Robust Low-Rank and Sparse Decomposition for Moving Object Detection: from Matrices to Tensors

Andrews Sobral

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE S2IM

Laboratoire Informatique, Image et Interaction (L3i)
Laboratoire Mathématiques, Image et Applications (MIA)

THÈSE présentée par :

Andrews CORDOLINO SOBRAL

soutenue le : **11/05/2017**
pour obtenir le grade de : **Docteur de l'Université de La Rochelle**
Discipline : **Informatique et Applications**

Détection d'objets mobiles dans des vidéos par décomposition en rang faible et parcimonieuse: de matrices à tenseurs

---

JURY :

Laure TOUGNE	Professeure, Univ. de Lyon (France), Présidente du jury.
Lucia MADDALENA	Chercheuse, Conseil National de la Recherche, Naples (Italie), Rapporteur.
Alfredo PETROSINO	Professeur titulaire, Univ. de Naples Parthenope, Naples (Italie), Rapporteur.
Francois BRÉMOND	Directeur de recherche, INRIA Sophia-Antipolis (France), Examinateur.
Jordi GONZÀLEZ	Professeur associé, Univ. Autonome de Barcelone (Espagne), Examinateur.
Thierry BOUWMANS	Maître de conférences (HDR), Univ. de La Rochelle (France), Co-directeur de thèse.
El-hadi ZAHZAH	Maître de conférences (HDR), Univ. de La Rochelle (France), Directeur de thèse.

Robust low-rank and sparse decomposition for moving object detection: from matrices to tensors

Thesis submitted by **Andrews Cordolino Sobral** at Université de La Rochelle to fulfill the degree of **Doctor of Philosophy**.

La Rochelle

Director	**Dr. El-hadi Zahzah** Laboratoire Informatique, Image et Interaction Université de La Rochelle (France)
Co-Director	**Dr. Thierry Bouwmans** Laboratoire Mathématiques, Image et Applications Université de La Rochelle (France)
Thesis committee	**Pr. Laure Tougne** Laboratoire d'Informatique en Image et Systèmes d'Information Université Lumière Lyon 2 (France) **Dr. Francois Brémond** INRIA Sophia-Antipolis (France) **Dr. Jordi Gonzàlez** Centre de Visió per Computador Universitat Autònoma de Barcelona (Spain)
European evaluators	**Dr. Lucia Maddalena** National Research Council Naples (Italy) **Pr. Alfredo Petrosino** Computer Vision & Pattern Recognition Laboratory University of Naples Parthenope (Italy)

Acknowledgement

First I would like to express my gratitude to my supervisors, Dr. Thierry Bouwmans and Dr. El-hadi Zahzah for their invaluable guidance and advices. I would also like to take this opportunity to thank the CAPES/Brésil for providing me a full Ph.D. scholarship. Thank you to the Computer Vision Centre (CVC) members for having welcomed me (Doctoral stage), especially Dr. Jordi Gonzàlez. Without the above supports, this thesis is impossible.

I would like to thank Dr. Lucia Maddalena from ICAR-CNR (Italy) and Prof. Alfredo Petrosino from Univ. of Naples Parthenope (Italy) for their acceptances to be the reviewers of the this European thesis manuscript and for sharing interesting comments and criticism that helped improve this manuscript.

Sincere thanks to my dissertation examiners, Prof. Laure Tougne from Univ. of Lyon (France), Dr. Francois Brémond from INRIA Sophia-Antipolis (France) and Mehran Yazdi from Univ. of Shiraz (Iran), for their interest in my work and for making their time available for me.

I am very grateful to my family for their unconditional love and support without which this journey would not have been possible. My gratefulness is also my beloved fiancee Caroline Pacheco for her inspiring support and for being my best companion in this challenging journey.

Last but not least, I would also like to thank my friends and colleagues from the University of La Rochelle, especially those from the L3I and MIA labs. Their support has been invaluable throughout my Ph.D. study, making my time both memorable and enjoyable.

Abstract

The detection of moving objects is an important step in computer vision field to develop numerous kinds of systems, such as intelligent video surveillance, motion capture, among the others. These systems are used in a wide range of applications, including retail, home automation, safety and security. The most commonly used equipment are stationary cameras or pan-tilt-zoom (PTZ) cameras to monitor activities in outdoor or indoor environments. Since the cameras are stationary (or almost stationary), the detection of moving objects can be achieved by building a representation of the scene background and comparing each new frame with this one. This process is called *background subtraction*, also named background/-foreground separation.

More recently, the research on decomposition into low-rank plus sparse matrices (or tensors) has been showing to be a suitable framework to deal with the background/foreground separation problem. This framework consider that the data to be processed satisfy two important assumptions: a) the inliers (latent structure) are drawn from a single or a union of low-dimensional subspaces, and b) the corruptions are sparse. This assumption holds a particular association to the problem of background/foreground separation, where the background model (almost static) is represented as a low-rank structure and the foreground objects are associated with the sparse residuals. However, the key issues and challenges in such approaches are their capabilities to handle complex and dynamic background scenarios, as well as performing in a real-time manner.

Given the importance of this subject, this thesis introduces the recent advances on decomposition into low-rank plus sparse matrices and tensors, as well as the main contributions to face the principal issues in moving object detection. First, we present an overview of the state-of-the-art methods for low-rank and sparse decomposition, as well as their application to background modeling and foreground segmentation tasks. Next, we address the problem of background model initialization as a reconstruction process from missing/corrupted data. A novel methodology is presented showing an attractive potential for background modeling initialization in video surveillance. Subsequently, we propose a double-constrained version of robust principal component analysis to improve the foreground detection in maritime environments for automated video-surveillance applications. The algorithm makes use of double constraints extracted from spatial saliency maps to enhance object foreground detection in dynamic scenes. We also developed two incremental tensor-based algorithms in order to perform background/foreground separation from multidimensional streaming data. These works address the problem of low-rank and sparse decomposition on tensors. Finally, we present a particular work realized in conjunction with the Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB).

Résumé

La détection d'objets mobiles dans des vidéos acquises à partir de caméras est une étape importante dans plusieurs systèmes automatiques de vision par ordinateur. Ces systèmes sont utilisés dans une large gamme d'applications, comprenant entre autres la Vidéo-Surveillance Intelligente (VSI) et la capture de mouvement optique. Les équipements les plus utilisés sont des caméras fixes ou pan-tilt-zoom (PTZ) pour surveiller les activités dans les environnements extérieurs ou intérieurs. Étant donné que les caméras sont fixes (ou presque stationnaires), la détection d'objets mobiles peut être obtenue en construisant une représentation du modèle du fond et comparant chaque nouvelle image avec ce modèle. Ce processus est appelé « soustraction de fond » ou séparation fond et objets du premier plan.

Les récentes avancées en ACP robuste par décomposition en matrices (ou tenseurs) de rang faible et parcimonieuse montrent un grand potentiel pour séparer les objets en mouvement dans des vidéos. Cette formulation de problème considère que les données à traiter ont deux hypothèses importantes: a) les « inliers » (structure latente) proviennent d'un seul sous-espace (ou d'une union de sous-espaces) de dimension faible, et b) les données corrompues « outliers » sont de caractères parcimonieuses. Cette hypothèse est particulièrement adaptée au problème de la séparation entre le fond et les objets du premier plan, où le modèle du fond (presque statique) est représenté par une structure de rang faible et les objets de premier plan sont associés aux résidus parcimonieux. Cependant, les principaux enjeux et défis dans telles approches sont ses capacités à gérer des scénarios de fond complexes et dynamiques ainsi que l'exécution incrémentale et en temps-réel.

Dans ce contexte, cette thèse introduit les avancées récentes sur la décomposition en matrices (et tenseurs) de rang faible et parcimonieuse ainsi que les contributions pour faire fâce aux principaux problèmes dans ce domaine. Nous présentons d'abord un aperçu des méthodes matricielles et tensorielles les plus récentes ainsi que ses applications sur la soustraction de fond. Ensuite, nous abordons le problème de l'initialisation du modèle de fond comme un processus de reconstruction à partir de données manquantes ou corrompues. Une nouvelle méthodologie est présentée montrant un potentiel intéressant pour l'initialisation de la modélisation du fond dans le cadre de VSI. Par la suite, nous proposons une version « double contrainte » de l'ACP robuste pour améliorer la détection de premier plan en milieu marin dans des applications de vidéo-surveillance automatisés. Nous avons aussi développé deux algorithmes incrémentaux basés sur tenseurs afin d'effectuer une séparation entre le fond et le premier plan à partir de données multidimensionnelles. Ces deux travaux abordent le problème de la décomposition de rang faible et parcimonieuse sur des tenseurs. A la fin, nous présentons un travail particulier réalisé en conjonction avec le Centre de Vision Informatique (CVC) de l'Université Autonome de Barcelone (UAB).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we provide the thesis context concerning the application of low-rank and sparse decomposition to the problem of moving object detection in videos.

1.1 Presentation

The detection of moving objects is an important step in computer vision to develop numerous kinds of systems, such as intelligent video surveillance and motion capture, among the others [26, 57, 173]. These systems are used in a wide range of applications, including retail, home automation, safety and security [1]. For example, in visual surveillance systems, the detection of moving objects can be important to identify useful insights from video data, such as intrusion/anomaly detection, abandoned objects, traffic data collection, etc. These insights are usually extracted after a sequence of video processing steps that are part of a more general module named Video Content Analysis (VCA), also known as Intelligent Video Analytics. Figure 1.1 summarizes the approach described here, where a VCA module is used to auto-

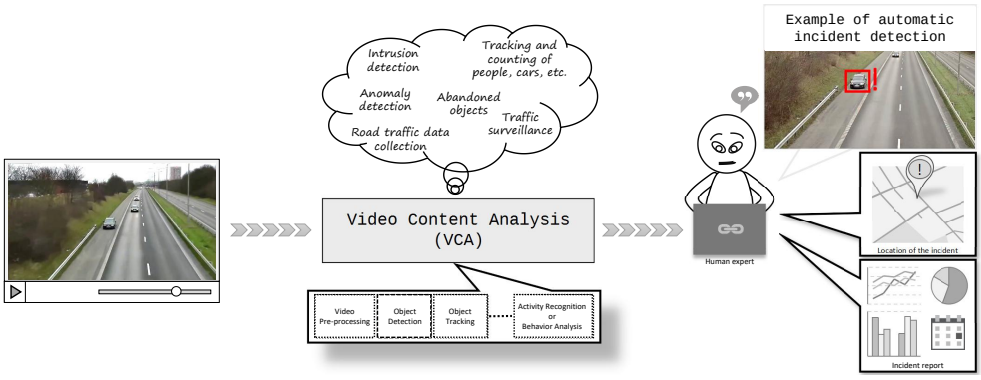


Figure 1.1: Illustration of an intelligent video surveillance system.

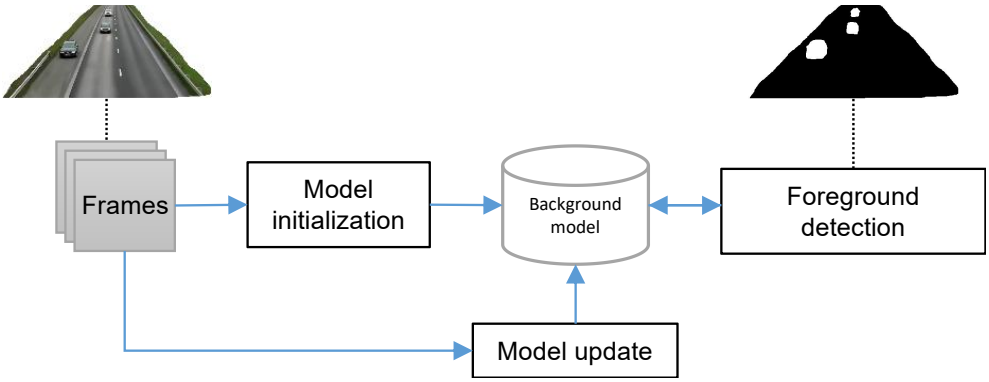


Figure 1.2: Block diagram of the background subtraction process.

matically report road traffic incidents. In many domains, VCA is implemented on CCTV systems, where the most commonly used equipments are stationary cameras or pan-tilt-zoom (PTZ) cameras to monitor activities in outdoor or indoor environments. Since the cameras are stationary (or almost stationary), the detection of moving objects can be achieved by building a representation of the scene background and comparing each new frame with this one. This process is called *background subtraction* (BS), also named background/foreground (B/F) separation, and the scene representation is called the *background model* (BM) [26, 57]. This basic operation works as a two-class classifier and it consists in separating the moving objects called "foreground" (FG), from the static (or quasi static) information, called "background" (BG). Typically the BS process includes three main steps: a) background model initialization, b) background model maintenance, and c) foreground detection (see block diagram in Figure 1.2). These steps work as follows:

- **Model initialization** - In general, this step consists in creating a BM that best represents the scene background. It is often assumed that initialization can be achieved by exploiting some "clean" frames (free of foreground objects) at the beginning of the sequence, and the scene here is assumed to be stationary or quasi stationary. However, this assumption is rarely encountered in indoor or outdoor scenarios, because several challenges appear and perturb this process, such as noise acquisition, dynamic factors, etc. [25, 135].

- **Model maintenance** - In real-life scenarios, there are changes that occur over time. These changes can be local, such as a moving object entering (or leaving) the scene, or global, such as day-light inference [26, 57]. It is important for any BM to adapt to these changes. The model maintenance step aims to preserve and maintain the BM learned in the initialization step to be as close as possible to the real scene background.

- **Foreground detection** - Given the representation of the scene background, the foreground detection step consists in comparing the learned background model with the input frame. This process depends on the type of changes expected in the scene background. These changes could be related to a specific object of interest or any other

factor, such as noise, illumination changes, among the others [26, 57]. The main challenge of this step is to minimize the number of false positive and false negative pixels.

The BS process must deal with a large number of challenges that may occur during its application for moving object detection, as described below:

- **Camera jitter:** In general, the camera jitter occurs when a fixed camera is affected by natural or environmental events, such as strong winds or earthquake. In such cases, the fixed camera presents a nominal motion that is usually indistinguishable from the motion of the foreground objects, leading to undesirable detection results.
- **Camera automatic adjustments:** Today, most of digital cameras have automatic adjustments, such as automatic exposure mode. This feature automatically determines the correct exposure for pictures. In automatic mode, the cameras make some decisions without any user input, including the aperture setting, the shutter speed and white balance. These settings may make difficult the task of segmentation.
- **Pan-Tilt-Zoom (PTZ) cameras:** Most of background subtraction research is focused on stationary cameras. However, the adoption of PTZ cameras for intelligent video surveillance became more frequent because of their ability to cover a wide field of view. These cameras are capable of (automatically or manually) remote directional and zoom control. In general, most of background subtraction algorithms fail in the case of moving cameras, due to the non stationarity of the background.
- **Video noise:** In general, a video signal can be contaminated by noise in the recording process. The noise is usually a random pattern that is caused by signal transmission/acquisition, coding, and between the processing steps. Usually, this phenomenon can produce undesirable effects or artifacts affecting the background scenes.
- **Intermittent object motion:** In some cases, moving objects stop for a long period of time or a background object starts moving. In such situations, the intermittent objects can produce "ghosting" artifacts in the background model. Typical examples include parking vehicles and abandoned objects. Dealing with these situations depends on the context. For some applications, motionless foreground objects must be incorporated into the background model and others not.
- **Dynamic backgrounds:** Dynamic factors of the environment are one of the main causes of dynamic backgrounds that are generally the outcome of an external event or a chain of events, such as flowing water and moving leaves caused by winds. In such environment, modeling a good representation of the background is a challenging task for a background subtraction algorithm, due to the separation of the dynamics of the foreground objects in comparison to the natural dynamics of the scene background.
- **Shadows:** Normally shadows are generated as result of a light source blocked by an opaque object. Shadows can be seen as a dark area that either may be attached or not to detected objects, causing objects merging and objects shape distortion. The presence of shadows usually does not allow a robust shape detection of moving objects. In general, the shadow areas are often misclassified as foreground objects, causing errors in the segmentation of moving objects.

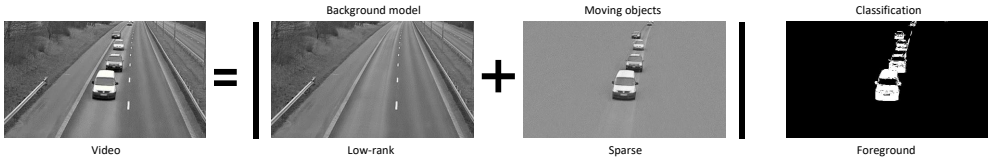


Figure 1.3: Background/foreground separation based on low-rank plus sparse decomposition.

- **Illumination changes:** Illumination changes often occur over time in outdoor (i.e. daylight) and indoor (i.e. light switch) scenes. In outdoor environments, the gradual changes in the appearance of the scene may result from Earth's rotation changing patterns of illumination of Earth's surface. Otherwise, in indoor scenes the occurrence of a sudden illumination change is a typical factor due to light switch. It is important for a background subtraction algorithm to adapt to these kind of changes by building a light invariant model of the background scene.
- **Bootstrapping:** It is often assumed that a representative model of the background can be produced by exploiting some clean frames (without moving objects) at the beginning of the sequence. However, this assumption is rarely encountered in real-life scenarios, because of continuous clutter presence. In such situations, a robust initialization process of the background model must be adopted, learning the correct background model over time.
- **Camouflage:** Moving objects can be visually similar to the background scene, or some portion of it. This effect is called camouflage, leading to erroneous distinction between foreground and background. The camouflage can be at color level, texture level, or any other appearance/depth feature level.
- **Night scenes:** Night videos are still a challenging task. Indeed, the low contrast between foreground and background causes many false detections due to the dramatic illumination change and low signal to noise ratio (SNR).

Many background subtraction methods facing these issues have been designed over the last decade [24, 183, 234], and they generally share the same scheme presented previously in the Figure 1.2. Conventional BS methods exploit the temporal (or spatio-temporal) variation of each pixel (or region) under many mathematical models, including probabilistic/statistical models, fuzzy models, neural/neuro-fuzzy models, subspace learning models, among the others [24, 183, 234].

More recently, the research on decomposition into low-rank plus sparse matrices (or tensors[1]) has been showing to be a suitable framework to deal with the background/foreground separation problem [27, 28]. This framework consider that the data to be processed satisfy two important assumptions: a) the inliers (latent structure) are drawn from a single (or a union of) low-dimensional subspace(s), and b) the corruptions are sparse. This assumption holds a particular association to the problem of B/F separation, where the background model

[1]The reader can also refer to Appendix C for an introduction on tensors

(almost static) is represented as a low-rank structure and the foreground objects are associated with the sparse residuals. Figure 1.3 illustrates the process described here. In general, the input video is converted into a matrix (or tensor) representation and then decomposed into a sum of low-rank and sparse components. The choice of the number of components is a free parameter and it varies according to the type of behavior needed to be modeled. Sometimes a third component that models the Gaussian noise is used to enhance the noise suppression, improving the foreground detection. However, the key issues and challenges in such approaches are their capabilities to handle complex and dynamic background scenarios, as well as performing in a real-time manner.

Given the importance of this subject, the thesis introduces recent advances in decomposition into low-rank plus sparse matrices and tensors, as well as the main contributions to face the principal issues in this domain. In the next sections, we present a list of the main contributions developed in the thesis and the outline of each chapter.

1.2 Contributions

In order to fit the above objectives, we have accomplished the following contributions summarized in this thesis[2]:

- **A new library, named LRSLibrary**[3]: that provides a collection of low-rank and sparse decomposition algorithms. The library was designed for background/foreground separation in videos and it contains a total of 104[4] *matrix*-based and *tensor*-based algorithms [180]. It has been fundamental for all the experiments conducted in the thesis.
- **A novel methodology for background model initialization**: that considers the background model initialization as a reconstruction problem from missing/corrupted data. Given a sequence of images, a simple joint motion-detection and frame-selection operation removes the redundant frames and induces missing entries from the moving regions. Next, the background model is recovered by matrix/tensor completion under partially observed data [178, 184].
- **A double-constrained Robust Principal Component Analysis (RPCA) method, named SCM-RPCA**: that takes the advantage of shape and confidence maps, both extracted from spatial saliency maps, to enhance object foreground detection in dynamic scenes [179].
- **An incremental tensor subspace learning (IMTSL) algorithm**: that handles the problem of background/foreground separation in streaming multidimensional data for intelligent video surveillance applications. Differently from the traditional tensor-based methods for background/foreground separation that only use the gray-scale or

[2]We suggest the reader to see the list of publications related to the thesis in the Appendix E.
[3]Please refer to the Appendix D for a complete description of the library.
[4]Up-to-date information on November 8, 2017.

color information, the proposed method constructs a multi-feature low-rank model for robust modeling of the scene background [176].

- **An online stochastic tensor decomposition (OSTD)**: that is more robust and faster than IMTSL algorithm for handling streaming multispectral video sequences. The OSTD algorithm makes use of RPCA on tensors for robust background/foreground separation. The proposed method was designed to be much faster than IMTSL and to address the major difficulties of multispectral imaging for video surveillance [182].
- **A survey of low-rank and sparse representation**: that covers the main aspects of the recent approaches for low-rank and sparse representation [27].
- **An evaluation of subspace clustering algorithms to the problem of human action recognition from 3D skeletal data**: that explores a particular approach for low-rank and sparse representation, named *subspace clustering* (SC). Instead of applying SC for background modeling and foreground separation as shown previously, here we evaluate the robustness of some subspace clustering algorithms to the problem of human action recognition from 3D skeletal data. This is a work realized in conjunction with CVC at UAB [73, 181].

1.3 Outline

The rest of the thesis is organized as follows:

- Chapter 2 provides an overview of the state-of-the-art methods for low-rank and sparse decomposition on matrices and tensors, as well as their application to the problem of background modeling and foreground segmentation. The methods were unified in a more general framework, named DLSM, that categorizes the matrix separation problem into three main approaches: implicit, explicit and stable.
- Chapter 3 presents a novel methodology for background model initialization, seen as a reconstruction problem from missing/corrupted data. This chapter is closely related to the first part of the DLSM framework introduced in Chapter 2, covering a wide range of methods for low-rank approximation on matrices and tensors.
- Chapter 4 describes a new double-constrained RPCA, named SCM-RPCA, to improve the object foreground detection in maritime scenes. This algorithm follows the third approach of the DLSM framework by adopting a stable decomposition. The algorithm makes use of double constraints extracted from spatial saliency maps to enhance object foreground detection in dynamic scenes.
- Chapters 5 and 6 present two incremental tensors-based algorithms in order to perform background/foreground separation from multidimensional streaming data. These chapters address the problem of low-rank and sparse decomposition on tensors. Chapter 5 introduces a new incremental method for higher-order decomposition on tensors, whereas Chapter 6 presents a new online stochastic algorithm that makes use of robust principal component analysis on tensors.

- Chapter 7 presents a particular approach of low-rank and sparse representation, named *subspace clustering*, for human action recognition from 3D skeletal data. This chapter address a particular work realized in conjunction with CVC at UAB.
- Chapter 8 summarizes the conclusions of the thesis, showing the advantages and limitations of the proposed approaches. It also discusses the open issues and future perspectives of the thesis.
- Appendices A and B provide a homogenized overview of all different mathematical notations, symbols and abbreviations found over all chapters in the thesis.
- Appendix C introduces the concept of tensors, as well as their basic operations.
- Appendix D presents the LRSLibrary, showing a brief overview of available algorithms and usage example.
- Appendix E presents a list of publications related to this thesis.

Chapter 2

Recent approaches via low-rank and sparse representation

This chapter introduces the principles of low-rank and sparse decomposition for the problem of B/F separation. Here, we present a concise overview based on our recently published survey (Computer Science Review, 2016, [27]) to cover the main aspects of the recent approaches of low-rank and sparse representation. In addition, an extension to tensors was also considered.

2.1 Introduction

Learning low-rank and sparse structures from corrupted or even incomplete observations has recently attracted wide attention in intelligent video surveillance to develop robust algorithms for background modeling and foreground segmentation [27]. As stated in Chapter 1, the main objective of these algorithms is to highlight the foreground (or moving) objects for further steps, such as detection, tracking and recognition. However, in this domain the observed data (images or videos) are rarely pure and often have high dimensionality.

A large number of approaches for robust low-rank and sparse modeling have been proposed in the last few years [27, 49, 117, 259]. These approaches are based on the assumption that the uncorrupted information lies in a low-dimensional subspace, whereas noise is sparse. This assumption holds a particular association to the problem of B/F separation, where the background model (almost static) is represented as a low-rank structure and the foreground objects are associated with the sparse residuals. However, the key issues and challenges in such approaches are their capabilities to handle complex and dynamic background scenarios, as well as performing in a real-time manner. Given the importance of this subject, several methods have been developed in order to perform B/F separation in a robust way [27].

In the next sections, we present an overview of the state-of-the-art algorithms for low-rank and sparse decomposition, as well as their application to background modeling and foreground segmentation tasks. First we start with recovering low-rank and sparse structures

on matrices in Section 2.2, then we present a more general case of RPCA, named subspace clustering, in Section 2.3. Finally, in Section 2.4, we show how to deal with the multidimensional case through tensor methods. The reader may refer to Appendix A for a complete description of the mathematical notations and symbols found in the current and next chapters of the thesis.

2.2 Decomposition into low-rank plus additive matrices

Let a sequence of n gray-scale images (or frames) $\mathbf{F}_1 \dots \mathbf{F}_n$ captured from a static camera, that is, $\mathbf{F} \in \mathbb{R}^{i_1 \times i_2}$ where i_1 and i_2 denote the frame resolution (rows by columns, a.k.a image height by image width), and considering that all frames are vectorized[1] into an observation matrix $\mathbf{A} = [vec(\mathbf{F}_1) \dots vec(\mathbf{F}_n)]$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $m = (i_1 \times i_2)$. The process of background/foreground separation can be regarded as a matrix separation problem. The background (almost static and highly correlated between frames) is assumed to lie in a low-dimensional subspace, where the sparse outliers usually represent the foreground (or moving) objects. We assume that this matrix separation problem can be unified in a more general framework formulated as follows:

$$\mathbf{A} = \sum_{y=1}^{Y} \mathbf{K}_y \tag{2.1}$$

where, in most of the cases, $Y \in \{1, 2, 3\}$, and for $Y = 1 \dots 3$, the matrices $\mathbf{K}_1 \dots \mathbf{K}_3$ are commonly defined as follows:

- **Implicit**: For $Y = 1$, the first matrix $\mathbf{K}_1$ is a low-rank matrix (e.g. $\mathbf{K}_1 = \mathbf{L}$). The matrix $\mathbf{L}$ is assumed to be the best **low-rank approximation** of the matrix $\mathbf{A}$, where $\mathbf{A} \approx \mathbf{L}$. The low-rank assumption for $\mathbf{A}$ comes from the fact that the uncorrupted data appear to be highly correlated to a certain degree. This means that we try to recover only the background component (almost stationary) from a sequence of vectorized images in the matrix $\mathbf{A}$. We call this decomposition as "implicit decomposition" due to the fact that we have any constraint with respect to the sparse components (or foreground objects). The sparse matrix $\mathbf{S}$ is recovered by performing the difference between the input matrix $\mathbf{A}$ and its low-rank component $\mathbf{L}$ (e.g. $\mathbf{S} = \mathbf{A} - \mathbf{L}$). Some methods are included in this category, such as Low-Rank Approximation (LRA), Non-negative Matrix Factorization (NMF), and Matrix Completion (MC).

  An alternative approach is to assume that the first matrix $\mathbf{K}_1 = \mathbf{S}$ is the best **sparse approximation** of the matrix $\mathbf{A}$ (also known as *sparse coding*), where $\mathbf{A} \approx \mathbf{S}$. In this case, we ignore the low-rank structure and we find only the sparse components that minimize the reconstruction error. This approach is widely used for sparse dictionary learning [138] and compressed sensing [158]. Some authors [232] considered the background model can be sparsely represented as a linear combination of a few atoms in the learned dictionary. However, in the context of this thesis we consider that our first matrix $\mathbf{K}_1$ (implicit decomposition) is recovered from a low-rank perspective.

[1] This operation consists of stacking vertically all columns of the frame $\mathbf{F}$.

The main drawback of the methods based on $Y = 1$ is that there is only one assumption about the structure of the approximated matrix $\mathbf{K}_1$ (either it is low-rank or sparse). In the case where $\mathbf{K}_1 = \mathbf{L}$, the foreground objects in the matrix $\mathbf{S} = \mathbf{A} - \mathbf{L}$ are mixed with dense or sparse noise, or anything else. Otherwise, if we consider that $\mathbf{K}_1 = \mathbf{S}$, any assumption about the structure of the background exists. For this reason, some authors proposed to "explicitly" specify a sparse component, resulting in $Y = 2$.

- **Explicit**: For $Y = 2$, the matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are usually assumed to be the low-rank and sparse representation of the data, respectively, such that $\mathbf{K}_1 = \mathbf{L}$ and $\mathbf{K}_2 = \mathbf{S}$. In this case, the input matrix $\mathbf{A}$ is decomposed in such way that $\mathbf{A} \approx \mathbf{L} + \mathbf{S}$. We call this decomposition as "explicit decomposition" due to the fact that we have two constraints: the first one enforcing a low-rank structure over the matrix $\mathbf{L}$, and the second one enforcing a sparse structure over the matrix $\mathbf{S}$. Usually we call the methods based on $Y = 2$ as robust methods, such as Robust Principal Component Analysis (RPCA), Robust Non-Negative Matrix Factorization (RNMF), Robust Dictionary Learning (RDL), among the others [27].

  Methods based on $Y = 2$ usually work better for the problem of background/foreground separation in comparison to the methods based on $Y = 1$. However, in real life surveillance videos the background is never completely stationary, and there is always measurement noise or corruptions. In order to deal with this, some authors [260] proposed to "explicitly" add a new component representing the noise term, resulting in $Y = 3$.

- **Stable**: For $Y = 3$, the matrices $\mathbf{K}_1$, $\mathbf{K}_2$ and $\mathbf{K}_3$ are usually assumed to be the low-rank, sparse and noise components, respectively, resulting in $\mathbf{K}_1 = \mathbf{L}$, $\mathbf{K}_2 = \mathbf{S}$ and $\mathbf{K}_3 = \mathbf{E}$, where $\mathbf{A} \approx \mathbf{L} + \mathbf{S} + \mathbf{E}$. The noise can be modeled by a Gaussian, a Mixture of Gaussians (MoG) or a Laplacian distribution [140]. This decomposition is called "stable decomposition" as it separates the outliers in $\mathbf{S}$ and the noise in $\mathbf{E}$. In the case of background/foreground separation, the noise matrix $\mathbf{E}$ can also represent some dynamic properties of the background, as well as it can capture the turbulence in thermal videos [152].

  Several methods based on $Y = 3$ were developed [28], and they are usually based on Stable Robust Principal Component Analysis (Stable RPCA) or Stable Principal Component Pursuit (Stable PCP) [260] and Three Term Decomposition (TTD) [74, 152]. In Chapter 4, we investigate the problem of moving object detection in maritime environment through a stable decomposition framework for separating the mixed dynamic behavior of the background (e.g. moving water, waves, etc) from the motion of the objects of interest (e.g. ships or boats).

From this homogenized overview, we call the above framework as Decomposition into Low-rank and Sparse Matrices (DLSM). In the next sections we introduce each part of this framework where the state-of-the-art methods based on $Y = 1 \ldots 3$ are presented in the Sections 2.2.1 (implicit approaches), 2.2.2 (explicit approaches) and 2.2.3 (stable approaches).

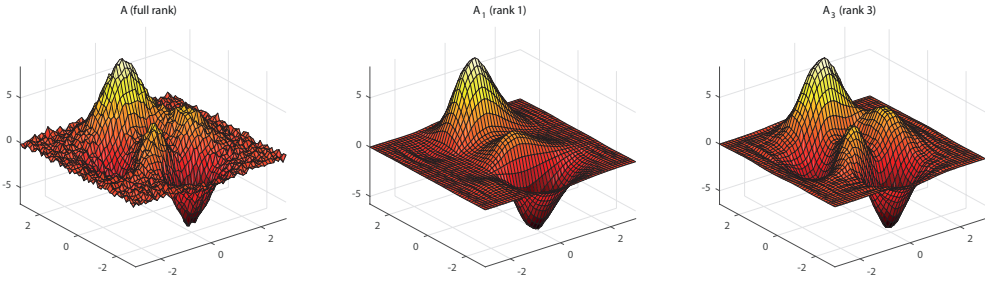


Figure 2.1: Example of a low-rank approximation from an input matrix contaminated by Gaussian noise. From left to right: the input matrix $\mathbf{A}$, its rank-1 approximation and its rank-3 approximation.

2.2.1 Implicit decomposition

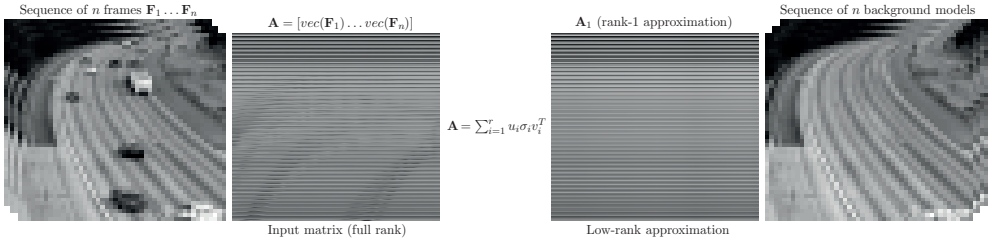
The implicit decomposition of the DLSM framework can be seen as a low-rank matrix recovery problem, where the uncorrupted data can be recovered from a low-dimensional representation of the input matrix. The low-rank approximation is formulated as a minimization problem, in which the cost function measures the fit between the input matrix $\mathbf{A}$ and an approximating matrix $\mathbf{L}$ (the optimization variable), subject to a constraint that the approximating matrix has reduced rank. This optimization, also known as *rank minimization* under *hard*-rank constraint, is defined as follows:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & f(\mathbf{A}-\mathbf{L}), \\ \text{subject to} \quad & rank(\mathbf{L}) = r, \end{aligned} \tag{2.2}$$

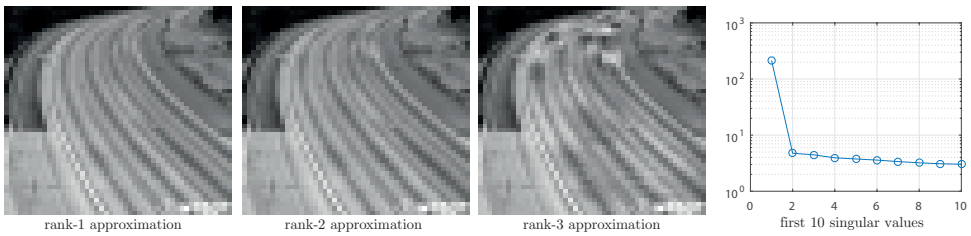
where $f(.)$ denotes a loss function and r $(1 \leq r < rank(\mathbf{A}))$ represents the desired rank. The minimum error can be given by the Frobenius norm or the ℓ_2-norm, due to their invariance to rotation. Solving (2.2) can be interpreted as finding the best rank r estimation of $\mathbf{A}$ in a least-squares sense, where the loss function is defined as $f(\mathbf{A}-\mathbf{L}) = ||\mathbf{A}-\mathbf{L}||_F^2$. This means that (2.2) does not have a local minimum and also a closed form solution can be estimated by computing the Singular Value Decomposition (SVD) of $\mathbf{A}$. Formally, the SVD of an $m \times n$ real or complex matrix $\mathbf{A}$ is a factorization of the form:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}} \tag{2.3}$$

where $\mathbf{U}$ is an $m \times m$ real or complex unitary matrix, $\mathbf{\Sigma}$ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{V}^{\mathbf{T}}$ is an $n \times n$ real or complex unitary matrix. The m columns of $\mathbf{U}$ and the n columns of $\mathbf{V}$ are called the left-singular vectors and right-singular vectors of $\mathbf{A}$, respectively. The diagonal entries $\mathbf{\Sigma}$ are known as the singular values of $\mathbf{A}$ and they are ordered in decreasing order. However, instead of taking all singular values (full SVD), the low-rank approximation problem, according to Eckart and Young [56] theorem, considers the existence of an optimal rank r approximation, denoted by



(a) Example of background model estimation through low-rank approximation.



(b) The influence of the rank approximation in the background model.

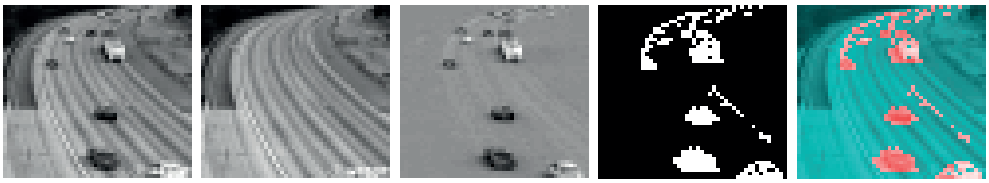
Figure 2.2: Application of low-rank approximation to the background model estimation in a sequence of images.

