ROBUST SUBSPACE ESTIMATION USING LOW-RANK OPTIMIZATION. THEORY AND APPLICATIONS IN SCENE RECONSTRUCTION, VIDEO DENOISING, AND ACTIVITY RECOGNITION.

by

OMAR OREIFEJ
B.S. University of Jordan, 2006
M.S. University of Central Florida, 2009

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2013

Major Professor: Mubarak Shah

ABSTRACT

In this dissertation, we discuss the problem of robust linear subspace estimation using low-rank optimization and propose three formulations of it. We demonstrate how these formulations can be used to solve fundamental computer vision problems, and provide superior performance in terms of accuracy and running time.

Consider a set of observations extracted from images (such as pixel gray values, local features, trajectories . . . etc). If the assumption that these observations are drawn from a liner subspace (or can be linearly approximated) is valid, then the goal is to represent each observation as a linear combination of a compact basis, while maintaining a minimal reconstruction error. One of the earliest, yet most popular, approaches to achieve that is Principal Component Analysis (PCA). However, PCA can only handle Gaussian noise, and thus suffers when the observations are contaminated with gross and sparse outliers. To this end, in this dissertation, we focus on estimating the subspace robustly using low-rank optimization, where the sparse outliers are detected and separated through the ℓ_1 norm. The robust estimation has a two-fold advantage: First, the obtained basis better represents the actual subspace because it does not include contributions from the outliers. Second, the detected outliers are often of a specific interest in many applications, as we will show throughout this thesis.

We demonstrate four different formulations and applications for low-rank optimization. First, we consider the problem of reconstructing an underwater sequence by removing the

turbulence caused by the water waves. The main drawback of most previous attempts to tackle this problem is that they heavily depend on modelling the waves, which in fact is ill-posed since the actual behavior of the waves along with the imaging process are complicated and include several noise components; therefore, their results are not satisfactory. In contrast, we propose a novel approach which outperforms the state-of-the-art. The intuition behind our method is that in a sequence where the water is static, the frames would be linearly correlated. Therefore, in the presence of water waves, we may consider the frames as noisy observations drawn from a the subspace of linearly correlated frames. However, the noise introduced by the water waves is not sparse, and thus cannot directly be detected using low-rank optimization. Therefore, we propose a data-driven two-stage approach, where the first stage "sparsifies" the noise, and the second stage detects it. The first stage leverages the temporal mean of the sequence to overcome the structured turbulence of the waves through an iterative registration algorithm. The result of the first stage is a high quality mean and a better structured sequence; however, the sequence still contains unstructured sparse noise. Thus, we employ a second stage at which we extract the sparse errors from the sequence through rank minimization. Our method converges faster, and drastically outperforms state of the art on all testing sequences.

Secondly, we consider a closely related situation where an independently moving object is also present in the turbulent video. More precisely, we consider video sequences acquired in a desert battlefields, where atmospheric turbulence is typically present, in addition to independently moving targets. Typical approaches for turbulence mitigation follow averaging or de-warping techniques. Although these methods can reduce the turbulence, they distort the independently moving objects which can often be of great interest. Therefore, we address the

problem of simultaneous turbulence mitigation and moving object detection. We propose a novel three-term low-rank matrix decomposition approach in which we decompose the turbulence sequence into three components: the background, the turbulence, and the object. We simplify this extremely difficult problem into a minimization of nuclear norm, Frobenius norm, and ℓ_1 norm. Our method is based on two observations: First, the turbulence causes dense and Gaussian noise, and therefore can be captured by Frobenius norm, while the moving objects are sparse and thus can be captured by ℓ_1 norm. Second, since the object's motion is linear and intrinsically different than the Gaussian-like turbulence, a Gaussian-based turbulence model can be employed to enforce an additional constraint on the search space of the minimization. We demonstrate the robustness of our approach on challenging sequences which are significantly distorted with atmospheric turbulence and include extremely tiny moving objects.

In addition to robustly detecting the subspace of the frames of a sequence, we consider using trajectories as observations in the low-rank optimization framework. In particular, in videos acquired by moving cameras, we track all the pixels in the video and use that to estimate the camera motion subspace. This is particularly useful in activity recognition, which typically requires standard preprocessing steps such as motion compensation, moving object detection, and object tracking. The errors from the motion compensation step propagate to the object detection stage, resulting in miss-detections, which further complicates the tracking stage, resulting in cluttered and incorrect tracks. In contrast, we propose a novel approach which does not follow the standard steps, and accordingly avoids the aforementioned difficulties. Our approach is based on Lagrangian particle trajectories which are a set of dense trajectories obtained by advecting optical flow over time, thus capturing the ensemble motions

of a scene. This is done in frames of unaligned video, and no object detection is required. In order to handle the moving camera, we decompose the trajectories into their camera-induced and object-induced components. Having obtained the relevant object motion trajectories, we compute a compact set of chaotic invariant features, which captures the characteristics of the trajectories. Consequently, a SVM is employed to learn and recognize the human actions using the computed motion features. We performed intensive experiments on multiple benchmark datasets, and obtained promising results.

Finally, we consider a more challenging problem referred to as complex event recognition, where the activities of interest are complex and unconstrained. This problem typically pose significant challenges because it involves videos of highly variable content, noise, length, frame size . . . etc. In this extremely challenging task, high-level features have recently shown a promising direction as in [53, 129], where core low-level events referred to as concepts are annotated and modelled using a portion of the training data, then each event is described using its content of these concepts. However, because of the complex nature of the videos, both the concept models and the corresponding high-level features are significantly noisy. In order to address this problem, we propose a novel low-rank formulation, which combines the precisely annotated videos used to train the concepts, with the rich high-level features. Our approach finds a new representation for each event, which is not only low-rank, but also constrained to adhere to the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Extensive experiments on large scale real world dataset TRECVID Multimedia Event Detection 2011 and 2012 demonstrate that our approach consistently improves the discriminativity of the high-level features by a significant margin.

To my parents.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Mubarak Shah and co-advisor Dr. Niels da Vitoria Lobo for their guidance and support. Their constant motivation and faith in my ability were the basic factors of this accomplishment. I am grateful to Dr. Xin Li, Dr. Imran Saleemi, and Dr. Vladimir Reilly for their valuable discussions and instructions during my research endeavor. I would like to thank my friends and colleagues: Berkan Solmaz, Afshin Dehghan, Amir Roshan, and Enrique Ortiz for their friendship and support. I am also grateful to Dr. Kenneth Stanley and Dr. Mingjie Lin for serving on my committee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Various fundamental applications in computer vision require finding the basis of a certain subspace. The obtained basis are then used to represent any observation in the subspace, often using only the major basis in order to reduce the dimensionality and suppress noise. Examples of such applications include face detection, motion estimation, activity recognition ... etc. An increasing interest has been recently placed on this area as a result of significant advances in the mathematics of matrix rank optimization. Interestingly, as we will discuss in more detail in the coming sections, robust subspace estimation can be posed as a low rank-optimization problem, which can be solved efficiently using techniques such as the method of Augmented Lagrange Multiplier [71].

In this thesis we propose novel formulations and extensions for low-rank optimization-based subspace estimation and representation. By minimizing the rank of the observations' matrix, we are able solve four fundamental computer vision problems including video denosing, background subtraction, motion estimation and decomposition, and complex event recognition.

1.1 Overview and Motivation

Subspace estimation is a very important problem with widespread applications in computer vision and pattern recognition. In computer vision, for example, the number of random

variables involved in a certain process can be significantly large (eg. pixels of an image), yet it is often possible to express the process using a parametric model, which employs a few parameters to describe, for instance, the appearance or geometry of the process. This low-dimensional representation can then be utilized in developing more efficient and robust solutions to problems such as detection and classification. This has motivated the researchers to develop various techniques for finding low-dimensional representations for high-dimensional data.

Conventional techniques, such as PCA [58] and GPCA [97], assume a Gaussian noise model. Therefore, they are significantly challenged when faced by data with missing entries and outliers. This issue can be addressed within a probabilistic framework, using mixture models [116] or hidden variable models [47]. However, the majority of theses approaches result in non-convex optimization problems, which are difficult to initialize and optimize.

Recently, sparse and low-rank representations have been dominating the literature, with a large amount of methods targeted towards subspace estimation in general, and various applications of that such as object classification, object detection, and tracking [22, 89, 20, 102, 33, 40, 88]. Given a vector of data entries, sparse representation seeks to minimize the number of non-zero entries in the vector. In comparison, low-rank representation seeks to minimize the rank of a matrix containing the data entries in its columns or rows. The two objectives can be viewed equivalently since they both seek a low-dimensional representation of the data. The objective in both sparse and low-rank representations is non-convex; however, there exist convex surrogates for these problems, which can be efficiently minimized. For instance, it was shown in [22] that when recovering low-rank matrices from sparse errors, if the rank of the matrix to be recovered is not too high and the number of non-zero error entries is not too large,

then minimizing the nuclear norm (sum of singular values) and ℓ_1-norm can recover the exact matrices. Therefore, the nuclear norm and the ℓ_1-norm are the natural convex surrogates for the rank function and the ℓ_0-norm, respectively.

More specifically, lets consider a noise corrupted data matrix $D = A + E$, with observations stacked in its columns. A is an unknown low-rank matrix and E represents the noise. PCA can be viewed as the problem of finding a low-rank approximation of D, which can be formulated as

$$\arg\min_{A} ||D - A||_F^2 \text{ s.t. } \text{rank}(A) \leq r. \tag{1.1}$$

The optimal solution to 1.1 is given by $A = U_r \Sigma_r V_r$, where U_r, Σ_r and V_r are obtained from the top r singular values and singular vectors of the data matrix D. When r is unknown, the problem of finding a low-rank approximation can be formulated as

$$\arg\min_{A} \text{rank}(A) + \frac{\beta}{2}||D - A||_F^2, \tag{1.2}$$

where $\beta > 0$ is a parameter. Since this problem is in general NP hard, a common practice (see [9]) is to replace the rank of A by its nuclear norm $||A||_*$, i.e., the sum of its singular values, which leads to the following convex problem

$$\arg\min_{A} ||A||_* + \frac{\beta}{2}||D - A||_F^2. \tag{1.3}$$

The optimal solution to 1.3 is obtained using the Singular Value Thresholding algorithm [19]

$$A = US_{\frac{1}{\beta}}(\Sigma)V^T, \tag{1.4}$$

where $U\Sigma V^T$ is the SVD of D, and $S_\alpha(\cdot)$ is the soft thresholding operator defined for a scalar x as:

$$S_\alpha(x) = \text{sign}(x) \cdot \max\{|x| - \alpha, 0\}. \tag{1.5}$$

Note that the latter solution does not coincide with the one given by PCA, which performs hard-thresholding of the singular values of D without shrinking them by $\frac{1}{\beta}$.

While the above methods work well for data corrupted by Gaussian noise, they break down for data corrupted by gross errors. In [22], this issue is addressed by assuming that the outliers are sparse, i.e., only a small percentage of the entries of D are corrupted. Hence, the goal is to decompose the data matrix D as the sum of a low-rank matrix A and a sparse matrix E

$$\arg\min_{A,E} \text{rank}(A) + \lambda||E||_0 \text{ s.t. } D = A + E, \tag{1.6}$$

where λ is a parameter that trades off the rank of the solution versus the sparsity of the error. Since 1.6 is non-convex, a convex relaxation is applied by replacing $\text{rank}(A)$ with the nuclear norm or sum of the singular values $||A||_* = \Sigma_i(\sigma_i)$, and replacing $||E||_0$ with its convex surrogate ℓ_1 norm $||E||_1$

$$\arg\min_{A,E} ||A||_* + \lambda||E||_1 \text{ s.t. } F = A + E. \tag{1.7}$$

While a closed form solution to this problem is not known, convex optimization techniques can be employed to find the optimal solution. We refer the reader to [71] for a review of the recent methods. The most popular approach to solve 1.7 is the Augmented Lagrange Multiplier (ALM) [71], and we will discuss its details further in the following chapters.

The recent discoveries in low-rank optimization and the ability to solve it efficiently with methods such as (ALM), encouraged investigating its applications in computer vision. In this thesis, we demonstrate how low-rank optimization can be adopted to solve several fundamental computer vision problems.

1.2 Contributions

In this thesis, we demonstrate four novel formulations of low-rank optimization. The developed approaches allow us to obtain superior solutions for fundamental computer vision problems including scene reconstruction, video stabilization, background subtraction, motion estimation and segmentation, and complex activity recogntion. In the coming subsections we discuss our specific contributions in each of the low-rank-based formulations.

1.2.1 Underwater Scene Reconstruction

First, we propose a novel method for recovering the original image of an underwater scene using a sequence distorted by water waves. In a sequence where the water is almost static, the frames of the sequence can be roughly considered linearly correlated. This is because the variations among the frames in this case are limited to slight effects such as illumination changes, which can be considered as multiplicative noise. Therefore, in the case where water

turbulence is present, minimizing the rank of the matrix containing the frames in its columns should reveal the underlying original frames. However, the noise introduced by the water waves is not sparse, and thus cannot directly be detected using low-rank optimization. Therefore, we propose a two stage approach. The first stage leverages the temporal mean of the sequence to overcome the structured turbulence of the waves through an iterative non-rigid registration process. Once the structured water turbulence is extracted, the remaining errors are sparse outliers, and thus can be eliminated by low-rank optimization, which resembles our second stage.

In the first stage of our technique, we register the frames to their temporal mean which is close to the original undistorted scene. However, the mean of an underwater sequence is blurry and noisy; therefore, a standard frame to mean non-rigid registration faces considerable challenges. The straightforward workaround is to deblur the mean; yet, the deconvolution operation involved in deblurring is generally ill-posed; therefore, even the latest deblurring techniques such as [127, 60] often fail and introduce undesirable edges. We show that registration-based reconstruction of an underwater scene could be greatly improved by a modified approach that we refer to as robust registration. The robust registration includes two advances: First, iterative refinement of the mean and the frames through iterative registration; second, aided mean-frame correlation step at which we blur the sharp frame instead of deblurring the blurry mean by estimating a blur kernel that brings the frame to the same blur level of the mean.

In a few iterations, the robust registration converges to a reconstructed mean and a new sequence which are free from most of the structured turbulence of the water waves. However, many frames will still contain unregistered components caused by three types of random noise:

First, light reflection on the water surface; second, occlusion of the underwater scene; third, the random behavior of the waves. Such unstructured random noise is however sparse and has a direct correlation with the rank of the matrix composed of the registered frames. In the second stage of our method, we eliminate the sparse noise through convex rank optimization. The result is a clear underwater sequence with significantly reduced noise.

1.2.2 Simultaneous Video Stabilization and Moving Object Detection

Next, we propose a framework to handle a more challenging scenario, where in addition to turbulence, objects of interest are also moving in the scene. The objects' motion is typically mixed up with the turbulence deformation in the captured images, rendering the problem of detecting the moving objects extremely difficult. In this thesis, we handle the dual problem of turbulence mitigation (stabilizing the sequence) and moving object detection under the turbulent medium. To the best of our knowledge, such a problem has never been explored before. Relevant previous approaches have either focused on detecting moving objects or dewarping a deformed sequence, but not on both tasks concurrently. Note that other than image deformation, atmospheric turbulence may cause blur if the camera exposure time is not sufficiently short. However, we focus only on image deformation because of the inherent confusion between the motion of the object and the motion caused by the turbulence.

We make three main contributions in this particular subject: First, we propose a new variant of matrix decomposition based on low-rank optimization and employ it to solve the novel problem of simultaneous moving object detection and turbulence mitigation in videos distorted by atmospheric turbulence. Second, we propose a turbulence model based on both

intensity and motion cues, where the motion distribution is derived from the Lagrangian particle advection framework [7]. The turbulence model is used to enforce an additional constraint on the decomposition which encourages the sparse solutions to be located in areas with non-Gaussian motion. Finally, we propose an additional force component in the particle advection framework in order to stabilize the particles in the turbulent medium and handle long sequences without discontinuities.

1.2.3 Motion Decomposition of Lagrangian Particle Trajectories

We propose a novel approach, targeted towards action recognition in videos acquired by a moving camera. In contrast to the traditional approaches for dealing with moving cameras, where video alignment is typically employed beforehand, we propose a novel algorithm based on low-rank optimization, which concurrently segments the trajectories corresponding to the moving object and eliminates their camera motion component; thus providing the relevant independent particle trajectories that correspond solely to the object's motion. It is important to notice that, in contrast to the other two low-rank-based methods we developed, this one finds the low-rank subspace of trajectories (compared to frames in the other methods). In other words, instead of stacking the frames as observations in the low-rank matrix, here the variables are motion trajectories (spatial coordinates), which we stack in the columns of the matrix to be decomposed.

Furthermore, in contrast to the traditional motion trajectory acquisition mechanisms, we propose to automatically extract a set of particle motion trajectories for action representation and recognition. Particle trajectories are a set of dense trajectories that capture the en-

semble motions of a scene through the motion of an overlaid grid of particles. The basis of particle trajectory acquisition lies in advecting the particles using optical flow. The advection-based particle trajectory acquisition follows a bottom-up method where neither pre-definition of interest points nor point correspondence across frames is required; hence, it is inherently different from traditional object tracking, and does not suffer from any of the aforementioned problems faced by tracking-based approaches. To the best of our knowledge, particle trajectories have been mainly used for crowded flow analysis [124], but they have never been used for action recognition.

The main contributions in this particular subject are summarized as: First, we propose a novel approach based on low-rank optimization to robustly extract the trajectories which merely correspond to the object's motion from the whole set of particle trajectories obtained in a moving camera scenario, thus avoiding the standard approach which requires explicit video alignment and moving object detection. Second, our method is the first to utilize the dense particle trajectories of the objects for action recognition.

1.2.4 Complex Event Recognition

Finally, we propose a novel approach, targeted towards complex event recognition in long and unconstrained videos. In this extremely challenging task, high-level features have recently shown a promising direction as in [53, 129], where core low-level events referred to as concepts are annotated and modelled using a portion of the training data, then each event is described using its content of these concepts. However, because of the complex nature of the videos, both the concept models and the corresponding high-level features are significantly

noisy. In order to address this problem, we propose a novel low-rank formulation, which combines the precisely annotated videos used to train the concepts, with the rich high-level features. Our approach finds a new representation for each event, which is not only low-rank, but also constrained to adhere to the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Extensive experiments on large scale real world dataset TRECVID Multimedia Event Detection 2011 and 2012 demonstrate that our approach consistently improves the discriminativity of the high-level features by a significant margin.

The main contribution in this particular subject is a novel low-rank formulation, through which we find a new representation for each event, which is not only low-rank, but also constrained by the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Our constrained low-rank representation is not restricted to a certain type of features; which allows us to employ a combination of state-of-the-art features including STIP [64], DTF-MBH [120], and DTF-HOG [120].

1.3 Organization of the Thesis

The thesis is structured as follows: **Chapter 2** reviews existing literature on all relevant topics including low-rank optimization, video denoising, turbulence mitigation, background subtraction, and action recognition. Moreover, we attempt to contrast the related works with our contributions. **Chapter 3** presents a novel approach for reconstructing an underwater sequence using low-rank optimization and non-rigid registration. **Chapter 4** proposes a novel approach for simultaneous moving object detection and video stabilization in a turbulent medium

using a method we refer to as Three-Way Low-Rank Decomposition. **Chapter 5** describes a framework for detecting the camera motion subspace and decomposing motion trajectories through a novel low-rank decomposition approach, which we also employ for activity recognition. **Chapter 6** describes our method for complex event recognition using a novel low-rank model which is constrained by manual user annotation. Finally, the thesis is concluded in **Chapter 7**, with a summary of contributions and a description of future work.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

In this chapter, we first review the problem of linear subspace estimation and present example problems where the conventional method (PCA) is typically used. Consequently, we discuss the most prominent advances in low-rank optimization, which is the main theoretical topic of this thesis. Since our various low-rank formulations fall into several computer vision domains, we additionally review the latest techniques in each domain, including video denosing, turbulence mitigation, background subtraction, and activity recognition.

2.1 Linear Subspace Estimation

Consider a set of points drawn from an unknown distribution, as illustrated in blue in Figure 2.1. For simplicity and ease of illustration, two dimensional points are considered; however, this discussion can be extended to any number of dimensions. The most popular method to estimate an orthogonal basis set is the Principal Component Analysis, where a set of orthonormal eigenvectors and their corresponding eigenvalues are computed such that the reprojection along these directions has a minimum reconstruction error. As discussed in the previous chapter, these eigenvectors (basis) can be sorted according to the variance along each basis direction using the corresponding eigenvalues. In practise, the basis vectors with insignificant variance are ignored since they typically correspond to noise. This also results in a major reduction of dimensionality, which is often desirable in many machine learning techniques

which scale exponentially with data dimension. Figure 2.1 illustrates the computed basis in yellow, where the length of the vectors are weighted using the eigenvalues (the longer vector corresponds to higher eigenvalue and higher variance).

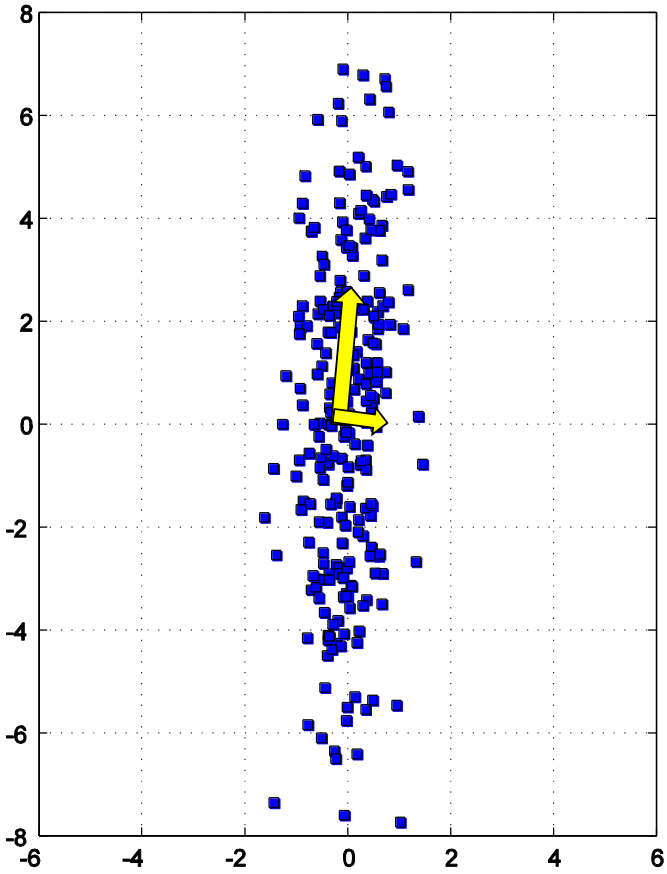

Figure 2.1: Blue: Sample data points drawn from an unknown distribution. Yellow: the estimated basis vectors computed using PCA. The length of the basis vector corresponds to the amount of variance in the data along the corresponding direction.

This forms the basic structure of a wide variety of computer vision and machine learning applications. For example, consider a set of images of size 100×100, which show the face of a human in different light conditions, poses, etc. $10K$ dimensions are required to represent each image. Alternatively, PCA can be applied, and the major k basis can be extracted. Con-

sequently, each image can be projected along each basis vector, and represented as only a set of k coefficients. These coefficients can then be employed for instance to compare or classify the faces.

A wide variety of extensions to PCA have been proposed, form which we briefly discuss the most prominent approaches

- Kernel PCA (KPCA) [100]: Kernel PCA is an extension for PCA to handle the case where the data lie in a non-linear subspace. Embedding this data into a higher dimensional space (feature space) allows it to behave linearly (can be linearly separated). In KPCA, this non-linear mapping is never found. Instead, since the data points always appear in a dot product form, the kernel trick is used, in which each point is represented using the distances to all other points in order to form a kernel matrix. Consequently, Eigen Value Decomposition is applied on this new representation. Since the actual high dimensional embedding is not explicitly computed, the kernel-formulation of PCA is restricted in that it does not compute the principal components themselves, but the projections of the data onto those components.

- Probabilistic PCA [117]: Probabilistic PCA formulates PCA in a Maximum Likelihood parameter estimation problem. The principal subspace is computed iteratively using Expectation Maximization (EM). Since probabilistic PCA estimates a generative model, it can handle the case where the data vectors has missing values.

- PCA for noise in exponential family [26]: PCA implicitly minimizes a squared loss function, which may be inappropriate for data that is not real-valued, such as binary,

integer, or non-negative data. This also means that PCA intrinsically assumes a Gaussian noise model. Here, a probabilistic approach is provided in which PCA is extended to handle loss functions that are better suited for several data types.

- Generalized PCA (GPCA) [97]: GPCA extends PCA to handle the case where the data points are drawn from multiple subspaces. This case is far more difficult, since the problem inherently includes data segmentation. GPCA is an algebraic geometric approach to data segmentation; therefore, it is different than probabilistic approaches, such as the Gaussian Mixture Models (GMM), or spectral clustering methods . . . etc.

One of the critical issues in PCA is robustness against outliers. PCA is sensitive to outliers; which often falsely contribute in the computed basis, and result in a potentially inaccurate basis. Popular outlier detection methods such as RANSAC cannot be employed since they often require prior knowledge of a model which fits the data, which is often unknown or non-existing. This is inarguably the most prominent reason behind the research for a method which can detect the outliers and estimate the basis simultaneously. In the coming section, we discuss the recent developments in low-rank optimization, which represents one of the most successful approaches in the area of robust subspace estimation.

2.2 Low-Rank Representation

A low-rank structure can be identified as a set of observations which can be represented using a low number of basis. We observe such low-rank structures frequently in our everyday life. For instance, building facades are typically composed of groups of repetitive structures,

which together form a low-rank space (each group corresponds to a certain basis vector in the space of the facade). Similarly, the frames of a video may also correspond to a low-rank subspace because in principle the video can be divided into groups of frames, where the frames of each group are similar or linearly correlated (the frames of a group may vary only in multiplicative factors such as illumination changes). Refer to Figure 2.2. It is also not difficult to think of other various examples in image processing, web data ranking, and bioinformatic data analysis.

Figure 2.2: Example low-rank structures. Left: Pictures of a human face under different illuminations. Middle: A building facade. Right: A sequences of frames (video).

When observing samples of such low-rank structures, they are often contaminated with noise which can be gross and difficult to model (eg. non-Gaussian sparse errors). For instance, Figure 2.3 shows a building facade occluded by a flag and a flyer. The rows (or columns) containing the occluded pixels do not lie in the low-rank space. Therefore, it is desirable

to recover the inherent low-rank structure using these noisy samples, and separate the noise, which also can often be of a specific interest.

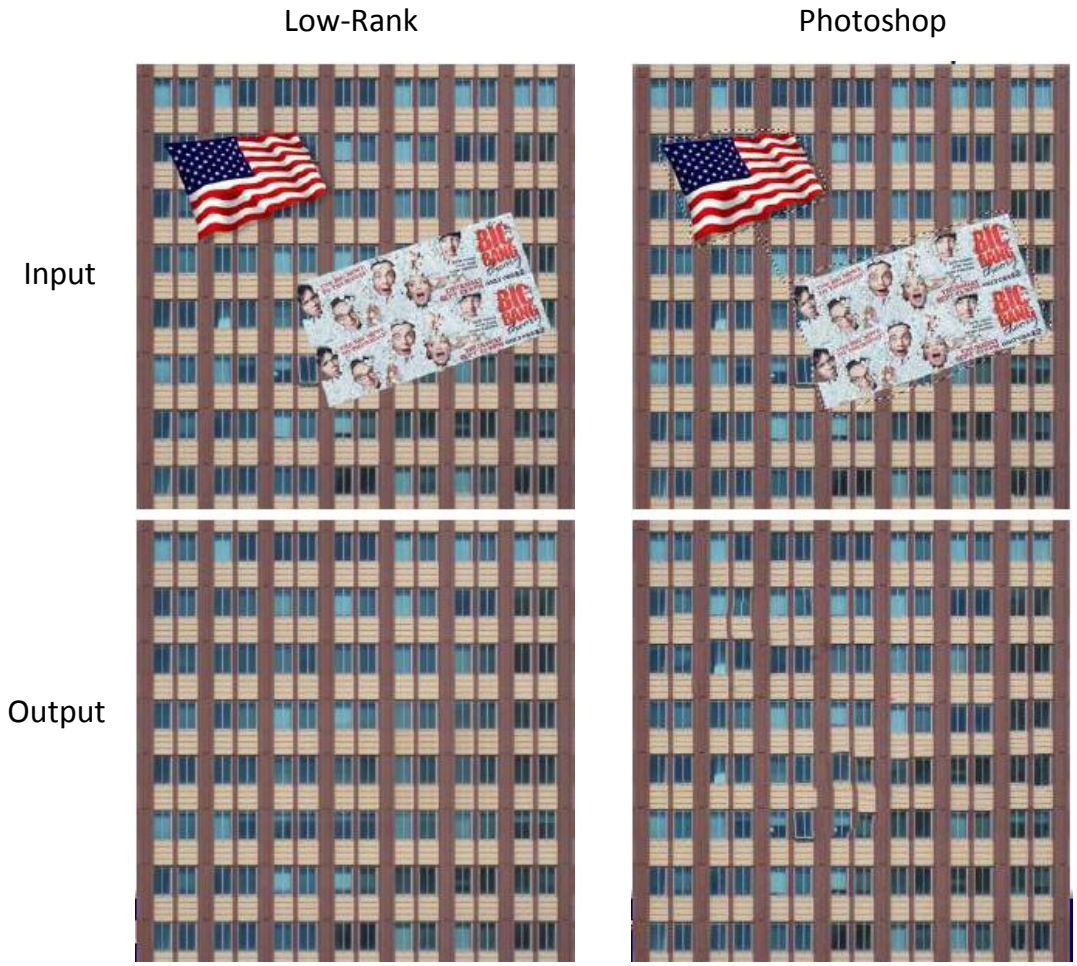

Figure 2.3: Example showing a building facade occluded by a flag and a flyer. The rank of the matrix containing this image is unnecessarily high due to the occlusion. Low-rank optimization makes it possible to detect the noise (the occlusion in this example), and thus recover the original low-rank structure (the image of the facade in this example). The picture is taken from [81].

In the previous chapter, we showed that noisy data points drawn from a linear subspace can be stacked in the columns of a matrix, and decomposed into a low-rank component and a sparse noise component by minimizing the nuclear norm and the ℓ_1 norm, respectively. Surprisingly, the Robust PCA problem can be solved under broad conditions via convex optimization

techniques such as the Augmented Lagrange Multiplier (ALM) [71] and the Accelerated Proximal Gradient (APG) [72]. These approaches fall in the computational methods for finding the desired decompositions based on the Alternating Directions Method of Multipliers (ADMM), which was first introduced in the mid-1970's by [42, 43], and is the current method of choice for large-scale non-smooth convex optimization, as in [112, 44, 50, 134]. It is also possible to solve RPCA with other methods such as for example [3], where sufficient conditions can be obtained in order to find an optimal solution, recovering the low-rank and sparse components of the matrix. In this thesis, we focus only on the ALM method to solve our Low-Rank formulations because of its stability and convergence speed.

The general method of ALM solves a constrained problem of this form:

$$\min f(X) \text{ s.t. } h(X) = 0, \tag{2.1}$$

where $f : \mathbb{R}^N \to \mathbb{R}$ and $h : \mathbb{R}^N \to \mathbb{R}^M$. The augmented lagrange function for equation (2.1) is defined as

$$L(X, Y, \mu) = f(X) + < Y, h(X) > + \frac{\mu}{2}||h(X)||_F^2, \tag{2.2}$$

where μ is a positive scalar. Under some rather general conditions, when $\{\mu_k\}$ is an increasing sequence and both f and h are continuously differentiable functions, the Lagrange multiplier converges to the optimal solution by following the algorithm outlined in algorithm 1.

