(4):741–745, October 2010.

[79] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.

[80] M.D. Craig. Minimum-volume transforms for remotely sensed data. *Geoscience and Remote Sensing, IEEE Transactions on*, 32(3):542–552, 1994.

[81] E.M. Winter and M.E. Winter. Autonomous hyperspectral end-member determination methods. *Sensors, Systems, and Next-Generation Satellites III*, 3870(1):150–158, 1999.

[82] J.M.P. Nascimento and J.M. Bioucas-Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(4):898–910, 2005.

[83] J.W. Boardman, F.A. Kruse, and R.O. Green. Mapping target signatures via partial unmixing of aviris data. In *Summaries of JPL Air-borne Earth Science Workshop*, 1995.

[84] J.M.P. Nascimento and J.M. Bioucas-Dias. Dependent component analysis: A hyperspectral unmixing algorithm. In *IbPRIA (2)*, pages 612–619, 2007.

[85] J. Li and J.M. Bioucas-Dias. Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data. In *IGARSS*, volume 3, pages III–250–III–253, jul. 2008.

[86] J.M. Bioucas-Dias. A variable splitting augmented lagrangian approach to linear spectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS '09. First Workshop on*, pages 1 –4, aug. 2009.

[87] Y.M. Masalmah and M. Velez-Reyes. Unsupervised unmixing of hyperspectral imagery. In *MWSCAS*, volume 2, pages 337–341, 2006.

[88] M. W. Berry, M. Browne, A.N. Langville, V. P. Pauca, and R.J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. In *Computational Statistics and Data Analysis*, pages 155–173, 2006.

[89] D.C. Heinz and Chein-I-Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(3):529–545, 2001.

[90] P. O. Hoyer and P. Dayan. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[91] A. Zare and P. Gader. Sparsity promoting iterated constrained endmember detection with integrated band selection. In *IGARSS*, pages 4045–4048, jul. 2007.

[92] S. Jia and Y. Qian. Spectral and spatial complexity-based hyperspectral unmixing. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(12):3867–3879, 2007.

[93] L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(3):765–777, 2007.

[94] A. Huck, M. Guillaume, and J. Blanc-Talon. Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(6):2590–2602, 2010.

[95] I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *The Third International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Aruba*, 2009.

[96] I. Ramirez and G. Sapiro. Sparse coding and dictionary learning based on the mdl principle. *arxiv.org*, http://arxiv.org/abs/1010.4751, 2010.

[97] Q. Du and R. Nekovei. Fast real-time onboard processing of hyperspectral imagery for detection and classification. *J. Real-Time Image Processing*, 4:273–286, 2009.

[98] M.C. Torres-Madronero, M. Velez-Reyes, and J.A. Goodman. Fusion of hyperspectral imagery and bathymetry information for inversion of bioptical models. In *SPIE, Remote Sensing of the Ocean, Sea Ice, and Large Water Regions*, 2009.

[99] M. S. West and R. G. Resmini. Hyperspectral imagery and LiDAR for geological analysis of Cuprite, Nevada. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7334, 2009.

[100] A. V. Kanaev, B. J. Daniel, J. G. Neumann, A. M. Kim, and K. R. Lee. Object level HSI-LIDAR data fusion for automated detection of difficult targets. *Opt. Express*, 19(21):20916–20929, 2011.

[101] M.A. Cho, L. Naidoo, R. Mathieu, and G.P. Asner. Mapping savanna tree species using carnegie airborne observatory hyperspectral data resampled to worldview-2 multispectral configuration. In *34th International Symposium on Remote Sensing of Environment*, 2011.

[102] L. Naidoo, M.A. Cho, R. Mathieu, and G.P. Asner. Spectral classification of savanna tree species, in the greater kruger national park region using carnegie airborne observatory

(cao) integrated hyperspectral and lidar data. In *1st AfricaGEO Conference, Cape Town*, 2011.

[103] D. Sarrazin, J.A. van Aardt, D.W. Messinger, and G.P. Asner. Fusing waveform lidar and hyperspectral data for species-level structural assessment in savanna ecosystems. In *SPIE, Laser Radar Technology and Applications XV*, 2010.

[104] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.