$svd_r(\mathbf{A})$, by truncating the SVD keeping the r largest singular values such that:

$$svd_r(\mathbf{A}) = \sum_{i=1}^{r} \mathbf{u}_i \boldsymbol{\sigma}_i \mathbf{v}_i^T \tag{2.4}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ denote the ith column of $\mathbf{U}$ and $\mathbf{V}$, respectively, and $\boldsymbol{\sigma}_i$ represents the diagonal entries of $\boldsymbol{\Sigma}$. Figure 2.1 shows an example of an input matrix $\mathbf{A}$ contaminated by a Gaussian noise and its rank-1 ($\mathbf{A}_1 = svd_1(\mathbf{A})$) and rank-3 ($\mathbf{A}_3 = svd_3(\mathbf{A})$) approximation, respectively. As it can be seen, the low-rank approximation can eliminate the noise component enough. However, some partial information in the rank-1 approximation is lost compared to the rank-3 approximation. For example, there are only 3 peaks in $\mathbf{A}_1$ instead of 4 peaks in $\mathbf{A}_3$, that is, $\mathbf{A}_3$ is closer to the original matrix (without noise) than $\mathbf{A}_1$.

Concerning the problem of background/foreground separation, the low-rank approximation can be used for the background model initialization task. As an example, Figure 2.2 (a) shows how to estimate the background model through low-rank approximation. It can be observed that the rank-1 approximation can recover, successfully, a good representation of the background model. Figure 2.2 (b) presents the influence of the rank approximation in the background model. It can be seen that the more the rank is increased the more artifacts are included into the background model. Taking into account the first 10 singular values, the high magnitude of the first singular value explains the high correlation between video frames and why the rank-1 approximation can give a good approximation of the background model. The best rank r approximation for background modeling is not always evident to find, and it depends on the scene.



(a) Input frame (b) Background model (low-rank aprox.) (c) Moving objects (sparse components) (d) Foreground mask (e) Segmentation

Figure 2.3: Example of moving vehicles segmentation after background model estimation from low-rank approximation.

The next step for estimating the foreground components is to find the sparse matrix $\mathbf{S}$, which can be recovered by performing the difference between the input matrix $\mathbf{A}$ and its low-rank component $\mathbf{L} = svd_r(\mathbf{A})$ for a given r, such that $\mathbf{S} = \mathbf{A} - \mathbf{L}$ (e.g. Figure 2.3 (c)). The foreground masks (e.g. Figure 2.3 (d)) are simply obtained by hard thresholding the sparse matrix $\mathbf{S}$ such that:

$$\begin{aligned} \mathbf{O} &= \mathbf{S}^2 < \sigma^2, \\ \sigma^2 &= var(\mathbf{s}), \end{aligned} \tag{2.5}$$

where $\mathbf{O}$ represents the outliers, $\mathbf{s} = vec(\mathbf{S})$ denotes the vectorization of the matrix $\mathbf{S}$ and $var(\mathbf{s})$ is the variance of the elements of the vector $\mathbf{s}$. Equation (2.5) is also known as variance threshold method, that removes all low-variance entries of $\mathbf{S}$. Finally the segmentation of the moving objects is obtained by coloring the elements of $\mathbf{O}$ (see Figure 2.3 (e)).

However, the low-rank approximation method presented previously is based on rank minimization under hard-rank constraint, and a closed form solution is obtained by SVD. Unfortunately, this approach has several limitations and drawbacks. It cannot handle affine transformations, missing entries, gross corruptions, etc. Consider the following example:

Affine transformation and missing entries: In many applications, we need to recover a minimal rank matrix subject to some problem-specific constraints, often characterized as an affine set. A typical situation is when the columns are i.i.d. samples of a random process with low-rank covariance [165], such as collaborative filtering [2] and latent semantic indexing [141]. This *affine rank minimization* problem is defined as follows:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & rank(\mathbf{L}), \\ \text{subject to} \quad & A(\mathbf{L}) = \mathbf{b}, \end{aligned} \tag{2.6}$$

where $A : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^p$ denotes a linear mapping and $\mathbf{b} \in \mathbb{R}^p$ represents a vector of observations of size p. The above minimization is equivalent to seeking the simplest model satisfying a given set of constraints. A special case of problem (2.6) is the matrix completion problem:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & rank(\mathbf{L}), \\ \text{subject to} \quad & P_\Omega(\mathbf{L}) = P_\Omega(\mathbf{A}), \end{aligned} \tag{2.7}$$

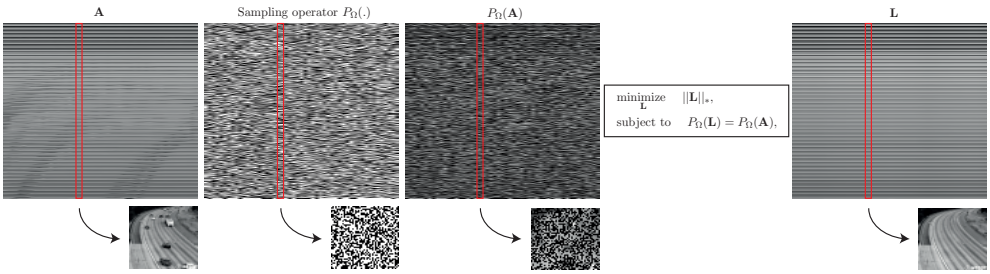


Figure 2.4: Example of low-rank matrix completion for background model estimation.

where $P_\Omega(.)$ denotes a sampling operator restricted to the elements of Ω (set of observed entries), i.e., $P_\Omega(\mathbf{A})$ has the same values as $\mathbf{A}$ for the entries in Ω and zero values for the entries outside Ω.

However, the problems (2.6) and (2.7) are NP-hard and all known finite time algorithms have at least doubly exponential running times in both theory and practice [40]. Candès and Recht [40] proposed to replace the $rank(.)$ function with the nuclear norm [63] (sum of singular values, see Appendix A) making the problem tractable, in such a way that the problem reduces to:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & ||\mathbf{L}||_*, \\ \text{subject to} \quad & P_\Omega(\mathbf{L}) = P_\Omega(\mathbf{A}). \end{aligned} \tag{2.8}$$

Given that the singular values are always positive, the nuclear norm can be regarded as an ℓ_1-norm of the singular values, while the $rank(.)$ function is the cardinality or ℓ_0-norm of the singular values. The advantages of using the nuclear norm relaxation are: a) the nuclear norm is convex[2], enabling to compute global optima efficiently, b) the nuclear norm is the tightest convex surrogate of the rank function [63], and c) due to its convexity, the minimization can be achieved tractably via several popular algorithms, such as semidefinite programming (SDP) [124], projected subgradient method [55], or low-rank parametrization [165]. Candès and Recht [40] theoretically proved that the solution of problem (2.8) can exactly recover the low-rank matrix with a high probability. However, in real applications, the input matrix can be contaminated by noise and the equality constraint in Equation (2.8) is too strict. For matrix completion with noise [39], a relaxed form of (2.8) it is often considered as follows:

$$\underset{\mathbf{L}}{\text{minimize}} \quad \frac{1}{2}||P_\Omega(\mathbf{L}) - P_\Omega(\mathbf{A})||_F^2 + \lambda||\mathbf{L}||_* \tag{2.9}$$

where λ is a trade-off parameter between the error and the low-rank regularization induced by the nuclear norm, and the selection of λ should depend on the noise level. This is an unconstrained convex optimization problem, and can be solved in a systematic way using a proximal algorithm [155, 200]. Figure 2.4 illustrates an example of low-rank matrix completion for background model estimation. In this example, the input matrix $\mathbf{A}$ is sampled from

[2]For instance, a (strictly) convex function on an open set has no more than one minimum.

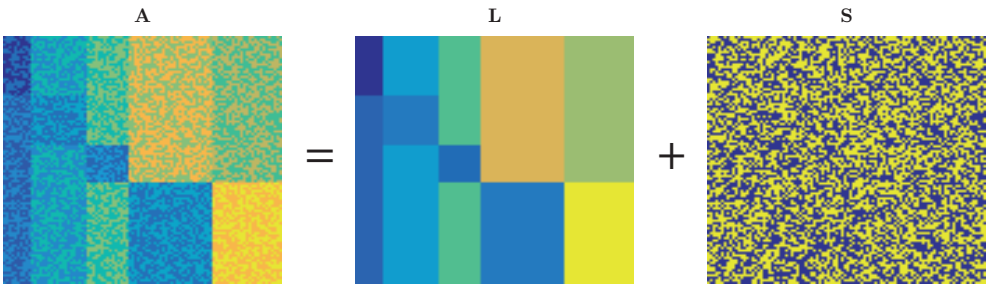


Figure 2.5: RPCA via decomposition in low-rank and sparse matrices.

a uniform distribution by $P_\Omega(.)$ where only 50% of its entries are revealed. As it can be seen, the low-rank matrix $\mathbf{L}$ was reconstructed successfully by using a Singular Value Thresholding (SVT) algorithm proposed by Cai et al. [37]. In Chapter 3 we develop the formulation of Matrix Completion for the recent approaches. In addition, we investigate the background model initialization as a reconstruction problem from missing/corrupted data.

2.2.2 Explicit decomposition

When considering the low-rank recovery problem in the case of strong noise, it seems that this problem is well solvable by the traditional Principal Component Analysis (PCA). However, the traditional PCA is effective in accurately recovering the underlying low-rank structure only when the noise is Gaussian. If the noise is non-Gaussian and strong, even a few outliers can make PCA fail. To overcome this issue, an extended model called "Robust PCA" or RPCA was considered by Wright et al. [229], Candès et al. [38] and Chandrasekaran et al. [41] when the gross errors are sparse, and we call this model as "explicit decomposition".

The explicit decomposition of the DLSM framework refers to the problem of decomposing an input data matrix $\mathbf{A}$ into the sum of two other matrices in such way that $\mathbf{A} = \mathbf{L} + \mathbf{S}$, where $\mathbf{L}$ is a low-rank matrix and $\mathbf{S}$ express the corrupted entries assumed to be sparse (see Figure 2.5). This definition is also known as Robust Principal Component Analysis (RPCA), and can be formulated as follows:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & rank(\mathbf{L}) + card(\mathbf{S}), \\ \text{subject to} \quad & \mathbf{A} = \mathbf{L} + \mathbf{S}, \end{aligned} \tag{2.10}$$

where $card(\mathbf{S}) = ||\mathbf{S}||_0$ denotes the number of non-zero entries of $\mathbf{S}$. Usually problem (2.10) is rewritten as:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & rank(\mathbf{L}) + \lambda||\mathbf{S}||_0, \\ \text{subject to} \quad & \mathbf{A} = \mathbf{L} + \mathbf{S}, \end{aligned} \tag{2.11}$$

where $\lambda > 0$, similar to the Equation 2.9, is a weight parameter that balances the significance between minimizing $||\mathbf{S}||_0$ and minimizing $rank(\mathbf{L})$. That is, for a larger λ the optimal

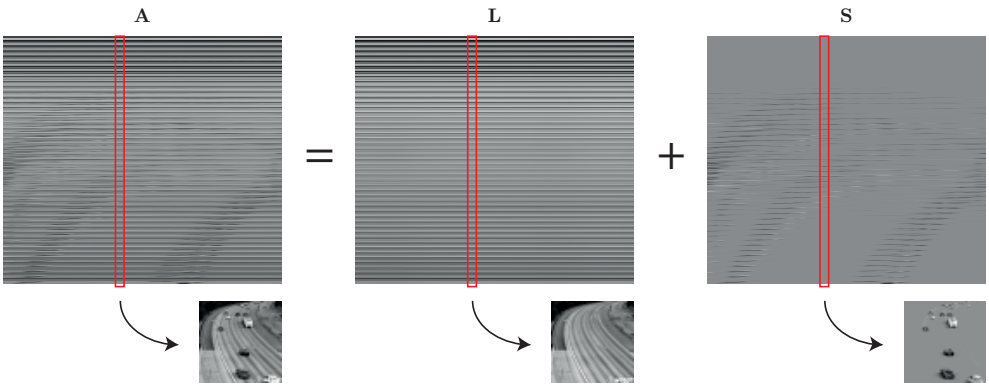


Figure 2.6: Background/foreground separation by RPCA via PCP.

solution will maximize the sparsity in $\mathbf{S}$ providing a less "low-rankness" in $\mathbf{L}$, whereas a smaller λ will result in a minimum sparsity in $\mathbf{S}$ and a more low-rankness in $\mathbf{L}$.

The low-rank minimization concerning $\mathbf{L}$ offers a suitable framework for background modeling due to the high correlation between frames. So, minimizing $\mathbf{L}$ and $\mathbf{S}$ implies that the background is approximated by a low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers which are contained in $\mathbf{S}$. The $rank(\mathbf{L})$ influences the number of "modes" of the background that can be represented by $\mathbf{L}$: if $rank(\mathbf{L})$ is too high, the model will incorporate the moving objects into its representation; if the $rank(\mathbf{L})$ is too low, the model tends to be uni-modal and then the multi-modality which appears in dynamic backgrounds will be not captured. The quality of the background/foreground separation is directly related to the assumption of the low-rank and sparsity of the background and foreground, respectively.

However, as stated in Section 2.2.1, $rank(\mathbf{L}) = ||\sigma(\mathbf{L})||_0$ and $||\mathbf{S}||_0$ yields a highly non-convex optimization problem. The problem (2.10) involves both low rank matrix recovery problem and ℓ_0-minimization problem, and both are NP-hard and hard to approximate [9, 38, 229]. In order to address this issue, a tractable optimization problem is obtained by relaxing (2.10) with convex envelopes that are easier to minimize [38, 229]. Usually the ℓ_0-norm is replaced with the ℓ_1-norm and the $rank(.)$ with the nuclear norm $||.||_*$, yielding the following convex surrogate:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1, \\ \text{subject to} \quad & \mathbf{A} = \mathbf{L} + \mathbf{S}, \end{aligned} \tag{2.12}$$

where $||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1$ is the convex envelope of $rank(\mathbf{L}) + \lambda ||\mathbf{S}||_0$ over the set of $(\mathbf{L}, \mathbf{S})$ such that $\max(||\mathbf{L}||_F, ||\mathbf{S}||_{1,\infty}) \leq 1$ (see Appendix A) [229]. Wright et al. [229] showed that, under natural probabilistic models, the low-rank matrix $\mathbf{L}$ and the sparse matrix $\mathbf{S}$ can be efficiently recovered by solving a convex program. However, the recovery depends on an appropriate choice of the regularizing parameter $\lambda > 0$. Usually, λ is widely assigned as $\lambda = \frac{1}{\sqrt{\max(m,n)}}$, becoming a universal choice [38, 229]. Shortly, Candès et al. [38] extended

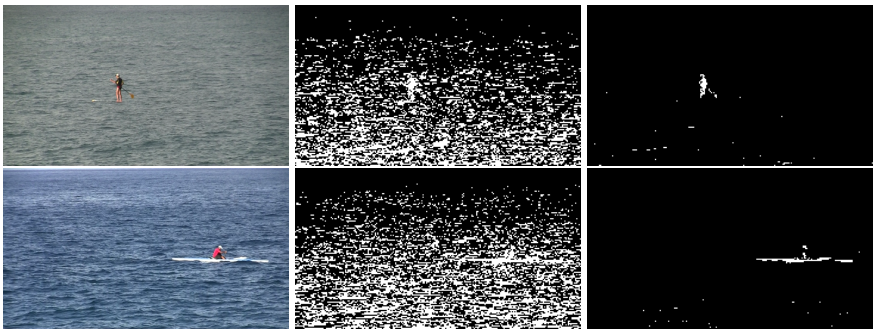


Figure 2.7: Visual comparison of foreground segmentation between PCP and Stable PCP for dynamic background. From left to right: input video, RPCA via PCP, and RPCA via Stable PCP.

the work of Wright et al. [229] for matrices with missing values and showed that it is possible to recover both the low-rank and the sparse components exactly by solving a convex program, called Principal Component Pursuit (PCP), by minimizing a weighted combination of the nuclear norm and of the ℓ_1-norm. The RPCA for matrices with missing entries is formulated as follows:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1, \\ \text{subject to} \quad & P_\Omega(\mathbf{A}) = P_\Omega(\mathbf{L} + \mathbf{S}). \end{aligned} \tag{2.13}$$

Figure 2.6 presents an illustration of the background/foreground separation by using RPCA via PCP proposed by Candès et al. [38]. Essentially, the nuclear-norm term corresponds to the low-frequency components while the ℓ_1-norm describes the high-frequency components. Usually, the low-frequency components (smooth variations) represent the background model and the high-frequency components are the foreground objects. However, this separation is not a trivial task. For example, low frequency components from foreground objects can leak into extracted background images for areas that are very crowded by moving objects. The leakage as ghost artifacts which appear in extracted background cannot be well handled by adjusting the weights between the two regularization parameters. An inverse problem occurs when we seek to separate the moving objects from a very dynamic background. Figure 2.7 shows a typical issue faced by RPCA via PCP for handling very dynamic background scenes (e.g. videos recorded by maritime video surveillance systems). As can be seen, the foreground segmentation is highly contaminated by sparse outliers coming from the dynamic factors in the background model. In order to deal with this issue, some authors [10, 260] proposed a stable version of PCP, discussed in the next section.

2.2.3 Stable decomposition

As previously shown, the PCP has some limitations, as the low-rank component needs to be exactly low-rank and the sparse component needs to be exactly sparse (e.g. consider the input matrix as the sum of a true low-rank matrix plus a true sparse matrix, see Figure 2.5).

However, in real applications the observations are often corrupted by noise. Zhou et al. [260] proposed a stable version of PCP, named Stable PCP (or SPCP), adding a third component that guarantees stable and accurate recovery in the presence of noise. The SPCP is defined by the following model:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}} \quad & ||\mathbf{L}||_* + \lambda_1 ||\mathbf{S}||_1 + \lambda_2 ||\mathbf{E}||_F^2, \\ \text{subject to} \quad & \mathbf{A} = \mathbf{L} + \mathbf{S} + \mathbf{E}, \end{aligned} \tag{2.14}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are weighting parameters, $\mathbf{E}$ is the noise term, and it is usually assumed to be $||\mathbf{E}||_F^2 \leq \epsilon$, where $\epsilon > 0$, allowing the existence of a Gaussian noise. The model (2.14) can be also represented as a relaxed version of PCP:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1, \\ \text{subject to} \quad & ||\mathbf{A} - \mathbf{L} - \mathbf{S}||_F^2 \leq \epsilon, \end{aligned} \tag{2.15}$$

resulting in a stable recovery of $\mathbf{L}$ and $\mathbf{S}$. SPCP offers a suitable framework for background/-foreground separation in real-life applications, as the background model is frequently contaminated by noise. Reconsidering the Figure 2.7, we can see a visual comparison between PCP and SPCP. We can note a relevant improvement given by SPCP compared to PCP in the foreground segmentation mask when dealing with scenes containing a highly dynamical background. In the SPCP model, the dynamical factors from the background model are usually included in the noise matrix $\mathbf{E}$, decreasing the number of wrong sparse components added in the matrix $\mathbf{S}$.

2.2.4 Solvers

In the last few years several algorithms (also named solvers) have been proposed for solving RPCA. All these algorithms require solving the following generalized model defined as follows:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & \lambda_1 f_{low}(\mathbf{L}) + \lambda_2 f_{sparse}(\mathbf{S}) + \lambda_3 f_{noise}(\mathbf{E}), \\ \text{subject to} \quad & \mathcal{C}, \end{aligned} \tag{2.16}$$

where $f_{low}(.), f_{sparse}(.)$ and $f_{noise}(.)$ are surrogate loss functions that are easier to minimize (usually convex functions), and $\mathcal{C}$ is a constraint on the matrices $\mathbf{L}, \mathbf{S}$ and $\mathbf{E}$. Depending on the choice of $f_{low}(.), f_{sparse}(.)$ and $f_{noise}(.)$, different instantiations of the problem (2.16) can be produced. Usually $f_{low}(.), f_{sparse}(.)$ and $f_{noise}(.)$ are taken to enforce the low-rank, sparsity, and noise constraints of $\mathbf{L}, \mathbf{S}$ and $\mathbf{E}$, respectively. Common choices for $f_{low}(.), f_{sparse}(.)$ and $f_{noise}(.)$ are the nuclear norm, ℓ_1-norm, and (squared) Frobenius norm, respectively. The constraint $\mathcal{C}$ is generally based on: a) an equality, such as $||\mathbf{A} - \mathbf{L} - \mathbf{S}||_{norm}^{\eta} = 0$ or $\mathbf{A} = \mathbf{L} + \mathbf{S}$, or b) an inequality, such as $||\mathbf{A} - \mathbf{L} - \mathbf{S}||_{norm}^{\eta} \leq \epsilon$ where η and ϵ are defined commonly as $\eta \in \{1, 2\}$ and $\epsilon = 0.5$. The $||.||_{norm}$ could be any norm, and the most used are the ℓ_2-norm and the Frobenius norm.

2.2.4.1 Optimization algorithms

A wide number of algorithms for solving RPCA both via PCP and via SPCP were proposed in the literature [27, 117]. Here we present an overview of the state-of-the-art algorithms, but for a more complete review, please refer to the recent surveys Bouwmans et al. [27] and Lin [117]. In Chapter 4 we present a special case of RPCA via SPCP in which a particular type of constraint is employed. We also propose a new variant of SPCP to deal with background/foreground separation in maritime video surveillance applications.

Given the original formulation of RPCA, where $f_{low}(\mathbf{L}) = rank(\mathbf{L}) = ||\sigma(\mathbf{L})||_0$ and $f_{sparse}(\mathbf{S}) = card(\mathbf{S}) = ||\mathbf{S}||_0$, this minimization problem yields a NP-hard discrete optimization problem. To overcome this difficulty, a common way is to convert it into a continuous optimization problem, and there are two principal ways to do this. The first way is by converting into a convex program. For example, $f_{low}(.)$ and $f_{sparse}(.)$ are replaced by the nuclear norm and the ℓ_1-norm, respectively. The second way is by converting into a non-convex program. More specifically, using a non-convex continuous function to approximate the $rank(.)$ and the $card(.)$ functions. For example, replacing the $rank(.)$ by the Schatten-p pseudo norm $||.||_{S_p}$ and $card(.)$ by ℓ_α pseudo norm $||.||_\alpha$ where $0 < \alpha < 1$. The principal advantage of convex programs is that a global optimal solution can be relatively easily obtained. The disadvantage is that the solution may not be strictly low-rank or sparse. In contrast, the advantage of non-convex optimization is that low-rank and/or sparse solutions can be obtained. However, their global optimal solution may not be reached. The quality of the solution may heavily depend on the initialization. So the convex and non-convex algorithms complement each other. In the next paragraphs we introduce both convex and non-convex algorithms for solving RPCA.

Convex algorithms: The nuclear norm in the PCP/SPCP problem can be represented as a semidefinite program and solved by interior point methods [29, 38, 117, 165]. These methods are implemented in some commercial solvers such as Mosek[3], SeDuMi[4], YALMIP[5] and CVX[6]. However, interior point methods are typically limited to small size problems (e.g. $n < 100$), due to the $O(n^6)$ complexity. In real applications such as computer vision and machine learning, we often require matrices of size $n > 10^4$, making interior point methods impractical. To overcome this issue, recent approaches focus on first-order optimization methods instead. In general, first-order methods have less numerical precision than interior point methods, but large scale problems can be solved efficiently because no second-order information needs to be stored. First-order methods, such as iterative thresholding algorithms for ℓ_1-minimization [37], perform nuclear-norm minimization by repeatedly shrinking the singular values of the input matrix. This approach reduces the complexity of each iteration to the cost of a SVD. However, iterative thresholding algorithms, such as SVT, converge very slowly [38]. Sub-gradient methods have also been used for convex minimization problems with very large number of dimensions [165]. The main advantages of sub-gradient methods are their simplicity to implement and their scalability to large-scale problems. However, as

[3] https://www.mosek.com/
[4] http://sedumi.ie.lehigh.edu/
[5] https://yalmip.github.io/
[6] http://cvxr.com/

it remains on SVD, the computation of the singular values can be computationally expensive. Currently, the majority of optimization methods for large scale computing are based on first order methods [117]. The most popular techniques include the Accelerated Proximal Gradient (APG) [16, 145], the Frank-Wolfe algorithm [65, 86], and the Alternating Direction Method (ADM) [119, 120].

Non-convex algorithms: RPCA is a popular convex optimization scheme for decomposing an observation matrix into its sparse and low rank components. However, the current methods based on convex optimization are computationally expensive as they require either matrix inversion and/or full (or partial) SVD. Moreover, replacing ℓ_0-norm by ℓ_1-norm to achieve sparsity may be suboptimal, since the ℓ_1-norm is a slack approximation of the ℓ_0-norm leading to an over-penalized problem [131]. Recently, some authors [77, 92, 94, 109, 116, 129, 131, 146, 190, 223, 242, 244, 250] developed a non-convex counter part to rank minimization and RPCA. In general, non-convex optimization problems are NP-hard, even if our goal is to compute a local minimizer [188]. However, some problems, such as deep neural networks (or deep learning) [18, 107], dictionary learning [187] and tensor decomposition [69] can be efficiently solved with heuristic algorithms such as (noisy) gradient descent and alternating directions [188]. Indeed, recent works demonstrate that some non-convex regularizers can outperform their convex counterparts [222, 231]. Several non-convex regularizers have been proposed, such as the ℓ_p-norm [133], Capped ℓ_1-norm [249], Logarithm [68], Exponential-Type Penalty (ETP) [67], Smoothly Clipped Absolute Deviation (SCAD) [62], Minimax Concave Penalty (MCP) [245], Geman [70] and Laplace [206].

The major limitation of the convex approaches for rank minimization (i.e. nuclear norm minimization) is that all the singular values are simultaneously minimized. As previously shown in Equation (2.8), the nuclear norm is essentially an ℓ_1-norm of the singular values and it has a shrinkage effect leading to a biased estimator. In order to deal with this issue, Hu et al. [85] developed a better approximation to the rank by Truncated Nuclear Norm (TNN), which is given by the nuclear norm subtracted by the sum of the largest few singular values. By minimizing TNN, the tailing singular values are influenced to be small, while the magnitudes of the first r singular values are unaffected. A weighted version of the TNN, named Weighted Nuclear Norm (WNN), was also proposed in Gu et al. [77], adding larger weights to smaller singular values. In Lu et al. [130], an Iteratively Reweighted Nuclear Norm (IRNN) algorithm was proposed to solve the non-convex non-smooth low-rank minimization problem. The authors developed a weighted version of the Singular Value Thresholding (SVT) algorithm, which has a closed form solution and is solved iteratively by IRNN. In Lu et al. [129], the authors used a non-convex continuous function to approximate the $rank(.)$ and the $card(.)$ functions, replacing the $rank(.)$ by the Schatten-p pseudo norm $||.||_{S_p}$ and $card(.)$ by ℓ_α pseudo norm $||.||_\alpha$ where $0 < \alpha < 1$. The authors have shown that the algorithm can be solved effectively by Iteratively Reweighted Least Squares (IRLS) [129]. In Lu et al. [131], the authors generalized the SVT, which is widely used in many convex low-rank minimization methods. A Generalized Singular Value Thresholding (GSVT) operator is proposed to solve the non-convex low-rank minimization problem in place of SVT.

Most recently, some authors [94, 146, 242] proposed fast methods for non-convex RPCA. For example, in Netrapalli et al. [146] the method has a linear convergence rate, low complexity, global convergence guarantee and a theoretical guarantee for exact recovery of the

low-rank matrix. The method consists of simple alternating (non-convex) projections onto low-rank and sparse matrices. When the rank r is small, the method nearly matches the complexity of the traditional PCA. In Kang et al. [94], the authors proposed a new matrix norm for non-convex rank approximation, named γ-norm. The γ-norm overcomes the imbalanced penalization by different singular values in convex nuclear norm. The authors adopted the difference of convex (DC) programming [196] to decompose a non-convex function as the difference of two convex functions. As the final solution might not be a globally optimal one, the experiments have shown that the algorithm produces promising results and converges more than twice faster than Netrapalli et al. [146] and 54 times faster than traditional convex RPCA solved by inexact augmented Lagrange multiplier (IALM) [38]. Finally, Yi et al. [242] proposed fast and efficient non-convex algorithms for RPCA via gradient descent. The method was also extended to solve robust PCA with partial observations (matrix completion). In short, the authors propose a projected gradient method that uses a novel sorting-based sparse estimator to produce a rough estimate of the sparse matrix based on the observed matrix. The *sparsification* operator keeps simultaneously a α-fraction of the entries of the residual matrix that have large magnitude. The algorithm outperforms previous non-convex RPCA approaches and shows a linear convergence rate under proper initialization.

2.3 Relation to low-rank/sparse subspace clustering

Subspace clustering via sparse [59] and/or low-rank representation [121] can be regarded as a particular case of RPCA. Differently from RPCA, where inliers[7] lie on a single low dimensional subspace, Low-rank/Sparse Subspace Clustering (L/S-SC) methods consider the inliers are drawn from the union of low-dimensional subspaces. These two common models can be summarized as follows:

- Sparse Subspace Clustering (SSC):

$$\min_{\mathbf{Z},\mathbf{E}} \ ||\mathbf{Z}||_1+\lambda||\mathbf{E}||_l, \ \ s.t. \ \mathbf{X}=\mathbf{AZ}+\mathbf{E}, \ \ diag(\mathbf{Z})=0, \tag{2.17}$$

- Low-Rank Representation (LRR):

$$\min_{\mathbf{Z},\mathbf{E}} \ ||\mathbf{Z}||_*+\lambda||\mathbf{E}||_l, \ \ s.t. \ \mathbf{X}=\mathbf{AZ}+\mathbf{E}, \tag{2.18}$$

In the above formulations, $\mathbf{Z} \in \mathbb{R}^{m\times n}$, $\mathbf{E} \in \mathbb{R}^{m\times n}$ and $\mathbf{A} \in \mathbb{R}^{m\times n}$ as the representation matrix, the noise matrix, and the dictionary matrix (linearly spans the data space), respectively, where $\lambda > 0$ is a parameter to balance the effects of two terms. By choosing $\mathbf{A} = \mathbf{X}$ (i.e., $\mathbf{X} = \mathbf{XZ} + \mathbf{E}$), we assume that the data matrix $\mathbf{X}$ is self-expressive. When $\mathbf{A} = \mathbf{I}$ ($\mathbf{I} = diag(1, 1, \ldots, 1)$), LRR degenerates to RPCA [38], which is suitable for the case that data are drawn from a single subspace. An appropriate dictionary $\mathbf{A}$ enables the low-rank representation to reveal the true subspace structure of the data lying near several subspaces [43, 121]. Usually, the minimization of $||\mathbf{Z}||_1$ enforces the sparsity in the represen-

[7] Data points that have strong mutual coherence.

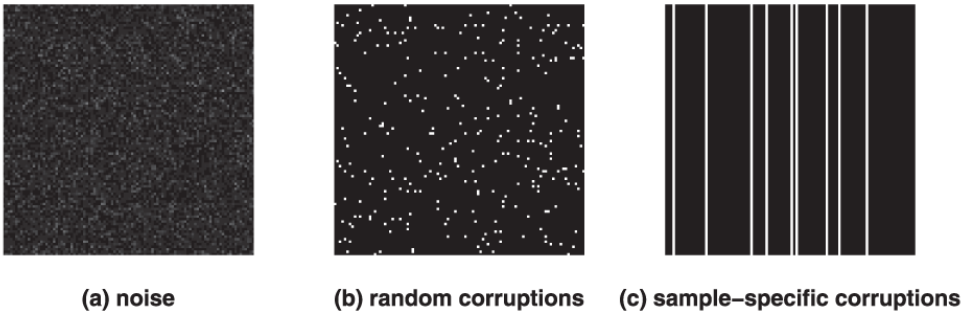


Figure 2.8: Illustration of three typical types of errors in the matrix data (from Liu et al. [121]): a) noise, b) random corruptions, and c) sample-specific corruptions.

tation matrix $\mathbf{Z}$ for SSC, and the minimization of $||\mathbf{Z}||_*$ enforces the low-rank assumption for LRR. The $||\mathbf{E}||_l$ can be replaced by:

- Squared frobenius norm ($||.||_F^2$) to specify the Gaussian disturbance (Figure 2.8(a)).
- l_1-norm ($||.||_1$) to characterize the sparse errors (entry-wise corruption) (Figure 2.8(b)).
- $l_{2,1}$-norm ($||.||_{2,1}$) to deal with sample-specific corruptions and outliers (Figure 2.8(c)).

Robustness of SSC and LRR algorithms has been reported previously in the works of Elhamifar et al. [59] and Liu et al. [121]. The SSC algorithm addresses the subspace clustering problem using techniques from sparse representation theory, while LRR aims to decompose the data matrix as the sum of a clean, self-expressive, low-rank dictionary plus a matrix of noise. Normally, the observed data is chosen to be the dictionary and the noise is assumed to be sparse.

2.3.1 Recent advances in subspace clustering

In the last few years, several authors have developed improved versions of the SSC and LRR given their successes in many computer vision applications. Several variants of SCC and LRR have been developed to deal with special cases when the data matrix can be corrupted by noise, missing entries, and outliers.

SSC variants: Wang et al. [222] developed a modified version of SSC that considers the problem of subspace clustering under noise. Specifically, when random noise is added to the unlabeled input data points, which are assumed to lie in a union of low-dimensional subspaces. Patel et al. [159] proposed a novel algorithm called Latent Space Sparse Subspace Clustering (LS3C) for simultaneous dimensionality reduction and clustering of data lying in a union of subspaces. Specifically, the method learns the projection of data and finds the sparse coefficients in the low-dimensional latent space. Cluster labels are then assigned by applying SC to a similarity matrix built from these sparse coefficients. Soltanolkotabi et al. [185] developed a robust version of SSC to cluster noisy data. In particular, the authors used geometric functional analysis to show that the algorithm can accurately recover the underlying subspaces under minimal requirements on their orientation and on the num-

ber of samples per subspace. Xu et al. [233] proposed a new subspace clustering algorithm, named re-weighted sparse subspace clustering (RSSC) that consists of an iterative weighting (reweighted) l_1 minimization framework which improves the performance of the traditional l_1 minimization framework used in the original SSC. Yang et al. [238] proposed a SSC variant to deal with missing entries outperforming the natural approach (low-rank matrix completion followed by sparse subspace clustering) when the data matrix is high-rank or the percentage of missing entries is large. Li et al. [111] proposed a unified optimization framework for learning both the affinity (affine transformation) and the segmentation (identification of multiple subspaces). The framework is based on expressing each data point as a structured sparse linear combination of all other data points, where the structure is induced by a norm that depends on the unknown segmentation.

LRR variants: Babacan et al. [12] considered the problem of clustering data points into low dimensional subspaces in the presence of outliers. The authors first developed an iterative expectation-maximization (EM) algorithm and then derived its global solution. While the first method is based on an alternating optimization scheme for all unknowns, the second method makes use of recent results in matrix factorization leading to fast and effective estimation. Both methods are extended to handle sparse outliers and missing values. Vidal and Favaro [214] proposed a framework, named Low Rank Subspace Clustering (LRSC), that considers the problem of fitting a union of subspaces to a collection of data points drawn from one or more subspaces and corrupted by noise and/or gross errors. The authors decomposed the corrupted data matrix as the sum of clean and self-expressive dictionary plus a matrix of noise and/or gross errors. The solution involves a novel polynomial thresholding operator on the singular values of the data matrix, which requires a minimal shrinkage. Chen et al. [43] proposed a new framework, named robust low-rank representation (Robust LRR), by considering the low-rank representation as a low-rank constrained estimation for the errors in the observed data. This framework aims to find the maximum likelihood estimation of the low-rank representation residuals and the experimental results have shown the robustness of this method to various type of noises (illumination, occlusion, etc) compared to the original LRR.

Combinations of SSC and LRR: Wang et al. [222] showed that SSC and LRR are fundamentally similar in that both are convex optimizations exploiting the intuition of "Self-Expressiveness". The authors proposed a new algorithm, named Low-Rank Sparse Subspace Clustering (LRSSC), by combining SSC and LRR taking the advantages of both methods in preserving the "Self-Expressiveness Property" and "Graph Connectivity" at the same time. Patel et al. [160] proposed three novel algorithms for simultaneous dimensionality reduction and clustering of data lying in a union of subspaces. Specifically, the authors described methods that learn the projection of the data points and find the sparse and/or low-rank coefficients in the low-dimensional latent space.