In the case of RPCA problem as in equation (1.7), the aforementioned ALM algorithm can be applied with $X = (A, E)$, $f(X) = ||A||_* + \lambda||E||_1$, and $h(X) = F - A - E$. Note that solving equation (1.7) using ALM involves solving a joint optimization for A and E, i.e.

Algorithm 1: The General Method of Augmented Lagrange Multiplier

while *not converged* **do**

Solve $X_{k+1} = \min_X L(X, Y_k, \mu_k)$;

$Y_{k+1} = Y_k + \mu_k h(X_{k+1})$;

Update μ_k to μ_{k+1};

end

Output X_k.

$(A_{k+1}, E_{k+1}) = \min_{A,E} L(A, E, Y_k, \mu_k)$. However, as shown in [71], solving for each of A and E separately is sufficient for them to converge to the optimal solution (in this case ALM is referred to as inexact ALM).

Ever since the basic formulation of Robust PCA was proposed in [22], where a low rank matrix was recovered from a small set of corrupted observations through convex programming, numerous applications followed that in Low-Rank and sparse optimization-based image and video processing. From these method, in this section, we briefly review the most related articles. Low-rank optimization was employed in [56] for video de-noising, where serious mixed noise was extracted by grouping similar patches in both spatial and temporal domains, and solving a low-rank matrix completion problem. Additionally, in [94], linear rank optimization was employed to align faces with rigid transformations, and concurrently detect noise and occlusions. In [133], Yu et. al. proposed an efficient solution to subspace clustering problems which involved the optimization of unitarily invariant norms. Another variant of such space-time optimization techniques is the total variation minimization, where for instance in [23], Chan et. al. posed the problem of video restoration as a minimization of anisotropic total varia-

tion given in terms of ℓ_1-norm, or isotropic variation given in terms of ℓ_2-norm. Consequently, the Lagrange multiplier method was used to solve the optimization function. Moreover, the low-rank constraint has been vigorously employed in other computer vision problems such as tracking [115], feature fusion [130], face recognition [25], and saliency detection [105].

2.3 Turbulence Mitigation and Video Denoising

Approaches for turbulence mitigation focused mainly on registration-based techniques. In [106, 135, 136], both the turbulence deformation parameters and a super-resolution image were recovered using area-based B-Spline registration. Moreover, in [114], Tian and Narasimhan proposed recovering the large non-rigid turbulence distortions through a "pull-back" operation that utilizes several images with known deformations. Averaging-based techniques are also popular for video de-noising and turbulence mitigation, including pixel-wise mean/median, non-local means (NLM) [16, 74], fourier-based averaging [122], and speckle imaging [98, 45, 46]. Another category of methods for turbulence mitigation is the lucky region approach [79, 31, 59], where the least distorted patches of the video are selected based on several quality statistics, then those selected patches are fused together to compose the recovered video.

The earlier work in reconstructing an underwater sequence focused on finding the center of the distribution of patches in the video through clustering [31, 32], and manifold embedding [34], or employing the bispectrum to average the frames in the fourier domain [122]. The state of the art in this area, however, is the model-based tracking [113], where the characteristics of the water waves were employed to estimate the water basis using PCA. In such work,

the optimal number and size of the basis remain vague; therefore, we argue that the estimated basis can be under-fitted or over-fitted depending on the selected basis. Other than requiring an orthographic camera and a given water height, the basis are additionally obtained by simulation using a single parameter differential equation for the waves, with the assumption that the surface fluctuations are small compared to the water's height. Such a simple model with low parameter space is quite limited, and does not fully represent the actual scenario which can be much more complicated and dependant over several other factors such as the container size, external forces, and camera calibration; hence, the results from [113] are not quite satisfactory. Later in [114], it was proposed that the large non-rigid distortion caused by effects such as the water waves cannot be overcome through traditional B-spline registration, but rather by utilizing training images with predefined deformations. While the distorted video is typically the only available piece of information in such a problem, [114] assumes additional given training images, or a template from which we can generate samples; therefore, their work is considered out of the scope of comparison with our method.

In the context of blur kernel estimation which is used in the robust registration stage of our underwater scene reconstruction, the latest advances focused on deblurring a single image [127, 60], or a motion blurred video [68, 4]. However, our problem layout is different in that the underwater sequence is not blurry, but its mean is extremely blurry and noisy such that it cannot be deblurred. Thus, we are interested in rather blurring the frames in order to aid the registration. For those reasons, we propose to estimate a spatially varying blur kernel which encodes the difference in blur between the mean and the frames by using the motion estimated at each iteration of our algorithm. Our robust registration is not very sensitive to the

frame blurring operation; therefore, in principal, other good blur estimation algorithms could be employed.

2.4 Moving Object Detection

Moving object detection is a widely investigated problem. When the scene is static, moving objects can be easily detected using frame differencing. A better approach would be to use the mean, the median, or the running average as the background [27]. The so-called eigenbackground [87] can also be obtained using PCA. However, when the scene is constantly changing because of noise, light changes, or camera shake, the intensities of image pixels can be considered as independent random variables, which can be represented using a statistical model such as a Gaussian, a mixture of Gaussians, or a kernel density estimator. The model can then be used to compute the probability for each pixel to belong to either the background or the foreground. Examples of such approaches include [109, 35, 137]. Additionally, the correlation between spatially proximal pixels could also be employed to improve the background modelling using a joint domain (location) and range (intensity) representation of image pixels such as in [104].

2.5 Motion Trajectories and Activity Recognition

Motion trajectories have been employed in a variety of problems for human action representation and recognition [84, 28, 1, 17, 6, 62, 92, 57]. Many tracking-based methods are used or could be adapted for trajectory acquisition (for a comprehensive review on tracking techniques, readers may refer to relevant surveys [2]). Usually, a single trajectory can be

acquired by simple techniques such as temporal filtering [1]. The tracking entity typically is either a human body part (head, hand, foot, etc.) or a person as a whole. For the simultaneous tracking of multiple points, KLT [118] is a popular choice [62, 92, 96]. Statistical mixture models are also developed for multi-trajectory tracking [28]. Some specific tracking strategies (e.g. SMP [84]) are designed to handle complicated and subtle full-body motions. A common drawback among tracking-based methods is that it is difficult to obtain reliable trajectories for the reasons discussed in the previous section. In addition, several studies assume that the motion trajectories are already available [123], or they rely on manual annotations [17], or the so-called semi-automatic manner [6]. In contrast, the particle advection in our work is fully automatic and is very easy to implement.

Particle trajectories have been previously used to model crowded scenes in [124], where the flows normally occupy the whole frame, and the camera is static; thus, such dense trajectories could be directly employed. We, in contrast, adopt the particle trajectories for recognizing actions in videos acquired from a moving camera, which imposes several challenges since the actions usually only cover a small part of the frame, and more importantly, the obtained trajectories combine both the camera motion and the object motion. Therefore, we propose a novel approach to detect the foreground trajectories and extract their object-induced component, which in principal requires estimating the background motion subspace. A large variety of subspace estimation methods exist in the literature such as PCA-based and RANSAC-based approaches. Such methods are, however, sensitive to noise which is considerably present in our scenario since a significant number of trajectories can be contaminated with the foreground motion. Fortunately, sparsity-based matrix decomposition methods such as [41, 21, 48] which

have been primarily employed in image denoising domain, proved that a robust estimation of an underlying subspace can be obtained by decomposing the observations into a low rank matrix and a sparse error matrix. Therefore, in this work, we show how Robust PCA [21] can be adopted to extract the object motion relevant to the action of interest.

The acquired motion trajectories can be represented by certain descriptors to identify the underlying spatio-temporal characteristics. Wu *et al.* [123] proposed a systematic signature descriptor that can provide advantages in generalization, invariants, and compactness, etc. Ali *et al.* [6] showed that the features based on chaotic invariants for time series analysis perform very well in modelling semi-automatically obtained trajectories. Meanwhile, "trajecton" was proposed in a Bag-of-Words context [92] for trajectory-based action recognition. Messing *et al.* [96] investigated the temporal velocity histories of trajectories as a more representative feature for recognizing actions. In this work, we employ the particle trajectories and choose the chaotic invariants [6] as a trajectory descriptor. It should be noted that we adopted the algorithms in [124] for computing the chaotic features as they have been shown more robust than [6].

Aside from trajectory features, a variety of feature representations have been developed for action recognition such as appearance features [29], shape-based representation [61]), volumetric features (e.g. Poisson equation-based features [80], 3D Haar feature [128]), spatiotemporal interest points ([91, 18, 93]), motion history image (MHI) [14], and kinematic features [8].

2.6 Complex Event Recognition

Compared to the traditional action recognition, complex event recognition is more challenging, mainly because the complex events have significantly longer lengths and diverse contents. Early methods for complex event recognition used low-level features such as Dollar [30], SIFT [78], MBH [120], and MFCC [64], and showed promising results as in [101, 13, 75, 121]. Additionally, pooling of these low-level features was proposed in [69], where features such as SIFT and color were fused in order to improve the complex event recognition. Moreover, [111] used seven different low-level features in a Bag-of-Words (BoW) framework. However, these approaches reveal limitation in representing semantic concepts because they seek high-level class labels using only low-level features.

On the other hand, concept-based complex event recognition [76, 77, 53, 129] has recently flourished and shown promising results in high-level semantic representation of videos with complicated contents. This approach is particularly appealing for the purposes of retrieval and filtering of consumer media. The semantic concepts inherently represent the building blocks of the complex events; therefore, they naturally fit the complex event recognition task. For instance, in [77] and [76], a large dataset is collected to train concept detectors. However, their concepts are not suitable for complex videos as they have been recorded in well constrained conditions. Additionally, Loui et al. in [77] collected a benchmark dataset containing 25 concepts; however, the concepts are based on static images, not videos. On the other hand, concepts have been also employed in other computer vision problems such as image ranking and retrieval [108], and object classification [39]. In that, the concepts were used in the form of attributes [39], which can be considered as concepts with small granularity [53].

The most recent works on complex event recognition are [129] and [53]. The former utilized 62 action concepts as well as low-level features in a latent SVM framework, while the latter used an unsupervised approach (deep learning [65]) to find the data driven concepts in order to describe the high level semantics of the complex videos. Data driven concepts showed promising performance; however, they do not provide any conceptual description for the video.

2.7 Summary

The review of existing literature provided a summary of the prominent approaches for low-rank optimization in general, and the other relevent problems of underwater scene reconstruction, turbulence mitigation, background subtraction, activity recognition, and complex event recognition.

The method we discuss here for underwater scene reconstruction is generic, simple, and robust, which in contrast to the pervious methods, does not require a known template such as [114], the camera's height [113], or a special illumination [66]. The method is similar to the previous state the art [113] in that it works on a short sequence (61 frames) rather than 800 in [34] and 120 in [122], but more importantly, superior to [113] in performance and processing time.

Furthermore, previous work in moving object detection in dynamic scenes mostly focused on detecting the objects and did not consider recovering the background. Inversely, previous work in turbulence mitigation did not consider the possible interest in detecting moving objects in the scene. In this thesis, we pose the two problems of moving object detection and turbulence mitigation as one application for our proposed three-term low-rank decomposition. We demonstrate how to decompose a turbulent video into separate background, foreground, and turbulence components. It is important to note that our method is not directly comparable to background subtraction or turbulence mitigation approaches; though, we do provide competitive results on each task separately.

Moreover, a common drawback among pervious tracking-based activity recognition methods is that it is difficult to obtain reliable trajectories for the reasons discussed earlier. In addition, several studies assume that the motion trajectories are already available [123], or they rely on manual annotations [17], or the so-called semi-automatic manner [6]. In contrast, the particle advection presented in this work is fully automatic and is very easy to implement. Additionally, in order to handle the moving camera, previous methods typically require motion compensation, moving object detection and object tracking. The errors from these daunting steps propagate, which further complicates the activity recognition task. In contrast, the method we discuss here does not follow these steps, and accordingly avoids the aforementioned difficulties through the use of low-rank optimization, where we decompose the trajectories into their camera-induced and object-induced components in one convex optimization step, which can be efficiently solved.

In the complex event recognition domain, one of the closest approaches to the method we present in the thesis is [130], where the low-rank constraint was enforced on the detection scores for the purpose of late fusion of different training models. In our method, we estimate the low-rank subspace of high-level features (concepts), which, to the best our knowledge, has never been used before. More importantly, not only we obtain a low-rank concept representation, but also show how to incorporate the user annotation in order to encourage the low-rank estimation to follow a semantic pattern.

CHAPTER 3: SEEING THROUGH WATER

Light rays reflected from objects go through several perturbations before being captured by the camera. If the rays pass through different media, they get affected by a complex series of reflections and refractions which can cause extreme distortion to the image. Imaging through water is an example of such a scenario, where reconstructing the original underwater scene still constitutes a big challenge. A fluctuating water surface poses significant difficulties to the process of image recovery mainly because the fluctuations tend to be high, random, and exhibit a complicated behavior especially near the edges of the container. In this context, traditional sparse or dense point correspondence and tracking methods [10, 95, 83, 106, 131], which could have been employed to learn the deformation function of the water, are in fact rendered unusable for three reasons: First, tracking over long periods is difficult in such a turbulent video; second, a noise-free template of the underwater scene is unavailable; third, even using a frame from the distorted video as a template for tracking will not capture the uninvertable distortion function of the water [113]. On the other hand, as a result of the high dimensionality and the embedded randomness of the waves, modelling the distribution of intensity in the sequence is ill-posed. For example, techniques such as pixel-wise mean/median, patch-wise clustering [31, 32, 34], and fourier-based averaging [122] usually suffer. Fortunately, the latest advances in this area such as [114] provided evidence that a better solution for such a problem can be obtained using methods with combined generative and discriminative properties. In

this chapter, we follow [114], and propose a generative-discriminative approach to robustly reconstruct the underwater sequence, however, without requiring the template as in [114].

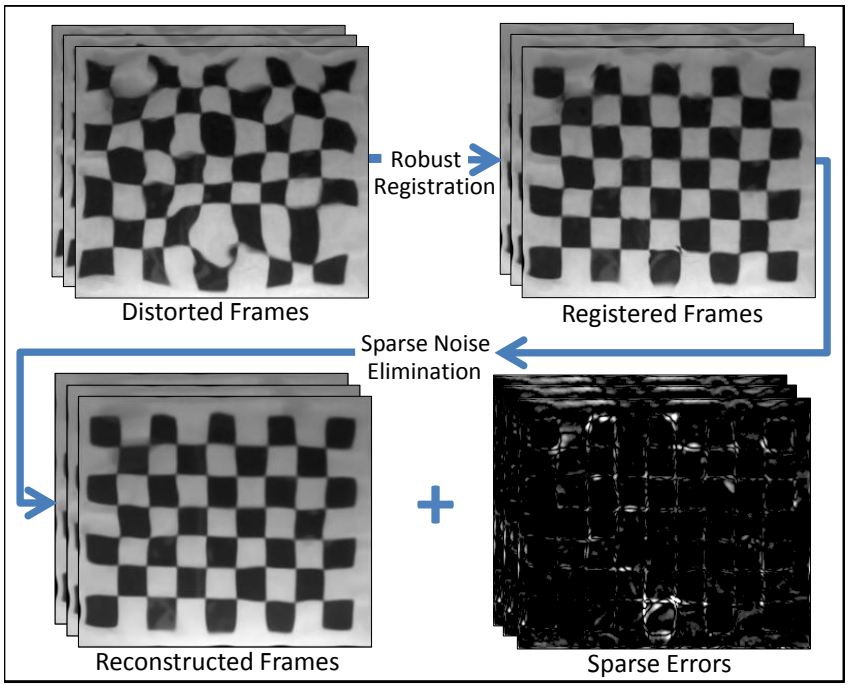

Figure 3.1: Two-Stage reconstruction of an underwater scene. Stage 1 (Robust Registration) aligns the sequence and overcomes the structured turbulence, while stage 2 (Sparse Noise Elimination) extracts the remaining unstructured random noise.

Figures Figure 3.1 and Figure 3.2 illustrate our two-stage framework which combines the powers of registration and low-rank representation. In a sequence where the water is almost static, the frames of the sequence are roughly linearly correlated. Therefore, when the water turbulence is present, minimizing the rank of the matrix containing the frames in its columns should reveal the underlying original frames. However, the noise introduced by the water waves is not sparse, and thus cannot directly be detected using low-rank optimization. Therefore, we propose a two stage approach. We refer to the first stage as the Robust Registration, where we leverage the temporal mean of the sequence to align the frames and overcome the structured turbulence of the waves through an iterative non-rigid registration process. After

Robust registration, the structured water turbulence is extracted, and the the sparse outliers are unveiled. Therefore, we employ a second stage, which we refer to as sparse noise elimination, in order to decompose the sequence into its low-rank and sparse error components.

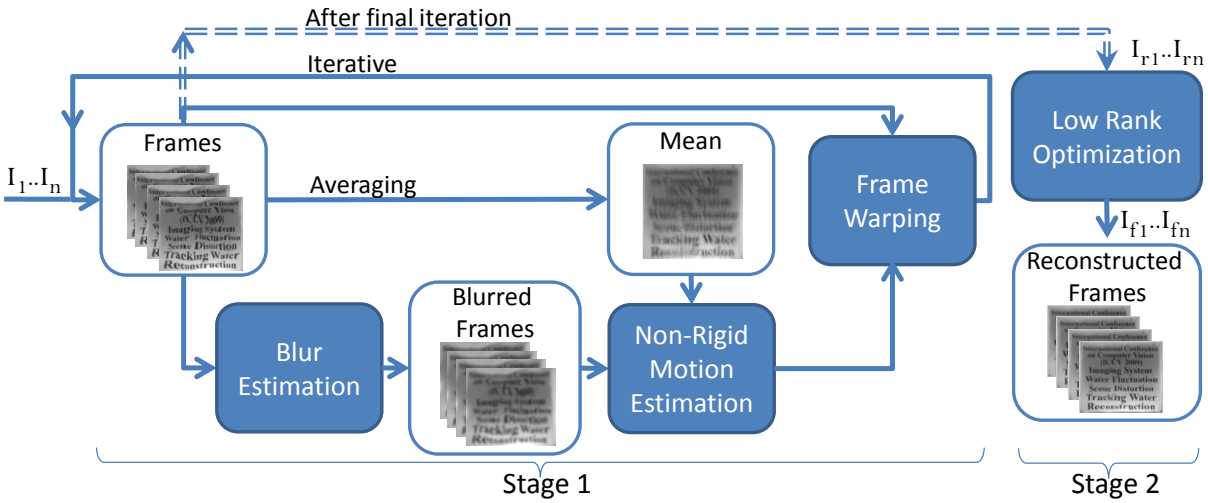

Figure 3.2: The various steps for seeing through water.

3.1 Robust Registration

For a video sequence $V \in \mathbb{R}^{w \times h \times n}$ of an underwater scene which consists of the frames $\{I_1 ... I_n\}$, where each frame I_i is distorted by an unknown water deformation $\Gamma(x)$, our goal is to recover a new wave-free sequence $V_f = \{I_{f1} ... I_{fn}\}$. Here, w and h are the width and height of the frame, respectively.

In this stage we register the frames such that the error introduced by the water is *sparsified*, and thus can be detected in the next stage by low-rank minimization. The main difficulty in that arises from the fact that a template for the underwater image is missing. Therefore, we consider the following minimization problem

$$\arg\min_{\Gamma,M} \sum_t \sum_{\mathbf{x}} (I_t(\mathbf{x} + \Gamma_t(\mathbf{x})) - M(\mathbf{x}))^2, \tag{3.1}$$

where Γ_t is the motion vector for point $\mathbf{x} = [x, y]^T$ in frame I_t, and M is the missing template. Equation 3.1 aims to find both the template and the frames' motion which will minimize the difference between the template and the warped frames.

In order to solve 3.1, we consider an iterative registration operation, where we alternate between optimizing with respect to the template and with respect to the motion. When optimizing for the template, the motion is set constant; therefore, it can be easily shown that the template is computed from 3.1 as

$$M(\mathbf{x}) = \frac{\sum_{t=1}^{n} I_t'(\mathbf{x})}{n}, \tag{3.2}$$

which is the temporal mean of the sequence. On the other hand, when the template is set to a constant, equation 3.1 becomes

$$\arg\min_{\Gamma} \sum_t \sum_{\mathbf{x}} (I_t(\mathbf{x} + \Gamma_t(\mathbf{x})) - M'(\mathbf{x}))^2, \tag{3.3}$$

which a standard non-rigid registration between each frame and the template. Therefore, at each iteration of our algorithm, the frames are shifted closer toward the correct mean (stage 1 in Figure 3.2).

In particular, we start by computing the temporal mean M; consequently, each frame is shifted to the mean through registration, thus generating a new sequence V_2. Since the mean is noisy, V_2 is not well registered; however, its mean M_2 is now better and shifted closer to the

true image underwater. Therefore, we re-register V_2 to M_2 generating the sequence V_3 and its mean M_3. We keep on performing this process for a few iterations until we settle on a robust mean and a better registered sequence.

Note that the computed mean at each iteration is blurry. In order to improve the motion estimation at each iteration, we use the motion estimated from the previous iteration to compute a blur kernel that brings the frames to the same blur level of the mean. The details about that will be discussed in the following subsection. Once the frames are blurred, they are used to compute the motion; however, the computed motion is then applied to warp the original unblurred frames in order not to introduce artifacts in the process.

The core motion estimation is a standard B-spline non-rigid registration and is similar to [99]. A frame I is registered to the mean M by estimating a motion vector Γ for each point $\mathbf{x} = [x, y]^T$ in I using a grid of control points of size $s_x \times s_y$

$$\Gamma(\mathbf{x}) = \sum_{m=0}^{3} \sum_{l=0}^{3} B_m(v) B_l(u) \Phi_{i+l,j+m}, \tag{3.4}$$

where $i = \lfloor x/s_x \rfloor - 1$, $j = \lfloor y/s_y \rfloor - 1$, $u = x/s_x - \lfloor x/s_x \rfloor$, and $v = y/s_y - \lfloor y/s_y \rfloor$. $\Phi_{i,j}$ is the motion vector of the ij-th control point on the grid, and B is a standard B-spline basis function which defines the weights of the control points contributing in the motion of $\mathbf{x}$

$$\begin{aligned} B_0(t) &= (1-t)^3/6, \\ B_1(t) &= (3t^3 - 6t^2 + 4)/6, \\ B_2(t) &= (-3t^3 + 3t^2 + 3t + 1)/6, \\ B_3(t) &= t^3/6. \end{aligned} \tag{3.5}$$

The B-spline weights are pre-computed for every image location given the image size and the spacing between the control points. Φ in equation 3.4 is estimated similar to [99] by minimizing the difference in the normalized mutual information between the mean M and the frame I that is warped by Φ, in addition to a smoothness constraint. At each iteration of our algorithm, the first frame of the sequence is registered using three refinement levels; starting from a coarse 16×16 grid, then 24×24, and ending with 32×32, where the motion is hierarchically transferred between the levels. The rest of the frames are registered using one level of 32×32 grid as their motion parameters are initialized using the corresponding previous frames; for instance, Φ_2 is initialized as Φ_1. We found such grid sizes adequate for our sequences (frame size of $\sim 256 \times 256$), but they could also be set adaptively [99].

At their current condition, the mean and the frames are at different levels of sharpness since averaging the frames introduces severe blur to the computed mean. In the following, we discuss how we aid the mean-frame correlation through blur kernel estimation.

3.1.1 Frame Blurring

The frame to mean registration process is impeded by the severely blurred mean. However, we found in our experiments that since deblurring the mean does not work, blurring the frame can be employed instead. The intuition behind this is that once the blurry regions in the mean are also blurred in the frame, the registration process is accordingly guided to focus on the sharper regions of the mean rather than the corrupted blurry regions; thus, improving the overall performance. The required blurring depends on several factors such as the amplitude of the waves and the depth of the water. Such factors are hard to model precisely; however, their

effect on the sequence is obvious which is the induced motion. Since the distribution of the positions which a certain tracked point $\mathbf{x}$ attains along the underwater sequence can be approximated with a Gaussian [34], the blur which is induced by the motion can also be approximated with a similar Gaussian. Therefore, without loss of generality, we assume that the blur kernel will be a 2D Gaussian centered at $\mathbf{x}$ with a covariance matrix equals to the covariance matrix of the tracked positions of $\mathbf{x}$ along the sequence

$$\Sigma(\mathbf{x}) = \begin{bmatrix} \sigma_x^2(\mathbf{x}) & \sigma_{xy}(\mathbf{x}) \\ \sigma_{xy}(\mathbf{x}) & \sigma_y^2(\mathbf{x}) \end{bmatrix}, \tag{3.6}$$

where

$$\sigma_x^2(\mathbf{x}) = \frac{\sum_{f=1}^{n}(x - x_f)^2}{n}, \quad \sigma_y^2(\mathbf{x}) = \frac{\sum_{f=1}^{n}(y - y_f)^2}{n},$$
$$\sigma_{xy}(\mathbf{x}) = \frac{\sum_{f=1}^{n}(x - x_f)(y - y_f)}{n},$$

where $(x_f, y_f)^T$ is the location of $\mathbf{x}$ at frame f. Since we have already estimated a dense motion field through registration, we do not need to further track the points, we can rather directly replace $(x - x_f)$ and $(y - y_f)$ with $\Gamma_x(\mathbf{x})$ and $\Gamma_y(\mathbf{x})$ at frame f, which are the x and y components of the estimated motion. Therefore, at each registration iteration k, we use the motion from the previous iteration $\Gamma^{k-1}(\mathbf{x})$ to compute the covariance of the blur kernel

$$\Sigma(\mathbf{x}, k) = \frac{1}{k^2 n} \begin{bmatrix} \sum_f \Gamma_x^2 & \sum_f \Gamma_x \Gamma_y \\ \sum_f \Gamma_x \Gamma_y & \sum_f \Gamma_y^2 \end{bmatrix}, \tag{3.7}$$

where we introduce an additional damping factor k^2 which will make sure that the blur reduces along the iterations. In the first iteration, we find $\Gamma^0(\mathbf{x})$ by an extra registration iteration without frame blurring. Once the kernel is computed, it is used to blur the frames using a fixed-size filter which is large enough to account for the maximum possible motion (we use a 20×20

filter). Figure 3.3 shows our blur estimation for two example sequences from [113]. It can be clearly seen from the figure that the determinant is high over the blurry regions of the mean; thus, the corresponding regions in the frame were blurred accordingly. Generally, the blur in the mean of an underwater sequence is less in the middle regions than the borders, such phenomenon is well captured by our spatially varying filter as illustrated in the figure.

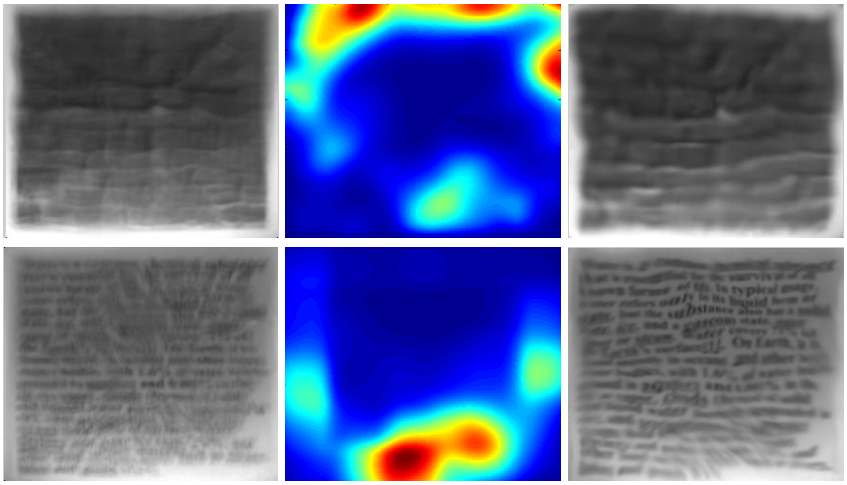

Figure 3.3: The estimated spatially varying filter at the first iteration of the robust registration for the brick sequence (top), and the small font sequence (down). Left to right: The mean, the per-pixel determinant for the motion covariance matrix, and a sample frame blurred using the computed spatially varying filter. Areas with high determinant correspond to areas with high amount of blur in the mean; therefore, the frames were blurred accordingly.

3.2 Sparse Noise Elimination

Applying the robust registration removes most of the distortion caused by the high fluctuations of the waves and generates a better mean image. However, the frames still include several unstructured and random noise caused by reflections and occlusions which the non-rigid registration cannot handle. Interestingly, after applying the first stage, the frames become generally aligned such that their difference can be considered as a sparse error. Therefore,

through rank minimization, we can decompose the matrix which has its columns as the registered frames from the previous stage $F = \{vec\{I_{r1}\}...vec\{I_{r1}\}\}$ into two components; the noise-free matrix A, and the sparse error matrix E

$$\arg\min_{A,E} rank(A) \text{ s.t. } F = A + E, ||E||_0 \leq \beta, \tag{3.8}$$

where β is a constant that represents the maximum number of corrupted pixels expected across all images. Using the Lagrangian form, equation 3.8 can be written as

$$\arg\min_{A,E} rank(A) + \lambda||E||_0 \text{ s.t. } F = A + E, \tag{3.9}$$

where λ is a parameter that trades off the rank of the solution versus the sparsity of the error, and we always set it to $1/\sqrt{(w \times h)}$ following the theoretical considerations in [22]. Consequently, we apply convex relaxation to the problem by replacing $rank(A)$ with the nuclear norm or sum of the singular values $||A||_* = \Sigma_i(\sigma_i)$, and replacing $||E||_0$ with its convex surrogate ℓ_1 norm $||E||_1$

$$\arg\min_{A,E} ||A||_* + \lambda||E||_1 \text{ s.t. } F = A + E. \tag{3.10}$$

Equation 3.10 is convex and can be solved with convex optimization methods such as the Augmented Lagrange Multiplier (ALM) algorithm [71] which we found robust and fast in our scenarios. The final output matrix A comprises in its columns the final reconstructed frames $\{I_{f1}...I_{fn}\}$.

3.2.1 From Stage One to Stage Two

At the first sight, our combination of stages might seem adhoc, while in fact the two stages are tightly connected and complement each other in recovering the underwater scene. As discussed earlier in this chapter, the water turbulence induces two noise components in

the underwater sequence; a local miss-alignment, and a random noise. The random noise can only be unveiled by rank minimization if it is reasonably sparse [22]. However, in the raw video, the local-miss alignment is dominant, and conceals the random noise. For that reason, the sparse noise elimination can only be utilized after first aligning the sequence through the robust registration. On the other hand, the robust registration itself fails to overcome the sparse errors, and that is where the second stage comes in handy. Figure 3.4 shows the result of applying the stages separately and then combined. It is clear that each stage plays an essential role in robustly reconstructing the underwater scene.

One issue arises in this formulation: When can we consider the sequence well-aligned such that the current errors are reasonably sparse? In other words, when can we move from stage 1 to stage 2? Our experiments indicate that the answer to such a question is still open-ended, because computing an exact parameter quantifying the current sparsity of error intrinsically requires the error to be first identified, which can only be available after stage 2. Interestingly, we found in our experiments that some quantities correlated to sparsity can be used as robust indicators of sparsity. Namely, we use the ℓ_1 difference between the frames and the mean of the current sequence V to make a stopping decision for stage 1

$$\ell_1(V) = \frac{\sum_i \sum_{\mathbf{x}} |M(\mathbf{x}) - I_i(\mathbf{x})|}{w \times h \times n}. \tag{3.11}$$

After normalizing the images to $[0 - 1]$, a fixed sparsity threshold of $.025$ on ℓ_1 worked quite well in all sequences. Technically, there is absolutely no penalty for reducing the threshold other than a possible waste of processing time. Such a threshold is directly related to the parameter λ in equation 3.10 since both are limits for the number of expected corrupted pixels.