2.3.2 Adequacy for the background/foreground separation

L/S-SC methods were widely applied to the motion segmentation (or motion clustering) problem by separating a video sequence into multiple spatio-temporal regions, as they correspond to different rigid-body motions in the scene [59, 93, 111, 164, 216, 233, 238]. Differently from

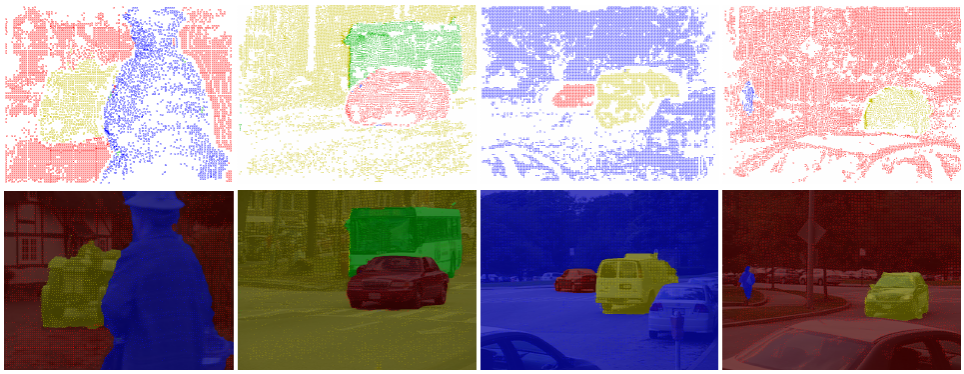


Figure 2.9: Turning sparse point trajectories into dense regions. Figure from Ochs and Brox [149].

background/foreground separation by RPCA where the entire video sequence is represented by a dense matrix decomposed into its low-rank and sparse components, L/S-SC methods work slightly differently. In general, they solve this problem by extracting a set of points in an image, and tracking these points through the video. Given a set of points drawn from a union of linear (or affine) subspaces, all the trajectories associated with a single rigid motion live in a low-dimensional subspace. Therefore, the motion segmentation problem reduces to clustering a collection of point trajectories according to multiple subspaces [59, 215]. Usually the algorithms assume that the feature points are visible in all the frames. However, some authors [164, 216] extended existing methods to the case of missing data, where some of the features are not visible in all the frames. Figure 2.9 (top) illustrates how motion trajectories are clustered into multiple subspaces using Hopkins 155 database [205]. As can be seen, the output from subspace clustering methods differs from the traditional foreground masks given by RPCA approaches. In general, clustering motion trajectories results in sparse trajectories and some additional efforts need to be done to obtain a foreground mask. Some authors [32, 149, 150] developed novel techniques for turning sparse point trajectories into dense regions, please see Figure 2.9 (bottom). Compared to the binary foreground masks obtained from RPCA methods, where the background is represented by a black color and the moving objects by a white color, subspace clustering approaches can provide a more complete information about the moving objects, splitting them into different class of motions. In addition, L/S-SC methods can deal with a particular limitation of the traditional B/F separation methods, that cannot perform well the case of moving cameras.

[7] Hopkins 155 dataset: http://www.vision.jhu.edu/data/hopkins155/

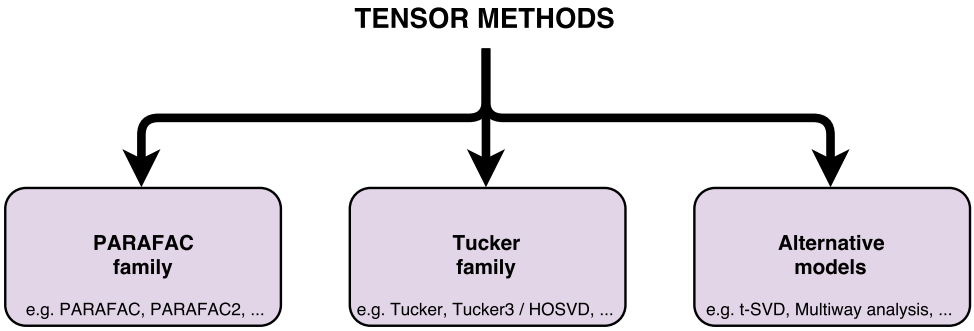


Figure 2.10: Families of tensor methods for multi-way data analysis. Adapted from Acar and Yener [3].

2.4 Extension to tensors

As seen in previous sections, matrix-based low-rank and sparse decomposition methods used for background subtraction work only on a single dimension and consider the image as a vector; hence, multidimensional data for efficient analysis can not be considered. In addition, the local spatial information is lost and erroneous foreground regions can be obtained. Some authors [83, 112, 176, 182, 192, 204] used a tensor representation to solve this problem.

In the thesis, we address some related works that employ robust tensor subspace learning for the background/foreground separation problem. First, we present the principal tensor decomposition tools in Section 2.4.1, then we describe the recent works that employ RPCA on tensors in Section 2.4.2. Moreover, we also present in Chapters 5 and 6 two different approaches for background modeling via tensor subspace learning.

2.4.1 Tensor decomposition and factorization

Tensor decompositions have been widely studied and applied to many real-world problems [76, 99, 132]. They were used to design low-rank approximation algorithms for multidimensional arrays [3, 72, 76, 105] taking full advantage of the multi-dimensional structures of the data.

In the next sections we introduce two widely-used models for low rank decomposition on tensors: the Tucker decomposition (Section 2.4.1.1) and the PARAFAC decomposition (Section 2.4.1.2). Other approaches were also developed [3, 45, 76] and usually they are classified as alternative models (see Figure 2.10). The reader may refer to [3, 45, 76, 99] for a deep literature review on tensor methods. We suggest the reader to refer to Appendix A for a summarized overview of mathematical notations and symbols used for tensors. The reader can also refer to Appendix C for an introduction on tensors, their properties and their operations.

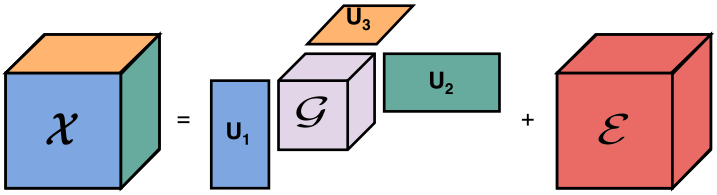


Figure 2.11: Illustration of a Tucker model for a third-order tensor. A third-order tensor is decomposed as the sum of a low-rank tensor (a core tensor multiplied by its factor matrices) and a residual tensor. Image adapted from [3, 99].

2.4.1.1 Tucker decomposition

The Tucker decomposition can be considered as a form of higher-order principal component analysis. It decomposes a tensor into a small tensor, named core tensor, multiplied by a matrix along each mode [99] (please refer to Figure 2.11 for a better illustration). For an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, the Tucker model is formulated as:

$$\mathcal{X} = \mathcal{G} \times_{i=1}^{N} \mathbf{U}_i + \mathcal{E} \tag{2.19}$$

where $\mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N$ is the Tucker model, $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \ldots \times r_N}$ represents the core tensor, $\mathbf{U}_i \in \mathbb{R}^{I_i \times r_i}$ are the factor matrices along the N modes, $r_1 \times r_2 \times \ldots \times r_N$ represents the rank of each mode, and $\mathcal{E}$ contains the residuals. Unlike the SVD for matrices, the core tensor $\mathcal{G}$ does not always result in a diagonal tensor. The columns of $\mathbf{U}_n$ are the principal components of the n-mode fibers on $\mathcal{X}$. For a third-order tensor, the core tensor of minimal size is defined by r_1 (the column rank), r_2 (the row rank), and r_3 (the tube rank). In other words, the multi-linear rank of an N-order tensor is represented by an N-tuple $(r_1, r_2, \ldots, r_N)$. The Tucker decomposition is also considered as a non-convex optimization problem. Several algorithms were developed to solve the Tucker model and the most popular are based on the Alternating Least Squares (ALS) framework [99], also named as Tucker-ALS. However, the ALS method is not guaranteed to converge to a global optimal. In Goldfarb and Qin [72], the authors solve the Tucker model under a convex optimization framework by using an alternating direction augmented Lagrangian (ADAL) method, also named as Tucker-ADAL.

Some authors [3, 104, 105] considered the Tucker model as the generalization of SVD to higher-order tensors[8]. Lathauwer et al. [104, 105] presented a Tucker model (also named as Tucker3) with orthogonality constraints on the components, and this approach is frequently referred to as Higher-Order Singular Value Decomposition (HOSVD). The HOSVD is computed by flattening the tensor in each mode and calculating the singular vectors corresponding to its mode. In other words, it considers the tensor as multiple matrices and forces the un-

[8] Is important to note that, unfortunately, there does not exist a higher order SVD that inherits all the properties of the matrix SVD [3, 99, 210, 211].

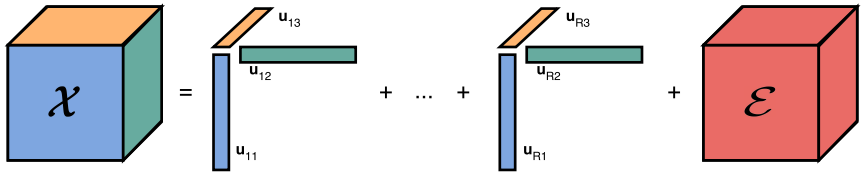


Figure 2.12: Illustration of the CP decomposition of a third-order tensor as the sum of rank-1 tensors $\mathbf{u}_{r1} \circ \mathbf{u}_{r2} \circ \mathbf{u}_{r3}$ for $r \in \{1, 2, \ldots, R\}$. Image adapted from [3, 99].

folding matrix along each mode of the tensor to be low rank as follows:

$$\mathcal{X} = \sum_{n=1}^{N} \mathbf{U}^{[n]} \mathbf{\Sigma}^{[n]} \mathbf{V}^{[n]T} + \mathcal{E} \tag{2.20}$$

where $\mathbf{U}^{[n]} \mathbf{\Sigma}^{[n]} \mathbf{V}^{[n]T}$ represents the SVD applied in the n-mode matricized tensor $\mathcal{X}^{[n]}$ (see Appendix C). This approach is usually referred to as Multilinear SVD [45, 104]. A major difference between SVD and HOSVD is that SVD represents a matrix as a sum of rank-one matrices, while HOSVD does not have this property.

2.4.1.2 CANDECOMP/PARAFAC decomposition

CANDECOMP/PARAFAC(CP)-decomposition can be seen as a special case of the Tucker model, where the core tensor is superdiagonal and the number of components in the factor matrices is the same [99]. The CP-decomposition expresses a tensor as the sum of a finite number of rank-one tensors (please refer to Figure 2.12). Given an N order tensor $\mathcal{X}$, the R-component CP model (also referred to as canonical decomposition) results into the following optimization problem:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{u}_{r1} \circ \mathbf{u}_{r2} \circ \mathbf{u}_{r3} = \mathbf{U}_1 \circ \mathbf{U}_2 \ldots \circ \mathbf{U}_R + \mathcal{E} \tag{2.21}$$

where $\circ$ denotes the outer product, $\mathbf{U}_i \in \mathbb{R}^{I_i \times R}$, $\mathbf{U}_1 \circ \mathbf{U}_2 \ldots \circ \mathbf{U}_R$ represents the PARAFAC model, and $\mathcal{E}$ contains the residuals. Differently from the matrix case, the rank of a tensor is a NP-hard problem [82, 99]. In practice, the rank of a tensor is determined numerically by fitting several rank-R CP models. It is important to note that the best rank-R approximation of a tensor of a rank higher than R is not guaranteed [45, 53]. In general, to compute the rank-R CP model in the presence of noise, the Frobenius norm of the difference between the data tensor and its CP approximation is minimized as follows:

$$\underset{\mathcal{L}}{\text{minimize}} \quad \frac{1}{2} ||\mathcal{X} - \mathcal{L}||_F^2, \tag{2.22}$$

where $\mathcal{L} = \mathbf{U}_1 \circ \mathbf{U}_2 \ldots \circ \mathbf{U}_R$. Usually the PARAFAC model is considered to be the method closest to SVD for matrices, because it decomposes an N-order tensor as the sum

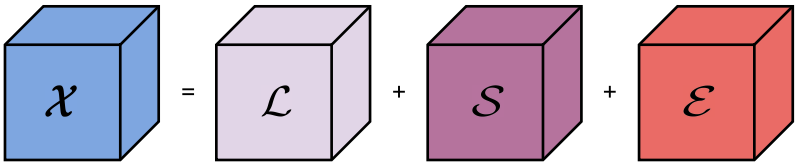


Figure 2.13: Extending robust principal component analysis on a third-order tensor.

of rank-one tensors. In general, the algorithms for solving the PARAFAC model use the ALS framework due to its simplicity. These algorithms were also named as CP-ALS. In the CP-ALS, each component matrix is optimized at a time, keeping the other component matrices fixed [45, 99]. Algorithms based on closed form solutions and gradient methods have also been proposed [202]. Other authors, e.g. Xu and Yin [236], imposed additional structure on the coefficients of the CP decomposition, such as non-negativity. In Zhou et al. [254], the authors proposed an accelerated and online algorithm for fitting the PARAFAC model. Their method achieves the solution much faster than the traditional PARAFAC solved by ALS. Algorithms for solving the PARAFAC model with missing entries were also proposed [201,243,252]. In Chapter 3, we investigate a particular problem of background model initialization. Not only matrix-based completion methods are evaluated, but also the recent approaches for tensor completion. We address the problem of background model initialization as a reconstruction problem from missing/corrupted data.

2.4.2 Robust Principal Component Analysis on tensors

In the last few years, some authors [72, 115, 128, 182, 204] extended the Robust PCA framework to the multilinear case. Basically, the RPCA for matrices was reformulated into its "tensorized" version. For an N-order tensor $\mathcal{X}$, it can be decomposed as:

$$\mathcal{X} = \mathcal{L} + \mathcal{S} + \mathcal{E}, \tag{2.23}$$

where $\mathcal{L}, \mathcal{S}$ and $\mathcal{E}$ represent the low-rank, sparse and noise tensors, respectively (please see Figure 2.13). Similarly to the matrix-case, problem (2.23) can be rewritten as the following optimization problem:

$$\begin{aligned} \underset{\mathcal{L},\mathcal{S}}{\text{minimize}} \quad & rank(\mathcal{L}) + card(\mathcal{S}) \\ \text{subject to} \quad & \mathcal{X} = \mathcal{L} + \mathcal{S}. \end{aligned} \tag{2.24}$$

Due to the intractability of problem (2.24), the $rank(.)$ and $card(.)$ are replaced by their convex envelopes, such as nuclear norm and the element-wise ℓ_1-norm. However, differently from matrices, the rank of a tensor is known to be NP-hard to compute. It is usually replaced by the tensor n-rank.

Definition 2.1. (Tensor n-rank). Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order tensor. The n-rank of $\mathcal{X}$, denoted as $rank_n(\mathcal{X})$, is defined as the number of linearly independent n-mode fibers of $\mathcal{X}$ by

$$rank_n(\mathcal{X}) = rank(\mathbf{X}^{[n]}) \tag{2.25}$$

For matrices, $rank(\mathbf{X}) = rank(\mathbf{X}^{\mathbf{T}})$. However, for a N-th order tensor this is not true. The convex envelope of a $rank_n(\mathcal{X}) = rank(\mathbf{X}^{[n]})$ is replaced by $||\mathbf{X}^{[n]}||_*$. Given an N-order tensor $\mathcal{X}$, its nuclear norm can be generalized as follows:

$$||\mathcal{X}||_* = \sum_{n=1}^{N} ||\mathbf{X}^{[n]}||_*. \tag{2.26}$$

Equation (2.26) represents the Sum of Nuclear Norms (SNN) [122] also referred to as tensor trace norm. So, replacing the $rank(.)$ by SNN and $card(.)$ by ℓ_1-norm in Equation (2.24), the tensor RPCA can be described by the following optimization problem:

$$\begin{aligned} \underset{\mathcal{L},\mathcal{S}}{\text{minimize}} \quad & \sum_{n=1}^{N} ||\mathbf{L}^{[n]}||_* + \lambda ||\mathcal{S}||_1 \\ \text{subject to} \quad & \mathcal{X} = \mathcal{L} + \mathcal{S}. \end{aligned} \tag{2.27}$$

This formulation, first introduced in Li et al. [115], was also extended to the Stable PCP problem:

$$\begin{aligned} \underset{\mathcal{L},\mathcal{S}}{\text{minimize}} \quad & \sum_{n=1}^{N} ||\mathbf{L}^{[n]}||_* + \lambda ||\mathcal{S}||_1 \\ \text{subject to} \quad & ||\mathcal{X} - \mathcal{L} - \mathcal{S}||_F^2 \leq \epsilon, \end{aligned} \tag{2.28}$$

resulting in a stable recovery of $\mathcal{L}$ and $\mathcal{S}$. Li et al. [115] proposed a multilinear extension of the PCP and SPCP problem to the tensor case. The tensor is decomposed into a low dimensional structure plus additive (sparse) component. Their method, named Rank Sparsity Tensor Decomposition (RSTD), employs the alternating direction method (ADM) for the optimization, leading to a block coordinate descent (BCD) algorithm. Some computer vision applications, such as image restoration, BS and face recognition, were also addressed [115]. Subsequently, Tran et al. [204] proposed a tensor-based method for video anomaly detection, applying the Stable PCP decomposition in each tensor mode. The proposed method uses the IALM framework [38] for each unfolded matrix of a tensor to determine which frames are anomalous in a video. Next, Tan et al. [192] proposed a method, named Low-n-rank Tensor Recovery Based on Multi-linear Augmented Lagrange Multiplier Method (LTR-MALM), to overcome the slowly convergence of the previous algorithm [115]. A new minimization method based on augmented Lagrange multiplier method (ALM) is used. The authors showed in the experimental results that the LTR-MALM algorithm is at least several times faster than the RSTD algorithm, while their results are comparable in terms of accuracy. Moreover, Donald and Qin [72] developed a rich framework with several variants of Higher-Order RPCA (HORPCA) methods for robust tensor recovery. Convergence guarantee and proofs of each method were also addressed. Recently, Zhao et al. [253] proposed a Robust Bayesian Tensor Factorization (BRTF) scheme for incomplete tensor completion data. BRTF provides a fast multi-way data convergence but tuning of annoying parameters and batch processing are the

major difficulties of this approach. Finally, Lu et al. [128] proposed a new approach for robust PCA on tensors. Their model is based on a new tensor Singular Value Decomposition (t-SVD) method developed by [97, 251]. t-SVD is the best close representation of SVD for third order tensors, decomposing $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$.

2.5 Conclusion

In summary, we presented an overview of the state-of-the-art methods for low-rank and sparse decomposition, as well as their application to the problem of background/foreground separation. The methods were unified in a more general framework, named DLSM, that categorizes the matrix separation problem into three main approaches: implicit, explicit and stable. In addition, we presented the matrix separation problem from a single low dimensional subspace to a union of low-dimensional subspaces, introducing the subspace clustering approach. We also showed its adequacy to the problem of background/foreground separation by clustering motion trajectories. Finally, we extended the matrix case to the tensor case for handling multidimensional data.

Chapter 3

Background model initialization via matrix and tensor completion

In this chapter, we investigate the problem of background model initialization as a reconstruction problem from missing/corrupted data. This problem can be formulated as a matrix or tensor completion task where the image sequence (or video) is revealed as partially observed data. This work is based on our publication (SBMI/ICIAP, 2015, [178]), and on its extended version for tensors (PRL, 2016, [184]). In addition, the majority of matrix and tensor completion algorithms presented here were made publicly available in the LRS library [180][1] (see Appendix D).

3.1 Introduction

As outlined in Chapter 1, background model initialization is commonly the first step of the BS process. It typically consists of creating a background model that best represents the scene background. In a simple way, this can be done by manually setting a static image that represents the background. Indeed, it is often assumed that initialization can be achieved by exploiting some clean frames at the beginning of the sequence, and the scene here is assumed to be stationary or quasi stationary. Naturally, this assumption is rarely encountered in real-life scenarios, because of continuous clutter presence. In addition, this procedure presents several limitations, because it needs a fixed camera with constant illumination, and the background needs to be static (commonly in indoor environments), and having no moving object in the first frames. In practice, several challenges appear and perturb this process, such as noise acquisition, bootstrapping, dynamic factors, etc. [25, 135].

The main challenge is to obtain a first background model when video frames contain foreground objects. Some authors perform the initialization of the background model by the arithmetic mean [102] (or weighted mean) of the pixels between successive images. Prac-

[1] LRSLibrary: https://github.com/andrewssobral/lrslibrary

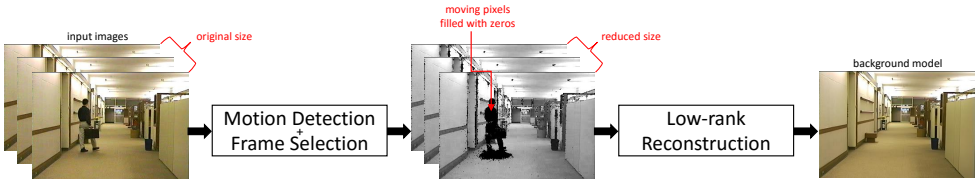


Figure 3.1: Proposed approach to background model initialization: given an input image, a joint motion-detection and frame-selection operation is applied. Next, a low-rank reconstruction process recovers the background model from the partially observed data.

Table 3.1: Classification of background model initialization methods according to Bouwmans et al. [25]. The approaches presented in this chapter are in bold face.

Type of methods	**Related works**
Temporal Statistics	Mean, Color Median, MoG [186, 262], KNN [263], BE-AAPSA [163]
Subintervals of Stable Intensity	WS2006 [220], IMBS-MT [21], LaBGen [106]
Model Completion	RSL2011 [166]
Optimal Labeling	Photomontage [4]
Subspace Estimation	Eigen [151], RSL [52], RPCA [38]
Missing Data Reconstruction	**Matrix Completion [178], Tensor Completion [184]**
Neural Networks	SC-SOBS [134], BEWiS [51]

tically, some algorithms are: (1) batch, using n training frames (consecutive or not), (2) incremental with known n or (3) progressive with unknown n, as the process generates partial backgrounds and continues until a complete background image is obtained. Furthermore, initialization algorithms depend on the number of background modes and the complexity of their background models. However, BS initialization has also been achieved by many other methodologies [24, 25, 135].

In 2014, Maddalena and Petrosino [135] initiated a first survey on background initialization models. This survey was extended in Maddalena and Petrosino [136] by adding new methods. Moreover, the authors assembled a dataset, named SBI 2015, consisting of sequences frequently adopted for background initialization. Second, a more complete survey was developed in Bouwmans et al. [25] by adding new methods and extending the SBI 2015 dataset. The main investigations used *methods based on temporal statistics* [163, 186, 262, 263], *subintervals of stable intensity* [21, 106, 220], *model completion* [166], *optimal labeling* [4], *subspace estimation* [38, 52, 151], and *neural networks* [51, 134]. Table 3.1 summarizes the type of methods according to the taxonomies presented in [25]. Concerning the works based on subspace learning (related to this chapter), we can cite for example the computation of eigen values and eigen vectors [151], and the robust subspace learning approach proposed by De La Torre and Black [52]. However, the recent research on subspace estimation by sparse representation and rank minimization [28] has been showing a suitable framework for background modeling. The background model is recovered by the low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers.

3.2 Proposed methodology

In this chapter, we present a novel methodology for background model initialization, classified as Missing Data Reconstruction in Table 3.1. The initialization of the background model is addressed as a reconstruction problem from missing data. Indeed, this problem can be formulated as a matrix or tensor completion task, where the image sequence (or video) is revealed as partially observed data. The missing entries are induced from the moving regions through a simple joint motion-detection and frame-selection operation. The redundant frames are eliminated, and the moving regions are represented by zeros in our observation model. The second stage involves evaluating twenty-three state-of-the-art algorithms including thirteen matrix completion and ten tensor completion algorithms. These algorithms aim to recover the low-rank component (or background model) from partially observed data. All experiments were performed by using the SBI dataset proposed by Maddalena and Petrosino [136][2]. Figure 3.1 shows the proposed framework. In this chapter, the processes described here are conducted in a batch manner.

3.2.1 Joint motion detection and frame selection

The elimination of redundant frames is an important step for a fast low-rank reconstruction process, removing the irrelevant information and decreasing the high computational cost of some matrix and tensor based methods. All algorithms evaluated in this thesis (except GROUSE) are batch methods, requiring all frames to be vectorized and stored in a column vector from a big matrix (usually, frame resolution $\times$ number of frames) before optimization.

Given a sequence of images, in order to reduce the number of redundant frames, a simple joint motion-detection and frame-selection operation is applied. First, the color images are converted into their gray-scale representation. So, let a sequence of n gray-scale images (frames) $\mathbf{F}_1 \dots \mathbf{F}_n$ captured from a static camera, that is, $\mathbf{F} \in \mathbb{R}^{I_1 \times I_2}$ where I_1 and I_2 denote the frame resolution (rows by columns). The difference between two consecutive frames (motion detection step) is calculated by:

$$\mathbf{D}_t = \begin{cases} \mathbf{0} & \text{if } t = 1 \\ \sqrt{(\mathbf{F}_t - \mathbf{F}_{t-1})^2} & \text{otherwise} \end{cases}, \tag{3.1}$$

where $t = 1, \dots, n$, $\mathbf{0} \in \mathbb{R}^{I_1 \times I_2}$ denotes a zero matrix[3] and $\mathbf{D}_t \in \mathbb{R}^{I_1 \times I_2}$ denotes the matrix of pixel-wise L_2-norm differences from frame $t-1$ to frame t. Next, the sum of all elements of $\mathbf{D}_t$ is stored in a data vector $\mathbf{d} \in \mathbb{R}^n$ whose t-th element is given by:

$$d_t = \sum_{x=1}^{I_1} \sum_{y=1}^{I_2} \mathbf{D}_t(x, y), \tag{3.2}$$

where $\mathbf{D}_t(x, y)$ is the matrix element located in the row $x \in [1, \dots, I_1]$ and column $y \in$

[2] http://sbmi2015.na.icar.cnr.it/SBIdataset.html

[3] A matrix with all its entries being zero.

Table 3.2: Number of selected frames after the frame-selection process.

#	Sequence	Frames	Selected	Reduction	τ
1	Board	228	64	71.93%	0.125
2	Candela_m1.10	350	84	76.00%	0.100
3	CAVIAR1	610	88	85.57%	0.100
4	CAVIAR2	460	83	81.96%	0.125
5	CaVignal	258	65	74.81%	0.125
6	Foliage	394	68	82.74%	0.600
7	Hall&Monitor	296	94	68.24%	0.075
8	HighwayI	440	59	86.59%	0.100
9	HighwayII	500	49	90.20%	0.075
10	HumanBody2	740	86	88.38%	0.050
11	IBMtest2	90	33	63.33%	0.100
12	People&Foliage	341	55	83.87%	0.100
13	Snellen	321	70	78.19%	0.125
14	Toscana	6	6	0.00%	-

$[1, \ldots, I_2]$. Then, the data vector $\mathbf{d}$ is normalized between 0 and 1 by:

$$\bar{\mathbf{d}} = norm(\mathbf{d}) = \frac{\mathbf{d} - \min(\mathbf{d})}{\max(\mathbf{d}) - \min(\mathbf{d})}, \tag{3.3}$$

where $\min(\mathbf{d})$ and $\max(\mathbf{d})$ denote the minimum and the maximum value of the vector, respectively. The frame-selection step is done by calculating the derivative of $\bar{\mathbf{d}}$ by:

$$\mathbf{d}' = \frac{d}{dt}\bar{\mathbf{d}}, \tag{3.4}$$

Next, the vector $\mathbf{d}'$ is also normalized as in Equation (3.3). Finally, the index of the more relevant frames is obtained by thresholding $\bar{\mathbf{d}}'$:

$$\mathbf{v} = \begin{cases} 1 & \text{if } |\bar{\mathbf{d}}' - \mu'| > \tau \\ 0 & \text{otherwise} \end{cases}, \tag{3.5}$$

where μ' denotes the mean value of the vector $\bar{\mathbf{d}}'$, and $\tau \in [0, 1]$ controls the threshold operator. In this chapter, $\dot{n} \leq n$ represent the set of all frames where $\mathbf{v} = 1$, and the parameter τ was chosen empirically for each scene. Figure 3.2 illustrates the frame selection operation in HallAndMonitor scene. The normalized vector (in blue) shows the difference between two consecutive frames. The derivative vector (in red) draws how much the normalized vector changes. Then, it is thresholded and the more relevant frames are selected (in orange). For this example, with $\tau = 0.075$, only 94 relevant frames are selected from a total of 296 frames ($68, 24\%$ of reduction). Table 3.2 shows the number of selected frames after frame selection process for the SBI dataset. As it can be seen, an average of 80% of reduction was achieved for each scene. The Toscana scene was ignored, due to its small amount of frames.

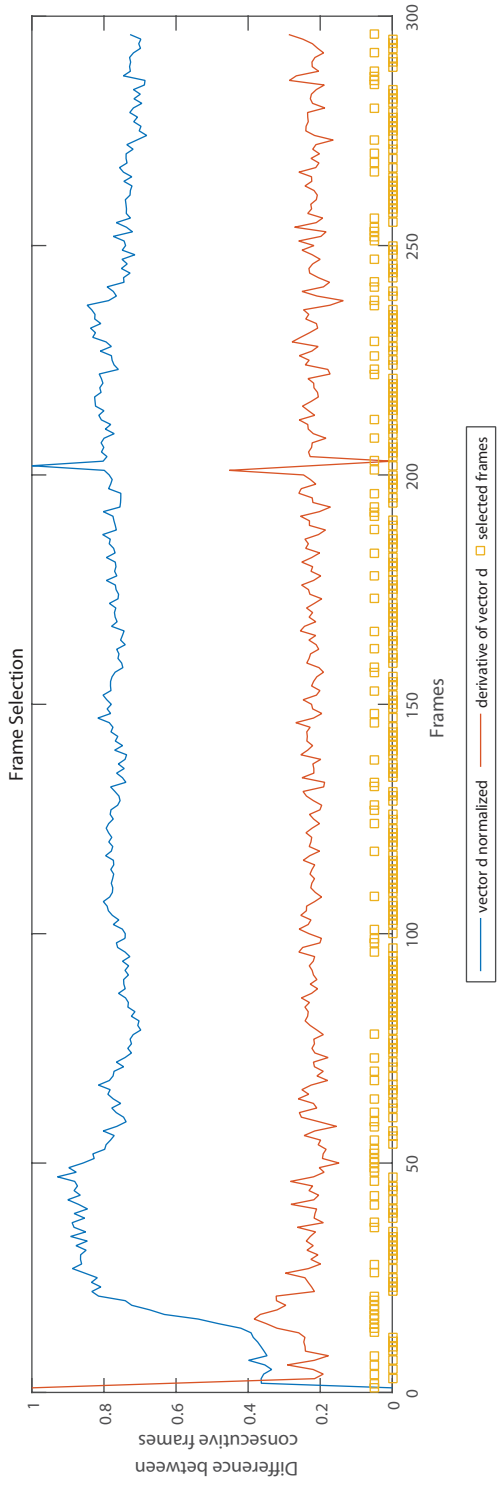


Figure 3.2: Illustration of the frame-selection operation.

Figure 3.3: Illustration of the low-rank reconstruction process. From top to bottom: matrix-based and tensor-based completion process. From left to right: a) the selected frames, b) the moving regions are represented by non-observed entries (black pixels), c) the moving regions filled with zeros, and d) the recovered data after low-rank reconstruction process.

3.2.2 Low-rank reconstruction from missing data

In this section the low-rank reconstruction is addressed from two points of view: matrix completion (MC) and tensor completion (TC). First, we start with a matrix concept of the completion process in Section 3.2.2, and next we describe a generalized concept with tensors in Section 3.2.2, providing brief descriptions of the methods adopted for the evaluation (Section 3.3).

The matrix completion case

As explained previously in Chapter 2, MC aims to recover a low rank matrix from partial observations of its entries. In recent years, several methods for low-rank matrix recovery have been proposed. Basically, they are divided into two categories based on their approaches to modeling the low-rank prior [259]. The first approach is to minimize the rank of the input matrix subject to some constraints. The second approach is to factorize the input matrix as the product of two factor matrices; the rank of the input matrix is upper bounded by the ranks of the factor matrices.

Matrix completion by rank minimization A direct approach to recover a low-rank matrix is to find a matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ with minimum rank that best approximates the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, as reported in Section 2.2.1, Equation (2.7). Candès and Recht [40] showed that this problem can be formulated as:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & rank(\mathbf{L}), \\ \text{subject to} \quad & P_\Omega(\mathbf{L}) = P_\Omega(\mathbf{A}), \end{aligned} \tag{3.6}$$

where $rank(\mathbf{L})$ indicates the rank of the matrix $\mathbf{L}$, and P_Ω denotes the sampling operator restricted to the elements of Ω (set of observed entries), i.e., $P_\Omega(\mathbf{A})$ has the same values as $\mathbf{A}$ for the entries in Ω and zero values for the entries outside Ω. Candès and Recht [40] proposed to replace the $rank(.)$ function with the nuclear norm:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & ||\mathbf{L}||_*, \\ \text{subject to} \quad & P_\Omega(\mathbf{L}) = P_\Omega(\mathbf{A}), \end{aligned} \tag{3.7}$$

where $||\mathbf{L}||_* = \sum_{i=1}^{r} \sigma_i$ such that $\sigma_1, \sigma_2, ..., \sigma_r$ are the singular values of $\mathbf{L}$ and r is the rank of $\mathbf{L}$. The nuclear norm makes the problem tractable and Candès and Recht [40] proved theoretically that the solution can be exactly recovered with a high probability. In addition, Cai et al. [37] proposed an algorithm based on soft Singular Value Thresholding (SVT) to solve this convex relaxation problem. However, in real world applications the observed entries may be noisy. In order to make the problem (3.7) robust to noise, Candès and Plan [39] proposed a stable matrix completion approach relaxing the equality constraint by:

$$\underset{\mathbf{L}}{\text{minimize}} \quad \frac{1}{2}||P_\Omega(\mathbf{L}) - P_\Omega(\mathbf{A})||_F^2 + \lambda||\mathbf{L}||_*, \tag{3.8}$$

where $||.||_F$ denotes the Frobenious norm and the parameter λ controls the rank of $\mathbf{L}$. The selection of λ should depend on the noise level [39].

A few recent works used an online formulation for matrix completion [15, 80, 127]. Online algorithms are useful because they are faster and need less storage compared to most batch techniques. In He et al. [80], the authors introduce GRASTA (Grassmannian Robust Adaptive Subspace Tracking Algorithm), an online robust subspace tracking algorithm that operates on highly subsampled data. In Balzano and Wright [15], the authors present GROUSE (Grassmanian Rank-One Update Subspace Estimation), a subspace identification and tracking algorithm that builds high quality estimates from very sparsely sampled vectors. In Lois and Vaswani [127], the authors introduce the ReProCS (Recursive Projected Compressive Sensing) algorithm for both online MC and online RPCA.

Matrix completion by matrix factorization Instead of minimizing rank, another approach for performing MC is through matrix factorization (MF). MF methods decompose the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ as the product of two factor matrices: $\mathbf{A} = \mathbf{W}\mathbf{H}^T$, where $\mathbf{W} \in \mathbb{R}^{m \times r}$, $\mathbf{H} \in \mathbb{R}^{n \times r}$, and r controls the rank of $\mathbf{W}$ and $\mathbf{H}$. Therefore, if r is small, $\mathbf{A}$ has a small rank. Using matrix factorization to model a low-rank matrix is based on the fact that $rank(\mathbf{W}\mathbf{H}^\mathbf{T}) \leq min(rank(\mathbf{W}), rank(\mathbf{H}))$. The problem of recovering a low-rank matrix can be converted into estimating two factor matrices $\mathbf{W}$ and $\mathbf{H}$. In the case of missing values, the factorization-based methods for MC aim to solve the following optimization problem:

$$\text{minimize} \quad \frac{1}{2}||P_\Omega(\mathbf{A}) - P_\Omega(\mathbf{W}\mathbf{H}^\mathbf{T})||_F^2. \tag{3.9}$$

A straightforward approach to solve the problem (3.9) is by minimizing the function over $\mathbf{W}$ or $\mathbf{H}$ alternately, fixing the other one. Each subproblem of estimating $\mathbf{W}$ or $\mathbf{H}$ turns into a least-squares problem, which admits a closed-form solution. Algorithms of this type have been well studied in many works in the recent matrix recovery literature [87, 193]. For example, the matrix completion solver LMaFit [227] also adopted the alternating strategy to solve the following equivalent form of the problem (3.9):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\mathbf{Z} - \mathbf{W}\mathbf{H}^\mathbf{T}||_F^2, \\ \text{subject to} \quad & P_\Omega(\mathbf{Z}) = P_\Omega(\mathbf{A}), \end{aligned} \tag{3.10}$$

where $\mathbf{Z}$ is an auxiliary variable. Additionally, LMaFit integrates a nonlinear successive over-relaxation scheme to accelerate the convergence of alternation.