The investigation of such a relation goes beyond the scope of this work; thus, we leave it for future work. The complete "Seeing through water" procedure is summarized in algorithm 2.

Algorithm 2: Seeing through water

input : Distorted image set $V \in \mathbb{R}^{w \times h \times n} = \{I_1, ..., I_n\}$

output: Undistorted image set $V_f \in \mathbb{R}^{w \times h \times n} = \{I_{f1}, ..., I_{fn}\}$

$M \leftarrow TemporalMean(V)$;

Stage 1: Robust Registration begin

while $\ell_1(V) > Sparsity\_Threshold$ **do**

$B \leftarrow ComputeBlurKernel(\Gamma)$;

$V_b \leftarrow$ Convolve the frames V with B;

$\Gamma \leftarrow$ Compute the warping from each frame in V_b to the mean M;

$V_w \leftarrow$ Warp the unblurred set V using Γ;

$V \leftarrow V_w$;

$M \leftarrow TemporalMean(V)$;

end

$V_r \leftarrow V$;

end

Stage 2: Sparse Noise Elimination begin

$F \leftarrow vec\{V_r\} = \{vec(I_{r1}), ..., vec(I_{rn})\}$;

$A, E \leftarrow \arg\min_{A,E} ||A||_* + \lambda||E||_1$ s.t. $F = A + E$;

$\{I_{f1}, ..., I_{fn}\} \leftarrow$ Reshape A to $w \times h \times n$;

end

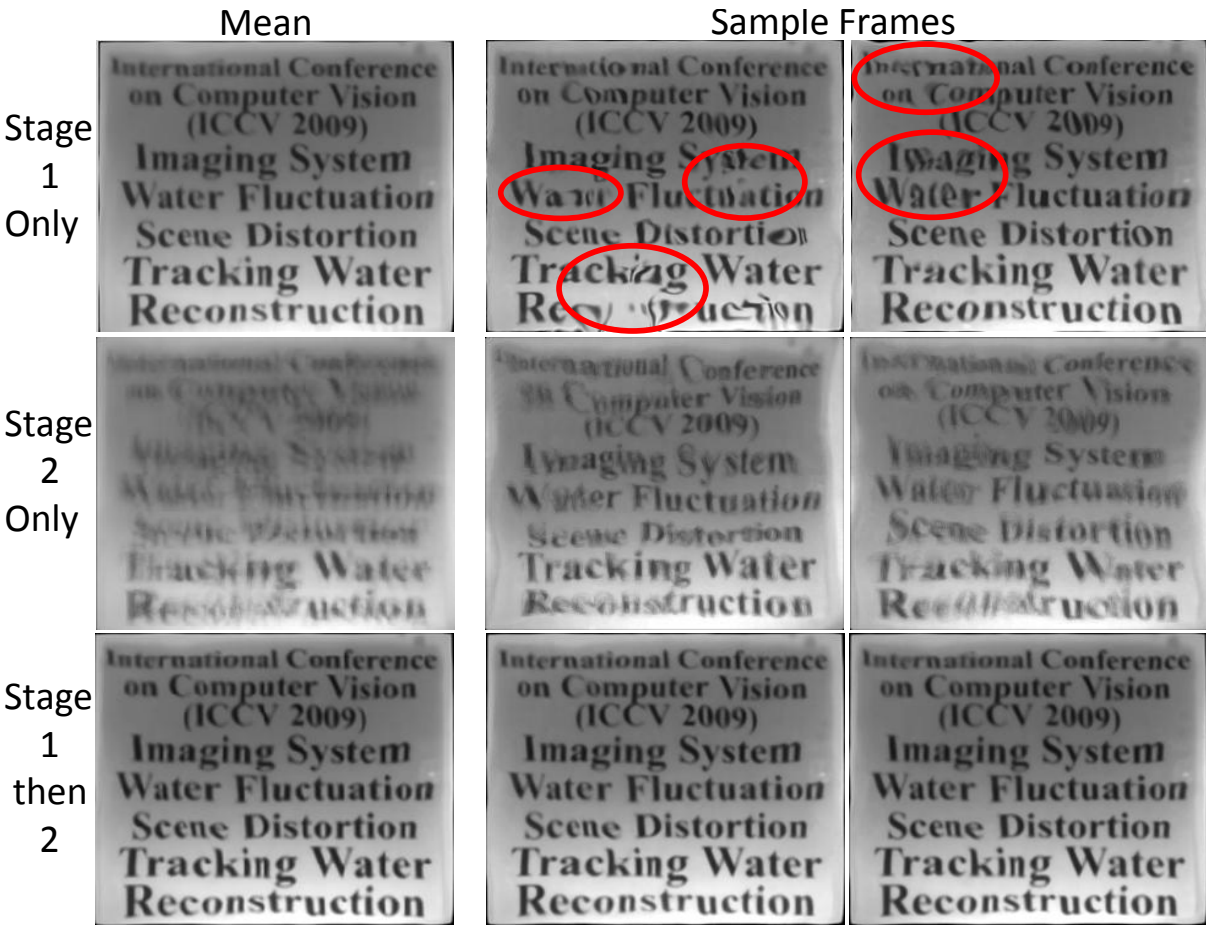

Figure 3.4: Mean image and sample frames after applying the robust registration without the sparse noise elimination (top), the sparse noise elimination without first applying the registration (middle), and both stages applied (bottom). The first stage itself is capable of obtaining a robust mean; however, the frames still contain several sparse errors (highlighted in red). Applying the second stage to the raw video fails for the reason discussed in the text. Combining the stages clearly achieves the best results.

3.3 Experiments

We extensively experimented on the proposed ideas using several standard underwater sequences from [113]. Each frame is 256×256 with 61 frames per sequence. The robust registration stage converges in $3-6$ iterations, while the sparse noise elimination stage quickly converges in $40 - 43$ ALM iterations. Overall, our algorithm takes about half of the time required by [113]. Figure 3.7 shows our output compared to the state of the art [113]. Our

algorithm is able to reconstruct all the underwater scenes and generate superior high quality means. It can be noticed from the figure that the mean of the sequence does not considerably improve after the sparse noise elimination, this essentially indicates that the majority of the remaining errors after the registration step belong to a zero mean additive noise. The effect of the sparse noise elimination, however, can be clearly observed on the frames as illustrated in Figure 3.5. Additionally, the role of each stage is better observed in the video stabilizing results provided on our website.

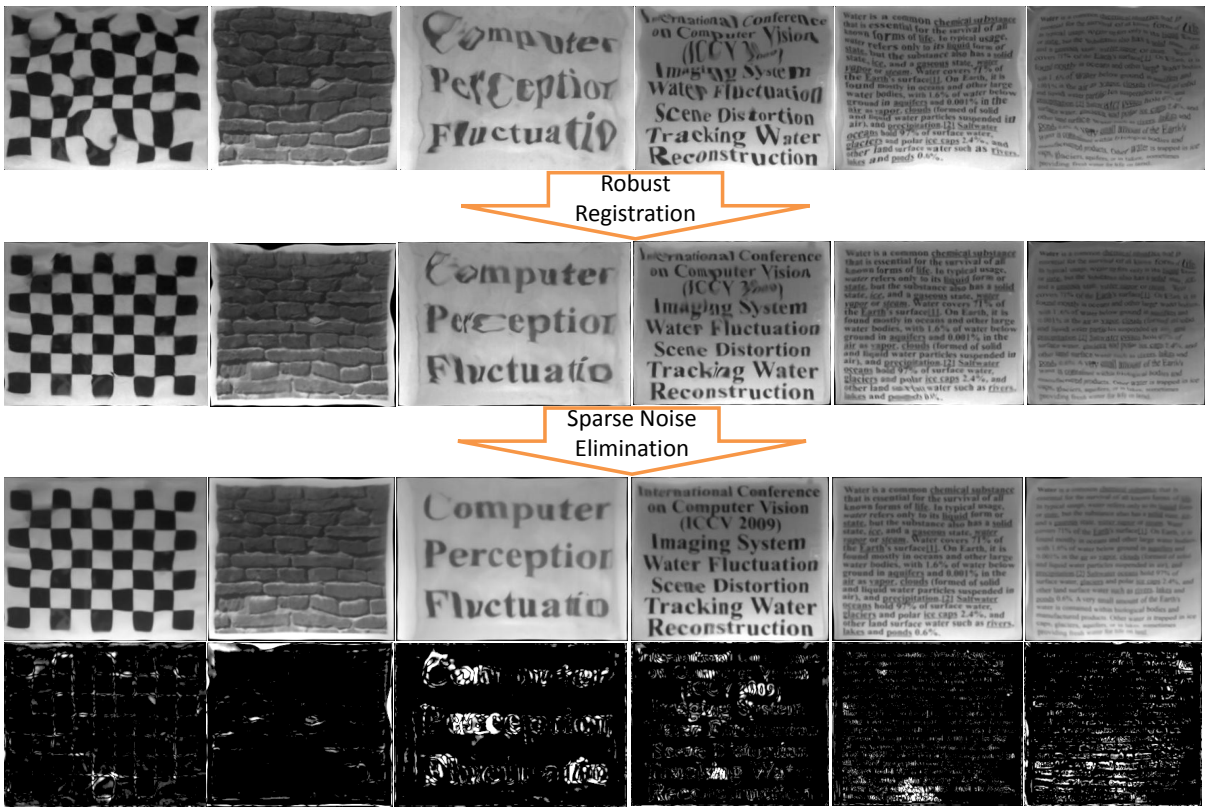

Figure 3.5: A sample frame from each sequence after applying each stage of our algorithm. The first stage overcomes most water turbulence; however, sparse errors are only eliminated after the second stage. The final two rows show the reconstructed images and the sparse errors respectively after rank minimization. (Please zoom in to see the details, and refer to our website for the complete videos).

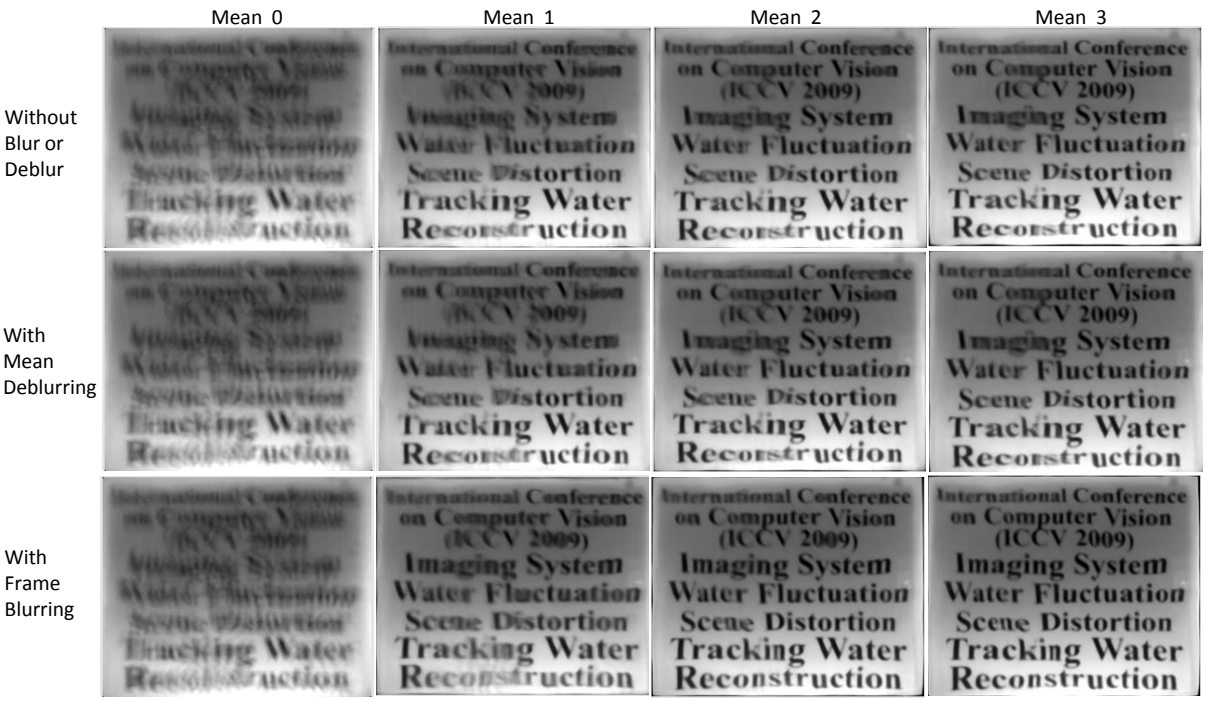

Figure 3.6: Evolution of the mean in stage 1. Left to right: The mean after each iteration of registration. Top to bottom: Stage 1 applied without blurring or deblurring, with mean deblurring, and with our frame blurring. After three iterations, the mean is significantly enhanced in all cases. However, underwater words on the left part of the image like “Imaging”, “Water”, “Scene”, “Tracking”, and “Reconstruction” are only correctly reconstructed using the frame blurring. (Please zoom in to see the details).

Figure 3.6 shows an example for the evolution of the mean for the middle font sequence from [113] during stage 1 of the algorithm under three cases: direct registration without blurring or deblurring, with mean deblurring using [127], and finally with frame blurring. It is clear from the figure that our robust registration algorithm is capable of finding a high quality mean in a few iterations in all cases. However, frame blurring evidently achieves the best results with all underwater words correctly reconstructed.

We use the reconstructed mean M and the template T provided from [113] to quantitatively compare with the results from [113] using four standard performance metrics:

- Sum of squared differences (SSD):

$$SSD(M,T) = \frac{\sum_{\mathbf{x}}(M(\mathbf{x}) - T(\mathbf{x}))^2}{w \times h}. \tag{3.12}$$

- Normalized Mutual Information (NMI):

$$NMI(M,T) = \frac{(H(M) + H(T))}{H(M,T)}, \tag{3.13}$$

where $H(M)$, $H(T)$ are the entropies of M and T, and $H(M,T)$ is the joint entropy of M and T. H is calculated from the histograms of the gray values of the images. This measure is heavily used in image registration such as [99]. Therefore, here it is a strong indication of how well the two images are aligned.

- Local Normalized Mutual Information (LNMI): This is similar to NMI except that it is computed for every patch of a 10×10 grid, and then normalized. LNMI captures the spatial relations among the image parts.

- SSD in Gradient (SSDG):

$$SSDG(M,T) = SSD(M_x, T_x) + SSD(M_y, T_y), \tag{3.14}$$

where M_x, T_x, M_y, T_y are the horizontal and vertical gradients for M and T. Gradient-based features were proved to be crucial in text recognition [107]. Since many of our testing sequences contain underwater text, we use this measure to compare text recognition accuracy.

Table Table 4.1 summarizes the obtained results for each of the sequences with an available template after normalizing the images to $[0-1]$. It is clear that our method drastically outperforms [113] in all sequences in terms of all measures. It is important to note that our

proposed method should not be compared to [114] since it assumes a different formulation where the template of the sequence is used.

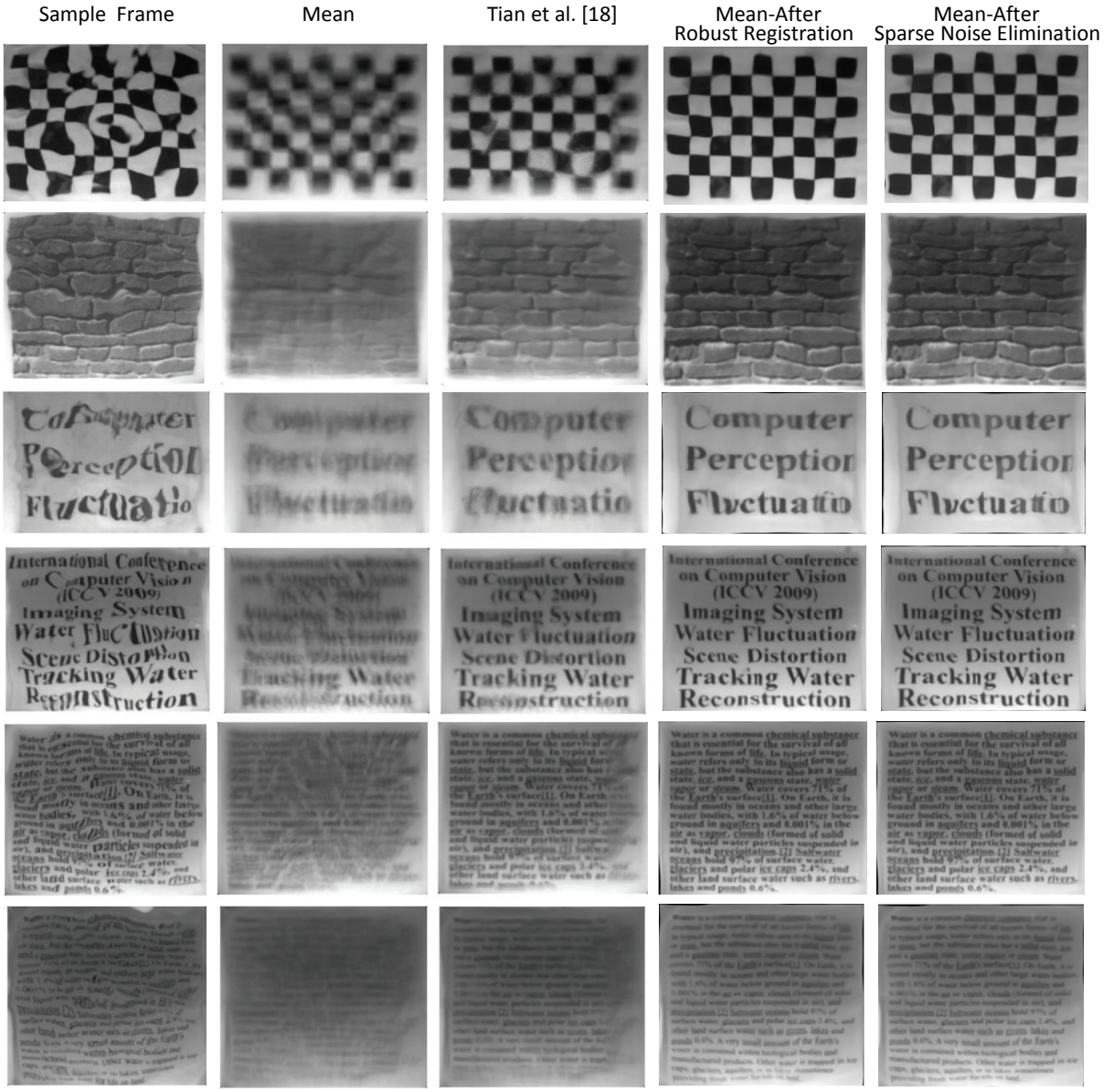

Figure 3.7: Image restoration results on standard sequences from [113]. The first column shows a sample frame from the input video, which is severely distorted. The second column shows the temporal mean of the sequence. The third column is the result from [113]. Finally our results are shown in the last two columns, after the first and second stages respectively. Results from our method clearly outperform [113] on all sequences even after the first stage. (Please zoom in to see the details).

Table 3.1: Performance of our method compared to [113].

	SSD	NMI	LNMI	SSDG
Sequence	Tian et al. / Ours	Tian et al. / Ours	Tian et al. / Ours	Tian et al. / Ours
Brick	0.019 / 0.009	1.083 / 1.101	1.116 / 1.138	0.007 / 0.004
Middle Font	0.027 / 0.011	1.072 / 1.178	1.147 / 1.223	0.012 / 0.005
Small Font	0.021 / 0.008	1.046 / 1.118	1.100 / 1.168	0.018 / 0.006
Tiny Font	0.017 / 0.011	1.088 / 1.134	1.091 / 1.133	0.006 / 0.004

3.4 Summary

In this chapter, we discussed a novel approach to reconstruct a sequence going through water turbulence by discovering its low-rank subspace. Our method inherently assumes that the original scene (without the turbulence) is static. Though this assumption is valid in many practical scenarios, it still restricts the method's applications to the cases where non-stationary objects exist in the scene. Therefore, in the next chapter, we propose a framework to handle this more challenging scenario, by extending the low-rank decomposition into a three-term optimization, which simultaneously discovers the turbulence, the moving objects, and the original scene background.

CHAPTER 4: SIMULTANEOUS TURBULENCE MITIGATION AND MOVING OBJECT DETECTION

The refraction index of the air varies based on several atmospheric characteristics including the air's temperature, humidity, pressure, carbon dioxide level, and dust density. Such conditions are typically not homogeneous; for instance, a non-uniform temperature distribution might be observed above a surface receiving sunlight. Therefore, light rays travelling through the air with such non-uniform changes in its relative refraction index, will go through a complex series of refraction and reflection causing extreme spatially and temporally varying deformations to the captured images [98, 67, 45, 79, 135].

On the other hand, if the objects of interest are additionally moving in the scene, their motion will be mixed up with the turbulence deformation in the captured images, rendering the problem of detecting the moving objects extremely difficult. In this chapter, we are interested in the dual problem of turbulence mitigation (stabilizing the sequence) and moving object detection under the turbulent medium. Relevant previous approaches have either focused on detecting moving objects or de-warping a deformed sequence, but not on both tasks concurrently.

Given a sequence of frames $\{I_1, ..., I_T\}$ acquired from a stationary camera residing in a turbulent medium while observing relatively tiny moving objects, we decompose the sequence into background, turbulence, and object components. More precisely, consider the frames matrix $F = [vec\{I_1\} \cdots vec\{I_T\}]$ for $I_k \in \mathbb{R}^{W \times H}$ $(k = 1, 2, ..., T)$, where $W \times H$ denotes the

frame resolution (width by height), and $vec : \mathbb{R}^{W \times H} \rightarrow \mathbb{R}^{M}$ is the operator which stacks the image pixels as a column vector. We formulate our decomposition of F as:

$$\min_{A,O,E} Rank(A) \text{ s.t. } F = A + O + E, \tag{4.1}$$
$$||O||_0 \leq s, \ \ ||E||_F \leq \sigma,$$

where F, A, O, and E are the matrices of frames, background, object, and error (turbulence), respectively. Here, the $\| \cdot \|_0-$norm counts the number of nonzero entries, $\| \cdot \|_F-$norm is the Frobenius norm which is equal to the square root of the sum of squared elements in the matrix, s represents an upper bound of the total number of moving objects' pixels across all images, and σ is a constant which reflects our knowledge of the maximum total variance due to corrupted pixels across all images.

Our decomposition is based on the intrinsic properties of each of the components:

1. The Background: The scene in the background is presumably static; thus, the corresponding component in the frames of the sequence has linearly correlated elements. Therefore, the background component is expected to be the part of the matrix which is of low rank. Minimizing the rank of the low-rank component of the frames matrix F emphasizes the structure of the linear subspace containing the column space of the background, which reveals the background.

2. The Turbulence: Previous work dealing with turbulence, such as [106, 90, 132, 24, 34], demonstrated that the fluctuations of fluids (for instance air and water) attain Gaussian-like characteristics such as being unimodal, symmetric, and locally repetitive; therefore,

the projected deformations in the captured sequence often approach a Gaussian distribution (we discuss this in more detail in Section 3.2). For this reason, the turbulence component can be captured by minimizing its Frobenius norm. The Frobenius norm of a matrix is the same as the Euclidean norm of the vector obtained from the matrix by stacking its columns. Therefore, as in the well-known vector case, constraining the error in the Euclidean norm is equivalent to controlling the sample variance of the error. Furthermore, theoretically, the estimate obtained by the Frobenius norm has several desirable statistical properties [12].

3. The Moving Objects: We assume that the moving objects are sparse in the sequence. This means that the number of pixels occupied by the moving objects is small (or can be considered as outliers) compared to the total number of pixels in the frames. This is a reasonable assumption for most realistic surveillance videos. For this reason, the moving objects are best captured by restricting the number of nonzero entries (denoted by the ℓ_0 norm of the matrix), which is desirable for finding outliers.

In practice, parts of the turbulence could also appear as sparse errors in the object matrix O. Therefore, an additional constraint needs to be enforced on the moving objects. We employ a simple turbulence model to compute an object confidence map which is used to encourage the sparse solutions to be located on regions exhibiting linear motion that is dissimilar from the fluctuations of the turbulence. Under the new constraint, the optimization problem (4.1) must be reformulated as:

$$\min_{A,O,E} Rank(A) \text{ s.t. } F = A + O + E, \tag{4.2}$$

$$||\Pi(O)||_0 \leq s, \ \ ||E||_F \leq \sigma,$$

where $\Pi : \mathbb{R}^{M \times T} \rightarrow \mathbb{R}^{M \times T}$ is the object confidence map, which is a linear operator that weights the entries of O according to their confidence of corresponding to a moving object such that the most probable elements are unchanged and the least are set to zero.

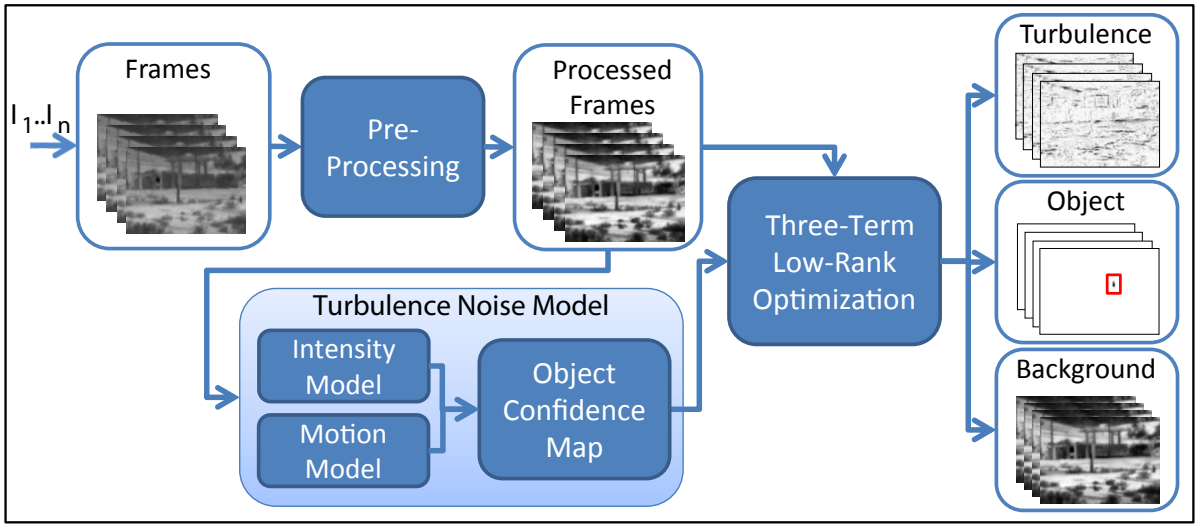

Figure 4.1: The various steps of the proposed algorithm.

Figure 4.1 shows a diagram of the proposed approach. We first apply a pre-precessing step to improve the contrast of the sequence, and reduce the spurious and random noise. Consequently, we obtain an object confidence map using a turbulence model which utilizes both the intensity and the motion cues. Finally, we decompose the sequence into its components using three-term rank minimization.

4.1 Proposed Approach

We decompose the matrix which contains the frames of the turbulence video, into its components: the background, the turbulence, and the objects. The decomposition is performed by solving the rank optimization in equation (4.2), which enforces relevant constraints on each component. In the next subsection we describe the details of the decomposition approach.

4.1.1 Three-Term Decomposition

When solving equation (4.2), it is more convenient to consider the Lagrange form of the problem:

$$\min_{A,O,E} Rank(A) + \tau||\Pi(O)||_0 + \lambda||E||_F^2 \qquad (4.3)$$
$$\text{s.t. } F = A + O + E,$$

where τ and λ are weighting parameters. The optimization of (4.3) is not directly tractable since the matrix rank and the ℓ_0-norm are nonconvex and extremely difficult to optimize. However, it was recently shown in [22] that when recovering low-rank matrices from sparse errors, if the rank of the matrix A to be recovered is not too high and the number of non-zero entries in O is not too large, then minimizing the nuclear norm of A (sum of singular values $\sum_i \sigma_i(A)$) and the ℓ_1-norm of O can recover the exact matrices. Therefore, the nuclear norm and the ℓ_1-norm are the natural convex surrogates for the rank function and the ℓ_0-norm, respectively. Applying this relaxation, our new optimization becomes:

$$\min_{A,O,E} ||A||_* + \tau||\Pi(O)||_1 + \lambda||E||_F^2 \text{ s.t. } F = A + O + E, \tag{4.4}$$

where $||A||_*$ denotes the nuclear norm of matrix A. We adopt the Augmented Lagrange Multiplier method (ALM) [71] to solve the optimization problem (4.4). Define the augmented Lagrange function for the problem as:

$$L(A, O, E, Y) = ||A||_* + \tau||\Pi(O)||_1 + \lambda||E||_F^2 + \tag{4.5}$$
$$\langle Y, F - A - O - E\rangle + \frac{\beta}{2}||F - A - O - E||_F^2,$$

where $Y \in \mathbb{R}^{M\times T}$ is a Lagrange multiplier matrix, β is a positive scalar, and $\langle , \rangle$ denotes the matrix inner product (trace($A^T B$)). Minimizing the function in equation (4.5) can be used to solve the constrained optimization problem in equation (4.4). We use the ALM algorithm to iteratively estimate both the Lagrange multiplier and the optimal solution by iteratively minimizing the augmented Lagrangian function:

$$(A_{k+1}, O_{k+1}, E_{k+1}) = \arg\min_{A,O,E} L(A, O, E, Y_k), \tag{4.6}$$
$$Y_{k+1} = Y_k + \beta_k(F_{k+1} - A_{k+1} - O_{k+1} - E_{k+1}).$$

When β_k is a monotonically increasing positive sequence, the iterations converge to the optimal solution of problem (4.4) [11]. However, solving equation (4.6) directly is difficult; therefore, the solution is approximated using an alternating strategy minimizing the augmented Lagrange function with respect to each component separately:

$$
\begin{aligned}
A_{k+1} &= \arg\min_{A} L(A, O_k, E_k, Y_k), \\
O_{k+1} &= \arg\min_{O} L(A_{k+1}, O, E_k, Y_k), \\
E_{k+1} &= \arg\min_{E} L(A_{k+1}, O_{k+1}, E, Y_k).
\end{aligned} \tag{4.7}
$$

Following the idea of the singular value thresholding algorithm [19], we derive the solutions for the update steps in equation (4.7) for each of the nuclear, Frobenius, and ℓ_1 norms. Please refer to the supplementary appendix for the complete derivations. Consequently, a closed form solution for each of the minimization problems is found:

$$
\begin{aligned}
UWV^T &= \text{svd}(F - O_k - E_k + \beta_k^{-1} Y_k), \\
A_{k+1} &= U S_{1/\beta_k}(W) V^T, \\
O_{k+1} &= S_{\tau/\beta_k \Pi}(F - A_{k+1} - E_k + \beta_k^{-1} Y_k), \\
E_{k+1} &= (1 + \frac{2\lambda}{\beta_k})^{-1} (\beta_k^{-1} Y_k + F - A_{k+1} - O_{k+1}),
\end{aligned} \tag{4.8}
$$

where $\text{svd}(M)$ denotes a full singular value decomposition of matrix M, and $S_\alpha(\cdot)$ is the soft-thresholding operator defined for a scalar x as:

$$
S_\alpha(x) = \text{sign}(x) \cdot \max\{|x| - \alpha, 0\}, \tag{4.9}
$$

and for two matrices $A = (a_{ij})$ and $B = (b_{ij})$ of the same size, $S_A(B)$ applies the soft-thresholding entry-wise outputting a matrix with entries $S_{a_{ij}}(b_{ij})$.