Nonnegative factors Non-negative Matrix Factorization (NMF) is a special case of the traditional MF, where the factor matrices $\mathbf{W}$ and $\mathbf{H}$ have no negative elements. The non-negativity makes the resulting matrices easier to inspect[4], as in many applications (e.g. images, texts, etc.) the data is non-negative. However, NMF is an NP-hard problem that requires to impose additional assumptions (e.g. lowrankness, convexity, etc.) on the data

[4]NMF learns a parts-based representation of the data, whereas PCA learn holistic representations [108].

points in order to reduce the original NMF to a tractable problem. In the literature, some authors [237,255] have addressed the non-negativity and low-rank completion to take the advantages of both, obtaining superior results than those of just using one of the two properties.

Randomized decomposition The factorization of large matrices becomes expensive and sometimes impractical for the traditional (deterministic) MF algorithms. In recent years, some authors focused on modern randomized matrix approximation techniques [78]. These algorithms use random sampling to identify a subspace that captures most of the underlying information of a matrix. Instead of computing the SVD of the whole matrix $\mathbf{A}$, the randomized low-rank SVD [61,228] consists of computing a rank-r approximation of $\mathbf{A}$, such that $\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^{\mathbf{T}}\mathbf{A}$, where $\mathbf{Q} \in \mathbb{R}^{m\times r}$ is orthonormal representing the economic QR decomposition[5] of $\mathbf{A}\boldsymbol{\Omega} = \mathbf{Q}\mathbf{R}$ such that $\boldsymbol{\Omega} \in \mathbb{R}^{m\times r}$ is a random sub-Gaussian[6] matrix. Then, the algorithm efficiently computes the SVD of a relatively small matrix $\mathbf{B} = \mathbf{Q}^{\mathbf{T}}\mathbf{A}$. Zhou and Tao [257] proposed a fast alternative way, named Bilateral Random Projections (BRP), that avoids the SVD for large matrices. The effectiveness and the efficiency of BRP was verified in the GoDec [256], SSGoDec [256] and GreGoDec [258] algorithms for low-rank matrix approximation and completion. Given r bilateral random projections of a $m \times n$ dense matrix $\mathbf{A}$, the low-rank approximation $\mathbf{L}$ can be rapidly built by $\mathbf{L} = \mathbf{Y}_1(\mathbf{X}_2^T\mathbf{Y}_1)^{-1}\mathbf{Y}_2^T$, where $\mathbf{Y}_1 = \mathbf{A}\mathbf{X}_1$, $\mathbf{Y}_2 = \mathbf{A}^T\mathbf{X}_2$, and $\mathbf{X}_1 \in \mathbb{R}^{n\times r}$ and $\mathbf{X}_2 \in \mathbb{R}^{m\times r}$ are random matrices.

Riemannian optimization Another widely-used regularization strategy in low-rank matrix factorization is to constrain the search space and optimize over manifolds. Keshavan et al. [96] proposed to solve the following matrix completion problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||P_\Omega(\mathbf{A}) - P_\Omega(\mathbf{W}\boldsymbol{\Sigma}\mathbf{H}^{\mathbf{T}})||_F^2, \\ \text{subject to} \quad & \mathbf{W} \in Gr(r,m), \mathbf{H} \in Gr(r,n), \Sigma \in \mathbb{R}^{r\times r}, \end{aligned} \tag{3.11}$$

where $Gr(r,p)$ denotes the set of r-dimensional subspaces in $\mathbb{R}^p$, which forms a Riemannian manifold, named Grassmannian. Keshavan et al. [96] proposed an algorithm named OptSpace to iteratively estimate the factor matrices, where $\mathbf{W}$ and $\mathbf{H}$ are updated by gradient descent over the Grassmannian, while $\boldsymbol{\Sigma}$ is updated by least squares.

Instead of exploring the geometries of search spaces of factor matrices, Vandereycken [209] proposed to directly optimize a function over the set of fixed-rank matrices:

$$\begin{aligned} \underset{\mathbf{L}}{\text{minimize}} \quad & \frac{1}{2}||P_\Omega(\mathbf{A}) - P_\Omega(\mathbf{L})||_F^2, \\ \text{subject to} \quad & \mathbf{A} \in M_r, \end{aligned} \tag{3.12}$$

where M_r denotes the set of rank-r matrices in $\in \mathbb{R}^{m\times n}$, which forms a smooth manifold. Vandereycken [209] developed a conjugate gradient descent algorithm named LRGeomCG

[5] If $\mathbf{A}$ is an m-by-n matrix with $m > n$, then QR computes only the first n columns of $\mathbf{Q}$ and the first n rows of $\mathbf{R}$.

[6] A sub-Gaussian distribution is a probability distribution with strong tail decay property [33].

Table 3.3: List of MC algorithms evaluated for BM initialization.

Type	Method	Main techniques	Author(s)
RM	IALM	Augmented Lagrangian	Lin and Wei (2010) [118]
	RMAMR	Augmented Lagrangian	Ye et al. (2015) [241]
MF	SVP	Hard thresholding	Jain et al. (2010) [87]
	OptSpace	Grassmannian	Keshavan et al. (2010) [96]
	MC-NMF	Non-negative factors	Xu et al. (2012) [237]
	LMaFit	Alternating	Wen et al. (2012) [227]
	ScGrassMC	Grassmannian	Ngo and Saad (2012) [148]
	LRGeomCG	Riemannian	Vandereycken (2013) [209]
	GROUSE	Online algorithm	Balzano and Wright (2013) [15]
	OR1MP	Matching pursuit	Wang et al. (2015) [224]
	GoDec	Randomized	Zhou and Tao (2011) [256]
	SSGoDec	Randomized	Zhou and Tao (2011) [256]
	GreGoDec	Randomized	Zhou and Tao (2013) [258]

RM - Rank Minimization
MF - Matrix Factorization

to efficiently optimize any smooth function over M_r.

Background modeling through matrix completion

Considering the background model initialization as a matrix completion problem, once the frame-selection process is done, the moving regions of the $\dot{n}$ frames, selected in the previous step (see Section 3.2.1), are determined by:

$$\mathbf{M}_k = thresh(\mathbf{D}_k) = \begin{cases} 1 & \text{if } 0.5(\mathbf{D}_k)^2 > \beta \\ 0 & \text{otherwise} \end{cases}, \tag{3.13}$$

where $k \in \{1, \ldots, \dot{n}\}$, $thresh(.)$ denotes a thresholding function, $\mathbf{D}_k$ is computed as in Equation (3.1) using only the $\dot{n}$ selected frames and β is a thresholding parameter (in this chapter, $\beta = 1e^{-3}$ for all experiments). Next, the moving regions of each selected frame are filled with zeros by $\mathbf{F}_k \circ \overline{\mathbf{M}_k}$, where $\overline{\mathbf{M}_k}$ denotes the complement of $\mathbf{M}_k$, and $\circ$ denotes the element-wise multiplication of two matrices. For color images, each channel is processed individually, then they are vectorized into a partially observed real-valued matrix $\mathbf{A} = [vec(F_1) \ldots vec(F_{\dot{n}})]$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m = (I_1 \times I_2)$, and $n = \dot{n}$. Figure 3.3 (top) illustrates our matrix completion process. It can be seen that the partially observed matrix can be recovered successfully even with the presence of many missing entries. So, let $\mathbf{L}$ be the recovered matrix from the matrix completion process, the background model is estimated by calculating the average value of each row, resulting in a vector $\mathbf{l} \in \mathbb{R}^m$, and then reshaped into a matrix $\mathbf{B} \in \mathbb{R}^{I_1 \times I_2}$.

Table 3.4: List of TC algorithms evaluated for BM initialization.

Type	Method	Main techniques	Author(s)
CP	NCPC	Non-negative factors	Xu and Yin (2013) [236]
	BCPF	Bayesian CP Factorization	Zhao et al. (2015) [252]
	TenALS	Alternating	Jain et al. (2014) [88]
	SPC	Smooth PARAFAC	Yokota et al. (2016) [243]
TD	HoRPCA-IALM	Augmented Lagrangian	Goldfarb and Qin (2014) [72]
	FaLRTC	Trace norm	Liu et al. (2013) [122]
	geomCG	Riemannian	Kressner et al. (2013) [100]
	TMac	Alternating	Xu et al. (2015) [235]
	t-SVD	Fourier domain	Zhang et al. (2014) [251]
	t-TNN	Nuclear norm	Hu et al. (2015) [84]

CP - CANDECOMP/PARAFAC decomposition.
TD - Tucker decomposition / HOSVD / N-mode SVD.

The tensor completion case

Differently from previous matrix-based methods that consider the image as a vector, so that the local spatial information is almost lost, some authors use a tensor representation to solve this low-rank reconstruction problem. Tensor decompositions have been widely studied and applied to many real-world problems [76, 99, 132]. As outlined in Chapter 2, Section 2.4, CP decomposition and Tucker decomposition are two widely-used low rank decompositions of tensors. Today, the Tucker model is better known as the Higher-Order SVD (HOSVD) from the work of Lathauwer et al. [105]. The HOSVD of a tensor $\mathcal{X}$ can be seen as the generalization of the matrix SVD, which involves the matrix SVDs of its unfolding matrices. In general, the low-rank completion for tensors is formulated as the following optimization problem:

$$\begin{aligned} \underset{\mathcal{L}}{\text{minimize}} \quad & rank(\mathcal{L}), \\ \text{subject to} \quad & P_\Omega(\mathcal{L}) = P_\Omega(\mathcal{X}), \end{aligned} \tag{3.14}$$

where $\mathcal{L} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is a tensor with minimum rank that best approximates the tensor $\mathcal{X}$. However, similarly to the matrix case, the optimization problem (3.14) is a non-convex optimization problem. This is commonly solved through trace norm optimization by:

$$\begin{aligned} \underset{\mathcal{L}}{\text{minimize}} \quad & ||\mathcal{L}||_*, \\ \text{subject to} \quad & P_\Omega(\mathcal{L}) = P_\Omega(\mathcal{X}). \end{aligned} \tag{3.15}$$

For a general tensor case, the definition of the trace norm is represented by a combination of the trace norms of all matrices unfolded along each mode as:

$$||\mathcal{X}||_* = \sum_{i=1}^{N} \alpha_i ||\mathcal{X}^{[i]}||_*, \tag{3.16}$$

where α_i's are constants satisfying $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i = 1$. However, unlike in the matrix case, computing the rank of a general tensor ($N > 2$) is an NP hard problem [82]. In tensor literature [76], three non-convex ways to deal with tensor completion problem can be found:

Tucker A natural approach is to analyze the tensor completion problem through Tucker model, introduced in Chapter 2 (see Section 2.4.1.1), in such a way that:

$$\begin{aligned} \underset{\mathcal{L}}{\text{minimize}} \quad & \frac{1}{2}||\mathcal{X} - \mathcal{L}||_F^2, \\ \text{subject to} \quad & P_\Omega(\mathcal{L}) = P_\Omega(\mathcal{X}). \end{aligned} \tag{3.17}$$

where $\mathcal{L} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_N \mathbf{U}_N$ is the Tucker model, $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \ldots \times r_N}$ represents the core tensor, $\mathbf{U}_i \in \mathbb{R}^{I_i \times r_i}$ are the factor matrices along the N modes and $r_1 \times r_2 \times \ldots \times r_N$ represents the rank of each mode. This can be solved by block coordinate descent method by iteratively optimizing two blocks $\mathcal{X}$ and $\mathcal{G}, \mathbf{U}_1, \ldots, \mathbf{U}_N$. Similar to the matrix case, approaches based on Riemannian optimization were also used for tensor completion problems. In Kasai and Mishra [95], the authors developed a novel Riemannian metric and explore the symmetry structure in Tucker decomposition.

CANDECOMP/PARAFAC A second approach is to use the CP model, introduced in Chapter 2 (see Section 2.4.1.2), resulting in the following optimization problem:

$$\begin{aligned} \underset{\mathcal{L}}{\text{minimize}} \quad & \frac{1}{2}||\mathcal{X} - \mathcal{L}||_F^2, \\ \text{subject to} \quad & P_\Omega(\mathcal{L}) = P_\Omega(\mathcal{X}). \end{aligned} \tag{3.18}$$

where $\mathcal{L} = \mathbf{U}_1 \circ \mathbf{U}_2 \ldots \circ \mathbf{U}_N$, $\circ$ denotes the outer product, $\mathbf{U}_i \in \mathbb{R}^{I_i \times r}$, and $\mathbf{U}_1 \circ \mathbf{U}_2 \ldots \circ \mathbf{U}_N$ represents the PARAFAC model and r represents the rank of the model. An alternative to the canonical tensor decomposition is the Tensor-Train (TT) format introduced by Oseledets [153]. Recent work employing TT in the context of tensor completion was released by Grasedyck et al. [75]. TT format offers a number of advantages over the canonical decomposition, and it is therefore attractive to consider its application to function approximation.

Tucker3 or Higher-Order SVD A third alternative is to consider the tensor as multiple matrices and force the unfolding matrix along each mode of the tensor to be low rank, (see Tucker model in Chapter 2, Section 2.4.1.1), as follows:

$$\begin{aligned} \underset{\mathcal{L}}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{N} \alpha_i ||\mathcal{X}^{[i]} - \mathcal{L}^{[i]}||_F^2, \\ \text{subject to} \quad & P_\Omega(\mathcal{L}) = P_\Omega(\mathcal{X}). \end{aligned} \tag{3.19}$$

where α_i's are constants satisfying $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i = 1$. This approach is also known as matrix SVD adapted for tensors. Recently Kilmer and Martin [97] and Zhang et al. [251] proposed a real representation of SVD for third order tensors called t-SVD (Tensor Singular

Value Decomposition). For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its SVD is formulated by $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{U} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$ and $\mathcal{V} \in \mathbb{R}^{I_2 \times I_2 \times I_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a rectangular f-diagonal tensor (see Appendix C, Section C.4.3) and $*$ denotes the t-product (see Appendix C, Section C.4.3).

A few recent works used non-convex functions instead of the nuclear norm as the surrogates of rank for rank minimization. Tomioka and Suzuki [203] proposed a structured version of Schatten norms for tensors. The authors consider the following more general *overlapped* $Sp/q - norm$ defined by: $||\mathcal{X}||_{Sp/q} = (\sum_{n=1}^{N} ||\mathcal{X}^{[n]}||_{Sp}^{q})^{1/q}$, where $||\mathbf{X}||_{Sp} = (\sum_{i=1}^{r} \sigma_i^p)^{1/p}$ is the Schatten p-norm for matrices. When $p \longrightarrow 0$ the minimization is intractable, and when $p = 1$ turns out to be the nuclear norm. For non-convex cases: $0 < p < 1$ [259].

3.3 Experimental results

In order to evaluate the proposed approach, twenty-three state-of-the-art low-rank reconstruction algorithms were selected. These algorithms include thirteen MC and ten TC algorithms, and they are listed in Table 3.3 and in Table 3.4, respectively. The algorithms were grouped into two categories, as well as their main techniques, following the same definition of Zhou et al. [259]. The parameters of each method were tuned for each sequence, so that the estimated background model is the best as possible.

Data set In this chapter, the SBI dataset[7] [136] was chosen for the background initialization task. This dataset contains 14 image sequences and their corresponding ground truth backgrounds. It provides also MATLAB scripts for evaluating background initialization results in terms of six metrics[8]: 1) Average Gray-level Error (AGE), 2) Percentage of Error Pixels (pEPs), 3) Percentage of Clustered Error Pixels (pCEPs), 4) Peak-Signal-to-Noise-Ratio (PSNR), 5) Multi-Scale Structural Similarity Index (MS-SSIM), and 6) Color image Quality Measure (CQM).

Methodology The algorithms are ranked as follows: 1) for each algorithm we calculate its rank position for each metric (Metric Rank); Next, 2) we sum the value of the rank position for each algorithm over the six metrics, and finally, 3) we calculate the rank position over the sum, and we call it as Scene Rank. For the Global Rank, first we sum the Scene Rank for each algorithm, then we calculate its rank position over the sum. MATLAB codes are publicly available at https://github.com/andrewssobral/mctc4bmi

Quantitative analysis Tables 3.5 and 3.6 show the rank of MC and TC algorithms, respectively, over the SBI dataset. Analyzing the scene rank of MC algorithms, LRGeomCG was the top-1 in 9 over 14 scenes, becoming the first algorithm in the Global Rank of its category. For TC algorithms, TMac was the top-1 in 11 over 14 scenes, becoming the first

[7] http://sbmi2015.na.icar.cnr.it/SBIdataset.html

[8] Please, refer to Maddalena and Petrosino [136] for a complete description of each metric.

algorithm in the Global Rank of its category. In order to compare the best MC and the best TC algorithms, Table 3.7 presents the top-5 matrix and top-5 tensor completion algorithms, respectively, over the SBI dataset. As it can be seen, the first four best ranked algorithms (headed by LRGeomCG) are based on the MC approach. This is an interesting factor because usually tensor-based methods are seen to be more robust for multidimensional data completion in comparison to matrix-based methods. However, given that SBI dataset scenes are based on RGB color images, this may not mean that they are multidimensional enough for the power of TC methods. In order to provide more detailed results, Tables 3.8, 3.9, 3.10 and 3.11 present the quantitative results of the top-10 best algorithms over all scenes from the SBI dataset. The results are ordered by the Scene Rank, and the Global Rank shows the best ranked algorithms for all scenes. Finally, Table 3.12 summarizes the top-1 best algorithms for each individual scene. The performance of tensor-based approaches has been highlighted only on two scenes: Candela˙m1.10 by SPC and HallAndMonitor by t-TNN.

Qualitative analysis Figures 3.4 and 3.5 compare the background estimated by the top-10 best ranked low-rank reconstruction algorithms. As it can be seen, almost all methods present similar visual results, except in some particular cases where the IALM method presents some color divergence artifacts. These color artifacts are expected because the matrix completion process is done for each color channel separately. The CQM can penalize such color artifacts, however it is averaged with other five metrics, decreasing its importance. Finally, we verified that for some scenes, in particular for Board, CAVIAR1, and CaVignal (columns 1, 3 and 5 of Figure 3.4) all low-rank reconstruction algorithms failed to remove some artifacts, showing some areas with shadings with different tones.

Table 3.5: Scene rank and global rank of each MC method over SBI dataset.

Method	Scene Rank														Global Rank
	Board	Candela_m1.10	CAVIAR1	CAVIAR2	CaVignal	Foliage	HallAndMonitor	HighwayI	HighwayII	HumanBody2	IBMtest2	PeopleAndFoliage	Snellen	Toscana	
LRGeomCG	3	**1**	2	**1**	**1**	2	**1**	2	**1**	**1**	2	**1**	**1**	**1**	**1**
LMaFit	2	5	**1**	2	2	3	4	4	3	2	**1**	3	3	10	2
RMAMR	4	2	3	4	3	4	3	**1**	4	3	3	5	4	2	2
MC-NMF	6	8	3	2	5	6	6	6	5	5	6	2	5	2	4
IALM	**1**	3	13	13	4	5	2	3	**1**	6	12	6	7	4	5
ScGrassMC	5	10	5	5	6	8	5	8	6	4	7	4	6	7	6
GROUSE	8	4	12	11	7	**1**	7	5	7	8	5	7	2	11	7
GreGoDec	9	7	7	7	10	10	9	9	8	9	8	8	8	5	8
OR1MP	7	11	10	7	8	9	8	7	9	7	4	10	10	8	9
SSGoDec	13	9	9	6	11	11	11	10	10	10	10	12	9	6	10
GoDec	10	6	6	9	9	12	10	11	13	11	9	11	13	9	11
OptSpace	11	13	11	12	11	7	13	13	12	13	13	9	11	12	12
SVP	12	12	8	10	13	13	12	12	11	12	11	13	12	12	13

Table 3.6: Scene rank and global rank of each tensor completion method over SBI dataset.

Method	Scene Rank														Global Rank
	Board	Candela_m1.10	CAVIAR1	CAVIAR2	CaVignal	Foliage	HallAndMonitor	HighwayI	HighwayII	HumanBody2	IBMtest2	PeopleAndFoliage	Snellen	Toscana	
TMac	**1**	5	**1**	**1**	**1**	**1**	3	**1**	**1**	**1**	**1**	**1**	**1**	3	**1**
SPC	2	**1**	2	2	5	3	6	3	3	2	2	2	3	**1**	2
t-SVD	2	3	5	4	4	2	2	2	2	3	3	3	2	4	3
t-TNN	4	4	6	6	6	4	**1**	5	4	5	4	5	4	2	4
FaLRTC	6	2	4	3	3	6	4	4	5	6	5	8	6	5	5
geomCG	4	9	2	4	2	10	9	10	9	4	9	3	9	9	6
BCPF	8	6	8	9	9	8	4	7	6	7	6	7	7	7	7
TenALS	7	7	9	7	8	5	8	6	8	9	8	6	5	6	7
NCPC	9	8	7	8	7	7	7	8	7	8	7	9	7	8	9
HoRPCA-IALM	10	10	10	10	10	9	10	9	10	10	10	10	10	10	10

Table 3.7: Comparison of the top-5 matrix completion with the top-5 tensor completion methods over SBI dataset.

	Method	Scene Rank													
		Board	Candela_m1.10	CAVIAR1	CAVIAR2	CaVignal	Foliage	HallAndMonitor	HighwayI	HighwayII	HumanBody2	IBMtest2	PeopleAndFoliage	Snellen	Toscana	Global Rank
M	**LRGeomCG**	3	5	2	**1**	**1**	**1**	3	2	**1**	**1**	2	**1**	**1**	**1**	**1**
M	LMaFit	2	8	**1**	2	2	2	6	4	3	2	**1**	2	2	10	2
M	RMAMR	4	6	3	4	3	3	4	**1**	4	3	4	5	3	4	3
M	MC-NMF	6	9	4	2	6	6	8	6	6	5	5	3	5	3	4
T	TMac	5	10	5	6	4	5	7	5	5	4	3	4	4	7	4
M	IALM	**1**	7	10	10	5	4	4	3	**1**	6	10	6	6	9	6
T	SPC	7	**1**	6	5	9	8	10	8	8	7	6	7	8	2	7
T	t-SVD	8	3	8	8	8	7	2	7	7	8	7	8	7	5	8
T	t-TNN	9	4	9	8	10	9	**1**	10	9	9	8	9	9	5	9
T	FaLRTC	10	2	7	7	7	10	9	9	10	10	9	10	10	8	10

M Matrix-based completion.
T Tensor-based completion.

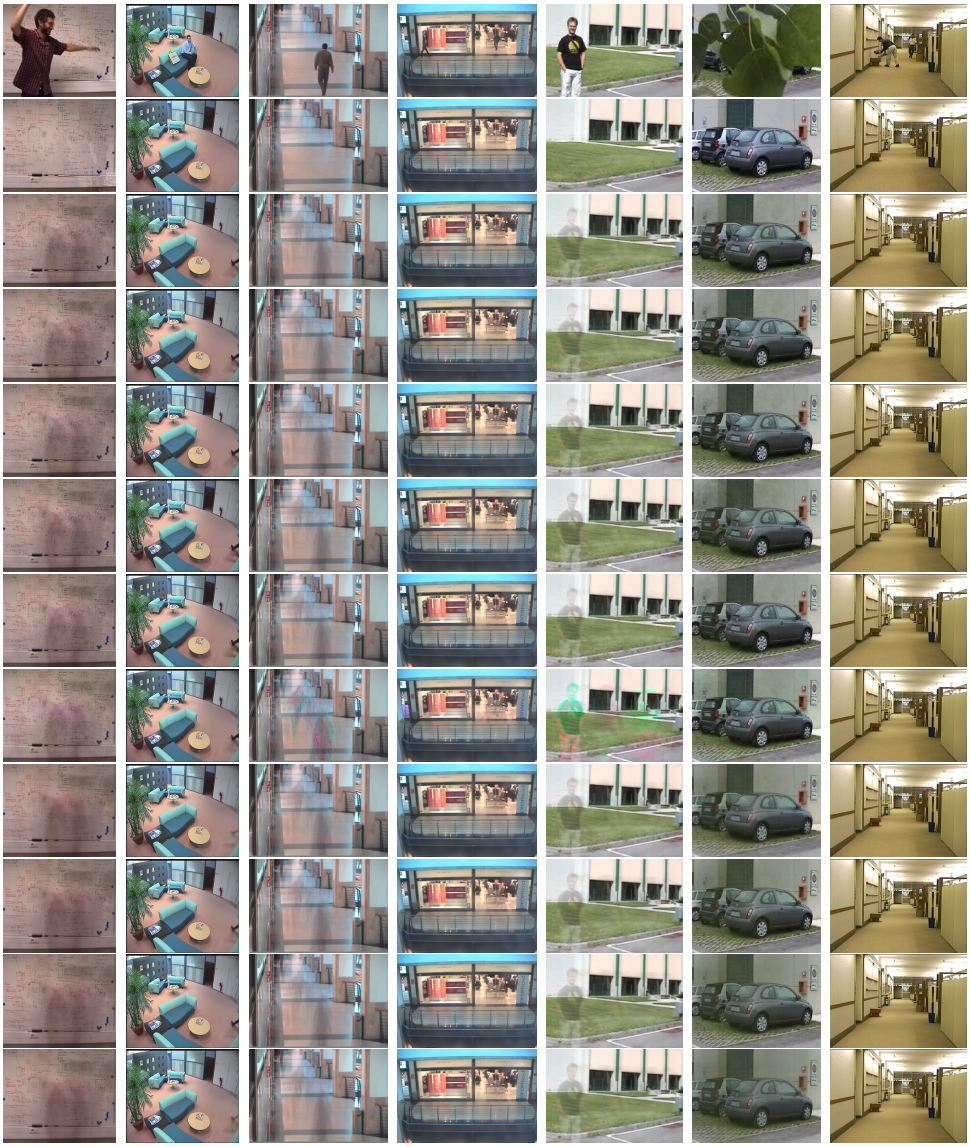


Figure 3.4: Part 1 - Visual comparison for the background model initialization over the first 7 scenes of the SBI dataset. From top to bottom: 1) example of input frame, 2) background model ground truth, and background model results for the top 10 best ranked low-rank recovery algorithms: 3) LRGeomCG, 4) LMaFit, 5) RMAMR, 6) MC-NMF, 7) TMac, 8) IALM, 9) SPC, 10) t-SVD, 11) t-TNN, and 12) FaLRTC.

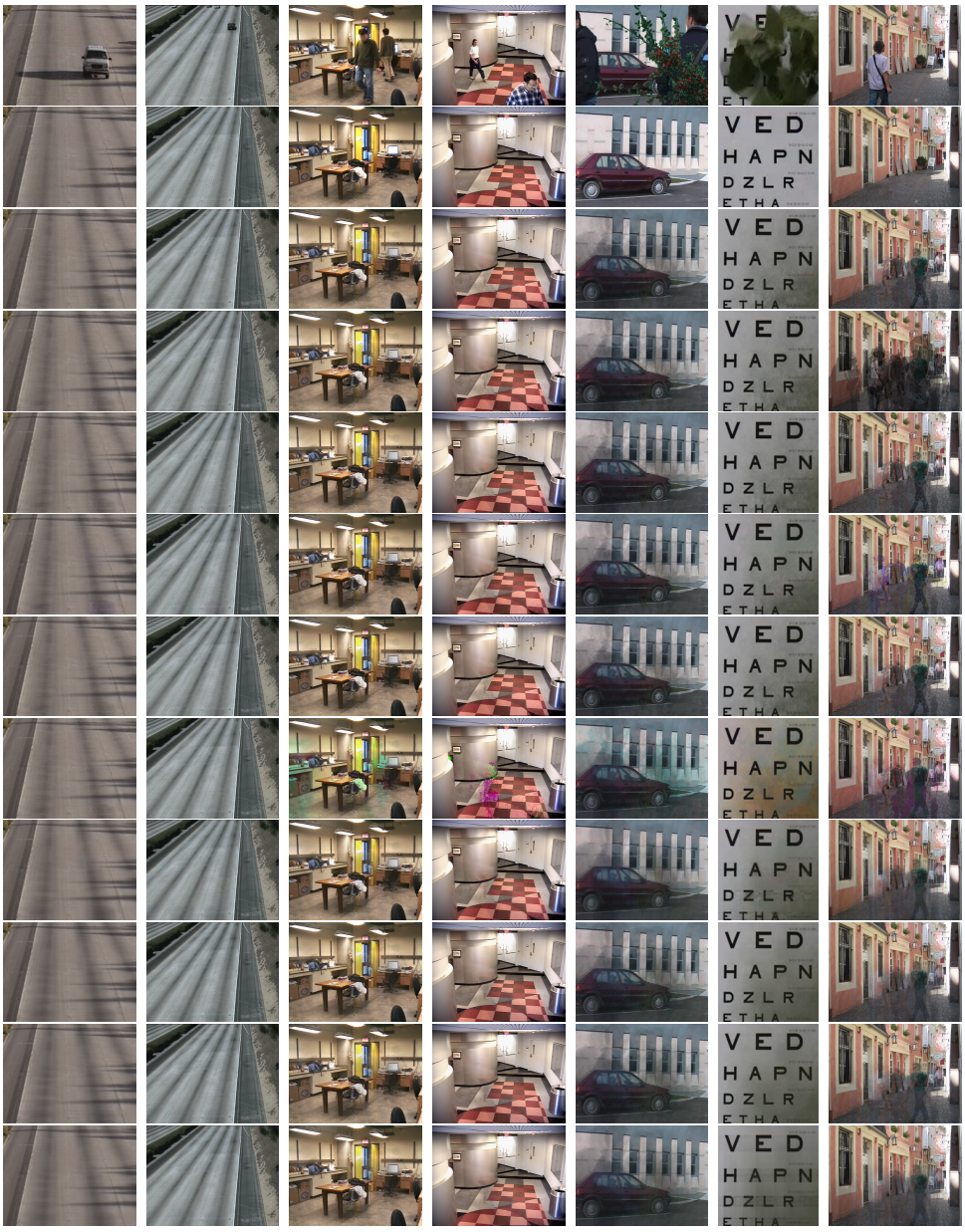


Figure 3.5: Part 2 - Visual comparison for the background model initialization over the last 7 scenes of the SBI dataset. From top to bottom: 1) example of input frame, 2) background model ground truth, and background model results for the top 10 best ranked low-rank recovery algorithms: 3) LRGeomCG, 4) LMaFit, 5) RMAMR, 6) MC-NMF, 7) TMac, 8) IALM, 9) SPC, 10) t-SVD, 11) t-TNN, and 12) FaLRTC.

Table 3.8: Part 1 - Quantitative results of the top-10 low-rank reconstruction algorithms over SBI dataset.

Board

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
M IALM	**16.0145**	**29.0732**	**22.6921**	**0.6704**	**21.2760**	**45.3854**	1
M LMaFit	18.8825	35.9543	27.5213	0.6754	20.1326	43.7285	2
M LRGeomCG	18.8829	35.9543	27.5213	0.6754	20.1325	43.7282	3
M RMAMR	18.9599	36.1311	27.6463	0.6752	20.1081	43.6978	4
T TMac	19.0408	36.2409	27.8293	0.6726	20.0735	43.7571	5
M MC-NMF	20.5415	39.8140	31.5610	0.6382	19.5552	43.3572	6
T SPC	24.2711	46.6494	37.8201	0.6181	18.4161	42.9340	7
T t-SVD	24.6558	47.7927	39.5244	0.6183	18.3139	43.1752	8
T t-TNN	24.8947	48.3994	40.0396	0.6162	18.2481	43.2070	9
T FaLRTC	25.0577	48.6067	40.3171	0.6165	18.1905	43.1227	10

Candela_m1.10

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
T SPC	**1.6942**	**0.2180**	**0.1263**	**0.9930**	**39.4553**	51.9223	1
T FaLRTC	1.8368	0.5455	0.3630	0.9917	37.6151	**51.9974**	2
T t-SVD	1.8697	0.5988	0.4745	0.9918	34.8058	48.0497	3
T t-TNN	1.8944	0.6185	0.4824	0.9916	34.5269	47.1672	4
M LRGeomCG	1.9037	0.6579	0.4991	0.9912	33.8805	45.0354	5
M RMAMR	1.8950	0.6609	0.5090	0.9912	33.7306	44.9021	6
M IALM	1.9144	0.6688	0.5060	0.9911	33.8414	44.9827	7
M LMaFit	1.9680	0.8079	0.6165	0.9903	33.2250	44.5333	8
M MC-NMF	2.0237	0.9746	0.7517	0.9892	32.7856	44.2348	9
T TMac	2.0456	1.0150	0.7842	0.9888	32.5507	43.7920	10

CaVignal

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
M LRGeomCG	**5.4839**	**6.4118**	**3.9890**	**0.9111**	**28.3288**	**52.9853**	1
M LMaFit	5.4853	6.4154	**3.9890**	**0.9111**	28.3273	52.9850	2
M RMAMR	5.4959	6.4301	**3.9890**	**0.9111**	28.3100	53.0123	3
T TMac	5.4979	6.5074	4.0625	0.9109	28.2877	53.0188	4
M IALM	5.9824	7.1875	3.8272	0.9002	27.7931	50.7009	5
M MC-NMF	5.9749	8.1949	5.4890	0.8997	27.4603	52.8197	6
T FaLRTC	7.1125	8.3309	5.4081	0.8876	26.9316	47.6830	7
T t-SVD	7.1949	8.3676	5.4669	0.8860	26.8741	47.8262	8
T SPC	7.1215	8.6176	5.6581	0.8846	26.8208	51.0394	9
T t-TNN	7.2519	8.4743	5.5184	0.8851	26.8285	47.4856	10

Foliage

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
M LRGeomCG	**11.6932**	**20.8924**	**14.9861**	**0.9535**	**23.9826**	39.0643	1
M LMaFit	11.7062	20.9271	15.0417	**0.9535**	23.9744	39.0737	2
M RMAMR	11.8031	21.1458	15.2222	0.9532	23.9134	**39.1203**	3
M IALM	11.8963	21.5417	15.6215	0.9528	23.8960	38.4825	4
T TMac	12.0335	21.9826	15.9861	0.9498	23.7072	38.7549	5
M MC-NMF	15.1040	25.4340	19.5069	0.9114	21.4930	36.5547	6
T t-SVD	16.2432	29.1424	22.2674	0.9033	21.4422	33.9587	7
T SPC	17.1535	33.2743	22.7951	0.8937	21.0282	32.9964	8
T t-TNN	18.5811	34.3993	24.7222	0.8734	20.5579	32.2173	9
T FaLRTC	22.1739	45.7639	31.2569	0.8197	19.2060	30.8085	10

M Matrix-based completion.
T Tensor-based completion.

Table 3.9: Part 2 - Quantitative results of the top-10 low-rank reconstruction algorithms over SBI dataset.