The steps of our decomposition are summarized in Algorithm 4. In the next subsection, we describe our method to obtain the moving object confidence map Π, which is employed as a prior in the rank minimization problem.

4.1.2 Turbulence Model

We employ a turbulence model to enforce an additional constraint on the rank minimization such that moving objects are encouraged to be detected in locations with non-Gaussian deformations. Exact modelling of the turbulence is in fact ill-posed as it follows a non-uniform distribution which varies significantly in time, besides having an additional complexity introduced during the imaging process; thus, rendering the problem of modelling turbulence extremely difficult. Although the refraction index of the turbulent medium is often randomly changing, it is also statistically stationary [45, 46, 15]; thus, the deformations caused by turbulence are generally repetitive and locally centered [106, 90, 132, 24, 52]; this encourages the use of Gaussian-based models as approximate distributions that are general enough to avoid overfitting, but rather capture significant portion of the turbulent characteristics.

We use a Gaussian function to model the intensity distribution of a pixel going through turbulence. This is similar to [109] which employs a mixture of Gaussians; however, we found that a single Gaussian worked better since more complicated models often require a period of training which is not available in our sequences. Therefore, the intensity of a pixel at location $\mathbf{x}$ is modelled using a Gaussian distribution:

$$I(\mathbf{x}) \sim \mathcal{N}(\mu_I, \sigma_I), \tag{4.10}$$

where μ_I and σ_I are the mean and the standard deviation at $\mathbf{x}$, respectively. On the other hand, the deformation caused by turbulence can be captured in the motion domain besides the intensity. Therefore, we combine the intensity and the motion features to obtain a better model of turbulence. In order to capture the ensemble motion in the scene, we use the concept of a "particle" in a Lagrangian particle trajectory acquisition approach. We assume that a grid of particles is overlaid onto a scene where each particle corresponds to a single pixel (the granularity is controllable). The basic idea is to quantify the scene's motion in terms of the motion of the particles which are driven by dense optical flow. A so-called particle advection [7, 125, 126] procedure is applied to produce the particle trajectories. Given a video clip $\in \mathbb{R}^{W\times H\times T}$, we denote the corresponding optical flow by (U_w^t, V_h^t), where $w \in [1, W]$, $h \in [1, H]$, and $t \in [1, T-1]$. The position vector (x_w^t, y_h^t) of the particle at grid point (w, h) at time t is estimated by solving the following differential equations:

$$\begin{aligned} \frac{dx_w^t}{dt} &= U_w^t, \\ \frac{dy_h^t}{dt} &= V_h^t. \end{aligned} \tag{4.11}$$

We use Euler's method to solve them, similar to [125]. By performing advection for the particles at all grid points with respect to each frame of the clip, we obtain the clip's particle trajectory set, denoted by $\{(x_w^t, y_h^t) | w \in [1, W], h \in [1, H], t \in [1, T]\}$.

We employ the spatial locations of the particle trajectories (i.e. (x_w^t, y_h^t)) to model the turbulence motion in the scene. The locations visited by a particle moving due to the fluctuations of the turbulence have a unimodal and symmetric distribution which approaches a Gaussian [106, 46, 34, 24]. This is dissimilar from the linear motion of the particles driven

by moving objects. Therefore, we associate each particle with a Gaussian with mean μ_M and covariance matrix Σ_M:

$$\mathbf{x} \sim \mathcal{N}(\mu_M, \Sigma_M). \tag{4.12}$$

By augmenting the intensity model in equation (4.10) with the motion model in equation (4.12), the total confidence of corresponding to the turbulence versus the moving objects for a particle at location $\mathbf{x}$ is expressed as a linear opinion pooling of the motion and the intensity cues

$$C(\mathbf{x}) = w\mathrm{P}(I(\mathbf{x})|\mu_I, \sigma_I) + (1-w)\mathrm{P}(\mathbf{x}|\mu_M, \Sigma_M). \tag{4.13}$$

The parameters of our model $\{w, \mu_I, \sigma_I, \mu_M, \Sigma_M\}$ can be learned by optimization using training sequences or set to constant values selected empirically. In the context of our three-term decomposition, the obtained confidence provides a rough prior knowledge of the moving objects' locations, which can be incorporated into the matrix optimization problem in equation (4.4). Interestingly, this prior employs motion information; therefore, it is complementary to the intensity-based rank optimization, and can significantly improve the result.

At frame t, we evaluate all the particles' locations against their corresponding turbulence models and obtain the turbulence confidence map $C_t \in \mathbb{R}^{W \times H}$. While C_t corresponds to the confidence of a particle to belong to turbulence, the desired Π in equation (4.4) corresponds to the confidence of belonging to the moving objects; therefore, we define the object confidence map Π as the complement of the stacked turbulence confidence maps:

$$\Pi = 1 - [vec\{C_1\} \cdots vec\{C_T\}]. \tag{4.14}$$

4.1.3 Restoring Force

The particles carrying the object's motion typically drift far from their original locations leaving several gaps in the sequence. In the presence of turbulence, the drifting also occurs as a result of the turbulent motion. Therefore, the particles need to be reinitialized every certain number of frames which, however, creates discontinuities. This is a typical hurdle in the Lagrangian framework of fluid dynamics [7, 125, 126], which constitutes a major impediment for the application of particle flow to turbulence videos. In order to handle the drifting and the discontinuity problems associated with the particle flow, we use a new force component in the advection equation:

$$\begin{aligned} \frac{dx_w^t}{dt} &= U_w^t + G(x, x_o), \\ \frac{dy_h^t}{dt} &= V_h^t + G(y, y_o). \end{aligned} \tag{4.15}$$

We refer to the new force as "Restoring Force" - a reference to a local restoration force acting in the direction of the original location of each particle. We use a simple linear function to represent the restoring force:

$$G(x, x_o) = \frac{x - x_o}{\gamma}, \tag{4.16}$$

where γ is a scaling factor which trades off the detection sensitivity and the speed of recovery for the particles. In other words, if γ is set to a high value, the effect of the restoring force will

be negligible, and therefore the particles will require a relatively longer time to return to their original positions. In this case, the sensitivity of moving object detection will be higher, but more prone to false positives. If γ is low, the particles will be more attached to their original location, thus less affected by turbulence, but will have lower detection sensitivity. In our experiments, we set γ to $0.5 \times W = 125$, which we found to be adequate for all sequences.

Using the restoring force allows continuous processing of the sequence without the need to reinitialize the particles. For instance, if an object moves to one side of the frame then comes back, we can still capture its motion when it returns. Additionally, the restoring force maintains the particles' motion within a certain range and provides robustness against random noise, thus reducing the number of false object detections. Figure (Figure 4.2) shows the overlaid particles on selected frames and the corresponding object particles with and without restoring force. It is clear that the restoring force stabilizes the particles and delivers better moving object confidence.

4.2 Algorithm and Implementation Details

4.2.1 Pre-Processing

The noise caused by turbulence often has several random and spurious components which are difficult to model. Therefore, we employ temporal averaging to mitigate such components. We use a small averaging window of 8 frames to avoid distorting the objects' motion. In order to improve the contrast of the sequences, we apply adaptive histogram equalization, which is similar to the ordinary histogram equalization, however it operates on small patches

of size 8×8. Deviation around the optimal values of such parameters only causes smooth and graceful degradation in the results.

Algorithm 3: Simultaneous Turbulence Mitigation and Moving Object Detection

---

[h!] **Input** : Distorted image set stacked as

$$F \in \mathbb{R}^{M \times T} = [vec\{I_1\}...vec\{I_T\}]$$

Initial Object Confidence Map

$$\Pi : \mathbb{R}^{M \times T} \rightarrow \mathbb{R}^{M \times T}$$

Output: Solution to equation (4.1)

Background $A \in \mathbb{R}^{M \times T}$

Turbulence $E \in \mathbb{R}^{M \times T}$

Moving Objects $O \in \mathbb{R}^{M \times T}$

while *not converged* **do**

Minimize the Lagrange function in equation (4.5)

$UWV^T = \text{svd}(F - O_k - E_k + \beta_k^{-1} Y_k)$, $A_{k+1} = US_{\frac{1}{\beta_k}}(W)V^T$,

$O_{k+1} = S_{\frac{\tau}{\beta_k}\Pi}(F - A_{k+1} - E_k + \beta_k^{-1} Y_k)$,

$E_{k+1} = (1 + \frac{2\lambda}{\beta_k})^{-1}(\beta_k^{-1} Y_k + F - A_{k+1} - O_{k+1})$,

$Y_{k+1} = Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_{k+1})$,

$\beta_{k+1} = \rho \beta_k$.

end

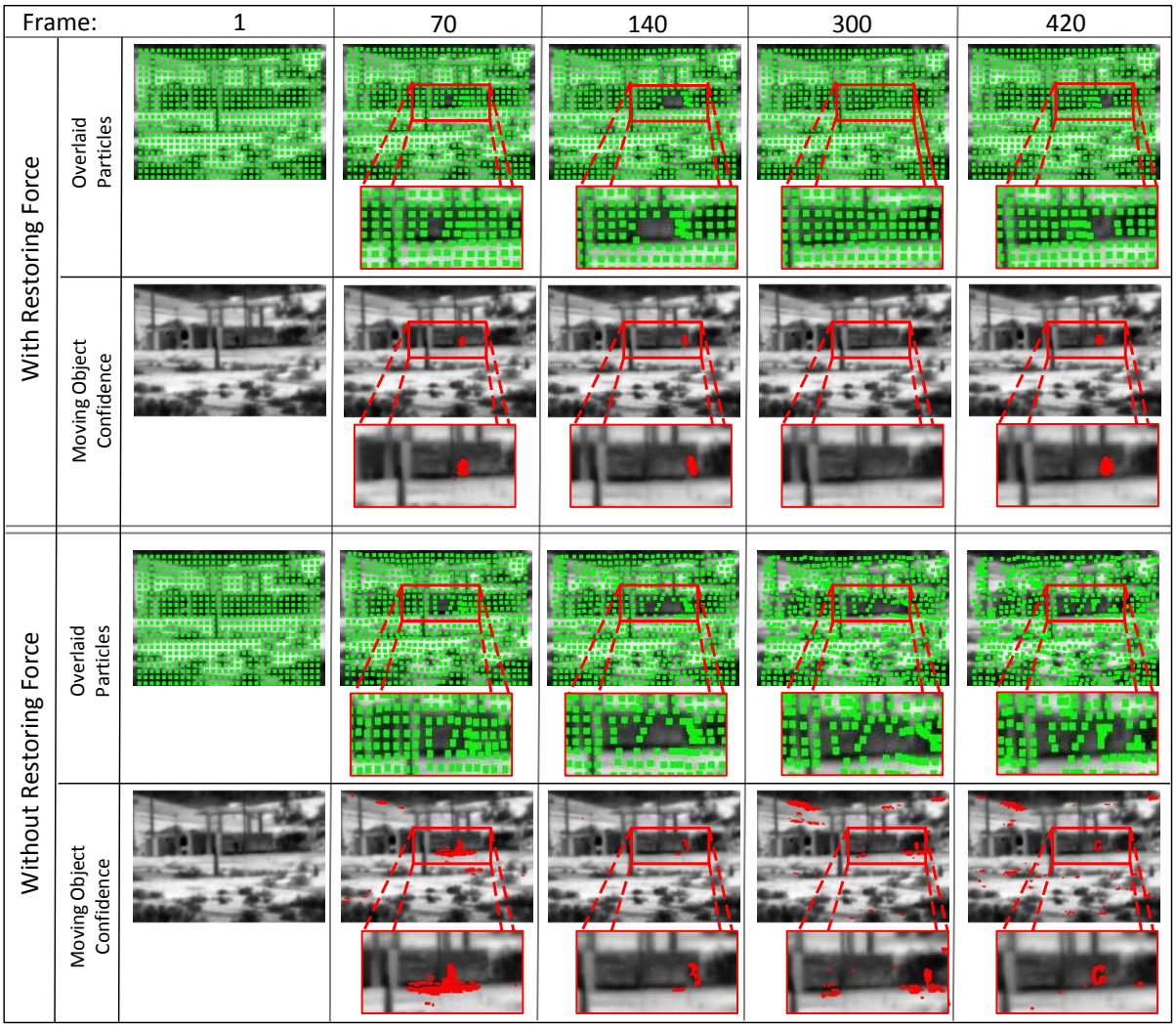

Figure 4.2: Overlaid particles (the end points of the particle trajectories), and the corresponding moving object confidence (equation (4.14)) for sample frames. Rows one and three show the overlaid particles (the granularity is reduced for better visualization). Shown in red, in rows two and four, are the particles with high confidence of belonging to a moving object rather than turbulence (a fixed threshold is used). After frame 1, the object starts moving to the right, gets occluded at frame 300, then moves back to the left at frame 420. In the top two rows, the restoring force is employed; therefore, the particles gain two new properties: First, they become intact and attached to their original position, which make them robust against drifting due to the turbulence. Second, the particles automatically return to their original positions after the moving object disappears, which can be seen on frame 300. In the bottom two rows, the restoring force is not employed and such properties are not available; therefore, the particles continue to float and drift along the sequence, resulting in a poor object confidence performance.

4.2.2 *Determining the Optimization Parameters*

The parameters τ and λ from equation (4.4) correspond to the total number of moving objects' pixels across all images, and the total variance due to corrupted pixels across all images, respectively. In other words, a higher τ leads to an increased significance in minimizing the O component, thus obtaining sparser moving objects. A higher λ leads to an increased significance in minimizing the noise in E component, thus obtaining less noise in E, and a more turbulent background. On the other hand, decreasing τ and λ leads to placing more emphasis on minimizing the rank, thus obtaining a more static background in A, large turbulence in E, and less sparse moving objects in O.

Several theoretical considerations were previously studied to derive optimal values for similar parameters in [22]. However, such analysis does not apply to all practical scenarios. In the context of our three-term decomposition, the matrix to be decomposed is not an exact composition of the expected components. In addition, the components do not correspond exactly to their expected model. For instance, the background is expected to be of low rank; however, the exact desired rank is debatable, as a background of rank 1 is often not desirable since it will be a repetition of a static image, which is not realistic. Similarly, the desired sparsity of error varies significantly among different applications. Therefore, we argue that, in practical scenarios, such parameters are problem-dependant and highly heuristic. To this end, we empirically set τ to 0.1 and λ to 2.0, which, as we will show, worked as a good compromise among the constraints of the optimization in all sequences.

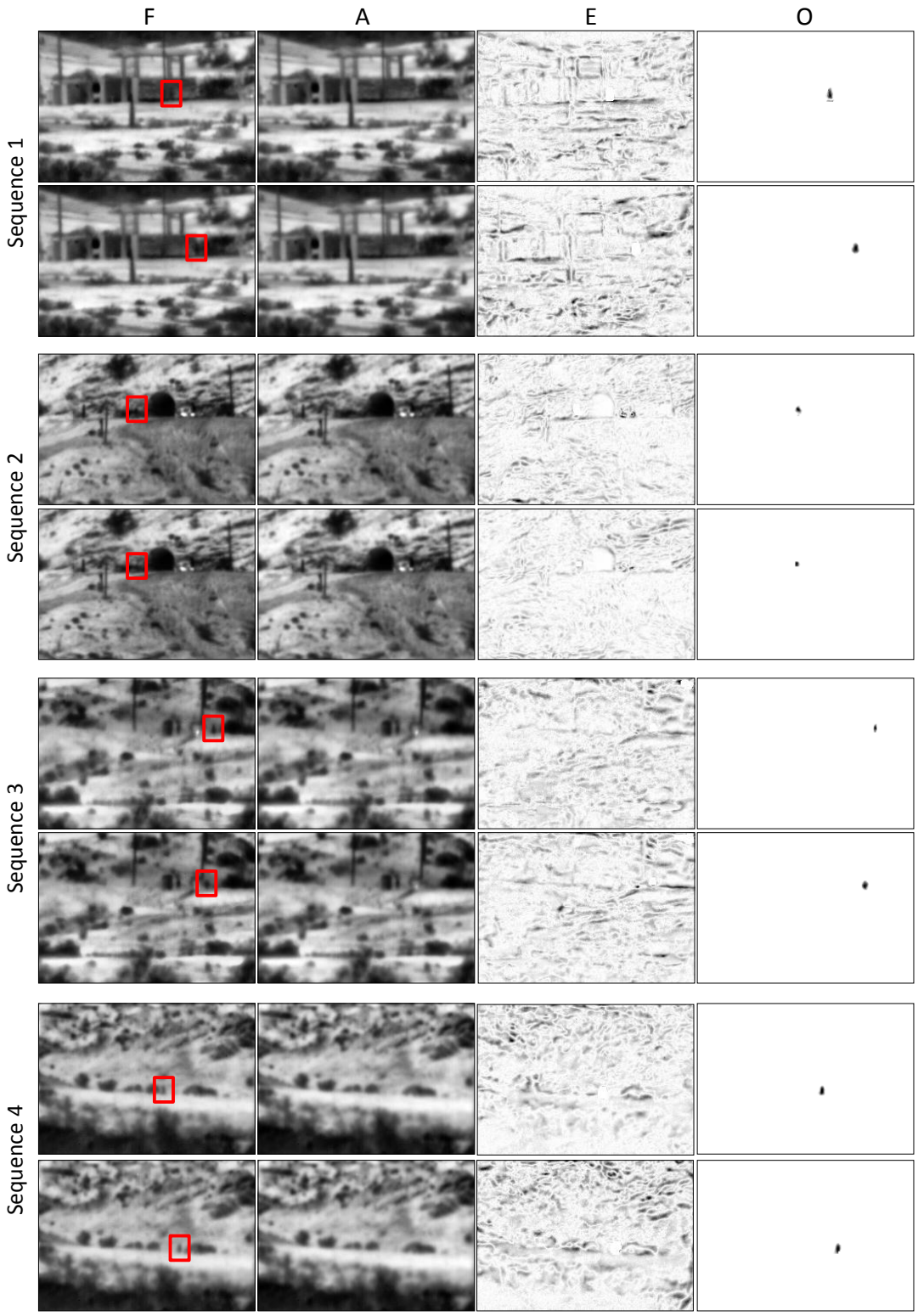

Figure 4.3: Three-term decomposition results for two example frames from four testing sequences. Column F shows the original sequence (after pre-processing) which was decomposed into background (column A), turbulence (column E, absolute value of E is shown), and moving object (column O). Please refer to our website for the complete videos as the results of correcting the deformations are difficult to observe in a single frame.

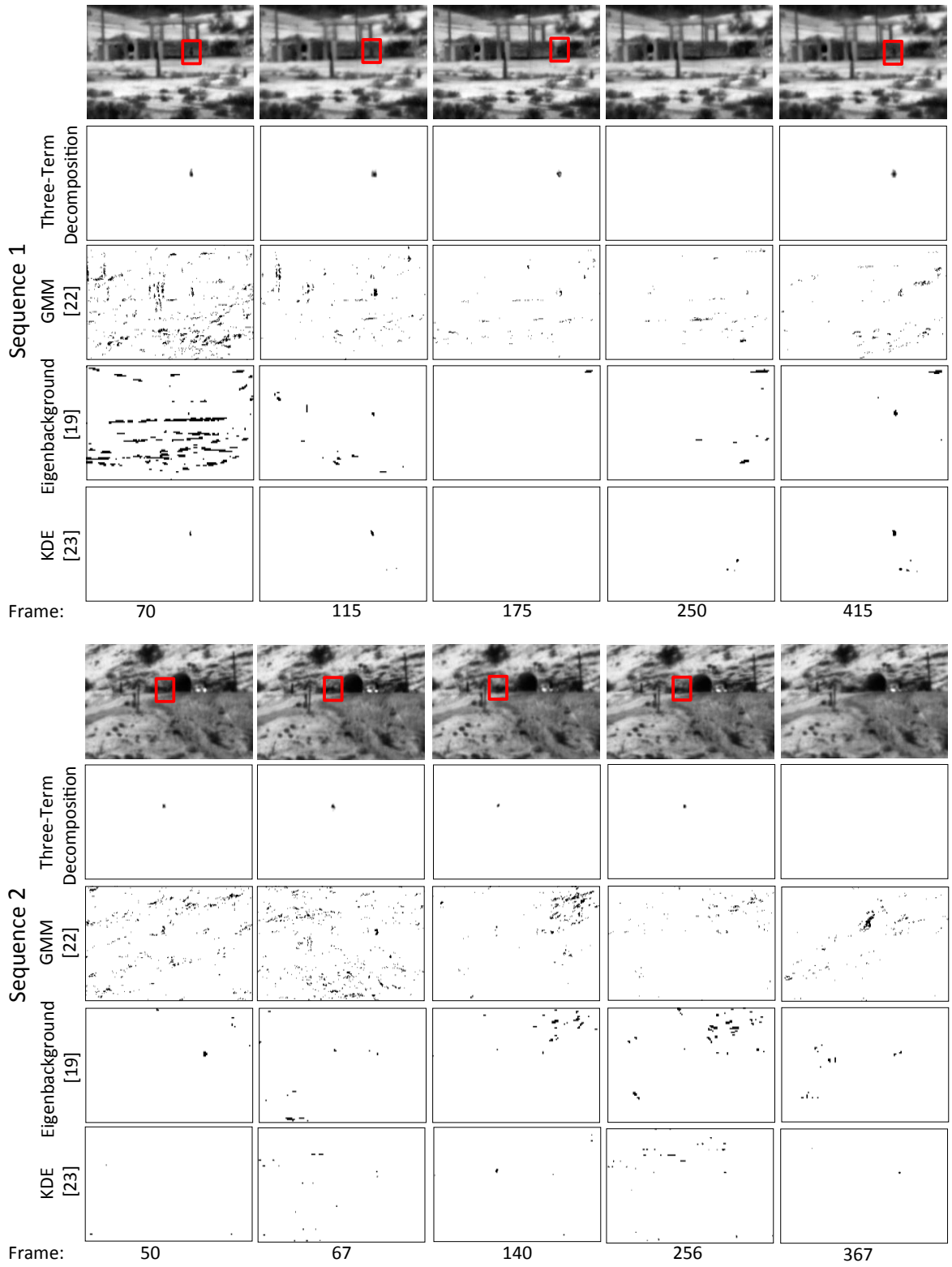

Figure 4.4: Our moving object detection results on sample frames from sequences 1 and 2 compared to [137], [87], and [104]. For every sequence, the first row shows the frames, the second row shows the result from our method (taken from matrix O after the decomposition), the third, fourth, and fifth rows show the background subtraction result obtained using [137], [87], and [104], respectively. Please zoom in to see the details.

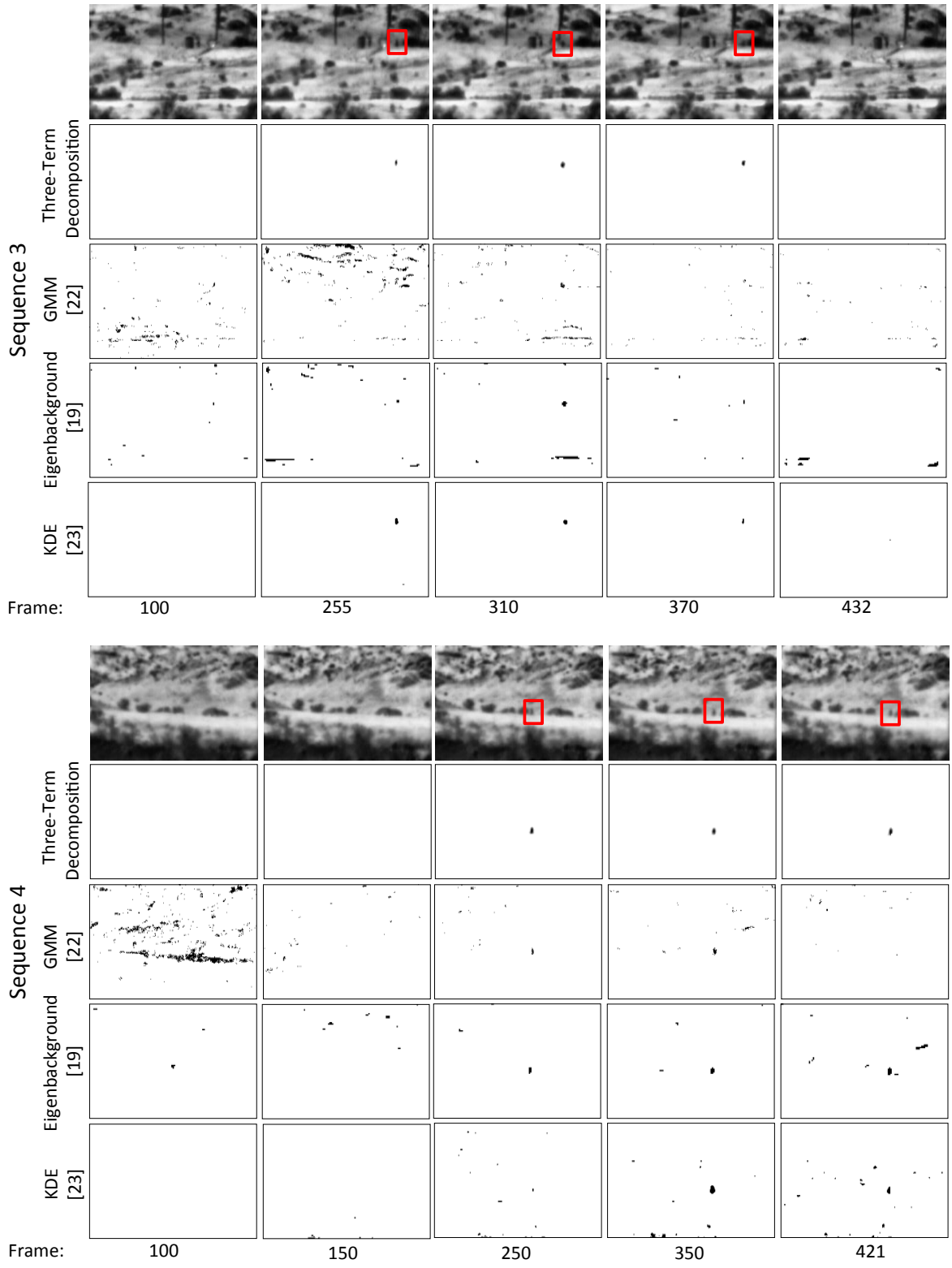

Figure 4.5: Our moving object detection results on sample frames from sequences 3 and 4 compared to [137], [87], and [104]. For every sequence, the first row shows the frames, the second row shows the result from our method (taken from matrix O after the decomposition), the third, fourth, and fifth rows show the background subtraction result obtained using [137], [87], and [104], respectively. Please zoom in to see the details.

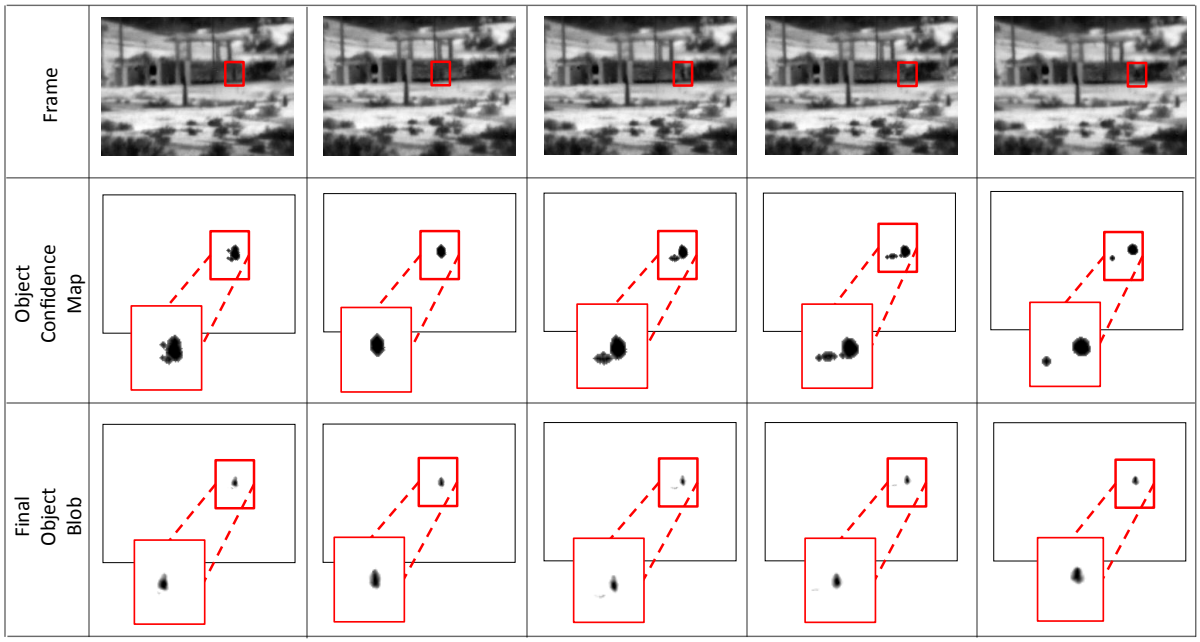

Figure 4.6: Example frames illustrating the contribution of the turbulence model and the low-rank optimization in the total moving object detection performance, separately. The first row shows the frames, the second row shows the object confidence map obtained from the turbulence model (confidence values are mapped to $[0 - 255]$, the highest confidence is black), and the third row shows the final object blob after the three-term low-rank decomposition. Clearly, the turbulence model provides a rough estimation of the object location, while the low-rank optimization refines the result to obtain an accurate detection.

4.2.3 Discussion of the Three-Term Model

Our method relies on a special three-term decomposition of matrices, which we formulated as the optimization problem (4). A similar optimization model has been proposed and studied very recently in [3], [112], and [71]. In [3], sufficient conditions are obtained in order to find an optimal solution to (4), recovering the low-rank and sparse components of F. In this work, however, we are interested in the computational methods for finding the desired decompositions. This leads us to scheme (7) via the Alternating Directions Method of Multipliers (ADMM), which was first introduced in the mid-1970's by [42, 43], and is the current method of choice for large-scale non-smooth convex optimization, as in [112, 44, 50, 134].

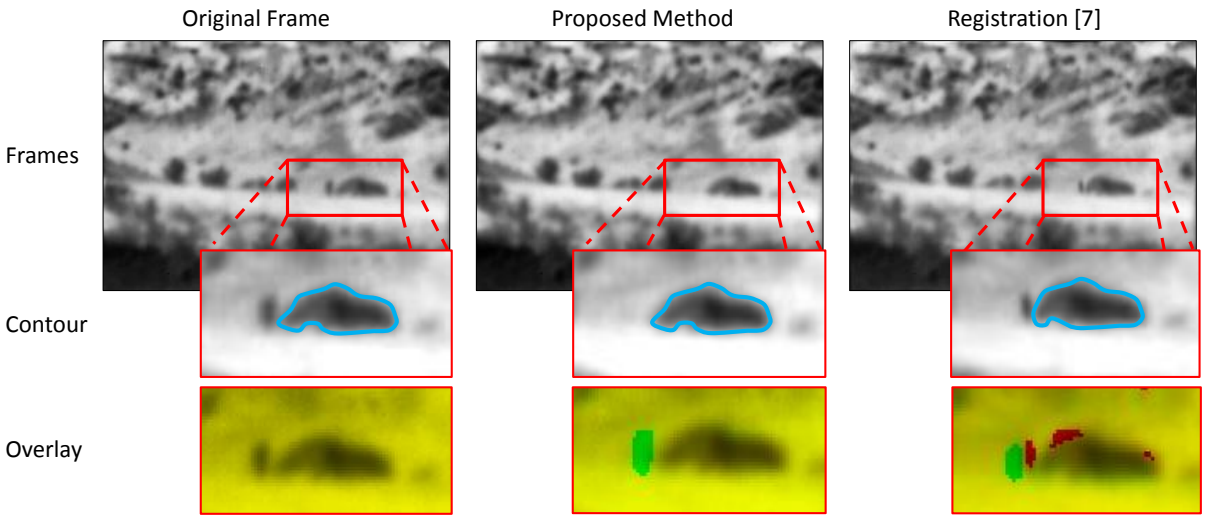

Figure 4.7: Our turbulence mitigation results compared to non-rigid registration [90] for an example frame. The first row shows from left to right: The original frame, the recovered background using our proposed method, and the recovered background using [90]. The second row shows a zoomed-in version of the frames with overlaid contours of a vehicle near the moving object. Note how the object's motion near the vehicle caused a deformation in the contour of the vehicle. The third row demonstrates another visualization of the artifacts experienced by registration due to the moving object. We inserted each of the frames into the R color channel with the original frame in the G channel, and set the B channel to zero. It is obvious that, using our proposed method, the object is eliminated from the background (appears as a green blob) and the vehicle's shape is maintained, while using registration methods such as, [90] and [106], do not handle moving objects and accordingly result in deformations in the region surrounding the object (appears as several green and red blobs). The figure is best observed in colors. Please zoom in to see the details and refer to our website for the complete videos.

A formulation similar to (7) can be obtained by replacing E with $F - A - O$ in the objective function in (4) in order to solve $\min_{A,O} \|A\|_* + \tau\|O\|_1 + \lambda\|F - A - O\|_F^2$. This was indeed done by Lin et. al. [71], in which they proposed two methods to solve this optimization: the Augmented Lagrange Multiplier (ALM) and the Inexact Augmented Lagrange Multiplier (IALM). The IALM is an alternating direction approach which converges almost as fast as the exact ALM, but with significantly fewer partial SVDs. This was further explored in [112] where an additional three-term alternation method was discussed. As noted in [112], the convergence of the scheme with more than two terms alternation is still an open problem in

general, although numerical experiments strongly suggest that the alternation scheme is globally optimal under relatively mild conditions. However, for special three-term alternations with very strict and potentially non-practical assumptions, there do exist proofs as given in [50] and [44].

The convergence of scheme (7) to the optimal solution of (4) clearly requires further study; however, we are able to prove the feasibility of the accumulation points produced by the iterations of the algorithm.

Theorem 1: The sequences $\{A_k\}$, $\{O_k\}$, and $\{E_k\}$ generated by Algorithm 1 are bounded, and any accumulation point (A^*, O^*, E^*) of (A_k, O_k, E_k) is a feasible solution: $F = A^* + O^* + E^*$.

Refer to the supplementary appendix for an outline of the proof of Theorem 1. Our theorem indicates that the iterations of our algorithm are guaranteed to yield a decomposition of low-rank, sparse, and turbulence components, which is sufficient in our problem. Proving that the solution that we arrive at is globally optimal is, however, still an open problem which we leave for future work. However, our experimental evaluations agree with [112] and suggest the optimality of the alternation scheme.

4.3 Experiments

We experimented extensively on the proposed ideas using four infrared sequences significantly distorted by atmospheric turbulence and also contain a moving human. The sequences and the code are available on our website: `http://www.cs.ucf.edu/~oreifej`. Each frame is 250×180 with 450 frames per sequence. The moving object occupies around

40 pixels in the frame and moves arbitrarily in the FOV. Typically, the object is static at the beginning of the sequence; therefore, we use the first 50 frames to compute the parameters of the intensity and the motion Gaussian models at every pixel. Our three-term decomposition converges quickly in about $25 - 35$ iterations, which takes approximately three minutes on a conventional laptop. Figure 4.3 shows our decomposition results. Our algorithm is able to decompose all the turbulence sequences generating a clear background and detecting the moving objects. The results are better observed in the videos available on our website http://www.cs.ucf.edu/~oreifej/.

Table 4.1: Comparison of the average PSNR in dB for the original sequences, after applying registration, and three-term decomposition.

	Original	Registration [90]	3-Term Decomp.
Sequence 1	26.55	32.20	31.62
Sequence 2	27.49	34.05	34.11
Sequence 3	27.71	31.79	32.25
Sequence 4	27.82	32.72	33.13

Since this problem is novel, there are no directly comparable approaches. Therefore, we consider comparing each of the moving object detection and the turbulence mitigation tasks separately. We compare our moving object detection results with the background subtraction method described in [137], where the background is modelled using a mixture of Gaussians. Moreover, we compare our result with an eigen-background model similar to [87], where the background basis for every 3×3 patch is found using PCA, then the patch is marked as foreground if it is not well represented by the PCA basis (i.e. its reconstruction error is above a threshold). We also compare our results with [104], which employs a nonparametric kernel

density estimation method (KDE) over the joint domain (location) and range (intensity) representation of image pixels. Figures Figure 4.4 and Figure 4.5 illustrate the results of the comparison on sample frames from our sequences. It is clear that even state-of- the-art methods suffer in turbulence sequences; therefore, our method outperforms such methods significantly. In Figure 4.6 we demonstrate the contribution of each of the turbulence model and the sparse optimization in the detection performance. As can be observed from the figure, our method leverages multiple constraints (sparsity, motion model, and intensity model) which complement each other to finally determine the object regions, thus significantly reducing miss-detection rates.

To evaluate the results quantitatively, we used a region-based measure where we applied connected components to the binary mask resulting from the background subtraction in order to obtain contiguous detection regions. Consequently, a detection region is considered correct if at least 50% of it is overlapping with the groundtruth, otherwise it is considered a miss-detection. The ROC curve in Figure 4.8 summarizes the obtained results for all sequences.

In addition, we compare our turbulence mitigation results with the robust registration algorithm presented in [90]. The robust registration is an iterative process which recovers the original de-warped sequence by registering the frames to their mean, and then updating the frames and the mean at every iteration. The registration is performed using non-rigid alignment by a means of control points overlaid on the frames. To evaluate the performance, we measured the peak signal-to-noise ratio (PSNR) between the first frame of the sequence and the rest of the frames, and reported the average results for all the frames in table Table 4.1. It is clear that both our method and registration can significantly stabilize the sequences and improve

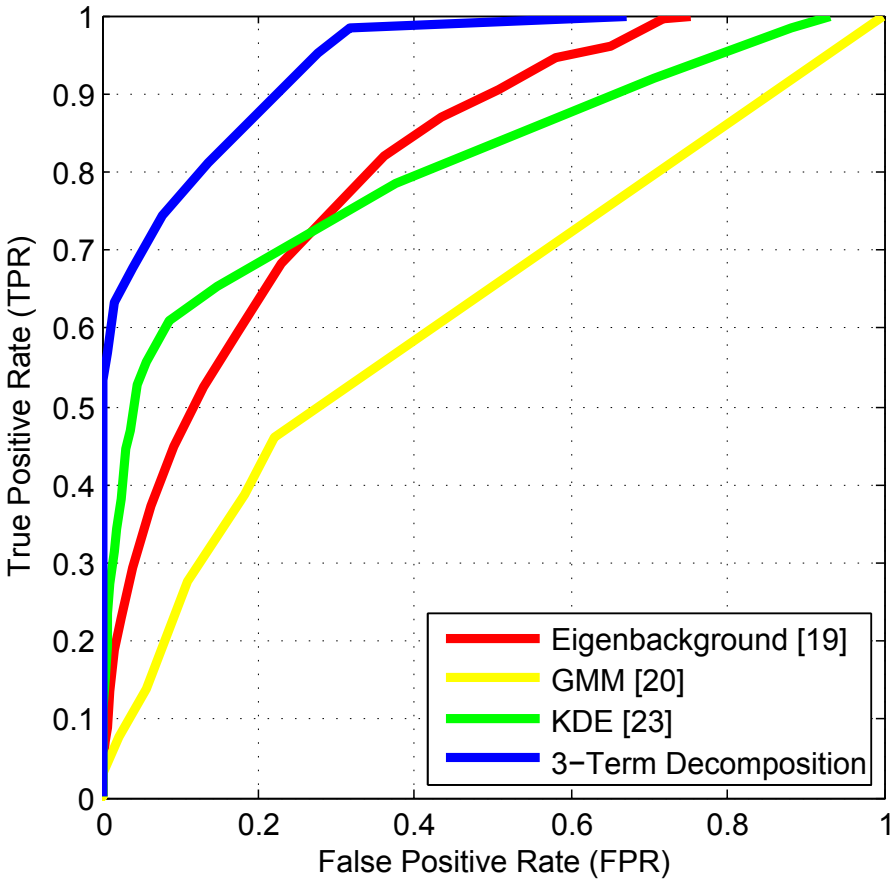

Figure 4.8: Performance of our method compared to background subtraction methods.

the PSNR. However, the moving object is not explicitly handled in the registration; therefore, it impedes the process by causing the control points to incorrectly shift in the direction of the object's motion, resulting in several artifacts in the surrounding area, as demonstrated in Figure 4.7. In contrast, our three-term decomposition handles such difficulties by separating the moving object; therefore, we recover a clear background without artifacts and with significantly reduced turbulence.

4.4 Summary

In the previous two chapters, we presented novel methods for reconstructing a scene going through water turbulence, then showed how we can extend that to handle the case when non-stationary objects additionally exist in the scene. In these two methods, each example is a gray-level frame reshaped as a vector. The low-rank formulations we discussed, however, are not limited to discovering the space of gray-level frames. In the coming Chapter, we discuss how to find the low-rank subspace of trajectories, where the variables are spatial coordinates of tracked points. We demonstrate how that is useful in detecting the camera motion subspace, and accordingly improve activity recognition.

CHAPTER 5: ACTION RECOGNITION IN VIDEOS ACQUIRED BY A MOVING CAMERA USING MOTION DECOMPOSITION OF LAGRANGIAN PARTICLE TRAJECTORIES

Action recognition from videos is a very active research topic in computer vision with many important applications for surveillance, human-computer interaction, video retrieval, robot learning, etc. Various action detection approaches are reported in the literature; however, they mostly tackle stationary camera scenarios. Recently, there has been an increasing interest in studying action recognition from moving cameras such as in aerial videos recorded by UAVs [5]. Action recognition from moving cameras poses significant challenges since the field of view is constantly changing with the camera motion. More importantly, the global camera motion and the local object motion are mixed up in the acquired frames (see Figure 5.1). Therefore, action recognition in such scenarios imposes a critical demand to eliminate the often dominant camera motion, and to recover the independent motions merely resulting from the performing actors.

Traditional approaches dealing with moving cameras usually need to go through a motion compensation step by performing video alignment [55, 54]. Consequently, the moving objects are detected by background subtraction, followed by the tracking of the detected moving blobs in order to compute certain motion features from the tracks to be employed in action recognition. However, this approach suffers from two inherent problems: First, video alignment is difficult and noisy due to the perspective distortions, and the errors in feature point

detection and localization; second, the errors from alignment and moving object detection further propagate to the tracking stage.

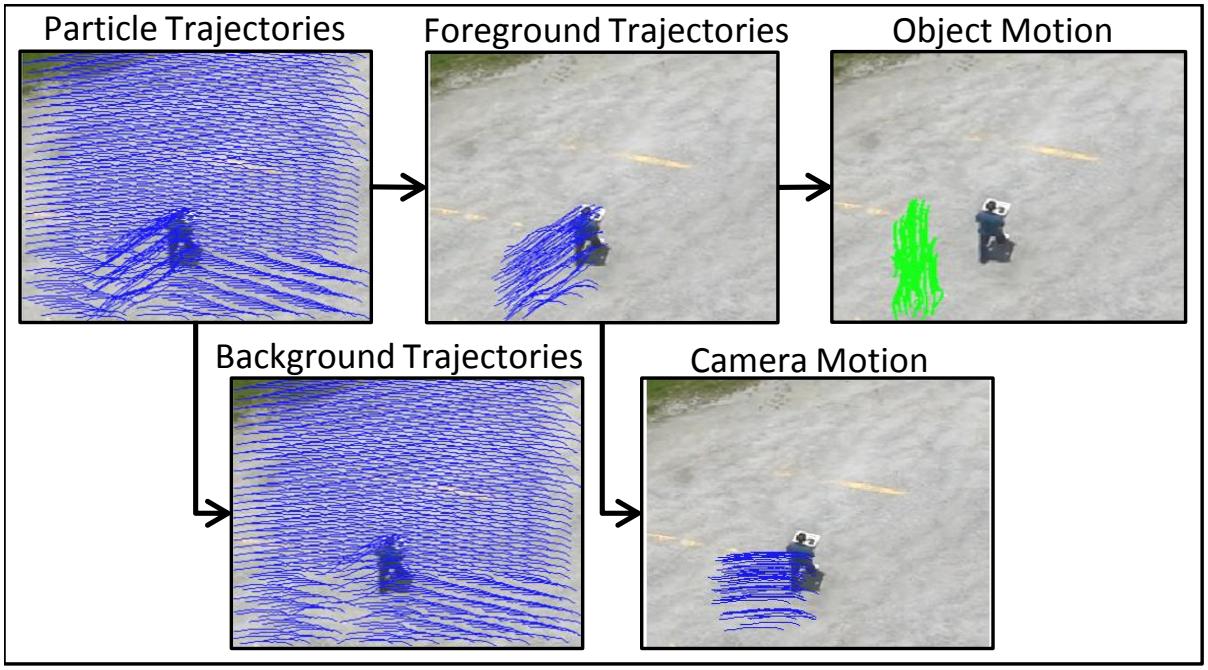

Figure 5.1: Motion decomposition for a moving camera sequence. The ensemble motions of a sequence is captured by particle trajectories, some of which solely correspond to the camera motion (background trajectories), and the others combine both the camera motion and the object motion (foreground trajectories). In this chapter, we show how to extract the object motion and employ it for action recognition. Note that the object motion component (green) appears displaced from the actor since the actor still carries the camera motion in the unaligned frame.

In addition, a fundamental problem in action recognition is to extract good features to describe the actions. In this chapter, we focus on motion features (trajectories). Motion trajectories are informative, compact, and spatiotemporally continuous, which makes them useful for action recognition [84, 28, 1, 17, 123]. Automatic trajectory acquisition can be performed by tracking. Although it is relatively easier to track the whole body or a part of a moving object and obtain a single trajectory corresponding to its centroid, the single trajectory is not able to provide semantically rich motion information for depicting complex and articulated

motions. Multiple-interest-point tracking using a tracker such as KLT as in [92] is possible yet very challenging due to three critical factors: First, good features for tracking (e.g. corners) need to be selected beforehand, which tends to be noisy in cluttered sequences. Second, the selected features may not be associated with the action of interest. Third, the obtained trajectories tend to be discontinuous due to the difficulty in maintaining consistent and correct point correspondence; therefore, the obtained tracks usually have variable lengths, which adds additional inconvenience for trajectory matching and alignment [85]. In addition, several methods employ explicit tracking markers attached to the objects to facilitate the tracking as in [84]; however, such invasive trajectory acquisition techniques are usually impractical.

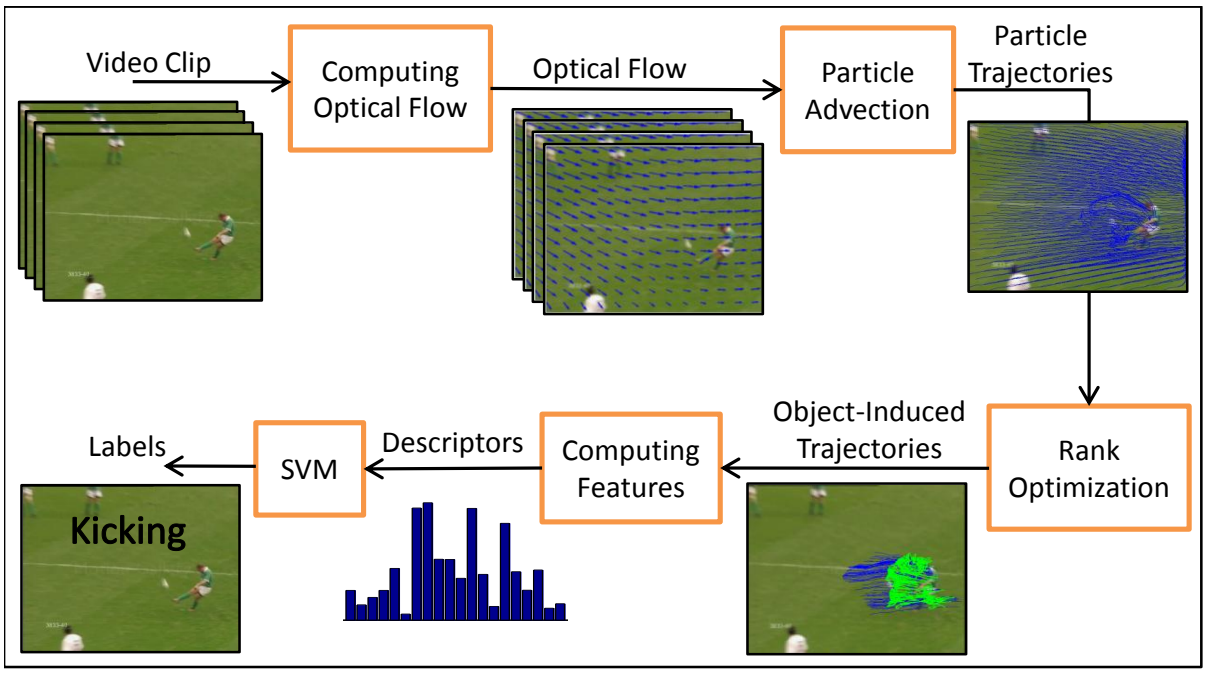

Figure 5.2: The various steps of our action recognition framework.

Our method is based on robust low-rank optimization that concurrently segments the trajectories corresponding to the moving object and eliminates their camera motion component; thus providing the relevant independent particle trajectories which correspond only to the

object's motion. Once we obtain the independent particle trajectories, we compute a compact set of motion features consisting of chaotic invariants and simple statistical features that describe the underlying motion properties. Consequently, a SVM is used for action learning and recognition. Figure 5.2 shows the overall workflow of the proposed framework.

5.1 Action Recognition Framework

We first employ particle advection to obtain particle trajectories, and then extract the independent trajectories that represent the object-induced motion. The extracted trajectories are then described by a set of chaotic invariants and simple statistical features, which are finally fed to a SVM.

5.1.1 Lagrangian particle advection

We use the concept of a "particle" to explain our Lagrangian particle trajectory acquisition approach. We assume that a grid of particles is overlaid on a scene where each particle corresponds to a single pixel (the granularity is controllable). The basic idea is to quantify the scene's motions in terms of the motions of the particles which are driven by dense optical flow. A so-called particle advection [124] procedure is applied to produce the particle trajectories. Given a video clip represented by a matrix of $T \times W \times H$, where T is the number of frames, and $W \times H$ denotes the frame resolution (width by height), we denote the corresponding optical flow by (U_w^t, V_h^t), where $w \in [1, W]$, $h \in [1, H]$, and $t \in [1, T-1]$.

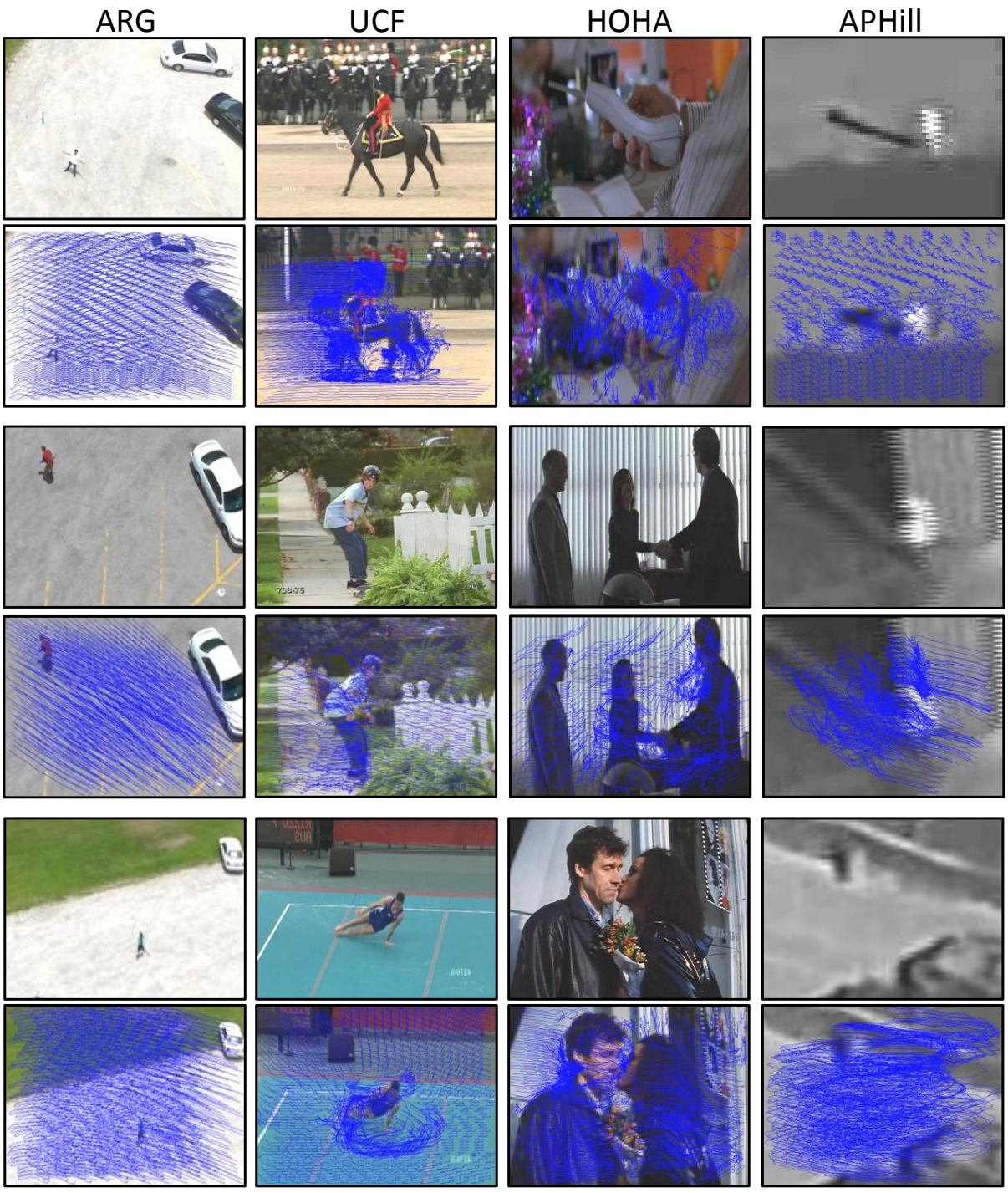

Figure 5.3: Three examples from each of our experimental datasets illustrating the obtained particle advection trajectories. Rows one, three, and five show the original frames, and rows two, four, and six show the corresponding overlaid trajectories respectively.

The position vector (X_w^t, Y_h^t) of the particle at grid point (w, h) at time t is estimated by solving the following equations:

$$\frac{dX_w^t}{dt} = U_w^t, \tag{5.1}$$

$$\frac{dY_h^t}{dt} = V_h^t. \tag{5.2}$$

We use Euler's method to solve them similar to [124]. By performing advection for the particles at all grid points with respect to each frame of the clip, we obtain the clip's particle trajectory set, denoted by $\{(X_w^t, Y_h^t) | w \in [1, W], h \in [1, H], t \in [1, T]\}$.

Figure 5.3 illustrates the obtained particle trajectories for three examples from each of our experimental datasets. The obtained particle trajectories almost occupy the full frame and therefore capture all the motions occurring in the scene. It is obviously unwise to use all of the particle trajectories for action recognition since the motion induced by the camera is irrelevant to the action of interest, and hence may significantly confuse the action recognition task. Therefore, in the coming subsection, we propose a robust method to extract the foreground trajectories and concurrently eliminate their camera motion component.

5.1.2 Independent Object Trajectory Extraction

The obtained particle trajectories are induced from two motion components: rigid camera motion, and object motion. When the action of interest includes global body motion (e.g. body translation in running action), the object motion can be further decomposed into two components: rigid body motion, and articulated motion. We employ the latest advances in sparse optimization to estimate each of these components, and extract the object trajectories

which solely correspond to the action of interest. Without loss of generality, we assume that the majority of the observed motion is induced by the camera motion (this is reasonable for most realistic datasets). Therefore, the trajectories should generally span a subspace determined by the scene structure and the camera's intrinsic and extrinsic parameters. In order to find the basis for the particle trajectory subspace, we first construct a $2T \times P$ (P is the number of particles, i.e. $P = W \times H$) measurement matrix M using the position vectors of the advected particle trajectories in a clip

$$M = \begin{bmatrix} X_1^1 & \cdots & X_P^1 \\ Y_1^1 & \cdots & Y_P^1 \\ \vdots & \vdots & \vdots \\ X_1^T & \cdots & X_P^T \\ Y_1^T & \cdots & Y_P^T \end{bmatrix}. \tag{5.3}$$

Through rank minimization, we can decompose M into two components: a low-rank matrix A, and the sparse error matrix E

$$\arg\min_{A,E} rank(A) \text{ s.t. } M = A + E, ||E||_0 \leq \beta, \tag{5.4}$$

where β is a constant that represents the maximum number of corrupted measurements expected across the sequence. Introducing the Lagrange multiplier λ, we get

$$\arg\min_{A,E} rank(A) + \lambda||E||_0 \text{ s.t. } M = A + E, \tag{5.5}$$

where λ trades off the rank of the solution versus the sparsity of the error, and we always set it to $1.1/\sqrt{(W \times H)}$ following the theoretical considerations in [21], and the results from our experiments. Consequently, we apply convex relaxation to the problem by replacing $rank(A)$

with the nuclear norm or sum of the singular values $||A||_* = \Sigma_i(\sigma_i)$, and replacing $||E||_0$ with its convex surrogate ℓ_1 norm $||E||_1$

$$\arg\min_{A,E} ||A||_* + \lambda||E||_1 \text{ s.t. } M = A + E. \tag{5.6}$$

Equation 5.6 is convex and can be solved with convex optimization methods such as the Augmented Lagrange Multiplier (ALM) algorithm [70] which we found robust and fast in our scenarios. The columns of the resulting low-rank matrix A define the basis of the low rank components in the trajectories. Since the camera motion is dominant, the subspace spanned by the major basis of A correspond to the desired background subspace which includes both the background trajectories and the camera motion component of the foreground trajectories. On the other hand, any rigid body motions in the scene will also contribute to A; therefore, the subspace spanned by the rest of the basis of A mostly correspond to rigid body motions. Since the camera motion subspace is approximately spanned by three basis [103, 36], the camera motion component can be estimated by $A_c = US^*V'$, where U and V are obtained by singular value decomposition $[U, S, V] = SVD(A)$, and S^* is equal to S except that all the singular values other than the most significant three are set to zero. Therefore, the rigid body motion component is expressed by $A - A_c$.

Moreover, the columns of the matrix E correspond to the deviation of each trajectory from the recovered low rank subspace, which captures the articulated motions. Therefore, the total object trajectories E_t which include the articulated and the rigid body motion is given by

$$E_t = E + A - A_c. \tag{5.7}$$

If the action of interest involves only articulated motions without a rigid motion component (e.g. boxing, waving, etc.), the object motion will be mostly captured in E while the rigid body component $A - A_c$ will be negligible. On the other hand, if the action of interest involves rigid body motion (e.g. running, walking, etc.), each of E and $A - A_c$ will contribute to the total object motion. Figure 5.4 illustrates the motion decomposition for two actions, "boxing" and "carrying".

Since additional noise is usually present, some object trajectories can correspond to noise. However, the motion in such trajectories is minor compared to the actual object's motion; therefore, they are easily eliminated by a simple threshold. In out experiments, we compute the sum of squared value for the columns of E, and accordingly select only the trajectories which attain at least 10% of the maximum value.

It is worth mentioning that we discard the boundary trajectories before constructing the measurement matrix M. The boundary trajectories are the trajectories that exhibit particles hung-up in the scene boundaries during the advection. Therefore, the points from such trajectories will remain stationary during the hung-up, and the resulting trajectories will not follow the complete camera motion. Hence, including such trajectories in M could deteriorate the performance of the rank minimization. Normally only a very small set of trajectories are excluded. Figure 5.5 depicts example object motion detection results for four sequences taken from each of our experimental datasets. It is clear from the figures that our method is able to robustly extract the object trajectories relevant to the action of interest.

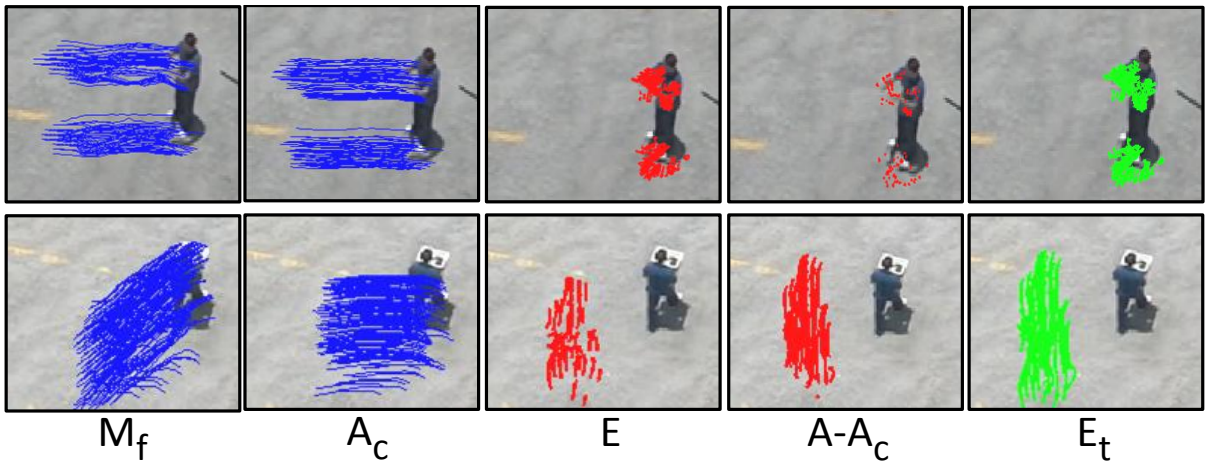

Figure 5.4: Two examples illustrating our proposed motion decomposition. From left to right: the detected foreground trajectories M_f, camera motion component A_c, articulated object motion component E, rigid object motion component $A - A_c$, total object motion E_t. The first row shows boxing action which only contains articulated motion component; thus, $A - A_c$ is negligible, and E and E_t are similar. The second row shows carrying action which contains both articulated and rigid body components; thus, E does not fully represent the motion, but E_t rather does. Note that the original foreground trajectories M_f is equal to $A_c + E_t$.