CAVIAR1

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕄 **LMaFit**	**5.6735**	**6.6274**	**5.4830**	**0.9120**	**28.1792**	**51.1680**	1
𝕄 LRGeomCG	**5.6735**	**6.6274**	**5.4830**	**0.9120**	28.1790	**51.1680**	2
𝕄 RMAMR	5.6641	6.6508	5.5094	0.9121	28.1736	51.1897	3
𝕄 MC-NMF	5.6737	6.6284	5.4830	0.9120	28.1787	51.1680	4
𝕋 TMac	5.6945	6.7017	5.5593	0.9116	28.1425	51.1681	5
𝕋 SPC	5.8531	7.2815	6.0628	0.9093	27.9196	51.1901	6
𝕋 FaLRTC	5.8699	7.3008	6.0801	0.9093	27.9042	51.1497	7
𝕋 t-SVD	5.8736	7.3222	6.0801	0.9092	27.9014	51.1434	8
𝕋 t-TNN	5.8792	7.3222	6.0801	0.9092	27.8956	51.1384	9
𝕄 IALM	6.7411	10.0464	7.8440	0.8802	26.5992	50.7632	10

CAVIAR2

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕄 **LRGeomCG**	**1.1822**	**0.3265**	**0.1038**	**0.9971**	**39.8982**	**59.1772**	1
𝕄 LMaFit	1.1823	**0.3265**	**0.1038**	**0.9971**	39.8978	**59.1772**	2
𝕄 MC-NMF	1.1823	**0.3265**	**0.1038**	**0.9971**	39.8978	**59.1772**	2
𝕄 RMAMR	1.1925	0.3276	0.1048	**0.9971**	39.8581	59.1748	4
𝕋 SPC	1.2136	0.3845	0.1312	0.9968	39.5137	59.1832	5
𝕋 TMac	1.1877	0.3286	0.1058	0.9970	39.8569	59.1736	6
𝕋 FaLRTC	1.2923	0.3845	0.1343	0.9967	39.2267	59.1819	7
𝕋 t-SVD	1.2896	0.3855	0.1363	0.9967	39.2218	59.1766	8
𝕋 t-TNN	1.3058	0.3866	0.1363	0.9967	39.1549	59.1797	8
𝕄 IALM	1.4858	0.6460	0.2431	0.9945	36.7150	58.4525	10

HallAndMonitor

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕋 **t-TNN**	**2.0417**	**0.1716**	**0.0000**	**0.9938**	**38.2894**	**46.4244**	1
𝕋 t-SVD	2.0469	0.1847	**0.0000**	**0.9938**	38.2084	46.3998	2
𝕄 LRGeomCG	2.0476	0.2237	**0.0000**	**0.9938**	38.0243	46.3813	3
𝕄 IALM	2.0476	0.2237	**0.0000**	**0.9938**	38.0242	46.3813	4
𝕄 RMAMR	2.0417	0.2261	**0.0000**	**0.9938**	38.0214	46.3837	4
𝕄 LMaFit	2.0498	0.2249	**0.0000**	**0.9938**	37.9193	46.3371	6
𝕋 TMac	2.0599	0.2415	**0.0000**	0.9937	37.5664	46.2214	7
𝕄 MC-NMF	2.0640	0.2415	**0.0000**	0.9937	37.4073	46.1720	8
𝕋 FaLRTC	2.2531	0.4025	0.0012	0.9925	36.4659	44.1228	9
𝕋 SPC	2.2763	0.5303	0.0024	0.9927	35.8033	43.8510	10

HighwayI

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕄 **RMAMR**	**2.6453**	0.1992	**0.0130**	**0.9779**	36.2748	**58.2536**	1
𝕄 LRGeomCG	2.6535	0.2018	**0.0130**	**0.9779**	**36.2808**	58.2359	2
𝕄 IALM	2.6598	**0.1953**	**0.0130**	0.9777	36.2798	58.2441	3
𝕄 LMaFit	2.6562	0.2018	**0.0130**	**0.9779**	36.2687	58.2344	4
𝕋 TMac	2.6788	0.1992	**0.0130**	0.9777	36.1917	58.2417	5
𝕄 MC-NMF	2.9940	0.5677	0.1328	0.9726	34.9387	57.8914	6
𝕋 t-SVD	4.2271	0.8138	0.3516	0.9643	32.4894	58.0333	7
𝕋 SPC	3.8228	1.2982	0.8568	0.9655	33.0111	57.8709	8
𝕋 FaLRTC	4.8294	0.8294	0.3516	0.9624	31.7239	57.8150	9
𝕋 t-TNN	4.7145	1.2474	0.6849	0.9611	31.6223	57.9345	10

𝕄 Matrix-based completion.
𝕋 Tensor-based completion.

Table 3.10: Part 3 - Quantitative results of the top-10 low-rank reconstruction algorithms over SBI dataset.

HighwayII

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
IALM	**2.7526**	**0.3555**	**0.0026**	**0.9908**	**35.3406**	**46.3161**	1
LRGeomCG	**2.7526**	**0.3555**	**0.0026**	**0.9908**	**35.3406**	**46.3161**	1
LMaFit	2.7528	**0.3555**	**0.0026**	**0.9908**	35.3361	46.3122	3
RMAMR	2.7687	**0.3555**	**0.0026**	**0.9908**	35.3015	46.3119	4
TMac	2.7697	0.3763	0.0039	0.9908	35.2287	46.2181	5
MC-NMF	2.8791	0.3672	0.0026	0.9901	34.9279	46.1025	6
t-SVD	3.8271	0.4596	0.0299	0.9864	33.0601	46.4679	7
SPC	3.4289	0.5091	0.0091	0.9878	33.5406	46.1876	8
t-TNN	4.1262	0.5013	0.0299	0.9850	32.4600	46.2045	9
FaLRTC	4.1368	0.5286	0.0247	0.9850	32.4087	46.2637	10

HumanBody2

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
LRGeomCG	**5.7621**	**4.6497**	**2.4414**	**0.9788**	**28.7047**	40.1984	1
LMaFit	5.7627	4.6510	**2.4414**	**0.9788**	28.7017	40.1985	2
RMAMR	5.7661	4.6628	2.4505	0.9787	28.6899	40.2114	3
TMac	5.8044	4.7292	2.4583	0.9786	28.6019	**40.2122**	4
MC-NMF	6.1841	5.3919	2.9102	0.9760	28.1319	39.7334	5
IALM	6.5515	5.6836	2.7969	0.9730	27.6232	39.1328	6
SPC	6.8470	6.2174	3.3984	0.9715	27.2413	38.4319	7
t-SVD	7.0378	6.2409	3.4922	0.9688	27.1468	38.2726	8
t-TNN	7.2657	6.5391	3.6341	0.9667	26.9194	37.8676	9
FaLRTC	7.3432	6.6953	3.7344	0.9670	26.8229	37.5329	10

PeopleAndFoliage

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
LRGeomCG	**29.2393**	57.7812	48.2135	0.8332	**16.8399**	**32.1823**	1
LMaFit	29.2520	57.8125	48.2539	0.8331	16.8367	32.1796	2
MC-NMF	29.5094	57.4115	**47.6497**	**0.8346**	16.7286	32.1034	3
TMac	29.4402	**57.1888**	47.6719	0.8233	16.6676	32.1267	4
RMAMR	29.3027	57.9049	48.3906	0.8329	16.8253	32.1624	5
IALM	30.6975	58.2839	49.0560	0.8011	16.2927	31.7307	6
SPC	32.2676	59.8932	51.2461	0.7850	15.8266	31.8834	7
t-SVD	32.7286	60.1901	51.9622	0.7755	15.7204	31.7717	8
t-TNN	33.1854	60.4258	52.3255	0.7724	15.6211	31.7342	9
FaLRTC	33.7695	61.0026	52.9258	0.7620	15.4770	31.6863	10

Snellen

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
LRGeomCG	**24.4846**	**50.4340**	**42.8337**	**0.9250**	**18.6585**	42.1307	1
LMaFit	24.5473	50.6318	42.9977	0.9249	18.6457	42.1328	2
RMAMR	24.6804	50.9790	43.3449	0.9245	18.6077	**42.1378**	3
TMac	24.8743	51.9965	44.3528	0.9206	18.5311	41.8552	4
MC-NMF	26.8535	58.6082	51.6541	0.9170	17.9515	41.5184	5
IALM	28.5543	59.2930	52.1943	0.8821	17.3378	39.2210	6
t-SVD	28.9166	58.9169	53.2407	0.8854	17.2301	39.8213	7
SPC	29.4769	59.9489	53.0478	0.8779	17.1026	37.3224	8
t-TNN	31.3521	65.6877	59.1001	0.8722	16.8427	38.3602	9
FaLRTC	36.3661	74.2911	66.3725	0.8258	15.7664	35.3544	10

Matrix-based completion.
Tensor-based completion.

Table 3.11: Part 4 - Quantitative results of the top-10 low-rank reconstruction algorithms over SBI dataset.

IBMtest2

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕄 **LMaFit**	**3.6413**	**1.4544**	**0.6081**	**0.9868**	**32.8931**	44.9527	**1**
𝕄 LRGeomCG	**3.6413**	**1.4544**	**0.6081**	**0.9868**	32.8930	44.9516	2
𝕋 TMac	3.6575	1.4831	0.6289	**0.9868**	32.8424	**44.9660**	3
𝕄 RMAMR	3.6715	1.4674	0.6146	**0.9868**	32.8495	44.9352	4
𝕄 MC-NMF	4.0120	2.4844	1.4427	0.9814	31.3615	43.9484	5
𝕋 SPC	4.3337	2.7161	1.6836	0.9810	30.9463	42.2799	6
𝕋 t-SVD	4.5889	2.8060	1.9362	0.9780	30.4336	42.3460	7
𝕋 t-TNN	4.7017	2.9089	2.0495	0.9772	30.2330	41.8205	8
𝕋 FaLRTC	4.7449	3.0091	2.0404	0.9776	30.1724	41.0093	9
𝕄 IALM	5.7583	5.2370	2.3164	0.9462	26.1977	38.9603	10

Toscana

Method	AGE	pEPs	pCEPS	MSSSIM	PSNR	CQM	Scene Rank
𝕄 **LRGeomCG**	**7.3009**	11.8569	**8.4394**	0.8959	24.2914	36.3206	**1**
𝕋 SPC	8.1087	12.9096	8.8235	0.9077	**24.5368**	**38.5918**	2
𝕄 MC-NMF	7.6612	**11.5710**	7.6317	0.8911	24.3671	36.2805	3
𝕄 RMAMR	7.3381	11.9000	8.4723	0.8959	24.2800	36.3043	4
𝕋 t-SVD	8.4022	14.1319	10.2058	0.9096	24.3111	38.1849	5
𝕋 t-TNN	8.6326	13.9075	9.7881	**0.9101**	24.2766	38.2244	5
𝕋 TMac	7.3742	12.0163	8.5815	0.8958	24.2609	36.2690	7
𝕋 FaLRTC	9.4138	15.6027	11.3146	0.9051	23.8897	38.1260	8
𝕄 IALM	9.5171	16.8137	13.3448	0.8887	23.0985	34.9645	9
𝕄 LMaFit	23.1736	42.7019	38.5535	0.7486	16.8984	25.6812	10

𝕄 Matrix-based completion.
𝕋 Tensor-based completion.

Global rank over all scenes

Method	Global rank
𝕄 **LRGeomCG**	**1**
𝕄 LMaFit	2
𝕄 RMAMR	3
𝕄 MC-NMF	4
𝕋 TMac	4
𝕄 IALM	6
𝕋 SPC	7
𝕋 t-SVD	8
𝕋 t-TNN	9
𝕋 FaLRTC	10

Table 3.12: Summary of the top-1 best algorithms for each scene. The columns Top-1 MC and Top-1 TC show the best algorithms among matrix and tensor completion methods, respectively. The last column highlights the winner algorithm among the top-10 best ranked low-rank recovery methods.

Scenes	Top-1 MC	Top-1 TC	Scene Top-1
Board	IALM	TMac	[M] IALM
Candela_m1.10	LRGeomCG	SPC	[T] SPC
CAVIAR1	LMaFit	TMac	[M] LMaFit
CAVIAR2	LRGeomCG	TMac	[M] LRGeomCG
CaVignal	LRGeomCG	TMac	[M] LRGeomCG
Foliage	GROUSE	TMac	[M] LRGeomCG
HallAndMonitor	LRGeomCG	t-TNN	[T] t-TNN
HighwayI	RMAMR	TMac	[M] RMAMR
HighwayII	IALM	TMac	[M] IALM
HumanBody2	LRGeomCG	TMac	[M] LRGeomCG
IBMtest2	LMaFit	TMac	[M] LMaFit
PeopleAndFoliage	LRGeomCG	TMac	[M] LRGeomCG
Snellen	LRGeomCG	TMac	[M] LRGeomCG
Toscana	LRGeomCG	SPC	[M] LRGeomCG

[M] Matrix-based completion.
[T] Tensor-based completion.

3.4 Conclusion

In this chapter, we have formulated the background initialization problem as a matrix or tensor completion task, and evaluated twenty-three recent low-rank recovery algorithms. The key idea is to first eliminate the redundant frames in a video, and consider their moving regions as non-observed values. This approach results in a data completion problem, which can be represented by a matrix or a tensor with missing entries. We show that the background model can be recovered even with partially observed data. The experimental results on the SBI dataset show the comparative evaluation of recent methods for matrix and tensor completion, and highlight the good performance of LRGeomCG method over its direct competitors. Finally, we note that matrix-based completion methods show an attractive potential for background modeling initialization in video surveillance. Moreover, in Bouwmans et al. [25], the proposed approach was classified in a new category of background initialization methods, named Missing Data Reconstruction methods. Future research may concern to evaluate incremental and real-time approaches of low-rank reconstruction algorithms for the background model initialization of streaming videos.

Chapter 4

Improving foreground detection by double-constrained robust PCA

This chapter investigates the problem of moving object detection in maritime environment for automated video-surveillance applications. To cope with this particular situation, a double-constrained robust principal component analysis algorithm, named SCM-RPCA (Shape and Confidence Map-based RPCA), is proposed. The work presented in this chapter is based on our publication (IEEE AVSS, 2015, [179]), and the related source code can be found in the SCM-RPCA website[1].

4.1 Introduction

As outlined in Chapter 2, the recent advances in low-rank and sparse decomposition offer a suitable framework for background modeling due to the high correlation between frames. However, the Robust Principal Component Analysis (RPCA) solved via Principal Component Pursuit (PCP) is limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse (see Section 2.2.2). However, the observations in real applications are often corrupted by noise that affects every entry of the data matrix. Therefore, Zhou et al. [260] proposed a stable PCP (SPCP) that guarantees stable and accurate recovery in the presence of entry-wise noise (see Section 2.2.3). SPCP assumes that the observation matrix $\mathbf{A}$ is represented as $\mathbf{A} = \mathbf{L} + \mathbf{S} + \mathbf{E}$ (also named as three-term decomposition), where $\mathbf{L}$ is a low-rank matrix, $\mathbf{S}$ is constrained to be a sparse matrix, and $\mathbf{E}$ is a noise term. To recover $\mathbf{L}$, $\mathbf{S}$ and $\mathbf{E}$, Zhou et al. [260] proposed to solve the following optimization problem, as a relaxed version of PCP: $\underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}} \; ||\mathbf{L}||_* + \lambda_1 ||\mathbf{S}||_1 + \lambda_2 ||\mathbf{E}||_F^2, \; \text{s.t.} \; \mathbf{A} = \mathbf{L} + \mathbf{S} + \mathbf{E}$, where $\lambda_1 > 0$ and $\lambda_2 > 0$ are arbitrary weighting parameters. This decomposition is called "stable" decomposition as it separates the outliers in $\mathbf{S}$ and the noise in $\mathbf{E}$.

[1] SCM-RPCA: https://sites.google.com/site/scmrpca/

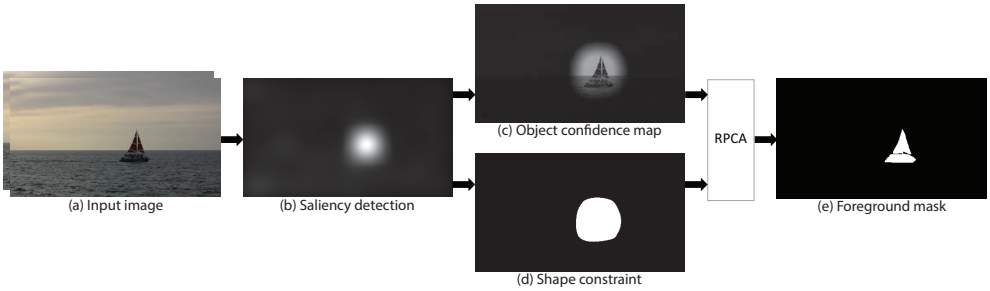


Figure 4.1: Block diagram of the proposed approach. Given an input image (a), a saliency detector is applied (b). Next, the confidence map (c) is built by normalizing the saliency map, while the shape constraint (d) is built by thresholding this one, and (e) the foreground mask obtained by thresholding the RPCA sparse component.

Maritime surveillance represents a challenging scenario due to the different background dynamics of the observed scenes, such as moving water, waves, etc [20]. Indeed, the motion of the objects of interest (i.e. ships or boats) can be mixed with the dynamic behavior of the background (non-regular patterns). Many algorithms have been designed to perform foreground detection, see surveys [24, 28, 183], but only a few of them have been designed for maritime scenes. Some related work can be found in Bloisi et al. [20]. The authors propose a multimodal approach for BS to deal with the water background. In addition, Liu et al. [125] propose an iterative approach for ship target segmentation in infrared images based on multiple features. However, the recent research on subspace estimation by sparse representation and rank minimization shows an interesting framework to separate moving objects from the background in videos. The background sequence is modeled by the low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers.

In scenes where the background is very dynamic (i.e. sea waves in maritime surveillance [20]), the motion of the objects of interest (i.e. boats) will be mixed with the dynamic behavior of the background (i.e. waves). SPCP-based methods try to deal with this problem under the term where the multi-modality of the background (i.e. waves) is considered as noise component ($\mathbf{E}$), while the moving objects (i.e. boats) are considered as sparse component ($\mathbf{S}$). The low-rank component ($\mathbf{L}$) represents the static part of the background.

In this chapter, a double-constrained RPCA, named SCM-RPCA (Shape and Confidence Map-based RPCA), is proposed to improve foreground detection in dynamic scenes. The sparse component is constrained by shape and confidence maps, both extracted from spatial saliency maps. One advantage of the SCM-RPCA in relation to its direct competitors, is the possibility of combining two types of source, which may come from: spatial, temporal, and spatio-temporal information; however, here we focus only on spatial saliency maps. Our motivation is to study how it improves RPCA in the task of foreground detection in maritime scenes. Fig. 4.1 highlights our proposed approach. Given an input image (a), a saliency detector is applied (b). Next, the confidence map (c) is built by normalizing the saliency map, while the shape constraint (d) is built by thresholding the saliency map, and (e) the foreground mask is obtained by thresholding the RPCA sparse component.

4.2 Related work

In the literature, there are several modifications which concern the original SPCP. Some authors [152, 239, 241] added constraint in the sparse term in order to improve the foreground detection. First, Oreifej et al. [152] used a turbulence model to enforce an additional constraint on the rank minimization. The authors quantify the scene's motion in terms of the motion of the particles, which are driven by dense optical flow. The obtained confidence map (a real-valued matrix) provides a rough prior knowledge of the moving objects' locations, which can be incorporated into the matrix optimization problem. Subsequently, Yang et al. [239] proposed a motion-assisted matrix restoration (MAMR) model for foreground-background separation. Thus, a dense motion field is estimated for each frame by dense optical flow, and mapped into a weighting matrix, which indicates the likelihood of each pixel belonging to the background. By incorporating this information, areas dominated by slowly-moving objects are suppressed, while the background that appears at only a few frames has more chances to be recovered in the foreground detection results. In addition, Ye et al. [241] extended MAMR (RMAMR) (also adopted in Chapter 3), which is robust to noise for practical applications.

4.3 Proposed method

In this chapter, we propose to combine some ideas proposed by Oreifej et al. [152] and Ye et al. [241]. The weighting matrix proposed by Ye et al. [241] can be used as a shape constraint (or region constraint), while the confidence map proposed by Oreifej et al. [152] reinforces the pixels belonging to the moving objects. A modified version of the original 3WD method proposed by Oreifej et al. [152] was implemented adding the shape constraint as done in RMAMR. Part of the reason we chose to modify the 3WD instead of RMAMR is it robustness to deal with the multimodality of the background. The second contribution of this chapter refers to the way of building the shape constraint and confidence map. Instead of using dense optical flow (temporal descriptor) as a preliminary step, we suggest using a saliency detector (spatial descriptor). In some cases where a) the object of interest moves very slowly (i.e long distance boats) or b) the background is very dynamic (i.e boats in the sea), the optical flow may not be enough to ensure the object detection. In addition, computing the dense optical flow requests high computational cost, while computing the saliency map is commonly much faster. Several saliency detection methods have been proposed in the literature [22]. In this chapter, the BMS[2] method proposed by Zhang and Sclaroff [247, 248] was selected, due to its speed performance and accuracy results.

Consider a sequence of n gray-scale images (frames) $\mathbf{F}_1 \dots \mathbf{F}_n$ captured from a static camera, that is, $\mathbf{F} \in \mathbb{R}^{I_1 \times I_2}$ where I_1 and I_2 denotes the frame resolution (rows by columns). All frames are vectorized into a observation matrix $\mathbf{A} = [vec(\mathbf{F}_1) \dots vec(\mathbf{F}_n)]$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $m = (I_1 \times I_2)$. The decomposition is formulated as:

$$\underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}} \; ||\mathbf{L}||_* + \lambda_1 ||\mathbf{\Pi}(\mathbf{S})||_1 + \lambda_2 ||\mathbf{E}||_F^2, \text{ s.t. } \mathbf{A} = \mathbf{L} + \mathbf{W} \circ \mathbf{S} + \mathbf{E} \tag{4.1}$$

[2] http://cs-people.bu.edu/jmzhang/BMS/BMS.html

Author(s)	Minimization	
Oreifej et al. (2013) [152]	$\underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}}$	$\|\|\mathbf{L}\|\|_*+\lambda_1\|\|\mathbf{\Pi}(\mathbf{S})\|\|_1+\lambda_2\|\|\mathbf{E}\|\|_F^2$
	subject to	$\mathbf{A}=\mathbf{L}+\mathbf{S}+\mathbf{E}$
Ye et al. (2015) [241]	$\underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}}$	$\|\|\mathbf{L}\|\|_*+\lambda_1\|\|\mathbf{S}\|\|_1+\lambda_2\|\|\mathbf{E}\|\|_F^2$
	subject to	$\mathbf{W}\circ\mathbf{A}=\mathbf{W}\circ(\mathbf{L}+\mathbf{S}+\mathbf{E})$
SCM-RPCA (proposed)	$\underset{\mathbf{L},\mathbf{S},\mathbf{E}}{\text{minimize}}$	$\|\|\mathbf{L}\|\|_*+\lambda_1\|\|\mathbf{\Pi}(\mathbf{S})\|\|_1+\lambda_2\|\|\mathbf{E}\|\|_F^2$
	subject to	$\mathbf{A}=\mathbf{L}+\mathbf{W}\circ\mathbf{S}+\mathbf{E}$

Table 4.1: Comparison of the proposed method and related works.

where $\mathbf{\Pi}\in\mathbb{R}^{m\times n}$ and $\mathbf{W}\in[0,1]^{m\times n}$ are the confidence map and shape constraint (binary map), respectively, and "$\circ$" denotes element-wise multiplication of two matrices. As explained previously, the confidence map $\mathbf{\Pi}$ reinforces the pixels belonging to the moving objects and the shape constraint $\mathbf{W}$ defines the region of interest. Table 4.1 compares the proposed method with those by Oreifej et al. [152] and Ye et al. [241]. These minimization problems are convex and can be solved by the Alternating Direction Method (ADM) under the Augmented Lagrangian Multiplier (ALM) framework [118].

4.3.1 Double-constrained robust PCA

To solve the problem in Equation (4.1), the ALM [118] is used. The ALM framework converts the constrained optimization problem in (4.1) to the minimization of the augmented Lagrange function:

$$\begin{aligned}\Gamma(\mathbf{L},\mathbf{S},\mathbf{E},\mathbf{Y}) &= ||\mathbf{L}||_*+\lambda_1||\mathbf{\Pi}(\mathbf{S})||_1+\lambda_2||\mathbf{E}||_F^2 \\ &+<\mathbf{Y},\mathbf{A}-\mathbf{L}-\mathbf{W}\circ\mathbf{S}-\mathbf{E})>+\frac{\beta}{2}||(\mathbf{A}-\mathbf{L}-\mathbf{W}\circ\mathbf{S}-\mathbf{E})||_F^2\end{aligned} \tag{4.2}$$

where $\mathbf{Y}\in\mathbb{R}^{m\times n}$ is a Lagrange multiplier matrix, $\beta>0$ is the penalty parameter for the violation of the linear constraint, and $<,>$ denotes the matrix inner product. Next, the ADM is used to update $\mathbf{L}$, $\mathbf{S}$, $\mathbf{E}$ and $\mathbf{Y}$ alternatively for each iteration t:

$$\begin{aligned}\mathbf{L}_{t+1} &= \arg\min_{\mathbf{L}}\Gamma(\mathbf{L},\mathbf{S}_t,\mathbf{E}_t,\mathbf{Y}_t), \\ \mathbf{S}_{t+1} &= \arg\min_{\mathbf{S}}\Gamma(\mathbf{L}_{t+1},\mathbf{S},\mathbf{E}_t,\mathbf{Y}_t), \\ \mathbf{E}_{t+1} &= \arg\min_{\mathbf{E}}\Gamma(\mathbf{L}_{t+1},\mathbf{S}_{t+1},\mathbf{E},\mathbf{Y}_t), \\ \mathbf{Z} &= (\mathbf{A}_{t+1}-\mathbf{L}_{t+1}-\mathbf{S}_{t+1}-\mathbf{E}_{t+1}), \\ \mathbf{Y}_{t+1} &= \mathbf{Y}_t+\beta_t\mathbf{Z}\end{aligned} \tag{4.3}$$

Algorithm 1 Algorithm for solving SCM-RPCA.

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$ (observation), $\mathbf{\Pi} \in \mathbb{R}^{m \times n}$ (confidence map), $\mathbf{W} \in [0,1]^{m \times n}$ (shape constraint), t_{max} max # of iterations, and ϵ error tolerance.
$t = 0$
while $(||\mathbf{Z}||_F / ||\mathbf{A}||_F) > \epsilon$ or $t < t_{max}$ **do**
$\mathbf{\Upsilon} = \beta_t^{-1} \mathbf{Y}_t$
$\mathbf{U}\mathbf{R}\mathbf{V}^{\mathbf{T}} = svd(\mathbf{A} - \mathbf{L}_t - \mathbf{E}_t + \mathbf{\Upsilon})$
$\mathbf{L}_{t+1} = \mathbf{U} s_{(1/\beta_t)}(\mathbf{R}) \mathbf{V}^{\mathbf{T}}$
$\mathbf{S}_{t+1} = \mathbf{W} \circ s_{(\lambda/\beta_t \mathbf{\Pi})}(\mathbf{A} - \mathbf{L}_{t+1} - \mathbf{E}_t + \mathbf{\Upsilon})$
$\kappa = (1 + \frac{2\lambda_2}{\beta_t})^{-1}$
$\mathbf{E}_{t+1} = \kappa(\mathbf{A} - \mathbf{L}_{t+1} - \mathbf{S}_{t+1} + \mathbf{\Upsilon})$
$\mathbf{Z} = \mathbf{A}_{t+1} - \mathbf{L}_{t+1} - \mathbf{S}_{t+1} - \mathbf{E}_{t+1}$
$\mathbf{Y}_{t+1} = \mathbf{Y}_t + \beta_t \mathbf{Z}$
$\beta_{t+1} = \rho \beta_t$
$t = t + 1$
end while
return $\mathbf{L} \in \mathbb{R}^{m \times n}$ (background), $\mathbf{S} \in \mathbb{R}^{m \times n}$ (foreground), and $\mathbf{E} \in \mathbb{R}^{m \times n}$ (noise).

where $\mathbf{Z} \in \mathbb{R}^{m \times n}$ is the residual. Then, a closed form solution for each of the minimization problems can be defined by:

$$
\begin{aligned}
&\mathbf{\Upsilon} = \beta_t^{-1} \mathbf{Y}_t, \\
&\mathbf{U}\mathbf{R}\mathbf{V}^T = svd(\mathbf{A} - \mathbf{L}_t - \mathbf{E}_t + \mathbf{\Upsilon}), \\
&\mathbf{L}_{t+1} = \mathbf{U} s_{(1/\beta_t)}(\mathbf{R}) \mathbf{V}^T, \\
&\mathbf{S}_{t+1} = \mathbf{W} \circ s_{(\lambda/\beta_t \mathbf{\Pi})}(\mathbf{A} - \mathbf{L}_{t+1} - \mathbf{E}_t + \mathbf{\Upsilon}), \\
&\kappa = (1 + \frac{2\lambda_2}{\beta_t})^{-1}, \\
&\mathbf{E}_{t+1} = \kappa(\mathbf{A} - \mathbf{L}_{t+1} - \mathbf{S}_{t+1} + \mathbf{\Upsilon})
\end{aligned}
\tag{4.4}
$$

where $svd(.)$ denotes a full singular value decomposition, and $s_{(.)}(.)$ is the soft thresholding operator defined by:

$$
s_{(\alpha)}(\mathbf{X}) = sign(\mathbf{X}) max(abs(\mathbf{X}) - \alpha, 0) \tag{4.5}
$$

and it is applied to a matrix $\mathbf{X}$ in an element-wise manner. The main steps of the proposed algorithm are shown in Algorithm 1. Usually the convergence is done when

$$
(||\mathbf{Z}||_F / ||\mathbf{A}||_F) \leq \epsilon, \tag{4.6}
$$

where ϵ is the error tolerance, or when the # of iterations is reached ($t = t_{max}$). The parameters λ and λ_2 are scalars and define the weighting parameter for the sparse and noise component, respectively, and ρ is a constant scalar and growth factor for the β parameter.

Oreifej et al. [152] shows when β_t is a monotonically increasing positive sequence, the iterations converge to the optimal solution of problem 4.1. Here, λ, λ_2, ρ, and β_0 were defined empirically as 2, $1/||\mathbf{A}||_2$, 1.25, and $5/\sqrt{m}$, respectively.

4.3.2 Definition of shape and confidence map

In this chapter, both the confidence map $\mathbf{\Pi}$ and the shape constraint $\mathbf{W}$ are constructed from spatial information given by saliency maps instead of optical flow, as proposed originally by Oreifej et al. [152] and Ye et al. [241]. Consider a sequence of n saliency maps denoted by $\mathbf{M}_1, \ldots, \mathbf{M}_n$ where $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$, so:

$$\mathbf{\Pi} = [vec(norm(\mathbf{M}_1)) \ldots vec(norm(\mathbf{M}_n))] \tag{4.7a}$$

$$\mathbf{W} = [vec(thresh(\mathbf{M}_1)) \ldots vec(thresh(\mathbf{M}_n))] \tag{4.7b}$$

where $norm(.)$ denotes the min-max normalization, scaling all entries of $\mathbf{M}$ between 0 and 1, as defined in Chapter 3, Equation 3.3. Subsequently, $thresh(.)$ denotes the thresholding function defined as:

$$thresh(\mathbf{M}) = \begin{cases} 1 & \text{if } (0.5\mathbf{M})^2 < \mu \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where $\mu = 0.5\eta(std(vec(\mathbf{M})))^2$, and $std(.)$ denotes the standard deviation of a data vector. Here, η was chosen experimentally and defined as 10.

4.4 Experimental results

In order to evaluate the performance of the proposed method for background subtraction, four videos extracted from the UCSD Background Subtraction Dataset[3] proposed by Mahadevan and Vasconcelos [137] and three videos from MarDT dataset[4] proposed by Bloisi et al. [19] were selected. The UCSD and MarDT datasets consist of 18 and 28 video sequences, respectively, both acquired from stationary and moving cameras; here, we have selected only the four sequences from UCSD and three sequences from MarDT, all sequences coming from stationary cameras.

We have compared the SCM-RPCA with its direct competitors: the original PCP proposed by Candès et al. [38], the stable PCP proposed by Aravkin et al. [10], the 3WD proposed by Oreifej et al. [152], and the RMAMR proposed by Ye et al. [241]. Note that the PCP and stable PCP are not constrained, while 3WD and RMAMR are single-constrained RPCA. It is important to note that for all constrained RPCA methods here evaluated have

[3] http://www.svcl.ucsd.edu/projects/background_subtraction/ucsdbgsub_dataset.htm

[4] http://www.dis.uniroma1.it/~labrococo/MAR/index.htm

Table 4.2: Precision, Recall and F-Measure metrics.

Metrics	Description
Precision (Pr)	$TP/(TP+FP)$
Recall (Re)	$TP/(TP+FN)$
F-Measure (F_1)	$2 \times (Pr \times Re)/(Pr+Re)$

TP = # of foreground pixels classified as foreground.
FP = # of background pixels classified as foreground.
TN = # of background pixels classified as background.
FN = # of foreground pixels classified as background.

used saliency maps as input constraint. In the next sections, we report the qualitative and quantitative evaluation, as well as the computational cost evaluation of the selected algorithms.

4.4.1 Qualitative and quantitative evaluation

Figures 4.2 and 4.3 show the visual results for background subtraction task in the UCSD and MarDT datasets, respectively. The true positive pixels (TP) are in white, true negative pixels (TN) in black, false positive pixels (FP) in red and false negative pixels (FN) in green. Is important to note that in the UCSD scenes we have used the original spatial saliency map provided by BMS, while for the MarDT scenes we have subtracted its temporal median due to the high saliency from the buildings around the river. The quantitative results in Table 4.3 show that the SCM-RPCA outperforms the previous methods, with the highest average F-measure over the selected video sequences. Each metric is described in Table 4.2. As can be seen from Figures 4.2 and 4.3, and Table 4.3, the combination with confidence map and shape constraint can reduce the amount of false positive pixels.

4.4.2 Computational cost

In Table 4.4, we report the computational cost evaluation over four videos of UCSD Background Subtraction Dataset [137]. The algorithms are implemented in MATLAB (R2014a) running on a laptop computer with Windows 7 Professional 64 bits, 2.7 GHz Core i7-3740QM processor and 32Gb of RAM. Note that in Table 4.4 the number of iterations (Iter) of the proposed method is slightly less than the 3WD and RMAMR, except for the Ocean scene. However, the computation time is slightly increased, except for the Boats scene. We noticed that the combination of shape constraint and confidence map did not changed significantly the number of iterations and computation time over original 3WD.

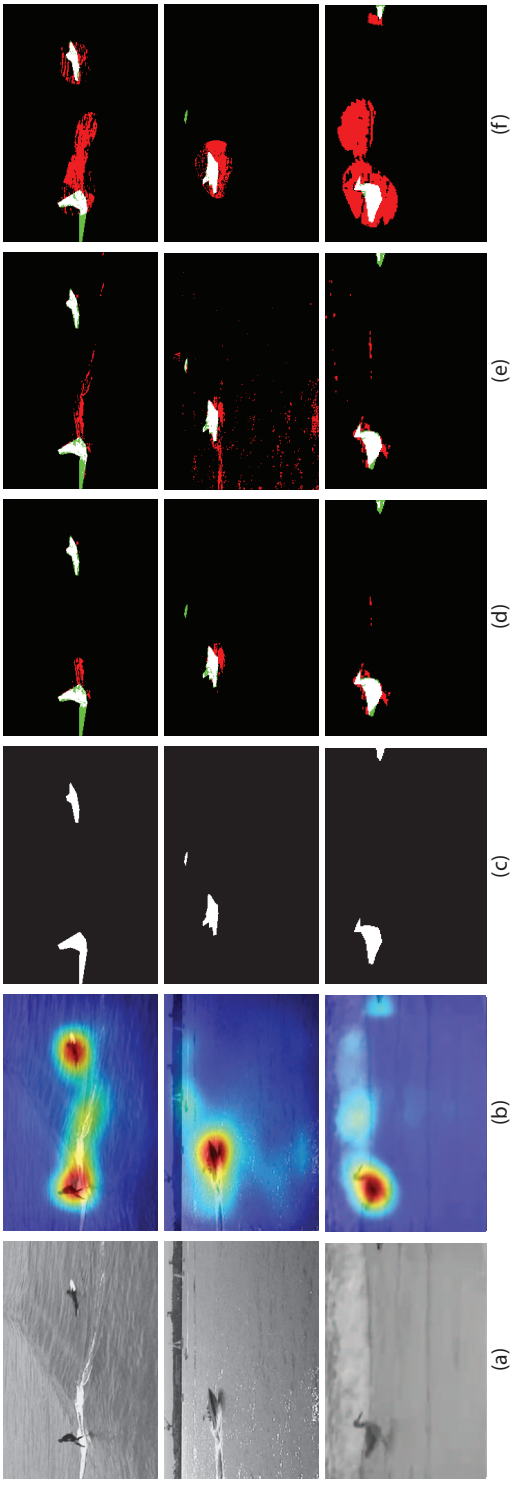


Figure 4.2: Visual comparison of background subtraction results over three scenes of UCSD dataset. From top to bottom: surfers, boats and birds. From left to right: (a) input frame, (b) saliency map generated by BMS, (c) ground truth, (d) proposed approach, (e) 3WD, and (f) RMAMR. The top 3 best algorithms (organized by rank) from Table 4.3 were chosen.

Figure 4.3: Visual results of SCM-RPCA over three scenes of MarDT dataset. From left to right: (a) input frame, (b) saliency map with its temporal median subtracted (due to the high saliency from the buildings around the river), (c) low-rank component, (d) sparse component, (e) foreground mask, and (f) ground truth.

Table 4.3: Quantitative results on four videos of UCSD Background Subtraction Dataset.

	Birds			**Surfers**			**Boats**			**Ocean**			**Rank**
	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1	Re	Pr	F_1	$Avg.F_1$
PCP	0.842	0.094	0.170	0.754	0.075	0.137	0.814	0.100	0.178	0.748	0.115	0.200	0.171
Lag-SPCP-QN	0.413	0.322	0.362	0.244	0.282	0.261	0.405	0.215	0.281	0.484	0.313	0.380	0.321
RMAMR	0.823	0.229	0.358	0.775	0.248	0.376	0.816	0.230	0.359	0.777	0.175	0.286	0.345
3WD	0.586	0.604	0.595	0.538	0.405	0.462	0.673	0.473	0.556	0.563	0.337	0.422	0.509
SCM-RPCA	0.573	0.638	**0.604**	0.518	0.565	**0.541**	0.663	0.550	**0.602**	0.457	0.544	**0.497**	**0.561**

Table 4.4: Computational cost evaluation over four videos of UCSD Background Subtraction Dataset.