5.1.3 Action Description and Recognition

We use the extracted object trajectories to describe and recognize actions. Since a particle is typically placed on each pixel, we obtain a large number of particle trajectories. In order to get a more compact representation, we cluster the obtained trajectories into 100 clusters using k-means, and accordingly select the cluster's centroid as the representative trajectory for each cluster. Consequently, we characterize an action by computing a compact set of descriptors of the trajectories for training and recognition. In that, we use the chaotic invariants features [124, 6] augmented with a simple statistical feature for each of the x and y time series of a trajectory

$$F = \{\sigma, L, C\}\,, \tag{5.8}$$

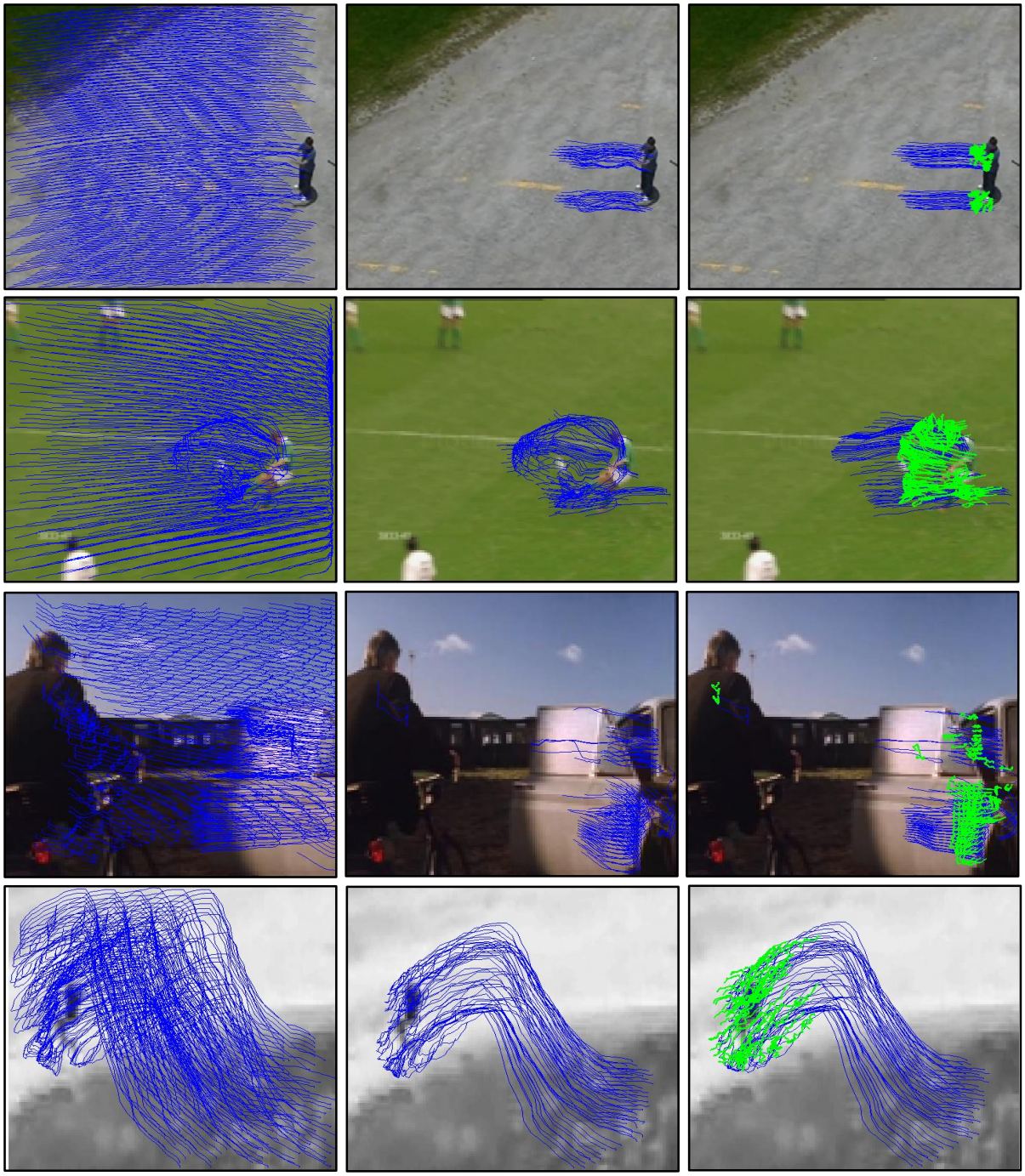

Figure 5.5: Each row shows an example illustrating our proposed object motion detection method. The examples are taken from four datasets, from top to down: ARG, UCF sports, HOHA, and APHill. The first column shows all the particle trajectories (excluding boundary trajectories). The second column shows the detected object trajectories. The third column shows the camera motion component (A_c) in blue, and the object motion component (E_t) in green. Please refer to our website for videos of the results.

where σ denotes the variance, L denotes the Largest Lyapunov Exponent (LLE), and C denotes the correlation dimension. σ has been proved to be a useful feature for time series description [6]. Meanwhile, L and C are typical chaotic invariants which are able to identify the underlying dynamics properties of a system (i.e., an action here). In the embedded phase space, L provides quantitative information about the orbits that start close together but diverge over time, and C measures the size of an attractor. We follow the algorithms described in [124] to calculate L and C. In order to estimate L for a time series X, we first locate all of the nearest neighbors ($\tilde{X}$) within the orbit in the embedding space. The nearest neighbors are assumed to diverge approximately at a rate L. We therefore have

$$\ln d_j(t_i) \approx \ln k_j + Lt_i. \tag{5.9}$$

where j is the index of the pair of nearest neighbors, $t_i = i\Delta t$, Δt is the sampling period, k_j is the initial separation, and $d_j(t_i)$ denotes the distance between the jth pair of the nearest neighbors after i discrete time steps. Equation (5.9) represents a set of approximately parallel lines and thus L can be approximated by the slope of a fitted average line denoted by $y(t_i) = \frac{\langle \ln d_j(t_i) \rangle}{\Delta t}$.

To estimate the correlation dimension, we first calculate the correlation sum

$$S(\delta) = \frac{2}{Q(Q-1)} \sum_{i \neq j} H(\delta - \|\tilde{X}_i - \tilde{X}_j\|). \tag{5.10}$$

where H denotes the Heaviside step function, δ is a threshold distance, Q is the number of points in the time series. Consequently, we can simply derive C by $S(\delta) \approx \delta^C$.

Finally, we use a radial basis SVM to learn action models from the feature set of training, and to recognize testing examples. It is worth mentioning that we experimented on several types of trajectory features, and found the selected set of features preferable.

5.2 Experiment Results

We extensively experimented on the proposed action recognition method using six datasets including four moving camera datasets (APHill, ARG, HOHA, and UCF sports), and two static camera datasets (KTH and Weizmann). For all of the datasets, we use the algorithm described in [73] for computing optical flow. To reduce the computational cost, we associate each particle with a 2×2 grid window.

5.2.1 APHill action recognition

APHill is a newly formed dataset of aerial videos. It includes 6 actions with 200 instances for each, except for "gesturing" action which has 42. This dataset is very challenging due to the low resolution (as low as 50×50) and the large intra-class variations (refer to Figure 5.3 for action examples). Using 20-fold cross validation, we obtained 41.8% recognition rate. In order to evaluate the contribution of our independent object motion estimation technique, we repeated the experiment using all of the initially obtained trajectories instead of using only the object-induced trajectories. In such case, we observed a significant decrease in performance (only 31.1% achieved), which provides a clear evidence of the contribution of our object motion detection method in action recognition. Table 5.1 shows the obtained confusion matrix. As can be seen, walking is mostly confused with running. Additionally, no actions were classi-

fied as gesturing which is mostly because the number of samples for this action is significantly less than the others. Moreover, standing is quite hard to distinguish as there are very minor motion features associated with such action. In general, given the difficulty of the dataset, the performance is quite promising.

Table 5.1: Confusion matrix for APHill dataset.

	Standing	Walking	Running	Digging	Gesturing	Carrying
Standing	21	06	16	32	00	25
Walking	02	19	62	02	00	14
Running	04	15	56	05	00	21
Digging	06	02	04	68	00	21
Gesturing	11	00	09	54	00	26
Carrying	05	08	24	22	00	42

5.2.2 ARG-aerial action recognition

ARG-aerial is a new multi-view dataset recorded from four viewpoints by a moving camera equipped in a freely floating balloon. It includes 9 actions, each is performed by 12 actors. We use a subset sequences selected from one of the viewpoints. Each sequence ranges from $\sim 30-50$ seconds, with several repetitions of the action pattern; thus, we divide it into multiple shorter clips (50 frames for "digging" and "throwing", and 30 for the rest). We obtain a total of 112 clips for our experiments.

A major challenge in ARG dataset arises from the large, dramatic, and fast camera motions in most of the videos due to the free floating nature of the balloon. In addition, the actors are extremely small occupying approximately only $\sim 2-5\%$ of the full frame (frame

size is 1920×1080). Such conditions are particularly challenging for articulated human action recognition.

We preprocess the clips by resizing them to $\sim 25\%$ of the original size, and cropping out a sub-window(ranging from $\sim 80 \times 80 - 300 \times 300$ pixels^2). Using 5-fold cross validation, we obtained an average recognition rate of 54.6%, and 30.8% when the independent motion estimation step is skipped, which provides additional support for the effectiveness of our method. Table 5.2 shows the obtained confusion matrix. As can be seen from the matrix, both walking and carrying actions are mostly confused with running. In fact, it is indeed very difficult to distinguish such actions relying on only motion features. In view of the discussed challenges, such performance is promising.

5.2.3 HOHA action recognition

HOHA (Hollywood Human Actions) dataset [51] includes 10 types of actions extracted from movies. Almost all of the sequences can be considered within the moving camera domain. HOHA is very challenging due to the complicated background of the realistic scenes, the large intra-class variation, and the existing changes of shots in a significant number of videos. The change of shots particularly is a considerable challenge for obtaining continuous particle trajectories; in fact, it raises the same challenge for any tracking-based method such as [92]. However, we found in our experiments that the shot change often slightly corrupts the trajectories such that no major spurious effects are introduced.

We use the "clean" training set for training and the separate testing set for testing. We compare the performance of our method with Trajecton [92] and space-time interest points

(STIP) [51] using the same experimental setup and performance measure (Average Precision). Table Table 5.3 shows the comparison results, from which it can be clearly observed that our method achieved a better performance than STIP. Additionally, our method significantly outperforms Trajecton which employs KLT to acquire trajectories; this particularly shows the advantage of our dense particle advection trajectories. Moreover, we repeated the same experiment except that we used all of the initially obtained particle trajectories (i.e., independent motion estimation step was skipped). The obtained performance, as can be seen from column 3 of the table, is still comparable to the case where only the object trajectories are employed. Such result is expected in this dataset since the camera motion is minor in many sequences, and more importantly, the actors occupy the majority of the frame such that most of particle trajectories are associated with the action of interest.

Table 5.2: Confusion matrix for ARG-aerial dataset.

	Boxing	Carrying	Clapping	Digging	Jogging	Running	Throwing	Walking	Waving
Boxing	51	10	00	13	08	04	02	06	00
Carrying	02	31	00	02	06	54	02	02	00
Clapping	04	02	67	04	10	04	04	02	02
Digging	02	00	00	79	02	02	15	00	00
Jogging	02	10	02	13	38	31	00	04	00
Running	00	06	02	02	02	85	00	02	02
Throwing	02	02	00	19	02	00	73	00	02
Walking	00	15	00	06	04	67	04	04	00
Waving	00	08	00	15	06	00	06	06	58

Table 5.3: Average Precision comparison for HOHA dataset.

Action	Ours	Ours(All Trajs)	STIP [51]	Trajecton [92]
Average	**47.6%**	**46.3%**	**38.4%**	**21.1%**
AnswerPhone	48.3%	46.9%	32.1%	4.5%
GetOutCar	43.2%	34.1%	41.5%	69.0%
HandShake	46.2%	49.7%	32.3%	71.4%
HugPerson	49.3%	49.6%	40.6%	0.0%
Kiss	63.6%	49.9%	53.3%	0.0%
SitDown	47.5%	50.0%	38.6%	5.3%
SitUp	35.1%	40.0%	18.2%	11.1%
StandUp	47.3%	50.0%	50.5%	7.7%

5.2.4 UCF sports action recognition

UCF sports is a challenging dataset with sequences mostly acquired by moving cameras. It includes 10 sport actions with a total of 150 sequences. We followed the same processing as in [49, 63] by adding a flipped version for all the videos in order to enlarge the dataset. We use a 5-fold cross validation strategy and Table Table 5.4 shows our results and the comparison to other methods. It is observed that our method slightly outperforms the state of the art. The performance is also comparable even when the independent motion estimation step was skipped.

Table 5.4: Recognition rate comparison for UCF sports dataset.

Method	Rate (%)
Ours	**89.7**
[63]	87.3
[49]	85.6
Ours (All Trajs)	85.8

5.2.5 Action recognition from static cameras

Though our proposed method is primarily designed for moving camera scenarios, we additionally experimented on KTH and Weizmann datasets which can generally be considered within the static camera domain though a small part of the videos are associated with a zoom-in and zoom-out operations in KTH. Each sequence is divided into multiple shorter clips ranging from $20 - 50$ frames per clip. We obtained an average recognition rate of 95.7% for KTH which is closely comparable to the state-of-the-art [62] with 96.7%. For Weizmann dataset, we obtained 92.8% which we particularly compare with the closely related work of [6] where a slightly inferior rate of 92.6% is achieved with manually obtained trajectories.

5.3 Summary

In this chapter, we presented a novel method for recognizing simple activities in videos recorded under constrained conditions. We discovered the low-rank structure of the camera motion, and used that to extract the object trajectories. The low-rank assumption for the camera motion is more valid when the scene is simple and the camera motion is dominant (for example in aerial videos). This is typically not the case in complex videos obtained for example from YouTube. These videos typically pose significant challenges because of their highly variable content, noise, length, frame size ... etc. In the coming chapter, we demonstrate that for complex activities (referred to as complex events), a low-rank structure exist in the feature space of the event. Discovering this low-rank subspace allows us to refine the event descriptors and obtain more semantically meaningful event models.

CHAPTER 6: COMPLEX EVENT RECOGNITION USING CONSTRAINED LOW-RANK OPTIMIZATION

The increasing popularity of digital cameras has been creating a tremendous growth in social media websites like Youtube. Along with the increased number of user-uploaded videos, the need to automatically detect and recognize the type of activities occurring in these videos has become crucial. However, in such unconstrained videos, automatic content understanding is a very challenging task due to the large intra-class variation, dynamic and heterogeneous background, and different capturing conditions. Therefore, this problem has recently gained a significant attention.

Most activity recognition methods are developed for constrained and short videos (5-10 seconds) as in [120, 76, 30, 64]. These videos contain simple and well-defined human actions such as waving, running, jumping . . . etc. In contrast, in this chapter, we consider more practical videos with realistic events, complicated contents, and significantly variable lengths. Therefore, standard activity recognition methods suffer when faced with such unconstrained videos. This is because it is difficult for a detector trained on low-level features to handle the considerably large content variation within each event category. To this end, the most recent approaches resorted to using low-level events called "concepts" as an intermediate representation [129, 53]. In that, a complex event is described in terms of the occurrence confidence of the concepts. For example, the event *Birthday Party* can be described as the occurrence of

singing, *laughing*, *blowing candles*, *jumping* . . . etc. The occurrence confidence of the concepts in a video forms a feature vector for the video, which we refer to as high-level feature.

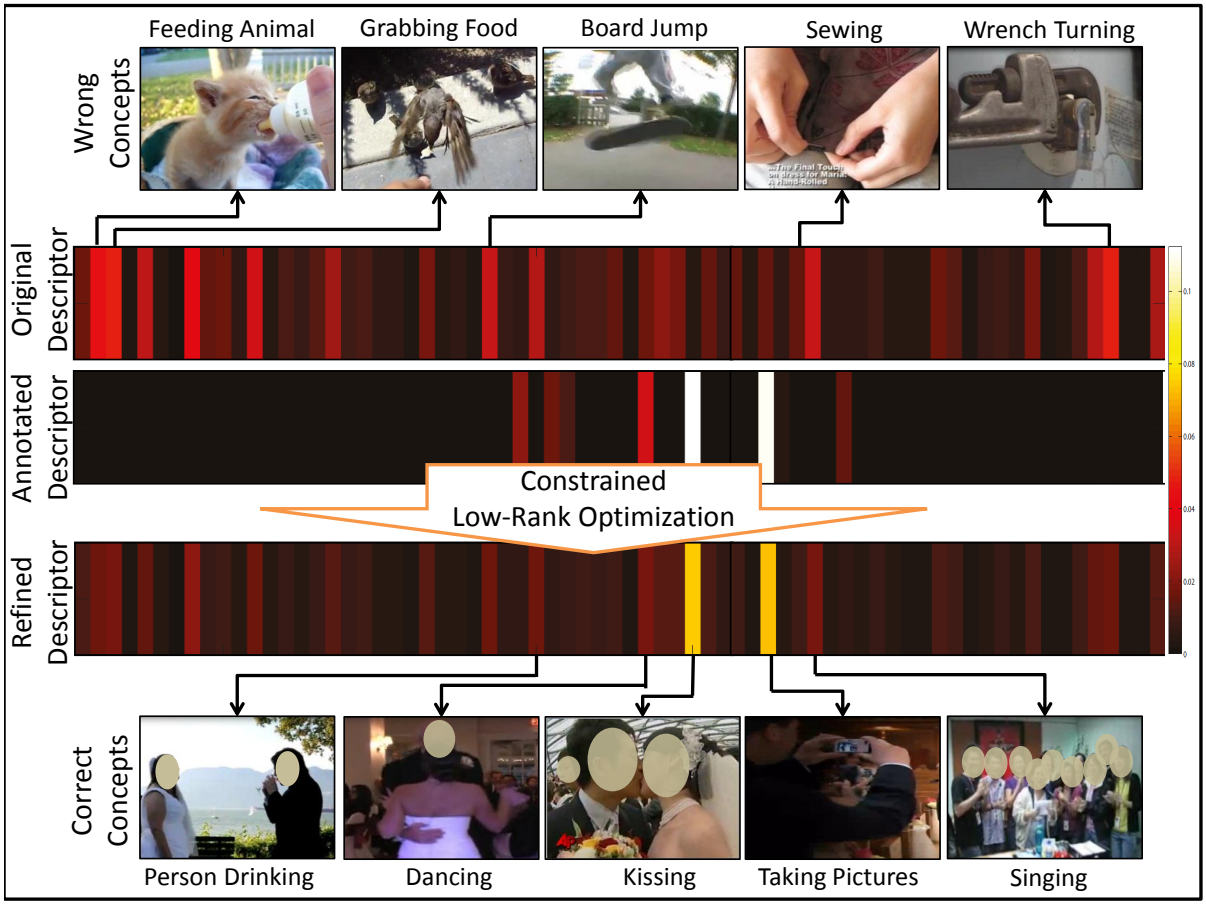

Figure 6.1: The figure shows an example event (*Birthday Party*) with its concept-based descriptors shown in Hot colormap, where each bar represents the confidence of a concept. The top row shows the original descriptor, which is automatically obtained using concept detectors. The middle row shows a manually annotated descriptor for the same event. Note that the two descriptors are surprisingly very different. We observed that, in such complex videos, the automatically detected concepts are strictly relying on local visual features, and lacking context and sematic cues, which humans naturally infer. Based on that, our method refines the concept-based representation of the complex events by encouraging the features to follow the human annotation. Thus, suppressing the falsely detected concepts (top), and inducing important concepts (bottom).

In the context of concept-based event representation, substantial consequences arise as a result of the complex nature of these unconstrained videos. First: The examples used to train each concept have significant variations, and thus the resulting concept models are noisy.

Second: The concept-content of each event may still significantly vary among the samples of each event, mainly because of the variable temporal lengths and capturing conditions. Therefore, the obtained high-level features (concepts' scores) used to describe each event are also significantly noisy. Third: The automatically obtained concept representation strictly relies on local visual features, and lacks context and sematic cues, which humans naturally infer (refer to Figure 6.1).

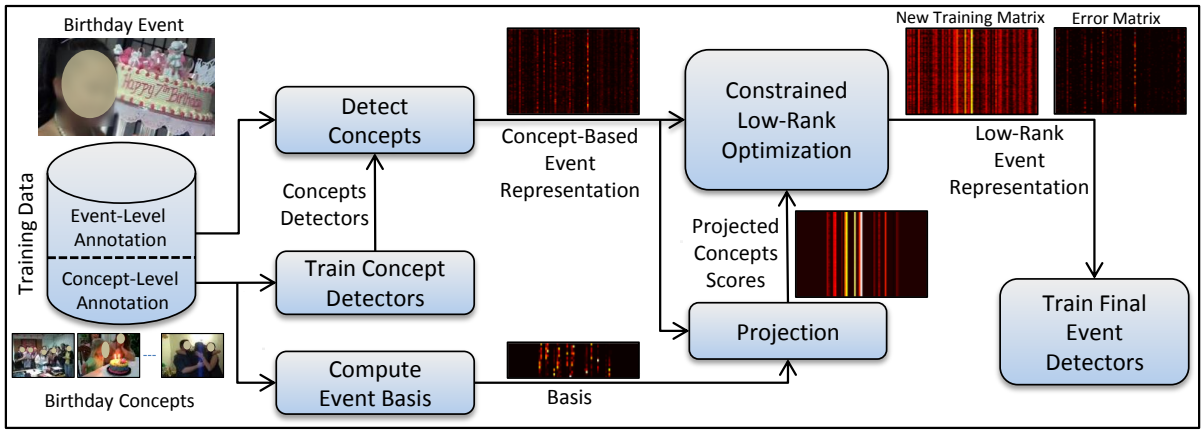

Figure 6.2: The block diagram of the proposed complex event recognition method. We manually annotate the concepts in a portion of the training data, and leave the other portion with only event-level annotation (labels). The concept-level annotation is used to train the concepts, which we run on the event-level annotated data and obtain their concept scores. Consequently, we compute the concepts' basis, and optimize the rank of the concept detection scores under the annotation constraint.

In this chapter, we address the discussed hurdles of the concept-based event representation using a novel low-rank formulation, which combines the precisely annotated videos used to train the concepts, with the rich high-level features. Our approach is based on two principles: First, the videos of the same event should share similar concepts, and thus should have consis-

tent responses to the concept detectors. Therefor, the matrix containing the high-level features of each event must be of low-rank. Second, since the videos used to train the concept models were manually and carefully annotated, the resulting low-rank matrix should also follow the annotation. For example, concepts like *person falling* or *person flipping* may falsely fire in *Birthday party* event. Therefore, by enforcing our constraints, such hurdles can be avoided.

Figure 6.2 summarizes the steps involved in our method. We split the training data into two sets: (1) the event-level annotated data, which has only event labels (2) the concept-level annotated data, which has both event-level and concept-level labels. We use the concept-level annotated data to train concept detectors, which we run on the event-level annotated data and obtain their concept scores. Consequently, we stack the concept scores for each event in a matrix and find their low-rank representation such that it also follows the basis of the concept annotation. The resulting training data combines the two sets in one rich and consistent training set.

The low-rank constraint has been vigorously employed in different computer vision problems such as tracking [115], feature fusion [130], face recognition [25], and saliency detection [105]. However, to the best of our knowledge, low-rank estimation of high-level features has never been used before. More importantly, our formulation is more general than the standard RPCA [22] in that we allow the estimated low-rank matrix to follow a prior pattern (the annotation in our scenario). On the other hand, since we exploit the low-rank constraint, our method is more robust against noisy concepts and cluttered background than [129, 53, 111], and significantly outperforms the state-of-the-art, as we demonstrate in the experiments.

The main contribution of this chapter is a novel low-rank formulation, through which we find a new representation for each event, which is not only low-rank, but also constrained by the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Our constrained low-rank representation is not restricted to a certain type of features; which allows us to employ a combination of state-of-the-art features including STIP [64], DTF-MBH [120], and DTF-HOG [120]. Additionally, we manually defined and annotated an action concept dataset based on TRECVID event collection, with 81 concepts for TRECVID MED11 and 93 for TRECVID MED12, using more than $20,000$ video clips. The rest of the chapter is organized as follows: Section 2 describes the process of computing the constrained low-rank event representation. In Section 3, we describe how to find the optimal solution for the low-rank optimization. The experimental results are presented in Section 4.

6.1 Low-Rank Complex Event Representation

Given training event samples $\mathcal{X} = \{\mathbf{x}_k\}$ with event labels $\mathcal{Y} = \{\mathbf{y}_k\}$, we manually annotate a portion of the data with 81 predefined low-level events, which occur frequently. These low-level events are called concepts, and they are similar to the concepts used in [53]. This generates two subsets: $\mathcal{M} = \{\mathbf{m}_k\}$, which has only event-level annotation (the labels), and $\mathcal{Z} = \{\mathbf{z}_k\}$, which has both event-level and concept-level annotations. A video is annotated by tagging the beginning and ending frame for every concept which occurs along the duration of the video. Therefore, we end up with training clips for each concept, from which we extract low-level features (DTF-MBH, DTF-HOG and STIP) and train a SVM model for each concept. Consequently, each video in the subset $\mathcal{M}$ is tested against each concept model. In that, the

video is uniformly divided into clips, and the features are extracted from each clip, then tested on each concept model. For each concept, the maximum confidence among the clips of a video is stored. This generates a confidence vector of length 81 for each video in subset $\mathcal{M}$, which represents our high-level feature vector.

Similarly, we construct confidence feature vectors for the samples in subset $\mathcal{Z}$ using the concepts' annotation. For each video in $\mathcal{Z}$, we set the confidence for each concept which was tagged in the video to the maximum possible confidence (we obtain that from $\mathcal{M}$), and we set the confidence for the non-existing concepts to zero. Therefore, we obtain two training subsets $\mathcal{Z}$ and $\mathcal{M}$, each with inherently different cues. The features in $\mathcal{Z}$ are exact and strict because they are collected manually. In contrast, $\mathcal{M}$ has automatically extracted features; therefore, it is noisy, however, contains rich descriptions of all the details in the videos.

Our goal is to combine the benefits of each subset in one rich and less noisy training set. To the best of our knowledge, this problem has never been explored before. One trivial solution is to simply combine the samples from the two sets and train a SVM on that. In this case, it is likely that the trained model will select support vectors from each of the subsets, and therefore the final decision will be a weighted sum of the dot product between the support vectors and the testing sample (in a linear SVM). However, it is clear that the weighted sum cannot denoise the noisy subset $\mathcal{M}$, or enrich the highly strict subset $\mathcal{Z}$. In fact, as we demonstrate later in the experiments, combining the subsets naively can decrease the classifier's discriminativity.

Since we train a model for each event separately, it is expected that the matrix containing all the features of an event stacked in the rows will be of low-rank (similarly if stacked in

the columns). We exploit this observation in our method. However, in our case, not only the matrix needs to be of low-rank, but also close to the annotation. We formulate this problem as

$$\min_{A_i, E_i} \text{Rank}(A_i) \;\; \text{s.t.} \;\; M_i = A_i + E_i, \qquad (6.1)$$
$$||E_i||_0 \leq s, \;\; f(A_i, Z_i) \leq \sigma,$$

where $M_i, A_i, E_i \in \mathbb{R}^{m_i \times d}$ are the matrices of the samples of event i from subset $\mathcal{M}$ stacked in the rows, the corresponding low-rank matrix, and the error matrix, respectively. d is the dimensionality of the feature vector (81 in our case), and m_i is the number of samples. $Z_i \in \mathbb{R}^{z_i \times d}$ is the matrix of the samples of event i from subset $\mathcal{Z}$ stacked in the rows, where z_i is the number of samples. s and σ are constants which control the relative weights in the minimization. The function $f : \mathbb{R}^{m_i \times d} \times \mathbb{R}^{z_i \times d} \rightarrow \mathbb{R}^1$ measures the similarity between the estimated low-rank matrix and the annotation matrix. Since there is no correspondence between the samples in M_i and Z_i, the function cannot be defined as $||A_i - Z_i||_F$, where $||.||_F$ is the Frobenious norm. Note that A_i and Z_i can also have different sizes. To this end, we consider the following distance measure

$$f(A_i, Z_i) = ||A_i - M_i U_i U_i^T||_F^2 \qquad (6.2)$$

where $U_i \in \mathbb{R}^{d \times q}$ is the matrix with the most significant q principal components of Z_i obtained by SVD(Z_i), and stacked in the columns. In other words, equation (6.2) represents the difference between the low-rank estimation A_i, and the components of M_i along the major directions of the annotation Z_i. Therefore, minimizing equation (6.1) under the constraint

in equation (6.2) encourages the low-rank estimation to additionally follow the concept-level annotation. Note that in order to constrain the low-rank solution using the annotation, it is possible to derive other forms of $f(A_i, Z_i)$. However, as we discuss further in the following section, having the similarity measure in the form $||A_i - X||$, where X is a constant matrix, allows us to obtain a closed form solution at each iteration of the optimization.

Figure 6.2 shows an example for the event "Birthday party" before and after the optimization. It is clear that the final training matrix is low-rank and also adopts the patterns in the annotation. In the coming subsection, we discuss our approach to solve the optimization in equation (6.1) using the method of Augmented Lagrange Multiplier (ALM) [71].

6.2 Optimizing the Constrained Low-Rank Problem

Our method decomposes the matrix containing the examples of an event by extracting the noise such that the resulting matrix is both low-rank and follows the concept-annotation. This is achieved using equations (6.1) and (6.2) as discussed in the previous section. When solving equation (6.1), it is convenient to consider the Lagrange form of the problem:

$$\min_{A_i, E_i} \text{Rank}(A_i) + \lambda ||E_i||_0 + \frac{\tau}{2} ||A_i - M_i U_i U_i^T||_F^2 \quad \text{s.t. } M_i = A_i + E_i, \tag{6.3}$$

where λ and τ are weighting parameters. The optimization of (6.3) is not directly tractable since the matrix rank and the ℓ_0-norm are nonconvex and extremely difficult to optimize. However, it was recently shown in [22] that when recovering low-rank matrices from sparse errors,

if the rank of the matrix A_i to be recovered is not too high and the number of non-zero entries in E_i is not too large, then minimizing the nuclear norm of A_i (sum of singular values) and the ℓ_1-norm of E_i can recover the exact matrices. Therefore, the nuclear norm and the ℓ_1-norm are the natural convex surrogates for the rank function and the ℓ_0-norm, respectively. Applying this relaxation, our new optimization becomes:

$$\min_{A_i,E_i} ||A_i||_* + \lambda||E_i||_1 + \frac{\tau}{2}||A_i - M_i U_i U_i^T||_F^2 \quad \text{s.t. } M_i = A_i + E_i, \tag{6.4}$$

where $||A_i||_*$ denotes the nuclear norm of matrix A_i. We adopt the Augmented Lagrange Multiplier method (ALM) [71] to solve the optimization problem (6.4), which is currently the method of choice among the computational methods for low-rank optimization [112, 44]. Define the augmented Lagrange function for the problem as:

$$\begin{aligned} L(A_i, E_i, Y_i) \;=\; & ||A_i||_* + \lambda||E_i||_1 + \frac{\tau}{2}||A_i - M_i U_i U_i^T||_F^2 \\ & + \langle Y_i, F_i - A_i - E_i \rangle \\ & + \beta/2||M_i - A_i - E_i||_F^2, \end{aligned} \tag{6.5}$$

where $Y_i \in \mathbb{R}^{m_i \times d}$ is a Lagrange multiplier matrix for event i, β is a positive scalar, and $\langle , \rangle$ denotes the matrix inner product (trace($A^T B$)). Minimizing the function in equation (6.5) can be used to solve the constrained optimization problem in equation (6.4). We use the ALM algorithm to estimate both the Lagrange multiplier and the optimal solution by iteratively minimizing the augmented Lagrangian function:

$$(A_i^{k+1}, E_i^{k+1}) = \arg\min_{A_i, E_i} L(A_i, E_i, Y_i^k), \tag{6.6}$$
$$Y_i^{k+1} = Y_i^k + \beta^k (F_i^{k+1} - A_i^{k+1} - E_i^{k+1}).$$

where k denotes the iteration number. When β_k is a monotonically increasing positive sequence, the iterations converge to the optimal solution of problem (6.4) [11]. However, solving equation (6.6) directly is difficult; therefore, the solution is approximated using an alternating strategy minimizing the augmented Lagrange function with respect to each component separately:

$$\begin{aligned} A_{k+1} &= \arg\min_{A} L(A, E_k, Y_k), \\ E_{k+1} &= \arg\min_{E} L(A_{k+1}, E, Y_k). \end{aligned} \tag{6.7}$$

Since the term $M_i U_i U_i^T$ can be pre-computed, it is considered a constant, and therefore we can directly use the singular value thresholding algorithm [19], and derive the update steps in equation (6.7), which can be easily shown to be in the following closed forms:

$$\begin{aligned} W\Sigma V^T &= \text{svd}[(\beta^k M_i - \beta^k E_i^k + Y_i^k + \\ &\qquad \tau M_i U_i U_i^T)/\alpha^k], \\ A_{k+1} &= U S_{1/\alpha^k}(\Sigma) V^T, \\ E_{k+1} &= S_{\lambda/\beta^k}(M_i - A_i^{k+1} - \frac{1}{\beta^k} Y_i^k), \end{aligned} \tag{6.8}$$

where $\alpha^k = \beta^k + \tau$, svd$(\cdot)$ denotes a full singular value decomposition, and $S_\alpha(\cdot)$ is the soft-thresholding operator defined for a scalar x as: $S_\alpha(x) = \text{sign}(x) \cdot \max\{|x| - \alpha, 0\}$. Algorithm 4 summarizes our proposed approach.

Algorithm 4: Complex Event Recognition

Input : Training samples for event i

$$M_i \in \mathbb{R}^{m_i \times d}$$

Annotated training samples for event i

$$Z_i \in \mathbb{R}^{z_i \times d}$$

Output: Annotation-constrained low-rank estimation

$$A \in \mathbb{R}^{m_i \times d}$$

Obtain the event basis from the annotation

$$W\Sigma V^T = \text{svd}(Z_i)$$

Select the most significant q vectors

$$U_i = W(1 \dots q)$$

while *not converged* **do**

Minimize the Lagrange function in equation (6.5)

$$W\Sigma V^T = \text{svd}[(\beta^k M_i - \beta^k E_i^k + Y_i^k + \tau M_i U_i U_i^T)/\alpha^k],\ A_{k+1} = U S_{1/\alpha^k}(\Sigma) V^T,$$

$$E_{k+1} = S_{\lambda/\beta^k}(M_i - A_i^{k+1} - \tfrac{1}{\beta^k} Y_i^k),$$

$$Y_i^{k+1} = Y_i^k + \beta^k (F_i^{k+1} - A_i^{k+1} - E_i^{k+1}),$$

$$\beta_{k+1} = \rho \beta_k.$$

end

6.3 Experiments

We extensively experimented on our method using the most challenging multimedia event datasets, TRECVID MED 2011 and 2012, which exhibit a wide range of challenges including camera motion, cluttered background and illumination changes. Inarguably, one of the biggest challenges in these datasets is the significantly varying video length, which ranges from 30 seconds to over 30 minutes. The frame rate also ranges from 12 to 30 fps, and the resolution ranges from 320×480 to 1280×2000. We report our performance using average precision (AP) and DET curves, similar to [111] and [53]. The datasets contain 4062 videos (2062 for MED 11 and 2000 for MED 12). Similar to [53], we split the data into 70% for training and 30% for testing. The concept models are learned using about half of the training data.

In order to compute the concept scores for a video, we divide it uniformly into overlapping clips, where the clip length is 180 frames, and the step size is 90 frames. Consequently, the confidence scores of each detector is computed for every clip, followed by max-pooling. Thus, our high-level feature for a video is a 93 dimensional vector, which contains the scores of the concept detectors. Consequently, we process the feature vectors for each event using our constrained low-rank optimization. The inputs to our method are two matrices. The first matrix contains the automatically obtained high-level features of the event. The second matrix contains the manually annotated high-level features of the same event, such that for every annotated video we construct an 93 dimensional vector which is set to zero for all the concepts except the annotated ones. We set the score for the annotated concepts to 0.2, which is the max-

imum score that a concept can have in our original training matrix. All the other parameters are selected using cross validation on the training data.

We compare our approach to seven methods:

- Our baseline: Here we use a SVM directly on the obtained high-level features (concept scores), and skip the constrained low-rank optimization. We refer to this setup in the tables as (Base).

- Naive Augmentation: We attempt to enforce the annotation in the event model by directly training a SVM using both the automatically obtained concept representation $\mathcal{M}$ and the manually annotated set $\mathcal{Z}$ (We augment the two sets and form one combined training set). We refer to this setup in the tables as (Naive).

- Structural SVM [110]: A structured SVM learns a discriminative classifier for a general structured output space instead of binary labels. In that, the model is trained using the joint input-output space, and thus can predict the structured output by evaluating the compatibility score of any input-output pair. In the context of our problem, where we aim to refine the concept scores such that they follow the annotation, we can consider the annotation as a structured output. In particular, we use the automatically annotated samples $\mathcal{M}$ as an input for training, and associate each sample in $\mathcal{M}$ with a corresponding structured output from the manual annotation set $\mathcal{Z}$. Consequently, we use SSVM to learn a classifier which predicts the annotation for the high-level features (the concept scores). Since the relation between the annotation in $\mathcal{Z}$ and the event label is predefined and one-to-one, we can predict the event label using the predicted annotation. We

use the SSVM implementation provided from [110], with margin rescaling and 1-slack algorithm as described in [110]. We refer to this approach in the tables as (SSVM).

- Low-Rank Optimization: Here we obtain a low-rank model for the events without using the annotation constraint (corresponds to setting τ to zero in equation 6.3). We refer to this setup in the tables as (LR).

- Bag of Words (BOW): In this, we cluster the low-level features (STIP, ISA, DTF-HOG) to obtain a dictionary. Consequently, we compute a histogram of word frequency for each feature. We use 10000 codebook size for all the features, and min-max normalization for the histograms. Note that the low-level features perform well; however, they lack the semantic cues contained in the concept-representation. We refer to this approach in the tables by the name of the features used in the BOW framework (i.e STIP, ISA, or DTF-HOG).

- Izadinia and Shah [53]: Here the presence/absence of the low-level events in the complex video is considered as a latent variable. Consequently, a Latent-SVM is employed to learn a discriminative event model. This is similar in concept to the Structural SVM formulation which we described above, except that here the annotation matrix $\mathcal{Z}$ is considered missing/unobservable instead of being considered as a structured output.

- Yang and Shah [129]: In that, deep learning is used to compute data-driven concepts. In particular, low-level features are first learned using Topography Independent Component Analysis (TICA) [65]. Consequently, a Deep Belief Network (DBN) is employed for

dimensionality reduction. Finally, the data-driven concepts are learned by clustering the training data in a low-dimensional space using vector quantization (VQ) [119].

6.3.1 TRECVID MED 2011 Event Collection

Table 6.1: Average precision for TRECVID MED 2011 event collection using only high-level features (concept representation).

Event	Base	Naive	SSVM	LR	ACLR
Boarding trick	83.8	78	63.7	83.5	83.5
Feeding animal	69.9	60	60.9	71.4	69.4
Landing fish	54.6	68	65.4	58.5	74.3
Wedding	78.6	76	66.0	85.7	85.5
Wood project	67.8	66	63.2	71.1	69.6
Birthday party	73.1	76	59.2	75.5	77.7
Changing tire	55.2	56	63.2	59.1	74.7
Flash mob	82.0	84	53.2	85.8	91.1
Vehicle unstuck	69.8	66	57.5	78.0	77.9
Grooming animal	78.1	52	59.2	81.4	85.3
Making sandwich	72.8	52	57.4	69.7	70.3
Parade	76.0	74	56.7	81.7	82.7
Parkour	56.4	82	64.0	62.5	69.1
Repairing appliance	77.6	74	53.0	79.9	80.5
Sewing project	71.0	64	57.4	72.9	75.9
Mean AP	**71.1**	**68.2**	**60.0**	**74.4**	**77.8**

TRECVID MED 2011 contains 15 complex events including, *Boarding trick*, *Feeding animal*, *Landing fish*, *Wedding*, *Woodworking project*, *Birthday party*, *Changing tire*, *Flash mob*, *Vehicle unstuck*, *Grooming animal*, *Making sandwich*, *Parade*, *Parkour*, *Repairing appli-*

ance and *Sewing project*. Each event has an event-kit, which includes a verbal description. We selected 110 mostly human action-centric concepts based on the description in the kits and by viewing example videos. Only the concepts which have over 40 examples were retained, and the rest were discarded. Therefore, we finally ended up with 93 action concepts. We extracted various low-level features to represent the concepts, including: Dense trajectory-based MBH and HOG [120], as well as STIP [64].

We present the performance results for TRECVID MED 11 in two tables: First, Table Table 6.1, which shows the average precision for the methods which are based on high-level features (concept representation). These include: the baseline (Base), the naive augmentation (Naive), Structured SVM (SSVM), Low-Rank optimization (LR), and our annotation constrained Low-Rank optimization (ACLR). As can be clearly observed, our method outperforms all other approaches by a significant margin, and improves the performance of the baseline concept representation by 7.7%. Second: Table Table 6.2, which shows the average precision for both the methods which are based on high-level features (concept representation) and low-level features (BOW of STIP/DTF-HOG/ISA features). In the table, we also demonstrate the performance of different feature combinations including: Combining all the low-level features (STIP+HOG+ISA), and combining all the low-level features with the high-level concept features obtained after applying our ACLR approach. All the feature combinations are computed by early fusion (concatenating the feature vectors). When fusing low-level and high-level features, we reduce the dimensionality of the low-level features (BOW histograms) from 10000 to 200 using PCA, such that it is compatible with the dimensionality of the high-level features (93).

Table 6.2: The average precision for TRECVID MED 2011 event collection using both high-level and low-level features. Columns 1-3 show the average precision for the low-level features. Column 4 shows the average precision obtained using our high-level concept models. Column 5 shows the average precision for the combination of columns 1, 2, and 3. Columns 6 and 7 show the average precision for [53] and [129], respectively. Finally, the last column shows the final result obtained by combining the features from columns 1, 2, 3, and 4.

Event	ISA (1)	HOG (2)	STIP (3)	ACLR (4)	(1+2+3)	[53]	[129]	(1+2+3+4)
Boarding trick	79.31	82.98	86.23	83.50	83.12	75.7	78	88.93
Feeding animal	85.42	70.24	69.64	69.41	71.25	56.5	60	84.04
Landing fish	83.39	95.45	84.44	74.30	91.48	72.2	68	94.54
Wedding	78.73	84.79	85.38	85.80	83.45	67.5	76	89.57
Wood project	79.84	77.91	76.38	69.66	82.64	65.3	66	85.75
Birthday party	77.71	86.09	77.61	77.67	82.00	78.2	76	82.49
Changing tire	63.74	64.52	81.95	74.73	67.32	47.7	56	62.95
Flash mob	84.63	95.22	92.38	91.13	92.21	91.9	84	94.89
Vehicle unstuck	78.15	86.44	80.30	77.95	84.88	69.1	66	89.75
Grooming animal	65.74	86.45	71.39	85.32	69.32	51.0	52	75.83
Making sandwich	67.42	69.14	75.97	70.37	80.85	41.9	52	79.00
Parade	77.30	84.60	74.63	82.71	86.21	72.4	74	85.14
Parkour	84.21	81.64	92.54	69.18	93.29	66.4	82	95.32
Repairing appliance	72.80	73.15	82.10	80.56	82.42	78.2	74	78.21
Sewing project	69.91	74.90	76.74	75.94	79.90	57.5	64	78.38
Mean AP	**76.56**	**80.91**	**78.52**	**77.8**	**82.02**	**66.10**	**68.20**	**84.32**

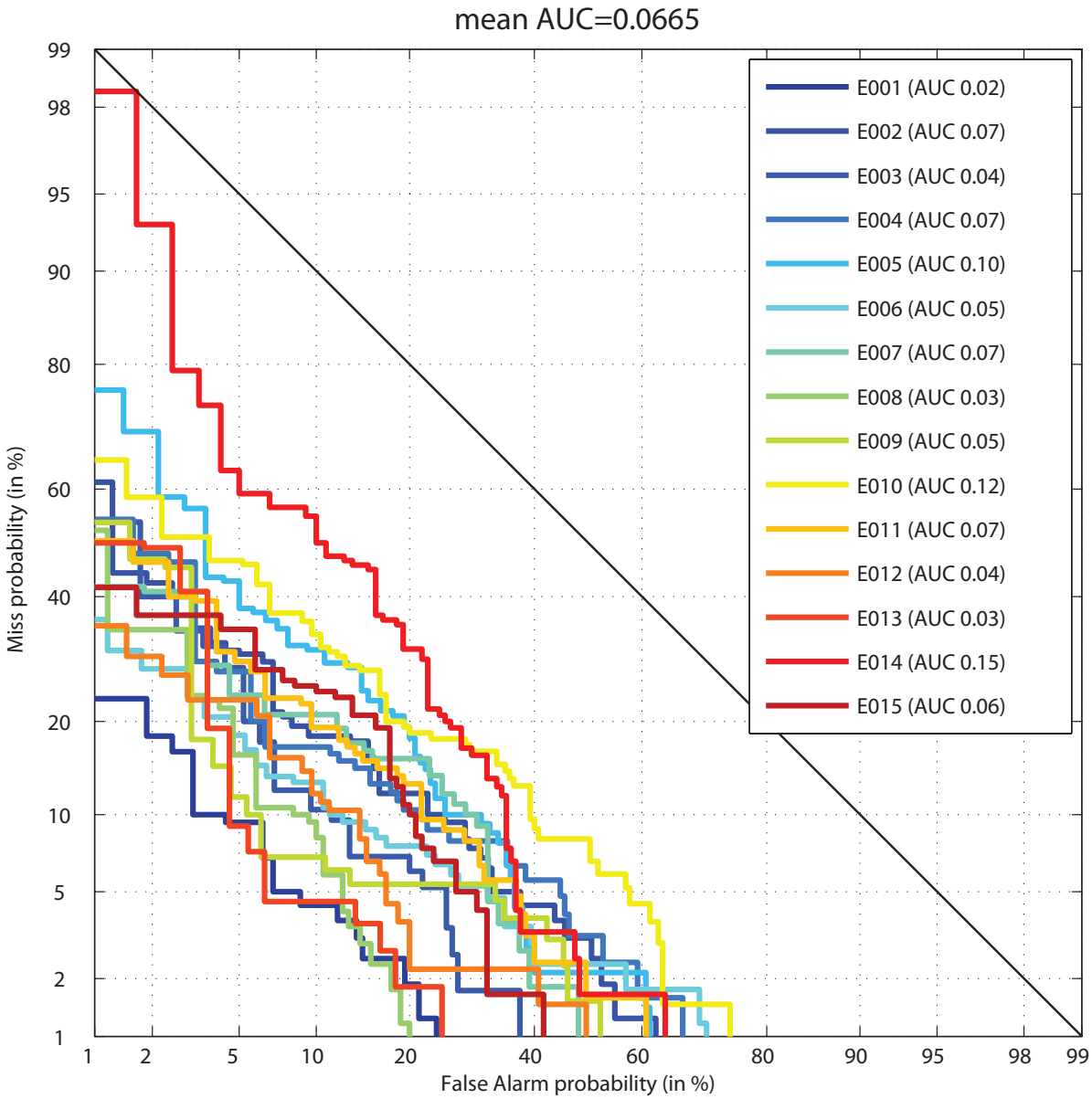

Figure 6.3: DET curves for TRECVID MED 2011 using DTF-HOG, DTF-MBH, and STIP.

It can be clearly observed in Table Table 6.2 that the performance of the combination of the low-level features and our concept representation attains the highest average precision on TRECVID MED 11. Note that the performance results of STIP and HOG features are slightly higher than our concept representation, which is expected since low-level features generally outperform concept-representation as also noted in [37] and [38]. However, the concept repre-

sentation is important and preferable because it carries semantically meaningful description of the video, which can be employed in other relevant tasks such as TRECVID Multimedia Event Recounting (MER), which recounts the important concepts that led to the conclusion that a particular multimedia clip contains an instance of particular event.

Table 6.3: Average precision for TRECVID MED 2012 event collection using only high-level features (concept representation).

Event	Base	Naive	SSVM	LR	ACLR
Bike trick	65.35	63.67	70.15	65.69	69.82
Cleaning appliance	58.49	71.52	53.81	60.25	66.11
Dog show	64.33	66.24	56.69	62.89	69.45
Giving direction	56.57	63.03	47.76	66.60	78.40
Marriage proposal	58.95	62.46	62.34	70.93	67.34
Renovating home	64.94	68.39	59.44	65.86	71.59
Rock climbing	63.05	80.49	64.65	66.55	69.26
Town hall meeting	67.84	75.30	63.45	60.05	72.56
winning a race	67.97	71.66	59.91	68.93	76.15
metal project	73.73	66.57	65.09	72.76	69.60
Mean AP	**64.08**	**68.93**	**60.33**	**66.05**	**71.03**

In Table Table 6.2, we also compare with the results from [53] and [129], which we significantly outperform. It is important to note that [53] and [129] use different features, which are not publicly available, and some of them are manually annotated and impossible to regenerate. However, it is necessary to compare with them since they constitute the state-of-the art on TRECVID 11. Figure Figure 6.3 shows the DET curves for events 1 to 15 from TRECVID MED11.

Table 6.4: The average precision for TRECVID MED 2012 event collection using both high-level and low-level features. Columns 1-3 show the average precision for the low-level features. Column 4 shows the average precision obtained using our high-level concept models. Column 5 shows the average precision for the combination of columns 1, 2, and 3. Finally, the last column shows the final result obtained by combining the features from columns 1, 2, 3, and 4.

Event	ISA (1)	HOG (2)	STIP (3)	ACLR (4)	(1+2+3)	(1+2+3+4)
Bike trick	65.52	63.67	67.30	69.82	74.55	75.77
Cleaning appliance	59.80	71.52	71.82	66.11	71.31	72.40
Dog show	52.48	66.24	58.52	69.45	64.89	68.32
Giving direction	66.39	63.03	63.32	78.40	68.66	67.49
Marriage proposal	59.53	62.46	77.33	67.34	75.84	75.31
Renovating home	70.19	68.39	68.86	71.59	73.56	77.06
Rock climbing	70.02	80.49	73.38	69.26	81.21	85.79
Town hall meeting	65.84	75.30	60.83	72.56	75.40	70.48
winning a race	62.25	71.66	69.62	76.15	68.04	71.17
metal project	71.45	66.57	75.79	69.60	74.70	78.89
Mean AP	**64.35**	**68.93**	**68.68**	**71.03**	**72.82**	**74.26**

6.3.2 TRECVID MED 2012

TRECVID MED 2012 is similar to TRECVID MED 2011, except that it contains 10 more events including: *Bike trick*, *Cleaning and appliance*, *Dog show*, *Giving direction*, *Mar-*

riage proposal, *Renovating a home*, *Rock climbing*, *Town hall meeting*, *Race winning*, *Metal craft project*. The concept detectors are trained similar to TRECVID MED 2011; therefore, we also ended up with 93 action concepts, and a corresponding 93-dimensional feature vector for each video.

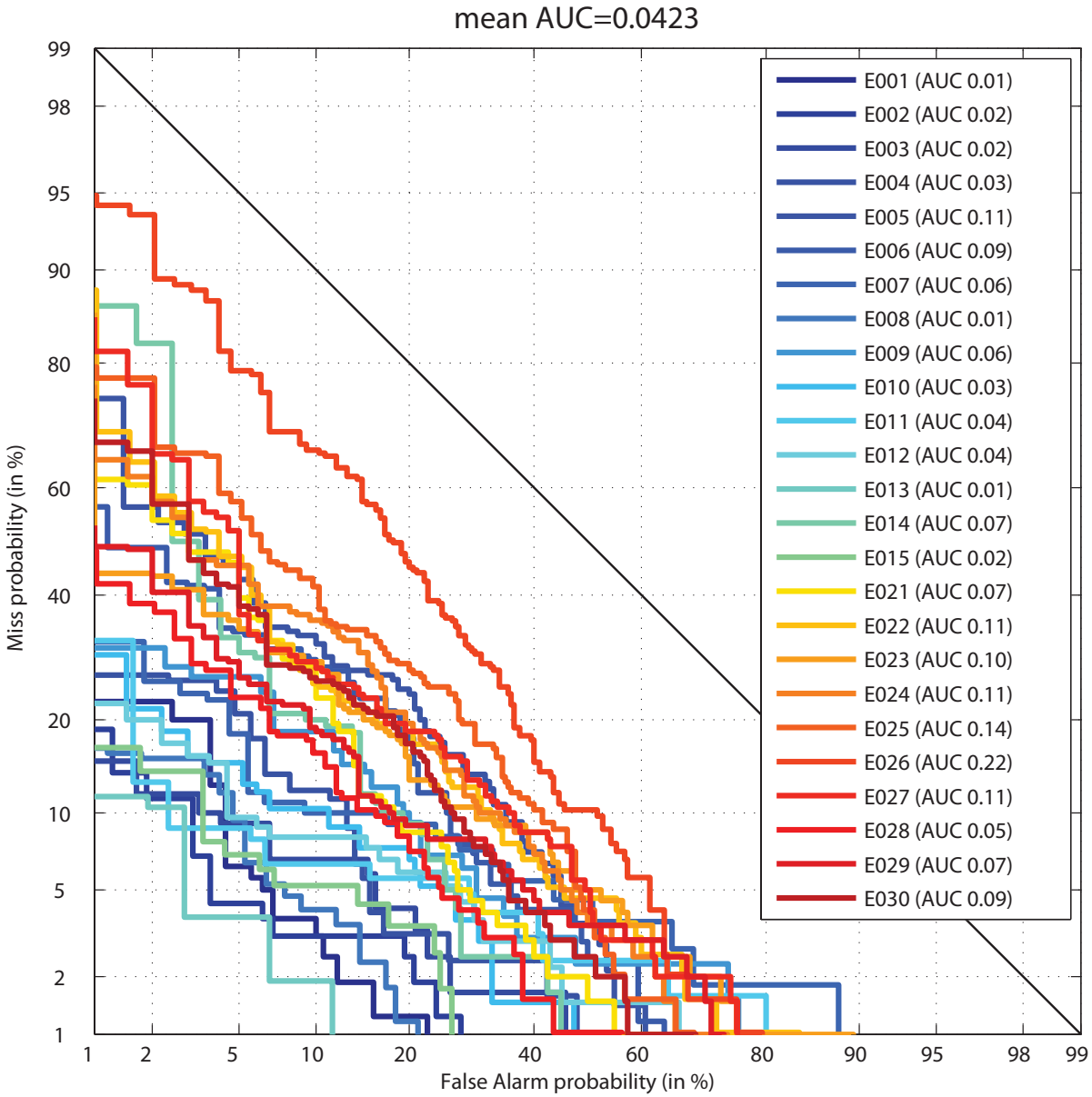

Figure 6.4: DET curves for TRECVID MED 2012 using DTF-HOG.

Similar to TRECVID MED 2011, we present the performance results in two tables: First, Table Table 6.3, which shows the average precision for the methods which are based on high-level features. Our method outperforms all other approaches by a significant margin, and improves the performance of the baseline concept representation by 6.95%. Second: Table Table 6.4, which shows the average precision for the methods which are based on high-level features, low-level features, and different feature combinations. It can be clearly observed in Table Table 6.4 that the performance of the combination of the low-level features and our concept representation attains the highest average precision on TRECVID MED 12. Additionally, unlike in TRECVID MED 11, here the low-level features are inferior to the concept representation, unless all the low-level features are combined. Figure Figure 6.4 shows the DET curves for events 1 to 25 from TRECVID MED12.

6.3.3 Refining the Concept Representation

Using our constrained low-rank optimization framework, we obtain more meaningful concept representations, in which the noisy concepts are suppressed, and replaced with new concepts which are more semantically meaningful. This is illustrated in Figure Figure 6.5, which shows the average concept scores of all the training samples before and after applying our method for the events *Attempting a Boarding Trick* from TRECVID MED 11 and *Attempting a Bike Trick* from TRECVID MED 12. In the event *Attempting a Boarding Trick*, several essential concepts are missing from the original features (have insignificant scores) such as *falling*, *flipping the board*, and *spinning the board*. Additionally, several irrelevant concepts are falsely firing, such as *riding bike on one wheel*, *running next to dog*, and *scaling walls*

trees. Similarly, in the event *Attempting a Bike Trick*, several essential concepts are missing from the original features such as *flipping the bike*, *spinning the bike handle*, and *standing on top of bike*. Moreover, several irrelevant concepts are falsely firing, such as *animal grabs food*, *running next to dog*, and *standing on the board*. Through our method, the concepts are enforced to follow the annotation. Therefore, the missing concepts are induced, while the irrelevant concepts are suppressed, thus generating a new concept representation which is less noisy and more semantically meaningful.

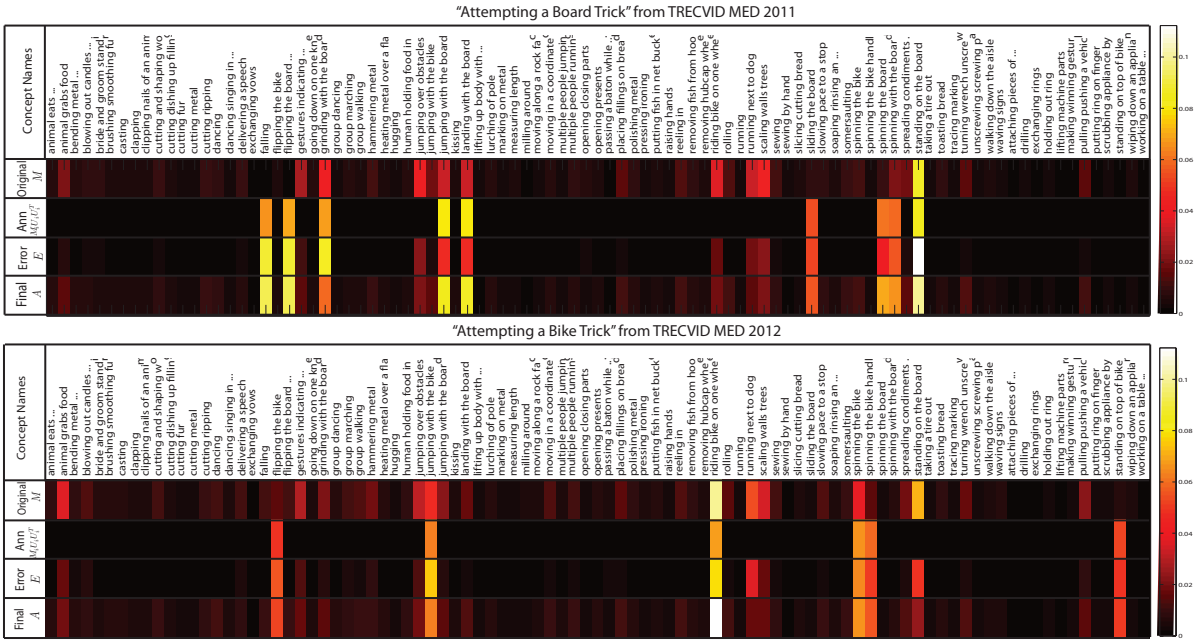

Figure 6.5: Refining the concepts for the event *Attempting a Board trick* from TRECVID MED 11 and the event *Attempting a Bike trick* from TRECVID MED 12. The average of the training examples are shown in the rows. In each of the two examples, the rows from the top to the bottom show: The original feature, the original feature projected on the basis of the annotation, the extracted noisy concepts, and the final concept representation. In the event *Attempting a Boarding Trick* (top), several essential concepts are missing from the original features such as *falling*, *flipping the board*, and *spinning the board*. Additionally, several irrelevant concepts are falsely firing, such as *riding bike on one wheel*, *running next to dog*, and *scaling walls trees*. Similarly, in the event *Attempting a Bike Trick* (bottom), several essential concepts are missing from the original features such as *flipping the bike*, *spinning the bike handle*, and *standing on top of bike*. Moreover, several irrelevant concepts are falsely firing, such as *animal grabs food*, *running next to dog*, and *standing on the board*. After applying our method (bottom row in each example), the missing concepts are induced, while the irrelevant concepts are suppressed, thus generating a new concept representation which is less noisy and more semantically meaningful.

6.4 Summary

We presented a novel, simple, and easily implementable method for complex event recognition. We first divide the training data into two sets, one where we annotate the concepts manually, and another where we detect the concepts automatically with models trained using the first set. Consequently, we exploit the inherent low-rank structure in the examples of an event, and combine the two training sets in one set which is not only low-rank but also encouraged to follow the annotation. Thus, combining the rich descriptors in the automatically annotated set, with the accurate manual annotations. This suppresses the miss-detected concepts and generates robust and discriminative event models, which, as we verified in the experiments, outperform all previous approaches on the relevant benchmark datasets.

CHAPTER 7: CONCLUSION

In this thesis, we proposed four low-rank optimization-based solutions for fundamental computer vision problems including scene reconstruction, turbulence mitigation and background subtraction, and action recognition.

In the first problem, we presented a novel, simple, and easily implementable method to reconstruct a sequence distorted by water waves. An iterative registration algorithm is first employed to recover a well aligned sequence and an enhanced mean. Consequently, sparse errors are extracted through rank minimization. We showed by experiments that the proposed method robustly recovers several sequences and highly outperforms state of the art. Our method is a general dewarping technique; therefore, in the future, we will investigate its application to further types of noise and turbulence.

For the second problem, we presented a novel method for concurrent turbulence mitigation and moving object detection. Our method leverages the low-rank, the Gaussian, and the sparse properties of the sequence, decomposing it into the background, the turbulence, and the moving objects, respectively.

Third, we proposed a novel method for recognizing human actions in videos acquired by moving cameras. To the best of our knowledge, this is the first work which employs Lagrangian particle trajectories for action recognition. Our method is able to extract a large number of particle trajectories corresponding to the motions; therefore, it better captures the

articulation of human actions which improves the recognition performance. Particle trajectories are easily obtained by advecting pixel-wise optical flow; thus, representing the ensemble motions of a scene, from which we extract the independent object motion through a novel method using rank optimization. This enables our method to avoid traditional trajectory acquisition techniques which require video alignment, object detection, and tracking. Through experiments, we have demonstrated the robustness of the proposed approach while outperforming the state-of-the-art on several benchmark datasets.

Finally, we presented a novel, simple, and easily implementable method for complex event recognition. We first divide the training data into two sets, one where we annotate the concepts manually, and another where we detect the concepts automatically with models trained using the first set. Consequently, we exploit the inherent low-rank structure in the examples of an event, and combine the two training sets in one set which is not only low-rank but also encouraged to follow the annotation. Thus, combining the rich descriptors in the automatically annotated set, with the accurate manual annotations. This suppresses the miss-detected concepts and generates robust and discriminative event models, which, as we verified in the experiments, outperform all previous approaches on relevant benchmark datasets.

7.1 Summary of Contributions

Our main contributions are summarized below.

1. Underwater scene reconstruction

(a) Proposing a new data-driven two-stage approach for recovering the original image of an underwater scene using a sequence distorted by water waves.

(b) Decomposing the noise caused by water turbulence into deformation effects and sparse errors.

(c) Developing a new iterative non-rigid registration approach, where the frames are registered to their mean iteratively.

(d) Formulating the water turbulence mitigation problem within a low-rank optimization framework and solving it efficiently.

2. Simultaneous video stabilization and background subtraction in turbulence

(a) Developing a new variant of low-rank matrix decomposition, where the turbulence video is decomposed into three components: Low-rank background, sparse moving objects, and Gaussian noise.

(b) Proposing a new turbulence noise model based on both intensity and motion cues, where the motion distribution is derived from the Lagrangian particle advection framework.

(c) Formulating an additional force component in the particle advection framework in order to stabilize the particles in the turbulent medium and handle long sequences without discontinuities.