	Birds ($242 \times 156 \times 71$)		**Surfers** ($344 \times 224 \times 41$)		**Boats** ($344 \times 224 \times 31$)		**Ocean** ($316 \times 196 \times 176$)	
	Iter	**Time***	**Iter**	**Time***	**Iter**	**Time***	**Iter**	**Time***
PCP	$^{+}100$	27.29	$^{+}100$	21.19	$^{+}100$	18.47	$^{+}100$	110.53
Lag-SPCP-QN	**29**	10.12	53	16.27	39	10.01	**18**	**29.49**
RMAMR	34	10.63	35	13.09	33	11.44	35	44.22
3WD	30	**4.53**	26	**4.28**	31	4.06	42	29.96
SCM-RPCA	**29**	4.59	**25**	4.37	**27**	**3.82**	43	33.02

($width \times height \times length$) denotes the frame resolution and the number of processed frames.

* Time for matrix decomposition (in seconds). Does not include the time to compute the input constraint (saliency maps).

$^{+}$ Iteration limit 100 reached.

4.5 Conclusion

In summary, a double-constrained version of RPCA is proposed to improve the foreground detection in dynamic scenes. The sparse component is constrained by shape and confidence maps both extracted from spatial saliency maps. The experimental results indicate a better enhancement of the object foreground mask when compared with its direct competitors. As shown in qualitative and quantitative evaluation, the combination with confidence map and shape constraint can reduce the amount of false positive pixels. In addition, the computational cost evaluation demonstrates that the proposed algorithm has a slightly change in the number of iterations and computation time compared to the original 3WD.

In further works, we plan to investigate how spatio-temporal saliency detectors can help the proposed approach to improve the foreground detection. In this chapter, the confidence map and shape constraint were built from the same source, specifically by saliency maps. We will explore how different sources can be used to build separately these constraints.

Chapter 5

Incremental tensor subspace learning using multiple features

In this chapter, we present an incremental multi-feature tensor subspace learning (IMTSL) algorithm for handling streaming multidimensional data in the case of intelligent video surveillance applications. The proposed method constructs a multi-feature low-rank model for robust modeling of the scene background. Moreover, the IMTSL method updates the low-rank model incrementally through an incremental learning of its unfolding matrices. This work is based on our publication (ICIAR, 2014, [176]), and the related source code can be found in the IMTSL website[1].

The remainder of this chapter is organized as follows. First we start with some related work in Section 5.1. Section 5.2 describes the incremental and multi-feature tensor subspace learning algorithm. Section 5.3 presents the foreground detection method. Finally, in Sections 5.4 and 5.5, the experimental results are shown, as well as conclusions.

5.1 Related work

In the literature, several authors have employed tensor decomposition for learning a low-rank representation of the data. [211] by Vasilescu and Terzopoulos was one of the first works to employ HOSVD (see Chapter 2, Section 2.4.1.1) for performing a multilinear analysis of facial images under different illumination conditions, expressions, viewpoints and person identities. The image sequence is represented as a higher-dimensional tensor and then decomposed in order to separate and parsimoniously represent the constituent factors, resulting in a "TensorFaces" representation. Wang and Ahuja [219] also employed HOSVD for learning the expression subspace and person subspace from an ensemble of facial images. The algorithm performs a simultaneous face and facial expression recognition, which can classify the given image into one of the basic facial expression categories. He et al. [81] presented

[1]IMTSL: https://github.com/andrewssobral/imtsl

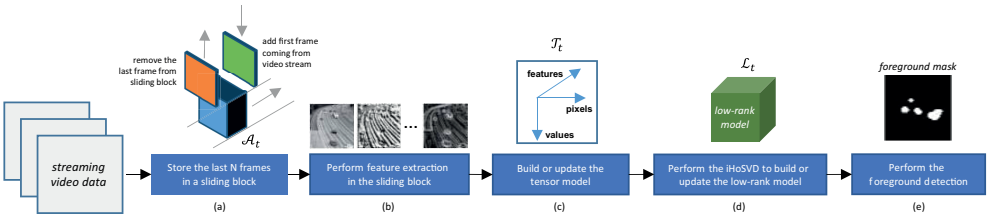


Figure 5.1: Block diagram of the proposed approach. In the step (a), the last N frames from a streaming video are stored in a sliding block or tensor $\mathcal{A}_t$. Next, a feature extraction process is done at step (b) and the tensor $\mathcal{A}_t$ is transformed in another tensor $\mathcal{T}_t$ (step (c)) . In (d), an incremental higher-order singular value decomposition (iHOSVD) is applied in the tensor $\mathcal{T}_t$ resulting in a low-rank tensor $\mathcal{L}_t$. Finally, in the step (e) a foreground detection method is applied for each new frame to segment the moving objects.

a tensor subspace analysis algorithm called TSA (Tensor Subspace Analysis), which detects the intrinsic local geometrical structure of the tensor space by learning a lower dimensional tensor subspace. Experiments on PIE and ORL databases demonstrated the efficiency and effectiveness of the method. However, in these last works any experiment was carried out for the background subtraction problem.

Recently, online tensor subspace learning approaches have been introduced. Sun et al. [189] proposed three tensor subspace learning methods: DTA (Dynamic Tensor Analysis), STA (Streaming Tensor Analysis) and WTA (Window-based Tensor Analysis). However, Li et al. [83] explained that the above tensor analysis algorithms cannot be applied to background modeling and object tracking directly. To solve this problem, some authors [83, 112, 113] proposed a high-order tensor learning algorithm, called incremental rank-(R1,R2,R3) tensor based subspace learning. This online algorithm builds a low-order tensor eigenspace model in which the mean and the eigenbasis are updated adaptively. The authors model the background appearance images as a 3-order tensor. Next, the tensor is subdivided into sub-tensors. Then, the proposed incremental tensor subspace learning algorithm is applied to effectively mine statistical properties of each sub-tensor. The experimental results show that the proposed approach is robust to appearance changes in background modeling and object tracking. The method described above only uses the gray-scale and color information. In some situations, only the pixels intensities may be insufficient to perform a robust foreground detection. To deal with this situation, an incremental and multi-feature tensor subspace learning algorithm is presented in this chapter.

5.2 Proposed method

Differently from previous related works, where the tensor model is built directly from the video data (i.e., each frontal slice of the tensor is a gray-scale image), in this chapter the tensor model is built from the feature extraction process. First, the last A_3 frames from a streaming video data are stored in a tensor $\mathcal{A}_t \in \mathbb{R}^{A_1 \times A_2 \times A_3}$, where t represents the tensor

$\mathcal{A}$ at time t. A_1 and A_2 is the frame width and frame height respectively, and A_3 is the number of stored frames ($A_3 = 25$ in the experiments). Subsequently, the tensor $\mathcal{A}_t$ is transformed into a tensor $\mathcal{T}_t \in \mathbb{R}^{T_1 \times T_2 \times T_3}$ after a feature extraction process, where T_1 is the number of pixels ($T_1 = A_1 \times A_2$), T_2 the number of feature values for each frame ($T_2 = A_3$) and T_3 the number of features. Here, 8 features are extracted: 1) red channel, 2) green channel, 3) blue channel, 4) gray-scale, 5) local binary patterns (LBP), 6) spatial gradients in horizontal direction, 7) spatial gradients in vertical direction, and 8) spatial gradients magnitude. All frames' resolution are resized to 160x120 (19200 pixels), so the dimension of the tensor model is $\mathcal{T}_t \in \mathbb{R}^{19200 \times 25 \times 8}$. The steps described here are shown in Figure 5.1 (a), (b) and (c). The steps (d) and (e) will be described in the next sections.

Incremental high-order singular value decomposition

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a matrix of full rank $r = \min(m, n)$. Its singular value decomposition can be expressed as: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$, where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthonormal matrices containing the eigenvectors of $\mathbf{A}\mathbf{A}^{\mathbf{T}}$ and $\mathbf{A}^{\mathbf{T}}\mathbf{A}$, respectively, (i.e. right and left singular vectors of $\mathbf{A}$), and $\mathbf{\Sigma} = diag(\sigma_1, \ldots, \sigma_r)$ is a diagonal matrix with the eigenvalues of $\mathbf{A}$ in descending order. However, the matrix factorization step in SVD is computationally very expensive, especially for large matrices. Moreover, the entire data may be not available for decomposition (e.g. streaming data when the full size of the data is unknown). Businger [36], and Bunch and Nielsen [34] are the first authors who have proposed to update SVD sequentially with the arrival of more samples, i.e. appending/removing a row/column. Subsequently, various approaches [15, 31, 110, 139, 169] have been proposed to update the SVD more efficiently and supporting new operations. Recently, Baker et al. [13] provided a generic approach to perform a low-rank incremental SVD. An implementation of the algorithm is freely available in the IncPACK MATLAB package[2].

In this chapter, we have used a modified version of the algorithm in [13]. The original version supports only the *updating* operation. As described previously, the tensor model $\mathcal{T}_t$ is updated dynamically. The last feature values are appended (i.e. *updating* operation) and the old feature values are removed (i.e. *downdating* operation) for each new frame. A simple change would be to modify the algorithm so that, instead of using a hard window, we insert an exponential forgetting factor $\lambda < 1$ ($\lambda = 1$ no forgetting occurs), weighting new columns preferentially over earlier columns. The forgetting factor is explained in the work of Ross et al. [169].

The proposed iHOSVD algorithm is shown in Algorithm 2. It creates a low-rank tensor model $\mathcal{L}_t$ with the dominant singular subspaces of the tensor model $\mathcal{T}_t$. $\mathcal{T}_t^{[n]}$ denotes the n-mode unfolding matrix (see Appendix C) of the tensor $\mathcal{T}$ at time t. $r^{[n]}$ and $t^{[n]}$ are the desired rank r and its thresholding value of the n-mode unfolding matrix ($r^{[1]} = 1$, $r^{[2]} = 8$, $r^{[3]} = 2$, and $t^{[1]} = t^{[2]} = t^{[3]} = 0.01$ in the experiments). $\mathbf{U}_{t-1}^{[n]}$, $\mathbf{\Sigma}_{t-1}^{[n]}$, and $\mathbf{V}_{t-1}^{[n]}$ denote the previous SVD of the n-mode unfolding matrix of the tensor $\mathcal{T}$ at time $t-1$.

[2] http://www.math.fsu.edu/~cbaker/IncPACK/

Algorithm 2 iHOSVD algorithm.

Input: $\mathcal{T}_t$, $r^{[n]}$, $t^{[n]}$
  $\mathcal{S}_t \leftarrow \mathcal{T}_t$
  if $t = 0$ **then**
    for $i = 1$ to n **do** {Performs the standard rank-r SVD}
      $[\mathbf{U}_t^{[i]}, \mathbf{\Sigma}_t^{[i]}, \mathbf{V}_t^{[i]}] \leftarrow \text{SVD}(\mathcal{T}_t^{[i]}, r^{[i]}, t^{[i]})$
    end for
  else
    for $i = 1$ to n **do** {Performs the incremental rank-r SVD}
      $[\mathbf{U}_t^{[i]}, \mathbf{\Sigma}_t^{[i]}, \mathbf{V}_t^{[i]}] \leftarrow \text{iSVD}(\mathcal{T}_t^{[i]}, r^{[i]}, t^{[i]}, \mathbf{U}_{t-1}^{[i]}, \mathbf{\Sigma}_{t-1}^{[i]}, \mathbf{V}_{t-1}^{[i]})$
    end for
  end if
  $\mathcal{S}_t \leftarrow \mathcal{T}_t \times_1 (\mathbf{U}_t^{[1]})^{\mathbf{T}} \ldots \times_n (\mathbf{U}_t^{[n]})^{\mathbf{T}}$ ($\times_n$ denotes the n-mode product between tensor $\mathcal{T}$ and matrix $\mathbf{U}$)
Output: $\mathcal{S}_t$, $\mathbf{U}_t^{[1]}$, ..., $\mathbf{U}_t^{[n]}$

5.3 Foreground detection

The foreground detection consists of segmenting all foreground pixels of the image to obtain the foreground components for each frame. As explained in the previous sections, a low-rank model $\mathcal{L}_t$ is built from the tensor model $\mathcal{T}_t$ incrementally. Then, for each new frame a weighted combination of similarity measures is performed. This process has two stages: first a similarity function is calculated, then a weighted combination is performed. Let $\mathcal{F}_t \in \mathbb{R}^{A_1 \times A_2 \times T_3}$ the feature's set extracted from the input frame at time t and $\mathcal{F'}_t$ the set of low-rank features reconstructed from the low-rank model $\mathcal{L}_t$ at time t; the similarity function $\mathcal{S}$ for the k-th feature ($k = \{1, \ldots, T_3\}$) at the pixel (i, j) is computed as follows:

$$\mathcal{S}_t(i,j,k) = \begin{cases} \frac{\mathcal{F}_t(i,j,k)}{\mathcal{F'}_t(i,j,k)} & \text{if } \mathcal{F}_t(i,j,k) < \mathcal{F'}_t(i,j,k) \\ 1 & \text{if } \mathcal{F}_t(i,j,k) = \mathcal{F'}_t(i,j,k) \\ \frac{\mathcal{F'}_t(i,j,k)}{\mathcal{F}_t(i,j,k)} & \text{if } \mathcal{F}_t(i,j,k) > \mathcal{F'}_t(i,j,k) \end{cases}$$

where $\mathcal{F}_t(i,j,k)$ and $\mathcal{F'}_t(i,j,k)$ are the feature value of pixel (i, j) for the feature k at time t, respectively. Note that $\mathcal{S}_t(i,j,k)$ assumes values in $[0, 1]$. Furthermore, $\mathcal{S}_t(i,j,k)$ is close to one if $\mathcal{F}_t(i,j,k)$ and $\mathcal{F'}_t(i,j,k)$ are very similar. Next, a weighted combination of similarity measures is computed as follows:

$$\mathbf{W}_t(i,j) = \sum_{k=1}^{T_3} w_k \mathcal{S}_t(i,j,k)$$

where T_3 is the total number of features and w_k weight for the k-th feature ($w_1 = w_2 = w_3 = w_6 = w_7 = w_8 = 0.125, w_4 = 0.225, w_5 = 0.025$ in the experiments). The weights

are chosen empirically in order to maximize the true positive pixels and minimize the false negative pixels in the foreground detection. The foreground mask $\mathbf{FG}$ at time t is obtained by applying the following threshold function:

$$\mathbf{FG}_t(i,j) = f(\mathbf{W}_t(i,j)) = \begin{cases} 1 & \text{if } \mathbf{W}_t(i,j) < \tau \\ 0 & \text{otherwise} \end{cases}$$

where τ is the threshold value ($\tau = 0.5$ in the experiments). In the next section we shows the experimental results of the proposed method.

5.4 Experimental results

In order to evaluate the performance of the proposed method for background modeling and subtraction, the BMC dataset[3] proposed by Vacavant et al. [207] is selected. We have compared our method with GRASTA algorithm proposed by He et al. [80] and BLWS algorithm proposed by Lin and Wei [118]. Tables 5.1 and 5.3 show the quantitative and the visual results (input image, ground-truth and foreground detection) with synthetic and real video sequences of the BMC dataset. The quantitative results in Table 5.1[4] show that the proposed method outperforms the previous methods, with the highest F-measure average and best scores over all video sequences except in 212, 312, 412 and 512. The visual results in Table 5.1 show the foreground detection for the frame #300 (Street) and frame #645 (Rotary), respectively. The experiments were performed on a computer running Intel Core i7-3740qm 2.7GHz processor with 16Gb of RAM. However, the proposed algorithm requires aprox. 2min per frame for background subtraction, where more than $> 95\%$ of time is used for low-rank decomposition. Further research consist in improving the speed of the incremental low-rank decomposition for real-time applications. Matlab codes and experimental results can be found in the iHOSVD homepage[5].

[3] http://bmc.iut-auvergne.com/

[4] In terms of Precision, Recall and F-Measure (defined in Chapter 4, Table 4.2)

[5] https://sites.google.com/site/ihosvd/

Table 5.1: Part 1 - Quantitative and visual results with synthetic videos of the BMC dataset.

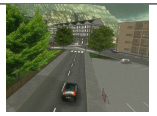
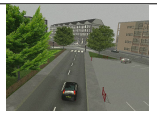
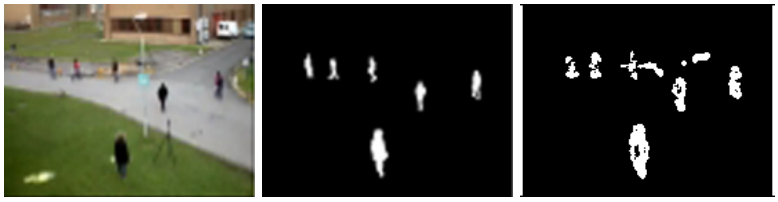
Scenes	Method	Recall	Precision	F-measure	Visual Results		
Street					**Image**	**GT**	**IMTSL**
112	IMTSL	**0.725**	0.945	**0.818**			
	GRASTA	0.700	0.980	0.817			
	BLWS	0.700	**0.981**	0.817			
212	IMTSL	0.692	0.845	0.761			
	GRASTA	**0.787**	**0.847**	**0.816**			
	BLWS	0.786	**0.847**	**0.816**			
312	IMTSL	0.566	0.831	0.673			
	GRASTA	0.695	0.965	0.807			
	BLWS	0.697	0.971	**0.812**			
412	IMTSL	0.637	0.838	0.723			
	GRASTA	**0.787**	0.843	0.814			
	BLWS	0.785	**0.848**	**0.815**			
512	IMTSL	0.652	0.893	0.753			
	GRASTA	**0.669**	0.960	**0.789**			
	BLWS	0.664	**0.966**	0.787			

Table 5.2: Part 2 - Quantitative and visual results with synthetic videos of the BMC dataset.

Scenes	Method	Recall	Precision	F-measure	Visual Results		
Rotary					**Image**	**GT**	**IMTSL**
122	IMTSL	**0.748**	**0.956**	**0.839**			
	GRASTA	0.680	0.902	0.776			
	BLWS	0.663	0.921	0.771			
222	IMTSL	**0.649**	**0.913**	**0.759**			
	GRASTA	0.637	0.548	0.589			
	BLWS	0.633	0.560	0.594			
322	IMTSL	0.555	**0.927**	**0.694**			
	GRASTA	**0.619**	0.530	0.571			
	BLWS	0.603	0.538	0.569			
422	IMTSL	0.548	**0.942**	**0.693**			
	GRASTA	**0.623**	0.778	0.692			
	BLWS	0.620	0.775	0.689			
522	IMTSL	0.677	**0.932**	**0.784**			
	GRASTA	0.791	0.714	0.751			
	BLWS	**0.793**	0.711	0.750			
Average							
	IMTSL	-	-	**0.749**			
	GRASTA	-	-	0.618			
	BLWS	-	-	0.742			

Table 5.3: Visual comparison with real videos of the BMC dataset.

Sequence Video "Wandering student"(frame #651)



Sequence Video "Traffic during windy day"(frame #140)

5.5 Conclusion

In summary, an incremental and multi-feature tensor subspace learning algorithm is presented. The multi-feature tensor model allows us to build a robust low-rank model of the background scene. Experimental results show that the proposed method achieves promising results for the background subtraction task. However, additional features can be added, enabling a more robust model of the background scene. Moreover, the proposed foreground detection approach can be changed to automatically select the best features allowing an accurate foreground detection. Further research consist of improving the speed of the incremental low-rank decomposition for real-time applications. Additional support for dynamic backgrounds might be interesting for real and complex scenes.

Chapter 6

Online stochastic tensor decomposition for multispectral video sequences

In this chapter, we propose an online stochastic tensor decomposition algorithm, named OSTD, to perform background/foreground separation in streaming multispectral video sequences. Differently from the IMTSL method presented in the previous chapter, that employed an incremental version of HOSVD, the OSTD algorithm makes use of RPCA on tensors for a robust background/foreground separation. In addition, OSTD was designed to be much faster than IMTSL and address the major difficulties of multispectral imaging for intelligent video surveillance applications. The work presented in this chapter is based on our publication (IEEE ICCV Workshop on RSL-CV, 2015, [182]), and the related source code can be found in the OSTD website[1].

6.1 Introduction

Until now, most of background subtraction algorithms were designed for mono (i.e. graylevel) or trichromatic cameras (i.e. RGB) within the visible spectrum or near infrared part (NIR). Recent advances in multispectral imaging technologies give the possibility to record multispectral videos for video surveillance applications [17]. In addition, this task becomes more complex when the data size grows (i.e. massive multidimensional data), since the real-world scenario requires larger data to be processed in a more efficient way, and in some cases, in a continuous manner (streaming data).

The primary advantage of multispectral cameras for video surveillance is the possibility to take into account the spatial (or spatio-temporal) relationships among the different spectra in a neighbourhood, allowing more elaborate spectral-spatial (and -temporal) models for a more accurate segmentation. However, the primary disadvantages are cost and complexity, due its massive and multidimensional characteristics.

[1] OSTD: https://github.com/andrewssobral/ostd

Usually a multispectral video consists of a sequence of multispectral images sensed from contiguous spectral bands. Each multispectral image can be represented as a three-dimensional data cube, or *tensor*, and here we call *frame* the measurements corresponding to a single spectral band (frontal slice of the tensor). Due to the specific nature of these data, many of the bands within multispectral images are often strongly correlated. In addition, processing multispectral images with hundreds of bands can be computationally burdensome.

In order to address these major difficulties of multispectral imaging for video surveillance (in particular, the detection of moving objects), this chapter proposes an online stochastic framework for tensor decomposition of multispectral video sequences. In short, the main contributions of this chapter are:

- an online stochastic framework for tensor decomposition to deal with multi-dimensional and streaming data, and
- the use of multispectral video sequences instead of standard mono/trichromatic images, enabling a better background subtraction.

First, we start with the related work in Section 6.2. The proposed method is described in Section 6.3. Finally, in Sections 6.4 and 6.5, the experimental results are shown, as well as conclusions.

6.2 Stochastic decomposition on tensors

Most of incremental tensor subspace learning approaches apply matrix SVD in the unfolded matrices. These approaches are usually an incremental version of the Tucker3 model (see Chapter 2, Section 2.4.1.1). However, the matrix factorization step in SVD is computationally very expensive, especially for large matrices. Therefore, real time processing is sacrificed, due to the major challenges discussed above. In order to address these problems, this chapter proposes a robust and fast online tensor-based algorithm for RGB videos, as well as for MSVS (multispectral video sequences). The proposed algorithm is based on stochastic decomposition of *low-rank* and *sparse* components. The idea of online stochastic RPCA optimization was previously proposed by Feng et al. [64] and Goes et al. [71], and it was successfully applied to background subtraction in [89–91]. In this chapter, we extend this approach to tensor analysis. The stochastic optimization is applied on each mode of the tensor and the individual basis[2] are updated iteratively followed by the processing of one video frame per time instance. In addition, a comparison of RGB and MSVS is provided, which shows that visible together with NIR spectral bands provide an improved foreground estimation compared to RGB features alone.

[2]Here, we refer basis as the set of elements (vectors) from a low-dimensional subspace.

6.3 Proposed method

Let say that $\mathcal{Y}$ is an input N-th order tensor, which is corrupted by outliers, say $\mathcal{E}$; then $\mathcal{Y}$ can be reconstructed by separating it into low-rank tensor $\mathcal{X}$ (that corresponds to BG), and sparse error $\mathcal{E}$ (that corresponds to FG objects), i.e., $\mathcal{Y} = \mathcal{X} + \mathcal{E}$, under the convex optimization framework developed in Goldfarb and Qin [72] as:

$$\underset{\mathcal{X},\mathcal{E}}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{N}||\mathcal{Y}^{[i]} - \mathcal{X}^{[i]} - \mathcal{E}^{[i]}||_F^2 + \lambda_1||\mathcal{X}^{[i]}||_* + \lambda_2||\mathcal{E}^{[i]}||_1, \tag{6.1}$$

where $||\mathcal{X}^{[i]}||_*$ and $||\mathcal{E}^{[i]}||_1$ denote the nuclear and l_1 norm of each i-mode unfolding matrices of $\mathcal{X}$ and $\mathcal{E}$, respectively. Efficient methods such as CP decomposition and Tucker decomposition [99] (a.k.a HOSVD) are used for low-rank decomposition of tensors (see Chapter 2, Section 2.4). In addition, APG, HORPCA-s based on ADAL and HORPCA-M based on I-ADAL were also developed in Goldfarb and Qin [72] to solve the problem in Equation (6.1). However, as mentioned above, these methods are based on batch optimization and are not suitable for scalable or streaming data.

In this chapter, an online optimization is considered to solve problem (6.1). The major challenge is the computation of HOSVD, because the nuclear norm keeps all the samples tightly and therefore all samples are accessed during optimization at each iteration. Therefore, it suffers from high computational complexity. In contrast, an equivalent nuclear norm is used in this chapter for each i-mode unfolding matrices of $\mathcal{X}$, whose rank is upper bounded, as shown in Recht et al. [165], as:

$$||\mathcal{X}^{[i]}||_* = \begin{array}{ll} \inf\limits_{\mathbf{L}_i,\mathbf{R}_i} & 0.5(||\mathbf{L}_i||_F^2 + ||\mathbf{R}_i||_F^2), \\ \text{subject to} & \mathcal{X}^{[i]} = \mathbf{L}_i\mathbf{R}_i^{\mathbf{T}}, \end{array} \tag{6.2}$$

where $\mathbf{L}_i \in \mathbb{R}^{p\times r}$, $\mathbf{R}_i \in \mathbb{R}^{q\times r}$, $p \times q$ denotes the dimension of the unfolding matrix $\mathcal{X}^{[i]}$, and r is the rank. Equation (6.2) shows that i-mode unfolding matrices of low-rank tensor $\mathcal{X}$ can be an explicit product of each low-dimensional subspace basis $\mathbf{L} \in \mathbb{R}^{p\times r}$ and its coefficients $\mathbf{R} \in \mathbb{R}^{q\times r}$ and this re-formulated nuclear norm is shown in [35,165,167]. Hence, Equation (6.1) is re-formulated by substituting Equation (6.2) by:

$$\begin{array}{ll} \underset{\mathcal{X},\mathcal{E},\mathbf{L},\mathbf{R}}{\text{minimize}} & \frac{1}{2}\sum\limits_{i=1}^{N}||\mathcal{Y}^{[i]} - \mathcal{X}^{[i]} - \mathcal{E}^{[i]}||_F^2 + \frac{\lambda_1}{2}(||\mathbf{L}_i||_F^2 + ||\mathbf{R}_i||_F^2) + \lambda_2||\mathcal{E}^{[i]}||_1, \\ \text{subject to} & \mathcal{X}^{[i]} = \mathbf{L}_i\mathbf{R}_i^{\mathbf{T}}. \end{array} \tag{6.3}$$

The objective function minimization, avoiding the constraints in Equation (6.3) and setting $\mathcal{X}^{[i]} = \mathbf{L}_i\mathbf{R}_i^{\mathbf{T}}$, is defined as follows:

$$\underset{\mathbf{L},\mathbf{R},\mathcal{E}}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{N}||\mathcal{Y}^{[i]} - \mathbf{L}_i\mathbf{R}_i^{\mathbf{T}} - \mathcal{E}^{[i]}||_F^2 + \frac{\lambda_1}{2}(||\mathbf{L}_i||_F^2 + ||\mathbf{R}_i||_F^2) + \lambda_2||\mathcal{E}^{[i]}||_1, \tag{6.4}$$

where λ_1 and λ_2 are regularization parameters for low-rank and sparsity patterns. Equation (6.4) is the main equation for stochastic tensor decomposition, which is not completely convex with respect to $\mathbf{L}_i$ and $\mathbf{R}_i$. However, Equation (6.3) gives the global optimal solution to the original optimization problem in Equation (6.2), as proved in Feng et al. [64]. The following cost function is required to be optimized for solving Equation (6.3) as:

$$f_n(\mathbf{L}) = \frac{1}{n}\sum_{i=1}^{N}\sum_{t=1}^{n} l(\mathcal{Y}^{(t)[i]}, \mathbf{L}_i) + \frac{\lambda_1}{2n}||\mathbf{L}_i||_F^2, \tag{6.5}$$

where n is the number of samples, and $\mathcal{Y}^{(t)[i]}$ denotes the i^{th} mode of a tensor $\mathcal{Y}$ at time instance t given by:

$$l(\mathcal{Y}^{(t)[i]}, \mathbf{L}_i) = \underset{\mathbf{L},\mathbf{R},\mathcal{E}}{\text{minimize}}\ ||vec(\mathcal{Y}^{(t)[i]}) - \mathbf{L}_i\mathbf{r} - \mathbf{e}||_2^2 + \frac{\lambda_1}{2}||\mathbf{r}||_2^2 + \lambda_2||\mathbf{e}||_1. \tag{6.6}$$

where $\mathbf{r} \in \mathbb{R}^r$ and $\mathbf{e} \in \mathbb{R}^p$ are vectors of coefficient and noise for matrix $\mathbf{R}_i^{\mathbf{T}}$ and unfolded matrix $\mathcal{E}^{[i]}$, respectively. Finally, the objective function $l_t(\mathbf{L}_i)$ for updating the basis $\mathbf{L}_i$ at time instance t is given by:

$$\begin{aligned} l_t(\mathbf{L}_i) = \frac{1}{n}\sum_{t=1}^{n}\Bigg\{ &\frac{1}{2}||vec(\mathcal{Y}^{(t)[i]}) - \mathbf{L}_i\mathbf{r}^{(t)} - \mathbf{e}^{(t)}||_2^2 + \frac{\lambda_1}{2}||\mathbf{r}^{(t)}||_2^2 \\ &+\lambda_2||\mathbf{e}^{(t)}||_1\Bigg\} + \frac{\lambda_1}{2n}||\mathbf{L}_i||_F^2, \end{aligned} \tag{6.7}$$

The main goal is to minimize the cost function in Equation (6.5) through stochastic optimization method, as shown in Algorithm 3. In case of BG modeling, one video frame (i.e. RGB image) at a time t is processed in an online manner. The coefficient $\mathbf{r}$, sparse outliers $\mathbf{e}$ and basis $\mathbf{L}_i$ are optimized in an iterative way. Moreover, $\mathbf{r}$ and $\mathbf{e}$ are estimated with fixed random basis $\mathbf{L}_i$ by projecting one sample using Equation (6.2). This subproblem requires to solve the following small-scale convex optimization problem at time instance t:

$$\mathbf{r}^{(t)} = (\mathbf{L}_i^{\mathbf{T}}\mathbf{L}_i + \lambda_1\mathbf{I})^{-1}\mathbf{L}_i^{\mathbf{T}}\left\{vec(\mathcal{Y}^{(t)[i]}) - \mathbf{e}^{(t-1)}\right\}, \tag{6.8}$$

$$e^{(t)} = \begin{cases} \mathbf{M}^{(t)}(k) - \lambda_2, & \text{if } \mathbf{M}^{(t)}(k) > \lambda_2, \\ \mathbf{M}^{(t)}(k) + \lambda_2, & \text{if } \mathbf{M}^{(t)}(k) < \lambda_2, \\ 0, & \text{otherwise,} \end{cases} \tag{6.9}$$

where $\mathbf{M}^{(t)} = vec(\mathcal{Y}^{(t)[i]}) - \mathbf{L}\mathbf{r}^{(t)}$ and $\mathbf{M}^{(t)}(k)$ is the k-th element in $\mathbf{M}^{(\mathbf{t})}$. The basis $\mathbf{L}_i$ is estimated using Equation (6.13) through minimizing the previously computed coefficients $\mathbf{r}$ and $\mathbf{e}$, and it is updated using Algorithm (4). If the rank r is given and the basis $\mathbf{L}_i$ is estimated as above, then $\mathbf{L}_i$ converges to the optimal solution asymptotically as compared to its batch counterpart, as shown in Feng et al. [64]. The BG sequence is then modeled by low-rank tensor $\mathcal{X}$ which changes at a time instance t. Finally, a hard thresholding scheme is applied on a sparse component to get the binary FG mask (see Equation 4.8).

Algorithm 3 Online Stochastic Tensor Decomposition

Input: $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$.

Initialize: $\mathcal{X} = \mathcal{E} = 0$ (*low-rank* and *sparse* components), $\mathbf{L} \in \mathbb{R}^{p \times r}$ (initial basis), $\mathbf{A} \in \mathbb{R}^{r \times r}$, $\mathbf{B} \in \mathbb{R}^{p \times r}$, $\mathbf{r} \in \mathbb{R}^{r}$, $\mathbf{R} \in \mathbb{R}^{q \times r}$, $\mathbf{e} \in \mathbb{R}^{p}$, $\mathbf{I} \in \mathbb{R}^{r \times r}$ (unitary matrix), $\lambda_1 = \frac{1}{\sqrt{\max(size(\mathcal{Y})}}$, and $\lambda_2 = 10\lambda_1$.

1: **for** $t = 1$ to n **do** {access each sample}
2: **for** $i = 1$ to N **do** {each tensor mode}
3: Access each sample from i^{th} mode of tensor $\mathcal{Y}$ by $\mathcal{Y}^{(t)[i]}$.
4: Compute the coefficients $\mathbf{r}$ and noise $\mathbf{e}$ by projecting the new sample as:

$$\left\{\mathbf{r}^{(t)}, \mathbf{e}^{(t)}\right\} = \arg\min \frac{1}{2}||vec(\mathcal{Y}^{(t)[i]}) - \mathbf{L}_i^{(t-1)}\mathbf{r} - \mathbf{e}||_2^2 + \frac{\lambda_1}{2}||\mathbf{r}||_2^2 + \lambda_2||\mathbf{e}||_1. \quad (6.10)$$

5: Compute the accumulation matrices $\mathbf{A}^{(t)}$ and $\mathbf{B}^{(t)}$:

$$\mathbf{A}^{(t)} \leftarrow \mathbf{A}^{(t-1)} + \mathbf{r}^{(t)}\mathbf{r}^{(t)\mathbf{T}}, \quad (6.11)$$

$$\mathbf{B}^{(t)} \leftarrow \mathbf{B}^{(t-1)} + (vec(\mathcal{Y}^{(t)[i]}) - \mathbf{e}^{(t)})\mathbf{r}^{(t)\mathbf{T}}. \quad (6.12)$$

6: Compute $\mathbf{L}_i^{(t)}$ with previous iteration $\mathbf{L}_i^{(t-1)}$ and update the basis using Algorithm (4).

$$\mathbf{L}_i^{(t)} = \arg\min \frac{1}{2} Tr[\mathbf{L}_i^{(t-1)\mathbf{T}}(\mathbf{A}^{(t)} + \lambda_1\mathbf{I})\mathbf{L}_i^{(t-1)}] - Tr(\mathbf{L}_i^{(t-1)\mathbf{T}}\mathbf{B}^{(t)}). \quad (6.13)$$

7: $\mathcal{L}^{(t)[i]} \leftarrow \mathbf{L}_i\mathbf{R}_i^{\mathbf{T}}$ (low-dimensional subspace for each i-th mode)
8: $vec(\mathcal{E}^{(t)[i]}) \leftarrow \mathbf{e}^{(t)}$ (*sparse* error)
9: **end for**
10: **end for**

Output: $\mathcal{X} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{X}^{[i]}$, $\mathcal{E} = \sum_{i=1}^{N}\mathcal{E}^{[i]}$.

Algorithm 4 Basis Update

Input: $\mathbf{L} = [\mathbf{l}_1, ..., \mathbf{l}_r] \in \mathbb{R}^{p \times r}$, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_r] \in \mathbb{R}^{r \times r}$, $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r] \in \mathbb{R}^{p \times r}$.

1: $\widetilde{\mathbf{A}} \leftarrow \mathbf{A} + \lambda_1\mathbf{I}$
2: **for** $j = 1$ to r **do** {access each column of $\mathbf{L}$}
3: Update each column of basis matrix $\mathbf{L}$

$$\mathbf{l}_j \leftarrow \frac{1}{\widetilde{\mathbf{A}}_{jj}}(\mathbf{b}_j - \mathbf{L}\widetilde{\mathbf{a}_j}) + \mathbf{l}_j \quad (6.14)$$

4: **end for**
5: **return** $\mathbf{L}$ (Updated basis)

6.4 Experimental results

In this section, we present our experimental results in detail. We first evaluate the proposed method performance on synthetic generated data; then the qualitative and quantitative analysis on MSVS is presented.