3. Action recognition by motion decomposition of Lagrangian particle trajectories

(a) Proposing a new action recognition method, which utilizes automatically obtained dense particle trajectories to obtain rich activity features.

(b) Developing a new approach to estimate the camera motion subspace using low-rank optimization.

(c) Using a novel low-rank optimization approach to decompose the particle trajectories into their camera-motion and object-motion components. Additionally, the proposed method is able to separate the object motion into its rigid and articulated motion components.

4. Complex event recognition using constrained low-rank optimization

(a) Proposing a novel low-rank formulation, through which we find a new representation for complex events, which is not only low-rank, but also constrained by the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event.

(b) Proposing a low-rank event model which is unrestricted to certain features, which allows us to employ a combination of state-of-the-art features including STIP [64], DTF-MBH [120], and DTF-HOG [120].

(c) Manually defining and annotating an action concept dataset based on TRECVID event collection, with 81 concepts for TRECVID MED11 and 93 for TRECVID MED12, using more than $20,000$ video clips.

7.2 Future Work

In this section we explore some of the possible improvements, extensions, and directions that can be explored.

The obvious direction to take in the underwater scene reconstruction framework, is to combine the two-stages of robust registration and sparse noise elimination in one coherent framework. However significant challenges face this direction because it is very difficult to integrate the highly non-linear operations employed in the non-rigid registration into the convex matrix optimization. The main limitation stems from the current math techniques available for matrix completion and factorization, which are limited to certain norms and formulations, the most popular being RPCA. In the literature, some constraints were integrated in the RPCA, such as image transformations as in [94]; however, such transformations are linear and directly correlated to the rank of the matrix. To include non-rigid registration in low-rank optimization is far more complicated, and thus needs significant further research, which we are planning to conduct.

On the other hand, we will investigate further applications of our three-term decomposition. In particular, we have shown how RPCA can be employed in action recognition to decompose the trajectories of an action in a moving camera scenario into low-rank components corresponding to camera motion and rigid object motion, and a sparse component corresponding to the articulated object motion. In an extension to that, we will investigate the use of the three-term decomposition to achieve finer decompositions of action sequences, and extract further components including the component corresponding to the Gaussian noise. We are also investigating the potential extension of the three-term-decomposition into a generic n-component decomposition, and its possible applications in clustering and segmentation problems.

APPENDIX: DERIVATION OF ALM EQUATIONS

In this chapter, we show the extended derivations for the solutions of the Low-Rank optimization problems which appeared in Chapter 4. In particular, we first provide the derivation of equation set (4.8), which is the solution for the minimization problem in equation set (4.7). Consequently, we provide the proof of theorem 1. In all derivations we refer to the Lagrange function L defined in equation (4.5).

- Derivation of the update step for A

$$A_{k+1} = \arg\min_{A} L(A, O_k, E_k, Y_k)$$

Dropping indices k and $k+1$ for simplicity

$$\begin{aligned} A &= \arg\min_{A} L(A, O, E, Y) \\ &= \arg\min_{A} ||A||_* + \langle Y, -A \rangle \\ &\quad + \frac{\beta}{2}||F - A - O - E||_F^2. \end{aligned}$$

The Frobenious norm is induced from the inner product, i.e. $||X||_F^2 = \langle X, X \rangle$. Therefore, replacing the Frobenious norm with an inner product then expanding the inner product and separating A we obtain

$$
\begin{aligned}
A &= \arg\min_A ||A||_* + \langle Y, -A\rangle \\
&\quad + \frac{\beta}{2}\{||A||_F^2 - 2\langle F - O - E, A\rangle\} \\
&= \arg\min_A ||A||_* + \\
&\quad \frac{\beta}{2}\{||A||_F^2 - 2\langle \beta^{-1}Y + F - O - E, A\rangle\} \\
&= \arg\min_A ||A||_* + \\
&\quad \frac{\beta}{2}||A - \beta^{-1}Y - F + O + E||_F^2 \\
&= \arg\min_A \beta^{-1}||A||_* + \\
&\quad \frac{1}{2}||A - (\beta^{-1}Y + F - O - E)||_F^2.
\end{aligned}
$$

Using the result from Singular Value Thresholding algorithm [19] we get

$$A = US_{\frac{1}{\beta}}(\Sigma)V^T,$$

where $U\Sigma V^T$ is the SVD of W, $W = \beta^{-1}Y + F - O - E$, and $S_\alpha(\cdot)$ is the soft thresholding operator defined in equation (4.9).

- Derivation of the update step for O

$$O_{k+1} = \arg\min_O L(A_{k+1}, O, E_k, Y_k)$$

Dropping indices k and $k+1$ for simplicity

$$\begin{aligned} O &= \arg\min_O L(A,O,E,Y) \\ &= \arg\min_O \tau||\Pi(O)||_1 + \langle Y, -O\rangle \\ &\quad + \frac{\beta}{2}||F-A-O-E||_F^2 \\ &= \arg\min_O \tau||\Pi(O)||_1 + \langle Y, -O\rangle \\ &\quad + \frac{\beta}{2}\{||O||_F^2 - 2\langle F-A-E, O\rangle\} \\ &= \arg\min_O \tau||\Pi(O)||_1 + \\ &\quad \frac{\beta}{2}\{||O||_F^2 - 2\langle \beta^{-1}Y + F-A-E, O\rangle\} \\ &= \arg\min_O \frac{\tau}{\beta}||\Pi(O)||_1 + \\ &\quad \frac{1}{2}||O-(\beta^{-1}Y+F-A-E)||_F^2. \end{aligned}$$

Let $X = \beta^{-1}Y + F - A - E$, then

$$O = \arg\min_O \frac{\tau}{\beta}||\Pi(O)||_1 + \frac{1}{2}||O-X||_F^2.$$

Using convex optimization theory [86, 82], 0 is in the subdifferential ∂ of the function:

$$\begin{aligned} 0 &\in \partial(\frac{\tau}{\beta}||\Pi(O)||_1 + \frac{1}{2}||O-X||_F^2) \\ &= \partial(||\frac{\tau}{\beta}\Pi(O)||_1) + O - X \\ &= \frac{\tau}{\beta}\Pi \text{ sign}(O) + O - X, \end{aligned}$$

which can be expressed as:

$$\begin{cases} 0 = \frac{\tau}{\beta}\Pi_{i,j} + O_{i,j} - X_{i,j} & \text{if } O_{i,j} > 0, \\ 0 = -\frac{\tau}{\beta}\Pi_{i,j} + O_{i,j} - X_{i,j} & \text{if } O_{i,j} < 0, \\ 0 \in [-1,1]\frac{\tau}{\beta}\Pi_{i,j} + O_{i,j} - X_{i,j} & \text{if } O_{i,j} = 0, \end{cases}$$

where $O_{i,j}$ is the (i,j)-element of the matrix O. Rearranging the above equation we get

$$O_{i,j} = \begin{cases} X_{i,j} - \frac{\tau}{\beta}\Pi_{i,j} & \text{if } X_{i,j} > \frac{\tau}{\beta}\Pi_{i,j} \\ X_{i,j} + \frac{\tau}{\beta}\Pi_{i,j} & \text{if } X_{i,j} < -\frac{\tau}{\beta}\Pi_{i,j} \\ 0 & \text{if } |X_{i,j}| \leq \frac{\tau}{\beta}\Pi_{i,j} \end{cases}$$

This piecewise function is equivalent to the soft thresholding operator S defined in equation (4.9); hence, we can rewrite the above equation as

$$\begin{aligned} O &= S_{\frac{\tau}{\beta}\Pi}(X), \\ &= S_{\frac{\tau}{\beta}\Pi}(\beta^{-1}Y + F - A - E). \end{aligned}$$

- Derivation of the update step for E

$$E_{k+1} = \arg\min_{E} L(A_{k+1}, O_{k+1}, E, Y_k)$$

Dropping indices k and $k+1$ for simplicity

$$\begin{aligned}
E &= \arg\min_E L(A, O, E, Y) \\
&= \arg\min_E \lambda||E||_F^2 + \langle Y, -E\rangle \\
&\quad + \frac{\beta}{2}||F - A - O - E||_F^2 \\
&= \arg\min_E \lambda||E||_F^2 + \\
&\quad \frac{\beta}{2}\{||E||_F^2 - 2\langle \beta^{-1}Y + F - A - O, E\rangle\} \\
&= \arg\min_E \ (\frac{2\lambda}{\beta} + 1)\{||E||_F^2 - 2 \\
&\quad \langle (\frac{2\lambda}{\beta} + 1)^{-1}(\beta^{-1}Y + F - A - O), E\rangle\} \\
&= \arg\min_E ||E - (\frac{2\lambda}{\beta} + 1)^{-1} \\
&\quad (\beta^{-1}Y + F - A - O)||_F^2.
\end{aligned}$$

Therefore, the minimum is obtained at

$$E = (\frac{2\lambda}{\beta} + 1)^{-1}(\beta^{-1}Y + F - A - O).$$

- **Proof of Theorem 1**

The proof of Theorem 1 in Section 4.3 can be derived from the following lemmas.

Let A_k, O_k, E_k, Y_k, and β_k be as generated by Algorithm 1.

Lemma 1: Let

$$\begin{aligned}
a_k &= Y_k + \beta_k(F - A_{k+1} - O_k - E_k), \\
b_k &= Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_k), \\
c_k &= Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_{k+1}),
\end{aligned}$$

then the sequences $\{a_k\}$, $\{b_k\}$, and $\{c_k\}$ are bounded.

Note that according to Algorithm 1, we have $Y_{k+1} = c_k$, and recall that $L(\cdot)$ is the Lagrange function defined in equation (5).

Lemma 2: Let

$$\begin{aligned} L_{k+1} &= L(A_{k+1}, O_{k+1}, E_{k+1}, Y_k, \beta_k), \\ e_k &= \|F - A_k - O_k - E_k\|_F^2. \end{aligned}$$

Then $e_k \leq c\beta_{k-1}^{-2}$ for some constant $c > 0$, and

$$L_{k+1} - L_k \leq \frac{\beta_k + \beta_{k-1}}{2} e_k, \;\; k = 1, 2, ...$$

At every iteration in Algorithm 1, we set β_{k+1} to $\rho\beta_k$ with $\rho > 1$. Therefore, the above inequality can be rewritten as:

$$L_{k+1} - L_k \leq \frac{1+\rho}{c\beta_{k-1}}, \;\; k = 1, 2, ...$$

Since $\{\beta_k\}$ is an increasing geometric sequence, we see that Lemma 2 implies the boundedness of the sequence $\{L_k\}$ and that $\lim_{k\to\infty}(F - A_k - O_k - E_k) = 0$, which implies that any accumulation points (if any) of (A_k, O_k, E_k) approaches a feasible solution to the desired decomposition. The following Lemma implies that such accumulation points exist.

Lemma 3: The sequences $\{A_k\}$, $\{O_k\}$, and $\{E_k\}$ are bounded.

Proofs of The Lemmas

Since we are dealing with finite dimensional Euclidean spaces, all norms are equivalent, and a bounded sequence in one norm is also bounded in any other norms. Therefore, we

do not specify the type of the norm unless needed. Additionally, we are assuming the positive sequence $\{\beta_k\}$ satisfies $\sum 1/\beta_k < \infty$.

***Proof of Lemma 1**:*

(i) Proof that $\{a_k\}$ is bounded:

We follow Lin et. al. [71] and note that

$$A_{k+1} = \arg\min_A L(A, O_k, E_k, Y_k)$$

$$\Rightarrow 0 \in \partial_A L(A_{k+1}, O_k, E_k, Y_k)$$

$$\Rightarrow 0 \in \partial \|A_{k+1}\|_* - Y_k - \beta_k(F - A_{k+1} - O_k - E_k)$$

Therefore,

$$a_k = Y_k + \beta_k(F - A_{k+1} - O_k - E_k) \in \partial \|A_{k+1}\|_*.$$

From this, according to Theorem 4 of [71], the sequence $\{a_k\}$ is bounded.

(ii) Proof that $\{b_k\}$ is bounded:

$$O_{k+1} = \arg\min_O L(A_{k+1}, O, E_k, Y_k)$$

$$\Rightarrow 0 \in \partial_O L(A_{k+1}, O_{k+1}, E_k, Y_k)$$

$$\Rightarrow 0 \in \partial(\tau \|\Pi(O_{k+1})\|_1) - Y_k - \beta_k(F - A_{k+1} - O_{k+1} - E_k)$$

Therefore,

$$b_k = Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_k) \in \partial(\tau \|\Pi(O_{k+1})\|_1).$$

Thus, $b_k(i,j) = 0$ if $(i,j) \notin \text{supp}(\Pi)$, and $b_k(i,j) \in \partial(\tau|Q_{k+1}(i,j)|)$ if $(i,j) \in \text{supp}(\Pi)$. Using Theorem 4 of [71] (for the scalar case), the sequence $\{b_k\}$ is bounded.

(iii) Proof that $\{c_k\}$ is bounded:

First, note that

$$\partial(\lambda\|E_{k+1}\|_F^2) = \{2\lambda E_{k+1}\}.$$

Therefore, using

$$E_{k+1} = \arg\min_E L(A_{k+1}, O_{k+1}, E, Y_k),$$

we have

$$0 \in \partial_E L(A_{k+1}, O_{k+1}, E_{k+}, Y_k)$$

$$\Rightarrow 0 \in \partial(\lambda\|E_{k+1}\|_F^2) - Y_k - \beta_k(F - A_{k+1} - O_{k+1} - E_{k+1})$$

Therefore,

$$c_k = Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_{k+1}) \in \{2\lambda E_{k+1}\}.$$

Thus,

$$c_k = 2\lambda E_{k+1}.$$

In Algorithm 1, $Y_{k+1} = c_k$. Hence, $Y_{k+1} = 2\lambda E_{k+1}$. Now, we have obtained

$$\begin{aligned} 2\lambda E_{k+1} &= Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_k) + \\ &\quad \beta_k(E_k - E_{k+1}) \\ &= b_k + \beta_k E_k - \beta_k E_{k+1}. \end{aligned}$$

Solving for E_{k+1}, we obtain

$$E_{k+1} = \frac{b_k}{\beta_k} \cdot \frac{1}{1 + \frac{2\lambda}{\beta_k}} + E_k \cdot \frac{1}{1 + \frac{2\lambda}{\beta_k}}.$$

Using the fact that $\{b_k\}$ is bounded and $\sum_{k=1}^{\infty} 1/\beta_k < \infty$, we conclude that $\{E_k\}$ must be bounded, and thus, the sequence $\{c_k\} = \{Y_{k+1}\} = \{2\lambda E_{k+1}\}$ is also bounded.

Remark. As a consequence of the proof, we obtained the boundedness of $\{E_k\}$ as well.

***Proof of Lemma 2**:*

Write $L_{k+1} = L(A_{k+1}, O_{k+1}, E_{k+1}, Y_k, \beta_k)$. Then

$$\begin{aligned}
L_{k+1} &\leq L(A_{k+1}, O_{k+1}, E_k, Y_k, \beta_k) \\
&\leq L(A_{k+1}, O_k, E_k, Y_k, \beta_k) \\
&\leq L(A_k, O_k, E_k, Y_k, \beta_k) \\
&= \|A_k\|_* + \tau\|\Pi(O_k)\|_1 + \lambda\|E_k\|_F^2 + \\
&\quad \langle Y_k, F - A_k - O_k - E_k\rangle + \\
&\quad \frac{\beta_k}{2}\|F - A_k - O_k - E_k\|_F^2 \\
&= L_k + \\
&\quad \langle Y_k - Y_{k-1}, F - A_k - O_k - E_k\rangle + \\
&\quad \frac{\beta_k - \beta_{k-1}}{2}\|F - A_k - O_k - E_k\|_F^2 \\
&= L_k + \\
&\quad \beta_{k-1}\|F - A_k - O_k - E_k\|_F^2 + \\
&\quad \frac{\beta_k - \beta_{k-1}}{2}\|F - A_k - O_k - E_k\|_F^2.
\end{aligned}$$

Therefore,

$$L_{k+1} - L_k \leq \frac{\beta_k + \beta_{k-1}}{2} e_k.$$

Finally, note that

$$e_k = \|F - A_k - O_k - E_k\|_F^2 = \left(\frac{Y_k - Y_{k-1}}{\beta_{k-1}}\right)^2 = O(\beta_{k-1}^{-2}).$$

The last equality is due to the fact that $Y_k = c_{k-1}$ and $Y_{k-1} = c_{k-2}$ are bounded by Lemma 1. This completes the proof of Lemma 2.

***Proof of Lemma 3**:*

Since $Y_{k+1} = c_k$, the sequence $\{Y_k\}$ is bounded by Lemma 1. Also the sequence L_k is bounded as implied by Lemma 2. Note that

$$\begin{aligned}
\|Y_{k+1}\|_F^2 &= \|Y_k + \beta_k(F - A_{k+1} - O_{k+1} - E_{k+1})\|_F^2 \\
&= \|Y_k\|_F^2 + 2\beta_k\langle Y_k, F - A_{k+1} - O_{k+1} - E_{k+1}\rangle \\
&\quad +\beta_k^2\|F - A_{k+1} - O_{k+1} - E_{k+1}\|_F^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\|Y_{k+1}\|_F^2 - \|Y_k\|_F^2}{2\beta_k} = \langle Y_k, F - A_{k+1} - O_{k+1} - E_{k+1}\rangle + \\
\frac{\beta_k}{2}\|F - A_{k+1} - O_{k+1} - E_{k+1}\|_F^2.
\end{aligned}$$

Since $\{Y_k\}$ is bounded and $\beta_k \to \infty$, we see that

$$\langle Y_k, F - A_{k+1} - O_{k+1} - E_{k+1}\rangle + \frac{\beta_k}{2}\|F - A_{k+1} - O_{k+1} - E_{k+1}\|_F^2$$

converges to 0 (and thus bounded). Consequently, since

$$\begin{aligned} \|A_{k+1}\|_* + \tau\|\Pi(O_{k+1})\|_1 + \lambda\|E_{k+1}\|_F^2 &= L_{k+1} \\ -\langle Y_k, F - A_{k+1} - O_{k+1} - E_{k+1}\rangle & \\ -\frac{\beta_k}{2}\|F - A_{k+1} - O_{k+1} - E_{k+1}\|_F^2, & \end{aligned}$$

we see that $\{A_k\}$ and $\{E_k\}$ are bounded. This, together with the fact that $\lim_{k\to\infty}(F - A_k - O_k - E_k) = 0$, yields that the sequence $\{O_k\}$ is also bounded.

LIST OF REFERENCES

[1] S. Gong A. Psarrou and M. Walter. Recognition of human gestures and behaviour based on motion trajectories. In *Image and Vision Computing*, 2002.

[2] O. Javed A. Yilmaz and M. Shah. Object tracking: a survey. In *ACM Journal of Computing Surveys*, 2006.

[3] A. Agarwal, S. Neghaban, and M. Wainwright. Noisy matrix decomposition: optimal rates in high dimensions. In *preprint*, 2011.

[4] Amit Agrawal, Yi Xu, and Ramesh Raskar. Invertible motion blur in video. In *SIGGRAPH*, 2009.

[5] R. Bhotika A.Hoogs, M. Chan and J. Schmiederer. Recognizing complex behaviors in aerial video. In *ICIA*, 2005.

[6] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, 2007.

[7] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, 2007.

[8] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. In *PAMI*, 2010.

[9] M. Fazel B. Recht and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. In *SIAM Review*, 2010.

[10] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. In *IJCV*, 2008.

[11] D. P. Bertsekas. Nonlinear programming. In *Athena Scientific*, 2004.

[12] A. Bjorck. Numerical methods for least squares problems. In *SIAM*, 1996.

[13] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[14] A. F. Bobick and J. Davis. The recognition of human movement using temporal templates. In *PAMI*, 2001.

[15] H. Branover and Y. Unge. Progress in turbulence research. In *AIAA*, 1994.

[16] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

[17] A. Yilmaz C. Rao and M. Shah. View-invariant representation and recognition of actions. In *IJCV*, 2002.

[18] I. Laptev C. Schuldt and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[19] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. In *SIAM J. on Optimization, Volume 20, Issue 4, pp. 1956-1982*, 2010.

[20] E. Cand'es and T. Tao. Decoding by linear programming. In *IEEE Transactions on Information Theory*, 2005.

[21] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? In *IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2009.

[22] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? In *Preprint*, 2009.

[23] Stanley H. Chan, Ramsin Khoshabeh, Kristofor B. Gibson, Philip E. Gill, and Truong Q. Nguyen. An augmented lagrangian method for total variation video restoration. In *IEEE Transactions on Image Processing*, 2010.

[24] M. Chang, C. Font, C. Gilbreath, E. Oh, E. Distefano, S. Restaino, C Wilcox, and F. Santiago. Comparing horizontal path c2n measurements over 0.6 km in the tropical littoral environment and in the desert. In *SPIE*, 2007.

[25] C. Chen, C. Wei, and Y. Wang. Low-rank matrix recovery with structural incoherence for robust face recognition. *CVPR*, 2012.

[26] Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*. MIT Press, 2001.

[27] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objectss, ghosts and shadows in video streams. In *PAMI*, 2003.

[28] J. Psl D. Meyer and H. Niemann. Gait classification with hmms for trajectories of body parts extracted by mixture densities. In *BMVC*, 1998.

[29] T. Darrell and A. Pentland. Classifying hand gestures with a view-based distributed representation. In *NIPS*, 1993.

[30] P. Dollar, V. Rabaud, V. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[31] A. Donate, G. Dahme, and E. Ribeiro. Classification of textures distorted by water-waves. In *ICPR*, 2006.

[32] A. Donate and E. Ribeiro. Improved reconstruction of images distorted by waterwaves. In *VISAPP*, 2006.

[33] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. In *Comm. on Pure and Applied Mathematics*, 2006.

[34] Alexei Efros, Volkan Isler, Jianbo Shi, and Mirko Visontai. Seeing through water. In *NIPS*, 2004.

[35] A. Elgammal, D. Harwood, and L.S. Davis. Non-parametric model for background subtraction. In *ICCV*, 1999.

[36] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009.

[37] H. Cheng et. al. Team sri-sarnoffs aurora system @trecvid 2011. *In Proceedings of NIST TRECVID, Workshop*, 2011.

[38] H. Cheng et. al. Sri-sarnoff aurora system at trecvid 2012 multimedia event detection and recounting. *In Proceedings of NIST TRECVID, Workshop*, 2012.

[39] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[40] Z. Lin G. Liu and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[41] Z. Lin G. Liu and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

[42] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. In *Computers and Mathematics with Applciations*, 1976.

[43] R. Glowinski and A. Marroco. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite, d'une classe de problemes de dirichlet non lineaires. In *Revue Francaise d'Automatique, Informatique et RechercheOperationelle*, 1975.

[44] D. Goldfarb and S. Ma. Fast multiple splitting algorithms for convex optimization. In *preprint*, 2009.

[45] J. Goodman. Statistical optics. In *Wiley-Interscience*, 2000.

[46] J. Goodman. Introduction to fourier optics. In *Roberts and Company Publishers*, 2004.

[47] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *CVPR*, 2004.

[48] Z. Shen H. Ji, C. Liu and Y. Xu. Robust subspace segmentation by low-rank representation. In *CVPR*, 2010.

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

[50] B. He and X. Yuan. Linearized alternating direction method with gaussian back substitutions for separable convex programming. In *preprint*, 2011.

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

[52] M. Isichenko and A. Gruzinov. Isotopological relaxation, coherent structures, and gaussian turbulence in two-dimensional (2-d) magnetohydrodynamics (mhd). In *Physics of Plasmas*, 1994.

[53] Hamid Izadinia and Mubarak Shah. Recognizing complex events using large margin joint low-level event model. *ECCV*, 2012.

[54] F. Han H. S. Sawhney J. Xiao, H. Cheng. Geo-spatial aerial video processing for scene understanding and object tracking. In *CVPR*, 2008.

[55] H. S. Sawhney J. Xiao, H. Cheng and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. In *CVPR*, 2010.

[56] Hui Ji, Chaoqiang Liu, and Yuhong Xu Zuowei Shen. Robust video denoising using low rank matrix completion. In *CVPR*, 2010.

[57] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC*, 1995.

[58] I. Jolliffe. Principal component analysis. In *Springer-Verlag, New York*, 1986.

[59] N. Joshi and M. Cohen. Lucky imaging for multi-image denoising, sharpening, and haze removal. In *ICCP*, 2010.

[60] Neel Joshiy, C. Lawrence Zitnicky, Richard Szeliskiy, and David J. Kriegman. Image deblurring and denoising using color priors. In *CVPR*, 2009.

[61] S. Baker K. M. Cheung and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003.

[62] M. B. Kaaniche and F. Bremond. Bremond, gesture recognition by learning local motion signatures. In *CVPR*, 10.

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

[64] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[65] Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and Andrew Y Ng. Tiled convolutional neural networks. *Advances in Neural Information Processing Systems*, 2010.

[66] Iosif M. Levin, Victor V. Savchenko, and Vladimir Ju. Osadchy. Correction of an image distorted by a wavy water surface: laboratory experiment. In *Applied Optics*, 2008.

[67] D. Li, R. M. Mersereau, and S. Simske. Atmospheric turbulence-degraded image restoration using principal components analysis. In *IEEE Geoscience and Remote Sensing Letters*, 2007.

[68] Yunpeng Li, Sing Kang, Neel Joshi, Steve Seitz, and Daniel Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *CVPR*, 2010.

[69] C. Liangliang, M. Yadong, N. Apostol, S. F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, 2012.

[70] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank completion. In *UIUC Technical Report*, 2009.

[71] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. In *UIUC Technical Report*, 2009.

[72] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. UIUC Technical Report, 2009.

[73] C. Liu. Beyond pixels: exploring new representations and applications for motion analysis. In *Doctoral Thesis, MIT*, 2009.

[74] C. Liu and W. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *ECCV*, 2010.

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

[76] X. Liu and B. Huet. Automatic concept detector refinement for large-scale video semantic annotation. In *IEEE Fourth International Conference on semantic computing (ICSC)*, 2010.

[77] A. Loui, J. Luo, S. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa. Kodak's consumer video benchmark data set: concept definition and annotation. In *Multimedia Information Retrieval*, 2007.

[78] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.

[79] G. Carhat M. Aubailly, M. Vorontsov and M. Valley. Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach. In *SPIE*, 2009.

[80] E. Shechtman M. Irani M. Blank, L. Gorelick and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[81] Yi Ma, John Wright, and Allen Y. Yang. Sparse representation and low-rank representation in computer vision. *ECCV Short Course*, 2012.

[82] G.G. Magaril-Ilyaev and V.M. Tikhomirov. Convex analysis: Theory and applications. In *AMS, Providence, RI*, 2003.

[83] I. Matthews and S. Baker. Active appearance models revisited. In *IJCV*, 2004.

[84] J. Min and R. Kasturi. Activity recognition based on multiple motion trajectories. In *ICPR*, 2004.

[85] R. Bhotika A.G.A. Perera J. Schmiederer G. Doretto M.T. Chan, A. Hoogs. Joint recognition of complex events and track matching. In *CVPR*, 2006.

[86] J. Nocedal and S.J. Wright. Numerical optimization (2nd ed.). In *Springer-Verlag, New York*, 2006.

[87] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. In *PAMI*, 2000.

[88] Teresa Pace Omar Oreifej, Guang Shu and Mubarak Shah. A two-stage reconstruction approach for seeing through water. In *CVPR*, 2011.

[89] Xin Liu Omar Oreifej and Mubarak Shah. Simultaneous video stabilization and moving object detection in turbulence. In *PAMI*, 2012.

[90] Omar Oreifej, Guang Shu, Teresa Pace, and Mubarak Shah. A two-stage reconstruction approach for seeing through water. In *CVPR*, 2011.

[91] G. Cottrell P. Dollar, V. Rabaud and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE VS-PETS*, 2005.

[92] M. Hebert P. Matikainen and R. Sukthankar. Trajectons: action recognition through the motion analysis of tracked features. In *ICCV*, 2009.

[93] V. Parameswaran and R. Chellappa. View invariance for human action recognition. In *IJCV*, 2006.

[94] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *PAMI*, 2010.

[95] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. In *IJCV*, 2008.

[96] C. Pal R. Messing and H.Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009.

[97] Y. Ma R. Vidal and S. Sastry. Generalized principal component analysis. In *PAMI*, 2005.

[98] M. Roggemann and B. Welsh. Imaging through turbulence. In *CRC Press*, 1996.

[99] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. In *Medical Imaging*, 1999.

[100] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Mller. Kernel principal component analysis. In *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*, pages 327–352. MIT Press, 1999.

[101] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[102] Omar Oreifej Shandong Wu and Mubarak Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV*, 2011.

[103] Y. Sheikh, O. Javed, , and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, 2009.

[104] Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. In *PAMI*, 2005.

[105] Xiaohui Shen and Ying Wu. A unified approach to salient object detection via low rank matrix recovery. *CVPR*, 2012.

[106] Masao Shimizu, Shin Yoshimura, Masayuki Tanaka, and Masatoshi Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *CVPR*, 2008.

[107] Palaiahnakote Shivakumara, Trung Quy Phan, and Chew Lim Tan. A gradient difference based technique for video text detection. In *ICDAR*, 2009.

[108] B. Siddiquie, R. Feris, and L. Davis. Image ranking and retrieval based on multiattribute queries. In *CVPR*, 2011.

[109] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[110] T. Finley T. Joachims and C.-N. Cutting-plane training of structural svms. *Machine Learning Journal*, 2009.

[111] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O.Javed, A. Divakaran, H. Cheng, and H. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.

[112] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. In *SIAM Journal on Optimization*, 2011.

[113] Y.D. Tian and S.G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *ICCV*, 2009.

[114] Yuandong Tian and Srinivasa Narasimhan. A globally optimal data-driven approach for image distortion estimation. In *CVPR*, 2010.

[115] Bernard Ghanem Tianzhu Zhang and Narendra Ahuja. Low-rank sparse learning for robust visual tracking. *ECCV*, 2012.

[116] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. In *Neural Computation*, 1999.

[117] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[118] C. Tomasi and J. Shi. Good features to track. In *CVPR*, 1994.

[119] Jan C van Gemert, Cor J Veenman, Arnold WM Smeulders, and J-M Geusebroek. Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010.

[120] H. Wang, A. Klaser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

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

[122] Z Wen, D. Fraser, A. Lambert, and H. Li. Reconstruction of underwater image by bispectrum. In *ICIP*, 2007.

[123] S. Wu and Y.F. Li. Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition. In *Pattern Recognition*, 2009.

[124] S. Wu, B. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, 2010.

[125] Shandong Wu, Brian Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, 2010.

[126] Shandong Wu, Omar Oreifej, and Mubarak Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV*, 2011.

[127] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010.

[128] R. Sukthankar Y. Ke and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

[129] Yang Yang and Mubarak Shah. Complex events detection using data-driven concepts. *ECCV*, 2012.

[130] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang. Robust late fusion with multi-task low rank minimization. *CVPR*, 2012.

[131] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. In *ACM Computing Surveys (CSUR)*, 2006.

[132] Y. Yitzhaky, I. Dror, and N. Kopeika. Restoration of atmospherically blurred images according to weather predicted atmospheric modulation transfer function (mtf). In *Optical Engineering*, 1997.

[133] Y. Yu and D. Schuurmans. Rank/norm regularization with closed-form solutions: Application to subspace clustering. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

[134] Y. Zhang. An alternating direction algorithm for nonnegative matrix factorization. In *preprint*, 2010.

[135] Xiang Zhu and Peyman Milanfar. Image reconstruction from videos distorted by atmospheric turbulence. In *SPIE*, 2010.

[136] Xiang Zhu and Peyman Milanfar. Stabilizing and deblurring atmospheric turbulence. In *ICCP*, 2011.

[137] Z.Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *ICPR*, 2004.