6.4.1 Evaluation on synthetic data

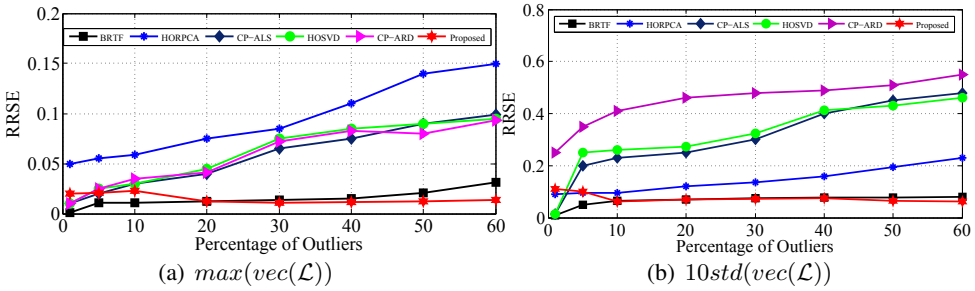
The proposed method is first quantitatively tested on synthetic data. For data evaluation, a true low-rank tensor $\mathcal{L}$ of size $30 \times 30 \times 30$ is generated by rank-3 factor matrices e.g., $\mathbf{Y}^{[k]} \in \mathbb{R}^{30\times 3}$ where $k = 1, 2, 3$. Each factor matrix $\mathbf{Y}^{[k]}$ consists of three components such as $[\sin(4\pi\frac{k}{30}), \cos(4\pi\frac{k}{30}), sgn(\sin(\pi))]$. The first two components are different and third one is common in all modes. A random entries of $\mathcal{L}$ is corrupted by outliers from uniform distribution and small noise $\mathcal{N}(0, 0.01)$. We used Root Relative Square Error (RRSE) as measure for evaluation, given by $\frac{||\widehat{\mathcal{L}}-\mathcal{L}||_2}{||\mathcal{L}||_2}$, where $\widehat{\mathcal{L}}$ is the estimated low-rank tensor. We compare our RRSE performance with other state of the art methods, such as BRTF [253], CP-ARD [142], CP-ALS [99], HORPCA [72] and HOSVD [72], respectively (see Chapter 2, Section 2.4). Figure 6.1 shows the value of RRSE for the recovered tensor $\widehat{\mathcal{L}}$. We consider two cases for robust tensor recovery for true data generation in Figure 6.1. First, the magnitude is considered within a range of true data (fully observed data) as shown in Figure 6.1 (a). However, Figure 6.1 (b) shows that the magnitude is taken larger for corrupting some entries in true low-rank (partially observed data). In each case, the proposed method shows a very significant improvement compared to its batch counter-part, such as BRTF.

6.4.2 Evaluation on multispectral video sequences

We evaluate the proposed method on MSVS dataset [17]. This is the first dataset on MSVS[3] available for research community in background subtraction. The main purpose of this dataset is to show the advantage of multispectral information for an efficient foreground-background separation when illumination variations and color saturation occurs. Both qualitative and quantitative results are presented.

The MSVS dataset contains a set of 5 video sequences with 7 multispectral bands (6 visible spectra and 1 NIR spectrum). Each sequence presents a well known BS challenge, such as color saturation and dynamic background. Figure 6.2 shows the visual comparison of the proposed approach for BS task over three scenes of MSVS dataset. The true positives pixels (TP) are in white, true negatives pixels (TN) in black, false positives pixels (FP) in red and false negatives pixels (FN) in green. Figure 6.3 shows the visual results of these sequences using individual band with RGB features. This qualitative evaluation shows that BS using stochastic tensor decomposition on 7 multispectral bands together with visible spectra provides a satisfactory FG segmentation. Figure 6.4 shows the result from RGB image, 6 visible spectrum and 1 NIR spectral band together with visible spectra.

[3] http://ilt.u-bourgogne.fr/benezeth/projects/ICRA2014/



(a) $max(vec(\mathcal{L}))$ (b) $10std(vec(\mathcal{L}))$

Figure 6.1: Performance of reconstructed low-rank tensor.

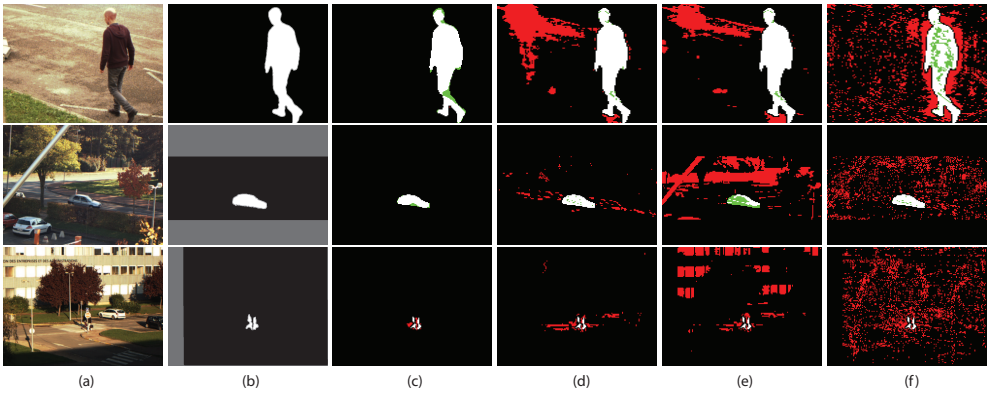
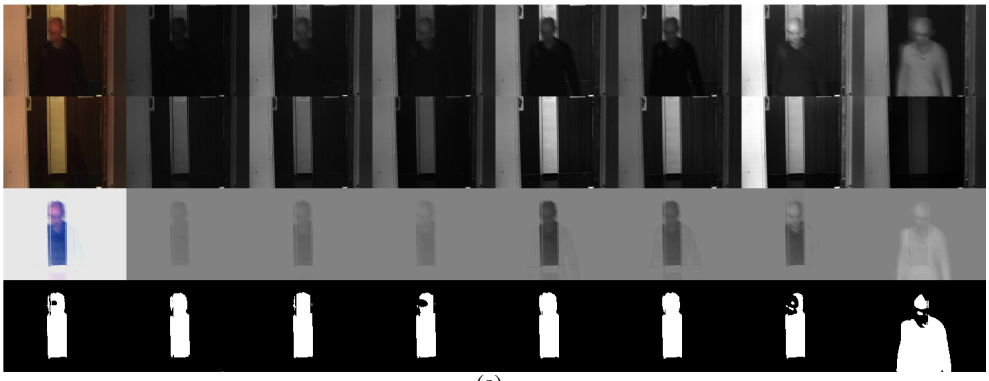


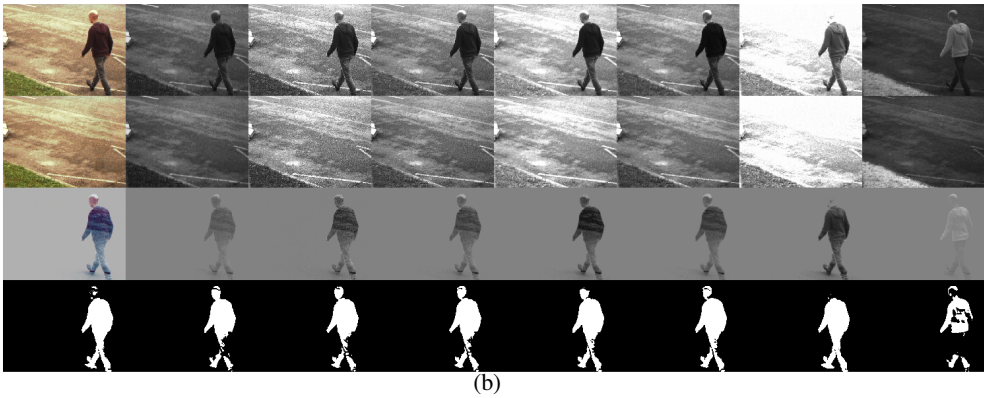
Figure 6.2: Visual comparison of background subtraction results over three scenes of the MSVS dataset. From left to right: (a) input RGB image, (b) ground truth, (c) proposed approach, (d) BRTF, (e) HORPCA, and (f) CP-ALS.

The proposed method is also tested for quantitative analysis. The MSVS dataset contains images of size 658×492 for each band. So, the size of the input tensor $\mathcal{A}$ with 7 multispectral bands is $658 \times 492 \times 7$ for each video frame. The F-measure value (see Table 4.2) is computed for each video sequence with its available ground truth images. Table 6.1 shows a comparison results achieved using the RGB bands and all the seven multispectral bands (MSB). The average F-measure score is compared for each video with 3 other methods: CP-ALS [99], HORPCA [72], and BRTF [253] (see Chapter 2, Section 2.4). The experimental evaluations show that the proposed methodology outperforms the other approaches.

The proposed scheme processes each multispectral or RGB image per time instance reaching almost real-time processing, whereas CP-ALS, HORPCA, and BRTF are based on batch optimization strategy. Due to this limitation, the CP-ALS, HORPCA, and BRTF were applied for each 100 frames at time (reducing the computational cost) of the whole video sequence (fourth-order tensor). In this chapter, the parameter r in Algorithm (3) was defined experimentally as 10. For CP-ALS, the rank was defined as 50 for better visual results. For HORPCA and BRTF, we used their default parameters. To obtain the foreground mask, the sparse component $\mathcal{E}$ was thresholded. We calculated the mean of $\mathcal{E}$ along the third dimension, generating a matrix $\mathbf{E}$, then a hard threshold function (see Equation 4.8) was applied.



(a)

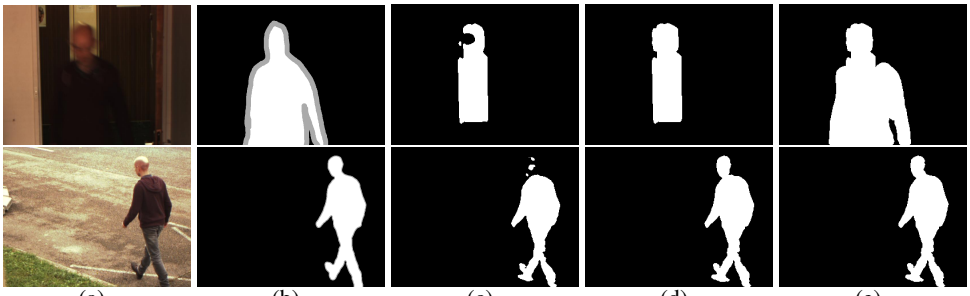


(b)

Figure 6.3: Visual results of the proposed method on each RGB and multispectral band. From top to bottom: input image, low-rank component, sparse component, and the foreground mask. From left to right: RGB image, set of 6 visible, and 1 NIR spectrum are shown in each column separately.

Table 6.1: MSVS dataset: Comparison of average F-measure score in (%) with other approaches.

Methods	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	**Avg**
CP-ALS	RGB 58.69 MSB 71.61	RGB 71.25 MSB 83.50	RGB 51.32 MSB 68.54	RGB 60.21 MSB 78.63	RGB 49.35 MSB 66.97	RGB 58.16 MSB 73.85
HORPCA	RGB 63.23 MSB 80.65	RGB 78.52 MSB 84.79	RGB 55.69 MSB 68.12	RGB 67.56 MSB 77.56	RGB 58.80 MSB 74.47	RGB 64.76 MSB 77.11
BRTF	RGB 68.56 MSB 85.30	RGB 79.21 MSB 89.63	RGB 63.56 MSB 68.11	RGB 73.22 MSB 84.65	RGB 62.51 MSB 77.91	RGB 70.32 MSB 82.76
Proposed	RGB **78.63** MSB **93.65**	RGB **85.96** MSB **95.17**	RGB **79.56** MSB **90.64**	RGB **76.32** MSB **89.29**	RGB **71.23** MSB **92.66**	RGB **76.69** MSB **92.28**



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

Figure 6.4: FG results on 1^{st} and 2^{nd} videos of the MSVS dataset. (a) input image, (b) ground truth, (c) results for only RGB, (d) for only 6 visible bands, and (e) for 1 NIR spectral band alone.

6.4.3 Basis initialization with bilateral random projections

Bilateral Random Projections (BRP) was first proposed by Zhou and Tao [257] as a fast low-rank approximation method for dense matrices. The effectiveness and the efficiency of BRP was verified in [256] for the GoDec algorithm to perform low-rank and sparse decomposition. Given r bilateral random projections of a $m \times n$ dense matrix $\mathbf{X}$, the low-rank approximation $\mathbf{L}$ can be rapidly built by:

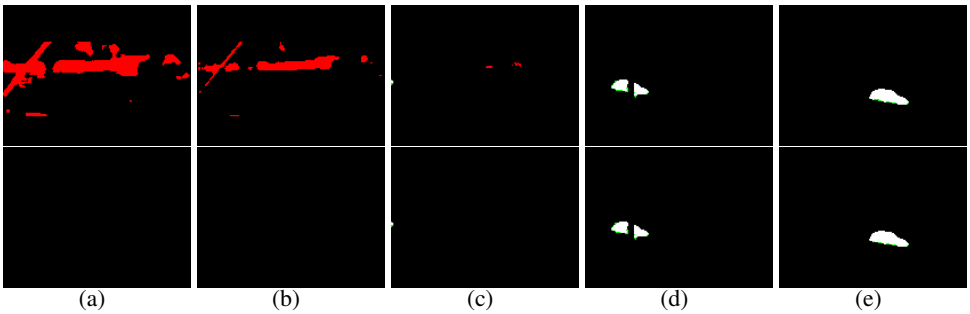
$$\mathbf{L} = \mathbf{Y}_1(\mathbf{A}_2^T\mathbf{Y}_1)^{-1}\mathbf{Y}_2^T \tag{6.15}$$

where $\mathbf{Y}_1 = \mathbf{X}\mathbf{A}_1$, $\mathbf{Y}_2 = \mathbf{X}^T\mathbf{A}_2$, and $\mathbf{A}_1 \in \mathbb{R}^{q\times r}$ and $\mathbf{A}_2 \in \mathbb{R}^{p\times r}$ are random matrices.

In this section, we evaluate the robustness of BRP for the basis initialization instead of the traditional uniformly distributed random numbers (UDRN). For demonstration, Figure 6.5 shows a fast background modeling convergence for the first 20 video frames on the 3^{rd} video of the MSVS dataset. As it can be seen, BRP enables a fast and effective low-rank approximation, reducing the amount of false positive pixels in the background model initialization task. Finally, the power scheme modification proposed by Zhou and Tao [257] can accelerate the low-rank recovery when the singular values of $\mathbf{X}$ decay slowly.

6.4.4 Computational time

Execution times have also been analyzed in our experiments. The time is recorded in CPU time as $[hh : mm : ss]$ and Table 6.2 shows the computational time of each method for the first 100 frames varying the image resolution. As it can be seen, the proposed algorithm is much faster than its direct competitors: it is almost 5 times faster than BRTF considering frames with size 160×120, and 10 times faster than CP-ALS for frames with size 320×240. The algorithms were implemented in MATLAB (R2014a) running on a laptop computer with Windows 7 Professional 64 bits, 2.7 GHz Core i7-3740QM processor and 32Gb of RAM. The MATLAB implementation of the proposed approach is available at https://github.com/andrewssobral/ostd, and the the evaluated algorithms are available in



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

Figure 6.5: FG results on the 3^{rd} video of the MSVS dataset (red = FP). From top to bottom: basis initialization with UDRN and BRP. From left to right, the FG mask at: (a) frame 1, (b) frame 5, (c) frame 10, (d) frame 15, and (e) frame 20.

Size	HORPCA	CP-ALS	BRTF	**Proposed**
160×120	00:01:35	00:00:40	00:00:22	**00:00:04**
320×240	00:04:56	00:02:09	00:03:50	**00:00:12**

Table 6.2: Execution times according to different image resolutions.

the LRS[4] [177] library.

6.5 Conclusion

In summary, we proposed an online stochastic tensor decomposition algorithm for robust BS application. Experimental results show that the proposed methodology outperforms the other considered approaches, and we have achieved almost real time processing, since one video frame is processed at time. The basis initialization with BRP can accelerate the low-rank approximation, reducing the amount of false positive pixels in the background model initialization step. In addition, the basis is updated incrementally, making it more robust against gross outliers. A future research may concern the recent advances on randomized principal component analysis [61, 78, 228]. Instead of making a full decomposition of the unfolded matrices, the randomized algorithms provide an efficient computational framework that computes a compressed representation of the data using random sampling.

[4] http://github.com/andrewssobral/lrslibrary

Chapter 7

Robust subspace clustering: from single subspace to multiple subspaces

In this chapter, we investigate a particular approach of low-rank and sparse representation, named *subspace clustering*. Differently from previous methods described in the last chapters, where inliers lie on a single low dimensional subspace, subspace clustering methods consider the inliers are drawn from the union of low-dimensional subspaces. Instead of applying subspace clustering for background modeling and foreground separation as shown in the previous chapters, we evaluate the robustness of some subspace clustering algorithms for human action recognition from 3D skeletal data. This chapter presents a particular work realized in conjunction with the Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB). The work presented here is currently under revision for publication [181]. This chapter is also related with a recently published survey (Sensors, 2016, [73]) on human pose estimation from monocular images in collaboration with researchers from China University of Petroleum and CVC.

7.1 Introduction

Human action recognition (HAR) is an important problem in computer vision. Application fields include video surveillance, automatic video indexing and human computer interaction. Most solutions for HAR learn action patterns from sequences of image features, like Space-Time Interest Points (STIP) [103], temporal templates [50], 3D SIFT [171], optical flow [6, 8], Motion History Volume [225], among the others. These features are commonly used to describe human actions, which are subsequently classified using techniques like Hidden Markov Models [6] and Support Vector Machines [170]. Recent and exhaustive reviews of methods for HAR can be found in [162, 226].

However, the development of advanced motion sensing devices, and especially the emergence of Microsoft Kinect [79], has enabled us to capture the human skeleton from the depth information in real-time, which inspired the research on activity recognition from 3D skeletal

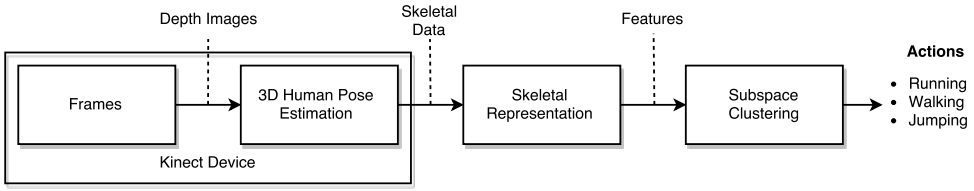


Figure 7.1: Proposed framework for robust subspace clustering of human activities through skeletal data.

data. An increasing number of algorithms have employed depth data in vision-based human action recognition [5, 44, 126]. The human body is represented as an articulated system of rigid segments connected by joints (human skeleton) [174, 175], and human action is considered as a continuous evolution of the spatial configuration of these segments. In essence, the problem of action recognition is based on the information extracted from a number of action descriptors calculated from a skeleton fitted to the body of a tracked subject.

On the one hand, approaches for recognizing human activities from skeletal data play an important role in human motion analysis using depth imagery. On the other hand, very few researches explore the recent advances in robust subspace clustering. In particular, we consider that the skeletal actions can be drawn from the union of low-dimensional subspaces. In accordance with the last advances on subspace clustering, Sparse Subspace Clustering (SSC) [59] and Low-Rank Representation (LRR) [121] are both considered as the state-of-the-art methods for subspace clustering [60, 185, 213, 214, 222] (see Chapter 2, Section 2.3). In the meantime, most of related works on subspace clustering were applied to motion segmentation [93, 111, 164, 216, 233, 238], face clustering [43, 111, 156, 233], and video summarization or scene categorization [58, 197]. Only a few works [58, 198] have explored the use of low-dimensional subspace approaches for human activity analysis from 3D skeletal data.

In this chapter, we present a methodology for robust subspace clustering of human activities from 3D skeletal data (whose black diagram is shown in Figure 7.1). In addition, we evaluate some LRR and SSC-*based* approaches for the 3D skeletal action recognition problem. A comparison between five skeletal representations is also covered in the experimental results. First, we start with the related work in Section 7.2. A brief introduction to subspace clustering is provided in Section 7.3. Next, the feature extraction process from skeletal data is described in Section 7.4. Finally, the experimental results on recent skeletal action datasets are reported in Section 7.5, as well as conclusions in Section 7.6.

7.2 Related works

Here, we present some works related to skeletal action recognition taking into account a supervised learning perspective (Section 7.2.1) and from an unsupervised one (Section 7.2.2).

7.2.1 Supervised skeletal-based action recognition

Recent skeletal-based action recognition approaches have incorporated new representations for describing actions, and some related works are here summarized.

Xia et al. [230] present an approach to human action recognition using histograms of 3D joint locations (HOJ3D) as a compact representation of postures. The 3D skeletal joint locations are extracted from Kinect depth maps using Shotton et al.'s method [174]. The HOJ3D computed from the action depth sequences are reprojected using LDA and then clustered into k posture visual words, which represent the prototypical poses of actions. The temporal evolutions of those visual words are modeled by discrete hidden Markov models (HMMs).

Devanne et al. [54] proposed a spatio-temporal motion trajectory representation for skeletal action recognition. Each trajectory consists of one motion channel corresponding to the evolution of the 3D position of all joint coordinates within frames of action sequence. The action recognition is achieved through a shape trajectory representation that is learnt by a K-NN classifier, which takes benefit from Riemannian geometry in an open curve shape space.

In Yang et al. [240], a feature descriptor is proposed for action recognition based on differences of skeleton joints (EigenJoints), which combine action information including static posture, motion property, and overall dynamics. An Accumulated Motion Energy (AME) method is proposed to perform informative frame selection, which is able to remove noisy frames and reduce computational cost. In addition, a non-parametric Naïve-Bayes-Nearest-Neighbor (NBNN) is employed to classify multiple actions.

In Vemulapalli et al. [212], a new skeletal representation is proposed that explicitly models the 3D geometric relationships between various body parts using rotations and translations in 3D space. Since 3D rigid body motions are members of the special Euclidean group $SE(3)$, the proposed skeletal representation lies in the Lie group $SE(3) \times ... \times SE(3)$, which is a curved manifold. Using the proposed representation, human actions can be modeled as curves in this Lie group. The classification is done using a combination of dynamic time warping (DTW), Fourier temporal pyramid representation and linear SVM.

In Pazhoumand-Dar et al. [161], a novel technique that automatically determines discriminative sequences of relative joint positions is proposed for each action class. The authors employ a combination of spatio-temporal based skeleton features and propose a new similarity function based on the longest common subsequence (LCSS) algorithm [217] for dealing with both simple and complex actions. The LCSS algorithm provide an intuitive notion of similarity between trajectories by giving more weight to similar portions of the sequences [217].

Tao et al. [195] proposed a novel body-part motion based feature called Moving Poselet, which corresponds to a specific body part configuration undergoing a specific movement. A simple algorithm for jointly learning Moving Poselets and action classifiers is also proposed.

7.2.2 Clustering human activities from skeletal data

To date, only a few works have been proposed to use subspace clustering approaches for human activity recognition from skeletal data.

Ball et al. [14] used the k–means algorithm for recognizing individual persons from their walking gait using three-dimensional skeleton data extracted from Microsoft Kinect.

Zhang et al. [246] proposed a subspace clustering approach, named SCAR, to recognize human activity and detect exceptional activities. However, different from previously described approaches, the proposed method was validated on data collected from RFID-based systems.

Oszust et al. [154] presented an approach for recognition of signed expressions based on visual and skeletal data obtained from Kinect sensor. Three clustering algorithms; k–means, k–medoids and minimum entropy clustering (MEC) [114], are used to isolated Polish sign language words from time series data.

Kitsikidis et al. [98] presented a method for body motion analysis in dance combining the skeletal tracking data of multiple sensors. A posture vocabulary is generated by performing k–means clustering on a large set of unlabeled postures. Then, body part postures are combined into body posture sequences and the Hidden Conditional Random Fields (HCRF) classifier is used to recognize motion patterns.

Finally, in Azis et al. [11] k–means clustering is applied to build a dictionary of frame representatives, and actions are encoded as sequences of frame representatives.

7.3 Introduction to subspace clustering

Subspace clustering, also referred to as spectral clustering, can be regarded as an extension of the traditional clustering algorithms that seeks to find clusters that best fit a collection of data points taken from a high-dimensional space [157,185,213]. Subspace clustering is defined as the problem of fitting a union of subspaces to a collection of data points drawn from one or more subspaces and corrupted by noise and/or gross errors. Mathematically, let $\mathbf{X} \in \mathbb{R}^{M \times N}$ be the data matrix consisting of N vectors $\{\mathbf{x}_i \in \mathbb{R}^M\}_{i=1}^N$ which are assumed to be drawn from the union of K linear (or affine) subspaces S_k of unknown dimensions $d_k = dim(S_k)$ with $0 < d_k < M$. The subspace clustering problem is to find the number K of subspaces, their dimensions $\{d_k\}_{k=1}^K$, the subspace bases, and the clustering of vectors x_i into these subspaces [12,213].

In the last few years, a large number of subspace clustering methods have been developed. Vidal et al. [213,214] presented four categories of subspace clustering algorithms: algebraic methods (i.e., Generalized PCA or GPCA [215]), iterative methods (i.e., k-plane clustering [30] — generalization of the k-means algorithm), statistical methods (i.e., "mixtures of PCA", and MPPCA [199]) and spectral clustering-based methods (ie. factorization-based affinity [23,48], sparse subspace clustering or SSC [59,60], and low-rank representation or LRR [43,121,214]). Among them, methods based on spectral clustering have been shown to

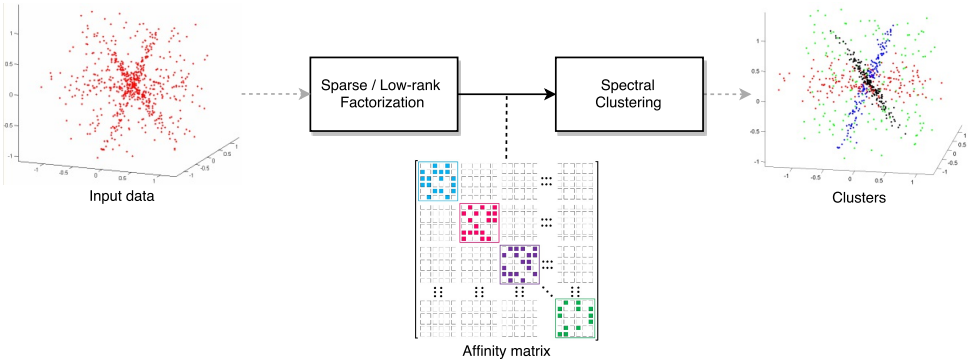


Figure 7.2: Illustration of the subspace clustering framework based on sparse and low-rank representation approaches for building the affinity matrix.

perform very well for several applications in computer vision. In general, these methods tries to first find a sparse or low-rank representation $\mathbf{Z}$ of the data matrix $\mathbf{X}$ and then apply a SC method on $\mathbf{Z}$ [147].

In the literature, von Luxburg [218] defined spectral clustering-based methods in two steps. First, a symmetric *affinity matrix* $\mathbf{C} \in [c_{ij}]$ is constructed, where $c_{ij} = c_{ij} \geq 0$ measures whether points i and j belong to the same subspace. Ideally $c_{ij} \approx 1$ if points i and j are in the same subspace and $c_{ij} \approx 0$ otherwise. The second step consists in building a weighted undirected graph where the data points are the nodes and the affinities c_{ij} are the weights. Finally, the segmentation of the data is found by clustering the eigenvectors of the graph Laplacian using central clustering techniques, such as k-means (see Figure 7.2). However, a good affinity matrix is the main challenge of this approach. Sometimes the data points could be very close to each other, even from different subspaces (e.g. near the intersection of two subspaces) [213, 214].

Previous works [23, 48] tried to build the affinity matrix of $\mathbf{X}$ by computing the SVD from data matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$ where $\mathbf{C} = \mathbf{V}_r\mathbf{V}_r^{\mathbf{T}}$ and $\mathbf{V}_r$ are the top $r = rank(\mathbf{X})$ singular vectors of $\mathbf{X}$. However, in real world applications, the data are often contaminated by noise and gross errors. In addition, selecting a good r becomes very difficult and many datasets are better modeled by affine subspaces [213, 214].

Recent advances on sparse and low-rank representation approaches have allowed the development of robust methods for building the affinity matrix in the case of data corrupted by noise and/or gross errors. As mentioned in Chapter 2, SSC [59] and LRR [121] are both considered as the state-of-the-art methods for subspace clustering. In this chapter we evaluate LRR and SCC (and their variants) for human activity recognition from 3D skeletal data.

7.4 Feature extraction on skeletal action datasets

Given a video sequence containing a specific human action, the 3D skeletal joint locations are inferred from depth maps via Kinect device using Shotton et al.'s method [174]. The 3D coordinates of each skeletal joint are represented as $x \in \mathbb{R}^D$, where $D = 3$. The J extracted skeletal joints are stored in a data vector $\mathbf{x} = \{x_1, x_2, \ldots, x_J\}^{\mathbf{T}} \in \mathbb{R}^P$, where $P = DJ$. For the whole video sequence, all skeleton joint locations are stored in a data matrix $\mathbf{X}^{(1)} \in \mathbb{R}^{P \times T}$ as:

$$\mathbf{X}^{(1)} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,T} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{P,1} & x_{P,2} & \cdots & x_{P,T} \end{bmatrix}, \tag{7.1}$$

where T is the number of frames. As T may vary per video sequence, a skeletal representation needs to be applied in data matrix $\mathbf{X}^{(1)}$, resulting in a feature matrix $\mathbf{X}^{(2)} \in \mathbb{R}^{F \times T^*}$ with fixed size ($T^* < T$). Finally, the skeletal representation of each action is grouped into an action matrix $\mathbf{X}^{(3)} \in \mathbb{R}^{M \times N}$ for clustering, where $M = FT^*$ and N is the total number of actions. The steps described here are shown in Figure 7.3. Several skeletal representations have been proposed in the literature (please refer to Tagliasacchi [191] for a complete survey). In this chapter, we have selected five well-known skeletal representations:

- AJP (Absolute Joint Positions) is the concatenation of 3D coordinates of all joints $x_1, \ldots, x_J$.
- RJP (Relative Joint Positions) is the concatenation of all vectors $\overrightarrow{x_i x_j}, 1 \leq i < j \leq J$.
- JAQ (Joint Angles Quaternions) is the concatenation of the quaternions corresponding to all joint angles.
- SE3AP (SE3 Lie Algebra with Absolute Pairs) and SE3RP (SE3 Lie Algebra with Relative Pairs), both proposed by Vemulapalli et al. [212], where each individual body part is represented as a point in a Lie group which is a curved manifold. Using this representation, human actions can be modeled as curves in this Lie group. For classification, the action curves are mapped from the Lie group to its Lie algebra, which is a vector space.

However, these skeletal representations are not sufficient for effective classification or clustering due to various issues, like rate variations, temporal misalignment, noise, etc. To deal with these problems, Dynamic Time Warping (DTW) [143] was first applied to handle rate variations. Next, the warped curves are represented using the Fourier temporal pyramid representation [221] removing the high frequency coefficients, handling the temporal misalignment and noise issues. This procedure is illustrated in Figure 7.4. Table 7.2 shows the length of each skeletal representation before and after the temporal modeling procedure. The length is represented by F times T^* of the feature matrix $\mathbf{X}^{(2)}$, where the difference of RAW length and Final length depends of the skeletal body model. It is important to note that the

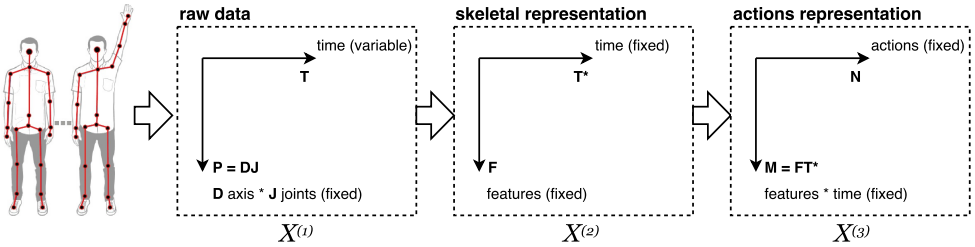


Figure 7.3: Steps behind the construction of the action representation matrix.

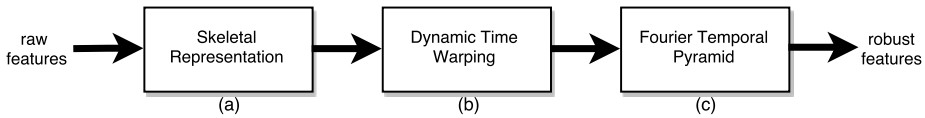


Figure 7.4: Temporal modeling procedure applied in the skeletal representation to deal with rate variations, temporal misalignment, and noise.

RAW data from a skeletal representation (output of Figure 7.4(a)) is not the same as the RAW data built up from 3D skeletal joint locations ($\mathbf{X}^{(1)}$ in Figure 7.3).

Pre-processing step: In this chapter, we have employed the same pre-processing step as adopted by Vemulapalli et al. [212]. This step work as follows:

- Invariance to absolute location: all 3D joint coordinates were transformed from the world coordinate system to a person-centric coordinate system by placing the hip center at the origin.
- Invariance to scale: one of the skeletons is used as reference, and all the other skeletons were normalized (without changing their joint angles) such that their body part lengths are equal to the corresponding lengths of the reference skeleton.
- Invariance to rotation: the skeletons were rotated so that the ground plane projection of the vector from left hip to right hip is parallel to the global x-axis.

7.5 Experimental results

In this section, we evaluate the performance of five state-of-the-art subspace clustering algorithms on two skeletal action datasets: UTKinect-Action [230][1] and Florence3D-Action [172][2]. Table 7.1 compares both datasets in term of number of actions, subjects and sequences. For all subspace clustering algorithms shown in Table 7.3, we followed the same pipeline:

[1] http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html
[2] http://www.micc.unifi.it/vim/datasets/3dactions/

Table 7.1: Datasets for human action recognition from 3D skeletal data.

Dataset	# of actions	# of subjects	# of sequences
UTKinect-Action	10	10	199
Florence3D-Action	9	10	215

Table 7.2: Length of each skeletal representation before (RAW column) and after (Final column) temporal modeling.

	AJP		RJP		JAQ		SE3AP		SE3RP	
Dataset	RAW	Final	RAW	Final	RAW	Final	RAW	Final	RAW	Final
UTKinect	4218	7182	42180	71820	11248	19152	8436	14364	151848	258552
Florence3D	215	2352	11025	17640	3920	6272	2940	4704	38220	61152

1. First, the*low-rank* or *sparse* representation of the action matrix $\mathbf{X}^{(3)}$ is obtained.
2. An undirected weighted graph $\mathbf{W}$ is constructed by using the *low-rank* or *sparse* representation to define the affinity matrix of the graph. $\mathbf{W} \in \mathbb{R}^{N\times N}$ is a symmetric non-negative similarity matrix representing the weights of the edges.
3. The clustering of the nodes is computed using a spectral clustering algorithm. We choose here the Ng et al.'s method [147] based on the normalized Laplacian as the standard SC method.

7.5.1 Evaluation protocol

In the evaluation, all the parameters are chosen so that the final average clustering error is the lowest (see Section 7.5.2). For all algorithms we varied the threshold ρ in the coefficient matrix in $[0, 1]$ increasing by 0.01. The best ρ is found when the clustering error is minimal. Then, to eliminate the effect of randomness, we repeated such trial 20 times and compared representative algorithms based on the average accuracy and standard deviation.

Implementation details: For all algorithms, we use the MATLAB code provided by their authors. All experiments are carried out using MATLAB 2015a on a laptop machine with Intel(R) Core(TM) i7-3740QM CPU at 2.70 GHz and 32 GB RAM.

7.5.2 Evaluation metrics

Following Chang et al. [42], we adopted clustering accuracy (ACC) as evaluation metrics in the experiments. Let q_i be the clustering label resulted from a clustering algorithm and p_i the corresponding ground truth label of an arbitrary data point x_i. Then, ACC is defined as follows:

$$ACC = \frac{\sum_{i=1}^{n} \delta(p_i, map(q_i))}{n} \tag{7.2}$$

Table 7.3: Selected subspace clustering algorithms for evaluation on skeletal action datasets.

Representation	Method	Author(s)
low-rank	LRR	Liu et al. (2013) [121]
	LRSC	Vidal and Favaro (2014) [214]
sparse	SSC	Elhamifar and Vidal (2009) [59]
	RSSC	Xu et al. (2015) [233]
	LS3C	Patel et al. (2013) [159]

where $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise. $map(q_i)$ is the best mapping function that permutes clustering labels to match the ground truth labels using the Kuhn-Munkres algorithm [144]. A larger ACC indicates better clustering performance.

7.5.3 Results on UTKinect-Action dataset

For this dataset, the action matrix $\mathbf{X}^{(3)}$ was projected into a $r = 2s$ dimensional subspace using PCA, where $s = 10$ represents the number of distinct actions. Thus, the number of rows of $\mathbf{X}^{(3)} \in \mathbb{R}^{M \times 199}$ is reduced, where $M = 20$ is the final row size of the action matrix before subspace clustering. Figure 7.5 shows the feature embedding visualizations using t-SNE [208]. Each clip is visualized as a point and clips belonging to the same action have the same color. Note that the features are better grouped after temporal modeling improving the clustering accuracy. Table 7.4 shows the performance comparison in terms of clustering accuracy and std of selected subspace clustering methods between the five skeletal representations, respectively. The best scores for each skeletal representation are in bold face. As it can be seen, LRSC and RSSC both using AJP and RJP as skeletal representation show the best results in terms of clustering ACC compared to their direct competitors. The confusion matrices for these two algorithms are shown in Figure 7.6.

7.5.4 Results on Florence3D-Action dataset

This is a challenging dataset due to the high intra-class variations, where the same action is performed using the left hand in some sequences and the right hand in others. In addition, the presence of actions like drink from a bottle and answer phone are quite similar to each other.

For this dataset, the action matrix $\mathbf{X}^{(3)}$ was projected into a $r = 10s$ dimensional subspace using PCA, where $s = 9$ represents the number of distinct actions. Thus, the row size of $\mathbf{X}^{(3)} \in \mathbb{R}^{M \times 215}$ is reduced, where $M = 90$ is the final row size of the action matrix before subspace clustering. Table 7.5 shows the performance comparison in terms of clustering accuracy and std of five subspace clustering methods between five skeletal representations, respectively. The best scores for each skeletal representation are in bold face. This is best viewed in Figure 7.7, that shows the feature embedding using t-SNE [208]. We note that

Table 7.4: Clustering accuracy and std of five subspace clustering methods between five skeletal representations extracted from UTKinect dataset.

Method	AJP	RJP	JAQ	SE3AP	SE3RP
SSC	0.913 ± 0.055	0.936 ± 0.048	**0.777 ± 0.027**	0.893 ± 0.040	0.820 ± 0.057
RSSC	0.921 ± 0.020	**0.951 ± 0.037**	0.760 ± 0.018	**0.900 ± 0.022**	**0.826 ± 0.026**
LRR	0.795 ± 0.035	0.788 ± 0.026	0.643 ± 0.034	0.659 ± 0.041	0.653 ± 0.042
LRSC	**0.951 ± 0.040**	0.812 ± 0.074	0.762 ± 0.013	0.768 ± 0.028	0.777 ± 0.042
LS3C	0.751 ± 0.034	0.723 ± 0.018	0.680 ± 0.013	0.675 ± 0.023	0.579 ± 0.020

some features are more overlapped than others, which results in a difficult task for clustering. As it can be seen, the RSSC using AJP as skeletal representation shows the best result in terms of clustering ACC compared to its direct competitors. The confusion matrix for this algorithm is shown in Figure 7.8. However, compared to the results obtained with UTKinect dataset, the Florence3D dataset seems to be more difficult even for the state-of-the-art methods, due to very similar actions such as drink from a bottle and answer phone, as can be seen in Table 7.6. Most of evaluated methods have clustering ACC decreased by a factor of approximately 10%.

7.5.5 Comparison to the state-of-the-art methods

As previously described, the methodology presented in this chapter explores an unsupervised learning approach through robust subspace clustering methods for skeletal action recognition. Table 7.6 compares the result of several state-of-the-art methods with the best subspace clustering method for both datasets. As can be seen, LRSC (low-rank based) and RSSC (sparse based) both achieved promising results compared with state-of-the-art supervised methods. The proposed work explores unlabeled data to find some intrinsic "natural" structures, organizing them into k-groups taking into account the recent advances on sparse and low-rank representation approaches. Evidently, supervised approaches usually outperforms the unsupervised ones in several key tasks.

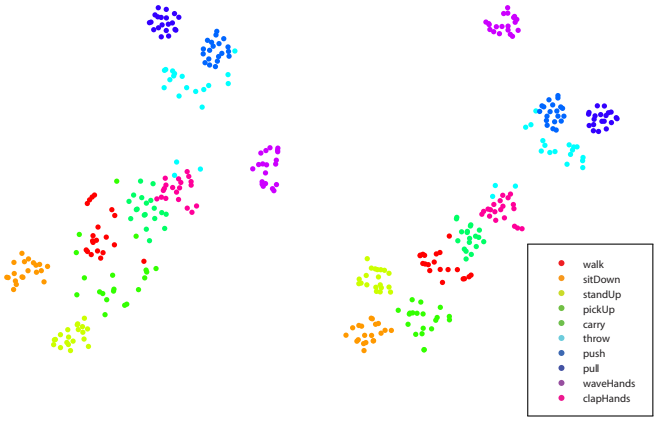


Figure 7.5: Feature embedding visualizations of AJP skeletal representation before (left) and after (right) temporal modeling procedure from UTKinect actions using t-SNE.

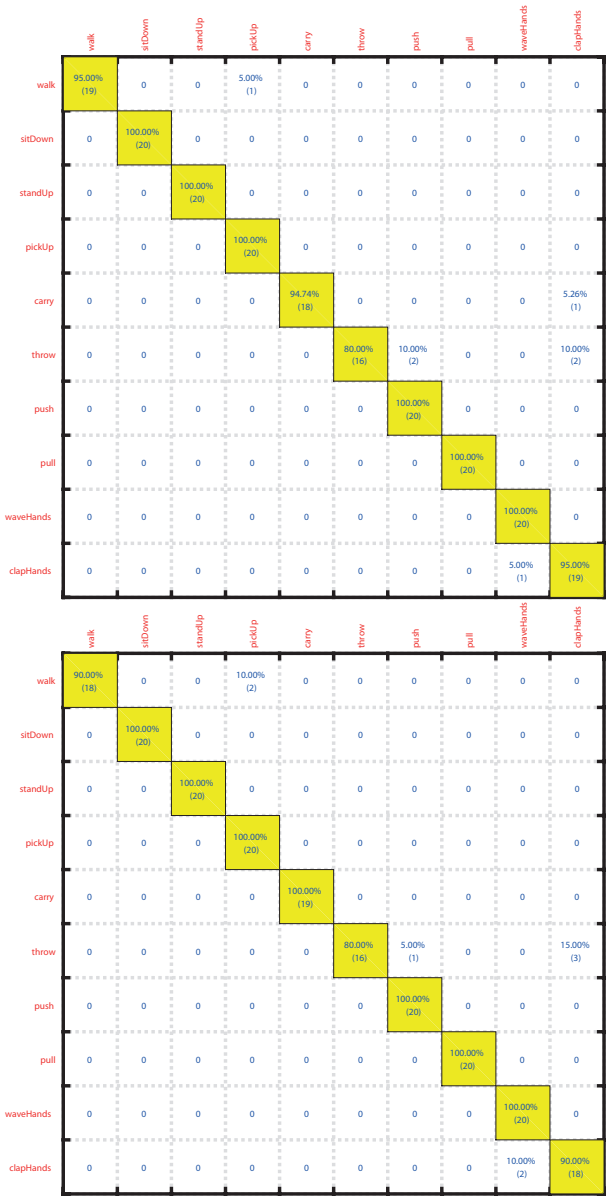


Figure 7.6: From top-down: confusion matrix for LRSC with AJP skeletal representation and RSSC with RJP skeletal representation in the UTKinect dataset.

Table 7.5: Clustering accuracy and std of five subspace clustering methods between five skeletal representations extracted from Florence3D-Action dataset.

Method	AJP	RJP	JAQ	SE3AP	SE3RP
SSC	0.788 ± 0.002	**0.742 ± 0.028**	0.708 ± 0.002	0.670 ± 0.005	0.642 ± 0.021
RSSC	**0.790 ± 0.001**	0.733 ± 0.045	0.706 ± 0.001	0.679 ± 0.004	**0.673 ± 0.010**
LRR	0.784 ± 0.002	0.503 ± 0.022	0.493 ± 0.012	0.522 ± 0.014	0.469 ± 0.014
LRSC	0.723 ± 0.007	0.730 ± 0.004	0.693 ± 0.001	**0.692 ± 0.009**	0.561 ± 0.026
LS3C	0.655 ± 0.020	0.624 ± 0.019	**0.732 ± 0.009**	0.473 ± 0.018	0.612 ± 0.039

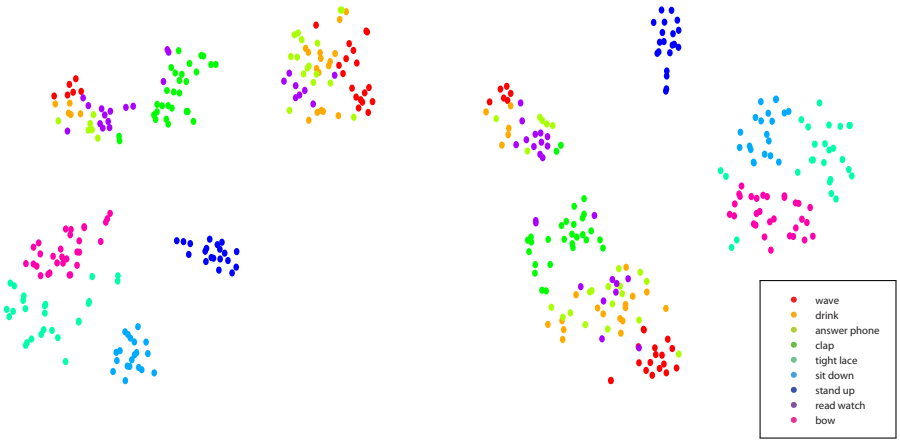


Figure 7.7: Feature embedding visualizations of AJP skeletal representation before (left) and after (right) temporal modeling procedure from Florence3D actions using t-SNE.

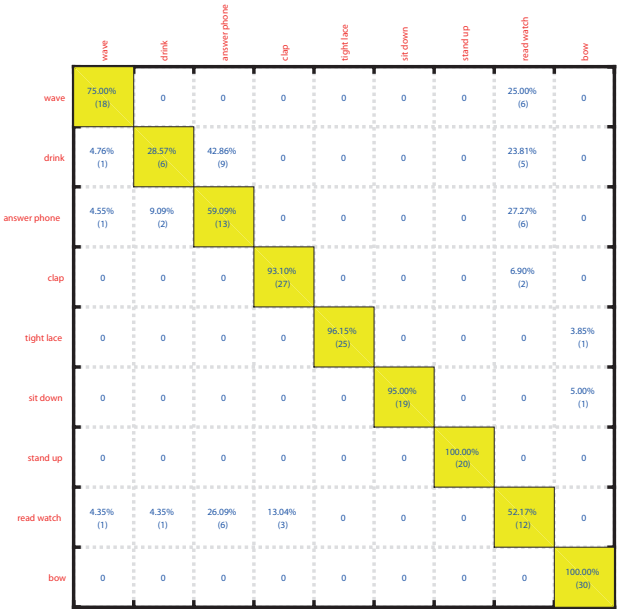


Figure 7.8: Confusion matrix for RSSC in the Florence3D dataset with AJP skeletal representation.

Table 7.6: Performance comparison with state-of-the-art methods.

Type	Author(s)	Approach	Recognition rate
	UTKinect-Action dataset		
S	Xia et al. (2012) [230]	Histograms of 3D joints	90.92%
S	Zhu et al. (2013) [261]	Random forests	87.90%
S	Vemulapalli et al. (2014) [212]	Points in a Lie Group	97.08%
U	**proposed**	LRSC + AJP or RSSC + RJP	95.10%
	Florence3D-Action dataset		
S	Seidenari et al. (2013) [172]	Multi-Part Bag-of-Poses	82.00%
S	Cippitelli et al. (2016) [47]	Key poses	82.10%
S	Vemulapalli et al. (2014) [212]	Points in a Lie Group	90.88%
U	**proposed**	RSSC + AJP	79.00%

S - *Supervised*, U - *Unsupervised*

7.6 Conclusion

In summary, we presented a methodology to recognize human activities from skeletal data through robust subspace clustering. The 3D skeletal joints locations are inferred from depth maps via the Kinect device using Shotton et al.'s method [174]. The experimental results showed that low-rank *based* (e.g. LRSC) and sparse *based* (e.g. RSSC) methods are both unsupervised approaches which provide interesting results for human action recognition from skeletal data compared with state-of-the-art supervised methods.

Ideas for future work include the possibility to apply recent Robust PCA methods for filtering the noise in the action matrix before the subspace clustering step. In addition, recent feature selection algorithms [168] can be evaluated to reduce the dimension of the action matrix instead of traditional PCA approach.

Chapter 8

Conclusions

In this chapter, we summarize the main contributions of the thesis covering the strengths and weaknesses of the proposed approaches presented in the main chapters of this thesis. Finally, we provide an outlook of the future research possibilities and directions.

8.1 Summary and contributions

The recent research on decomposition into low-rank plus sparse matrices or tensors shows a general-purpose framework that covers a wide range of applications where the data to be processed have two important assumptions: a) the inliers are drawn from a single (or a union of) low-dimensional subspace(s), and b) the corruptions are sparse. In the thesis, we have explored the fact that this assumption holds a particular association to the problem of B/F separation where the background model (almost static) is represented as a low-rank structure and the foreground objects are associated with the sparse residuals. However, the key issues and challenges in such approaches are their capabilities to handle complex and dynamic background scenarios, as well as performing in a real-time manner. Given the importance of this subject, the thesis presented here has brought the following contributions:

- In Chapter 1, we introduced the problem of moving object detection under background/foreground separation for visual-surveillance applications. We highlighted that the recent research on decomposition into low-rank plus sparse matrices shows a suitable framework to separate moving objects from the background.
- In Chapter 2, we gave an overview of the state-of-the-art methods for low-rank and sparse decomposition, as well as their application to background modeling and foreground segmentation tasks. The methods were unified in a more general framework, named DLSM, that categorizes the matrix separation problem into three main approaches: implicit, explicit and stable. In addition, we developed the matrix separation problem from a single low dimensional subspace to the union of low-dimensional subspaces, introducing the subspace clustering approach. We showed also its adequacy

to the problem of background/foreground separation by clustering motion trajectories. Finally, we extended the matrix case to the tensor case for handling multidimensional data.

- In Chapter 3, we presented a novel methodology for background model initialization seen as a reconstruction problem from missing/corrupted data. The redundant frames are eliminated and the moving regions are set to be non-observed values. Next, twenty-three matrix and tensor low-rank recovery algorithms were evaluated for the background initialization problem. The experimental results on the SBI dataset highlighted the good performance of LRGeomCG method over its direct competitors. Finally, we note that matrix-based completion methods show an attractive potential for background modeling initialization in video surveillance.

- In Chapter 4, we proposed a double-constrained version of RPCA to improve the foreground detection in maritime environments for automated video-surveillance applications. The sparse component is constrained by shape and confidence maps, both extracted from spatial saliency maps. The experimental results indicate a better enhancement of the object foreground mask when compared with its direct competitors.

- In Chapter 5, an incremental and multi-feature tensor subspace learning algorithm (IMTSL) was presented. Different from previous related works where a tensor model is built directly from the video data (i.e., each frontal slice of the tensor is a gray-scale image), in this work the tensor model was built from a previous feature extraction process. The multi-feature tensor model allows us to build a robust low-rank model of the background scene. In addition, an incremental high-order singular value decomposition was proposed, making our method able to process streaming data when the full size of the data is unknown. The experimental results have shown that the proposed method achieves promising results for the background subtraction task.

- In Chapter 6, we proposed an online stochastic tensor decomposition algorithm, named OSTD, for handling streaming multispectral video sequences for intelligent video surveillance applications. Differently from the IMTSL algorithm, the OSTD algorithm makes use of robust principal component analysis on tensors for a robust background/-foreground separation. The experimental results have shown that the proposed method outperforms its direct competitors, and we have achieved almost real time processing, since one video frame is processed in an online optimization scheme. Moreover, it is shown that the basis initialization with BRP can accelerate the low-rank approximation, reducing the amount of false positive pixels in the background model initialization step.

- Finally, in Chapter 7 we presented a particular work realized in conjunction with Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB). In this chapter, we have shown a methodology to recognize human activities from 3D skeletal data through robust subspace clustering approaches. The experimental results showed that LRSC and RSSC methods were both unsupervised approaches which provided interesting results compared with state-of-the-art supervised methods.

8.2 Limitations and future perspectives

The strengths of the contributions introduced in the thesis have been demonstrated through many experimental evaluations. However, there are limitations which could be open opportunities for future research.

- The methodology presented in Chapter 3 has two main drawbacks. First, it makes use of a simple joint motion-detection and frame-selection operation that removes the redundant frames and induces missing entries from the moving regions. This joint operation cannot deal with many real-world challenges of background model initialization, due to its sensitivity to noise, inability to deal with dynamic background and stopped objects, among the others. Secondly, the matrix and tensor completion approaches evaluated in this work make use of a batch optimization process, requiring that all video frames be stored in memory in advance. This is an important issue that limits the application in the case of streaming data or high resolution images.

  Future researches may concern: a) the investigation of a more robust approach for frame-selection that can handle the major challenges of video surveillance applications, and b) the development or evaluation of incremental and real-time approaches for low-rank reconstruction, enabling the algorithm to perform the background model initialization in streaming videos.

- The double-constrained RPCA algorithm presented in Chapter 4 makes use of shape and confidence maps, both extracted from spatial saliency maps. Thus it strongly depends on the robustness of the saliency extractor, that could present incorrect segmentation in the presence of high visual saliency objects coming from the background scene. In addition, both confidence map and shape constraint were built from the same source, instead of two complementary sources. Finally, the proposed SCM-RPCA works in a batch manner, requiring all frames be stored in memory, restricting the application in streaming or high resolution videos.

  Future work may concern the investigation of how spatio-temporal saliency detectors can help the proposed approach to improve the foreground detection, or how different sources could be used to build complementary shape and confidence maps. Furthermore, the development of an incremental version of the proposed algorithm could be desirable for streaming applications.

- The incremental and multi-feature tensor subspace learning algorithm presented in Chapter 5 has two main drawbacks. The first one is related to the high computational cost of the incremental SVD method, making it infeasible for real-time applications. The second one is related to the foreground detection method that relies on three basic steps: a) similarity function, b) weighted combination of features, and c) hard thresholding. The major limitation concerns the set of weights for each feature, that are calculated manually, making the foreground detection step unable to automatically adjust to new conditions.

  Further research may consists in improving the speed of the incremental low-rank decomposition for real-time applications. Additional supports for dynamic backgrounds

might be interesting for real and complex scenes. Finally, the investigation of fast optimization algorithms for finding the most appropriate weights could be an important research direction for this work.

- The online stochastic tensor decomposition algorithm proposed in Chapter 6 makes use of an online stochastic optimization algorithm to decompose the unfolded matrices of a tensor into a low-rank and sparse representation. The main drawback of the proposed algorithm is the computational cost required to process each unfolded matrix.

  A future research may concern the exploitation of recent advances on randomized principal component analysis [61, 78, 228]. Instead of making a full decomposition of the unfolded matrices, the randomized algorithms provide an efficient computational framework that computes a compressed representation of the data using random sampling. In other words, it captures the essential information that can then be used to obtain a low-rank matrix/tensor approximation. Finally, the implementation of the OSTD algorithm in C/C++ language with GPU support could improve its scalability for high resolution and real-time applications.

- Finally, concerning Chapter 7, ideas for future work include the possibility to apply recent Robust PCA methods for filtering the noise in the action matrix before the subspace clustering step. In addition, recent feature selection algorithms [168] can be evaluated to reduce the dimension of the action matrix instead of the traditional PCA approach.

Appendix A

Notations and symbols

In this section we provide a homogenized overview of all different mathematical notations and symbols found over all chapters in this thesis. Table A.1 presents a summarized overview of the adopted symbols.

Matrices For matrices, $\mathbf{A}$ stands for the observation matrix, $\mathbf{L}$ is the low-rank matrix, $\mathbf{S}$ is the unconstrained (residual) matrix or sparse matrix, and $\mathbf{E}$ is the noise matrix. $\mathbf{I}$ is the identity matrix. For the specific matrices, the notations are given in the section of the corresponding method.

Tensors Similar to matrices, but represented by calligraphic letters, such as $\mathcal{A}$, $\mathcal{L}$, $\mathcal{S}$, and $\mathcal{E}$.

Norms The different norms used for vectors and matrices in this thesis are classified as follows:

- Vector ℓ_α-norm, with $0 \leq \alpha \leq 2$: $||\mathbf{v}||_0$ is the ℓ_0-norm of the vector $\mathbf{v}$, and it corresponds to the number of non-zero entries. $||\mathbf{v}||_1 = \sum_i \mathbf{v}_i$ is the ℓ_1-norm of the vector $\mathbf{v}$, and it corresponds to the sum of the vector elements. $||\mathbf{v}||_2 = \sqrt{\sum_i (\mathbf{v}_i)^2}$ is the ℓ_2-norm of the vector $\mathbf{v}$, and it corresponds to the Euclidean distance.

- Matrix ℓ_α-norm, with $0 \leq \alpha \leq 2$: $||\mathbf{A}||_0$ is the ℓ_0-norm of the matrix $\mathbf{A}$, and it corresponds to the number of non-zero entries. $||\mathbf{A}||_1 = \max_j \sum_i |\mathbf{A}_{ij}|$ is the ℓ_1-norm of the matrix $\mathbf{A}$, and it corresponds to the maximum absolute column sum norm. $||\mathbf{A}||_2 = \sqrt{\sigma_{max}(\mathbf{A}^\mathbf{T}\mathbf{A})} = \sigma_{max}(\mathbf{A})$ is the ℓ_2-norm of the matrix $\mathbf{A}$, and it corresponds to the largest singular value of the matrix $\mathbf{A}$ or the square root of the maximum eigenvalue of $\mathbf{A}^\mathbf{T}\mathbf{A}$. The ℓ_2-norm for matrices is also known as spectral norm. The ℓ_2-norm is also employed in its squared version such that $||\mathbf{A}||_2^2 = \sigma_{max}(\mathbf{A}^\mathbf{T}\mathbf{A})$.
- Matrix ℓ_∞-norm: $||\mathbf{A}||_\infty = \max_i \sum_j |\mathbf{A}_{ij}|$ is the ℓ_∞-norm of the matrix $\mathbf{A}$, and it corresponds to the maximum absolute row sum norm. The ℓ_∞-norm of

the matrix $\mathbf{A}$ is equivalent to the ℓ_1-norm of the transposed matrix, such that $||\mathbf{A}||_\infty = ||\mathbf{A}^\mathbf{T}||_1$.

- Matrix $\ell_{\alpha,\beta}$-norm, with $0 \leq \alpha, \beta \leq 2$: $||\mathbf{A}||_{\alpha,\beta}$ is the $\ell_{\alpha,\beta}$-mixed norm of the matrix $\mathbf{A}$, and it corresponds to the ℓ_β-norm of the vector formed by taking the ℓ_α-norms of the columns of the matrix $\mathbf{A}$. The norm $\ell_{1,1}$ is equivalent to $\sum_{i,j}|\mathbf{A}_{i,j}|$, that is the sum of all absolute values of the matrix elements. The norm $\ell_{1,2}$ is equivalent to $\sigma(\sum_i|\mathbf{A}_i|)$, that corresponds to the singular value of the vector formed by taking the ℓ_1-norms of the columns of the matrix $\mathbf{A}$. The norm $\ell_{2,1}$ is equivalent to $\sum_i||\mathbf{A}_i||_2$ or $trace(\sqrt{\mathbf{A}^\mathbf{T}\mathbf{A}})$. The norm $\ell_{2,2}$ is equivalent to $\sqrt{trace(\mathbf{A}\mathbf{A}^\mathbf{T})}$ (also known as Frobenius norm).
- Matrix Frobenius norm: $||\mathbf{A}||_F = \sqrt{\sum_i\sum_j|\mathbf{A}_{ij}|^2} = \sqrt{trace(\mathbf{A}\mathbf{A}^\mathbf{T})}$ is the Frobenius norm of the matrix $\mathbf{A}$, and it is defined as the square root of the sum of the squared absolute values of its elements. The Frobenius norm is equivalent to $\ell_{2,2}$-norm and it is sometimes also called the Euclidean norm, which may cause confusion with the vector ℓ_2-norm. The Frobenius norm is also employed in its squared version such that $||\mathbf{A}||_F^2 = \sum_i\sum_j|\mathbf{A}_{ij}|^2 = trace(\mathbf{A}\mathbf{A}^\mathbf{T})$, representing the sum of squares of all entries.
- Matrix max norm: $||\mathbf{A}||_{max} = \max|\mathbf{A}_{ij}|$ is the max norm of the matrix $\mathbf{A}$, and it corresponds to the maximum absolute value of the matrix $\mathbf{A}$. The max norm is equivalent to the $\ell_{\infty,\infty}$-norm such that $||\mathbf{A}||_{max} = ||\mathbf{A}||_{\infty,\infty}$.
- Matrix nuclear norm: $||\mathbf{A}||_* = \sum_i \sigma_i(\mathbf{A})$ is the nuclear norm of the matrix $\mathbf{A}$, and it corresponds to the sum of the singular values of the matrix $\mathbf{A}$. The nuclear norm is equivalent to the ℓ_1-norm applied on the vector whose elements are the singular values of the matrix, such that $||\mathbf{A}||_* = ||\sigma(\mathbf{A})||_1$. For a desired rank r in low-rank minimization, the nuclear norm is also defined by $||\mathbf{A}||_* = \sum_{i=1}^{r}\sigma_i(\mathbf{A})$.
- Matrix Schatten-p norm, with $0 \leq p \leq 2$: the Schatten-p norm is the p-norm applied to the vector of singular values of a matrix. The Schatten p-norm is defined by $||\mathbf{A}||_{S_p} = (\sum_i(\sigma_i(\mathbf{A}))^p)^{1/p}$, where $\sigma_i(\mathbf{A})$ represents the i-th singular value of the matrix $\mathbf{A}$. For $p = 1$ and $p = 2$, it yields the nuclear norm and the Frobenius norm, respectively. The case $p = \infty$ yields the spectral norm.

- Tensor Frobenius norm: $||\mathcal{X}||_F = \sqrt{\sum_i\ldots\sum_N|\mathcal{X}_{i\ldots N}|^2}$ is the Frobenius norm of an N^{th}-order tensor $\mathcal{X}$, and it is defined as the square root of the sum of the absolute values of its elements. The Frobenius norm is sometimes also called the Euclidean norm, which may cause confusion with the vector ℓ_2-norm. The Frobenius norm is also employed in its squared version representing the sum of squares of all entries, such that $||\mathcal{X}||_F^2 = \sum_i\ldots\sum_N|\mathcal{X}_{i\ldots N}|^2$.

Table A.1: Summary of symbols used in this thesis.

x, y, z, X, Y, Z	Scalars (lowercase or uppercase letters)
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Vectors (lowercase bold letters)
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	Matrices (uppercase bold letters)
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	Tensors (uppercase calligraphic bold letters)
$\boldsymbol{x}_i$	ith element of vector $\mathbf{x}$
$\mathbf{X}(i,j), \mathbf{X}_{ij}$	Entry at position (i,j) of matrix $\mathbf{X}$
$\mathcal{X}(i,j,k), \mathcal{X}_{ijk}$	Entry at position (i,j,k) of 3^{th}-order tensor $\mathcal{X}$
$\mathcal{X}(i_1,i_2,\ldots,i_N), \mathcal{X}_{i_1 i_2 \ldots i_N}$	Entry at position $(i_1,i_2,\ldots,i_N)$ of N^{th}-order tensor $\mathcal{X}$
$\mathbf{x}_i$	ith vector
$\mathbf{x}_i^{(t)}$	ith vector at time instance t
$\mathbf{X}_i$	ith matrix
$\mathbf{X}_i^{(t)}$	ith matrix at time instance t
$\mathcal{X}_i$	ith tensor
$\mathcal{X}_i^{(t)}$	ith tensor at time instance t
$\mathbf{X}_{i:}$	A vector formed by all columns of the ith row of a matrix $\mathbf{X}$
$\mathbf{X}_{:j}$	A vector formed by all rows of the jth column of a matrix $\mathbf{X}$
$\mathcal{X}_{:jk}, \mathcal{X}_{i:k}, \mathcal{X}_{ij:}$	Column, row, and tube fibers of a third-order tensor $\mathcal{X}$
$\mathcal{X}_{i::}, \mathcal{X}_{:j:}, \mathcal{X}_{::k}$	Horizontal, lateral and frontal slices of a third-order tensor $\mathcal{X}$
$\mathbf{x}^\mathbf{T}, \mathbf{X}^\mathbf{T}$	Transpose of vector $\mathbf{x}$ and matrix $\mathbf{X}$
$\overline{\mathbf{X}}$	Denotes the complement of the matrix $\mathbf{X}$
$\mathcal{X}^{[n]}$	n-mode matricization of tensor $\mathcal{X}$
$\mathbf{X}^{[n]}$	Matrix representing the n-mode matricization of tensor $\mathcal{X}$
$\mathcal{X}^{(t)[n]}$	n-mode matricization of tensor $\mathcal{X}$ at time instance t
$\mathbf{X}^{(t)[n]}$	Matrix representing the n-mode matricization of tensor $\mathcal{X}$ at time instan
$\mathbf{0} \in \mathbb{R}^{m \times n}$	Zero matrix. A matrix with all its entries being zero.
$\mathbf{1} \in \mathbb{R}^{m \times n}$	All-ones matrix. A matrix where every element is equal to one.
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	The set of all real vectors of length n
$\mathbb{R}^{m \times n}$	The set of all real matrices of size $m \times n$
$\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$	The set of all N^{th}-order real tensors of size $I_1 \times I_2 \times \ldots \times I_N$
$\{\ldots\}$	A set, depending on context
$[x, y]$	Closed interval from x to y
$\lvert . \rvert$	Absolute value of a real number
$\lvert\lvert . \rvert\rvert$	Norm (in general)
$\lvert\lvert . \rvert\rvert_0$	ℓ_0-norm (number of non-zero elements)
$\lvert\lvert . \rvert\rvert_\alpha$	Elementwise ℓ_α-norm
$\lvert\lvert . \rvert\rvert_\infty$	Infinity norm
$\lvert\lvert . \rvert\rvert_{\max}$	Max norm
$\lvert\lvert . \rvert\rvert_F$	Frobenius norm
$\lvert\lvert . \rvert\rvert_{\alpha,\beta}$	Elementwise $\ell_{\alpha,\beta}$-mixed norm (matrices only)
$\lvert\lvert . \rvert\rvert_{S_p}$	Schatten p-norm (matrices only)
$\lvert\lvert . \rvert\rvert_*$	Nuclear norm (matrices only)
$\langle ., . \rangle$	Inner product

$var(\mathbf{x})$	Variance of the elements of the vector $\mathbf{x}$
$std(\mathbf{x})$	Standard deviation of the elements of the vector $\mathbf{x}$
$card(\mathbf{S})$	Denotes the number of non-zero entries of matrix $\mathbf{S}$
$rank(.)$	Matrix or tensor rank
$rank_r(\mathbf{X})$	rank-r approximation of matrix $\mathbf{X}$ (general case)
$svd_r(\mathbf{X})$	rank-r approximation of matrix $\mathbf{X}$ by SVD
$vec(\mathbf{X}), vec(\mathcal{X})$	Vectorization of matrix $\mathbf{X}$ or tensor $\mathcal{X}$
$\mathbf{x} \otimes \mathbf{y}$	Outer product between vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{X} \circ \mathbf{Y}$	Element-wise multiplication between matrices $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{X} \otimes \mathbf{Y}$	Kronecker product between matrices $\mathbf{X}$ and $\mathbf{Y}$
$\mathcal{X} \times_n \mathbf{U}$	n-mode product between tensor $\mathcal{X}$ and matrix $\mathbf{U}$
$\mathcal{X} \times_{i=1}^{N} \mathbf{U}_i$	Shorthand for $\mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \ldots \times_N \mathbf{U}_N$
$P_\Omega(.)$	Sampling operator
$\min(x_1, x_2, \ldots, x_N)$	The smallest among scalars $\{x_i\}$
$\min(\mathbf{x}), \min(\mathbf{X})$	The smallest element of a vector $\mathbf{x}$ or matrix $\mathbf{X}$
$\max(\mathbf{x}), \max(\mathbf{X})$	The biggest element of a vector $\mathbf{x}$ or matrix $\mathbf{X}$
$\min_x f(x)$	The minimum value of real function f with respect to x
$\arg\min_x f(x)$	The minimizer of real function f with respect to x

Appendix B

List of abbreviations

ADM	Alternating Direction Method
ALM	Augmented Lagrange Multipliers
ALS	Alternating Least Squares
APG	Accelerated Proximal Gradient
B/F	Background/Foreground
BCD	Block Coordinate Descent
BM	Background Model
BMI	Background Model Initialization
BMM	Background Model Maintenance
BRP	Bilateral Random Projections
BS	Background Subtraction
CANDECOMP	CANonical DECOMPosition
CP	CANDECOMP/PARAFAC
DLSM	D ecomposition into Low-rank and Sparse Matrices
FD	Foreground Detection
FS	Foreground Segmentation
GSVT	Generalized Singular Value Thresholding
HOSVD	Higher-Order Singular Value Decomposition
IALM	Inexact Augmented Lagrange Multiplier
iHOSVD	Incremental HOSVD
IMTSL	Incremental Multi-feature Tensor Subspace Learning
IRLS	Iteratively Reweighted Least Squares
IRNN	Iteratively Reweighted Nuclear Norm
L/S-SC	Low-rank/Sparse Subspace Clustering
LRA	Low-Rank Approximation
LRR	Low-Rank Representation
MC	Matrix Completion
MF	Matrix Factorization
NMF	Non-negative Matrix Factorization
MoG	Mixture of Gaussians

OSTD	Online Stochastic Tensor Decomposition
PARAFAC	PARAllel FACtors
PCA	Principal Component Analysis
PCP	Principal Component Pursuit
RDL	Robust Dictionary Learning
RM	Rank Minimization
RNMF	Robust Non-negative Matrix Factorization
RPCA	Robust PCA
SC	Subspace Clustering
SCM-RPCA	Shape and Confidence Map-based RPCA
SDP	Semidefinite Programming
SNN	Sum of Nuclear Norms
SSC	Sparse Subspace Clustering
SVD	Singular Value Decomposition
SVT	Singular Value Thresholding
TC	Tensor Completion
TD	Tucker Decomposition
TNN	Truncated Nuclear Norm
t-SVD	Tensor Singular Value Decomposition
TTD	Three Term Decomposition

Appendix C

Introduction to tensors

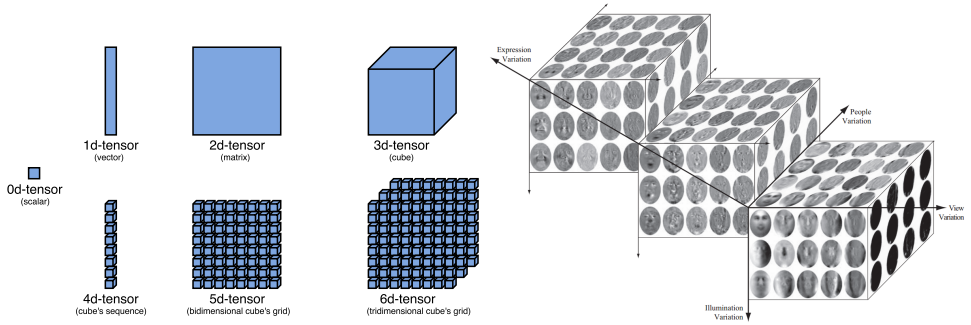


Figure C.1: From left to right: illustration of tensor's dimensionality, and partial visualization of the TensorFaces representation (image from Vasilescu thesis's [210]).

From the point of view of multi-linear algebra, a *tensor* can be considered as a multi-dimensional or multi-way array of data, usually seen as a generalization of the vector concept [99, 104]. For example, a scalar is represented as a 0th-order tensor, a vector as a 1st-order tensor, and a matrix (a 2-dimensional array) as a 2nd-order tensor. The order (also degree or rank) of a tensor is the dimensionality of the array needed to represent it. Tensors of order three or higher are usually called higher-order tensors. Figure C.1 (left) gives an example of tensor's dimensionality. Since the human brain's is limited to three dimensional perception, the visualization of high dimensional data is still a non trivial task. However, several approaches have been proposed in the literature for visualizing data with four or more dimensions [123, 194]. An easy way to interpret the fourth dimension case is to consider a cube's sequence, for example a sequence of color images (a spatio-temporal volume) where its first three dimensions are represented by its width, height and color channels (i.e. RGB color model). In the case of the fifth (or more) dimension, an interesting example is the representation of the TensorFaces proposed by Vasilescu and Terzopoulos [211] (Figure C.1 right). The image database consists of 28 male subjects imaged in 15 different views, under 4 different illuminations, performing 3 different expressions. In Vasilescu and Terzopoulos [211],

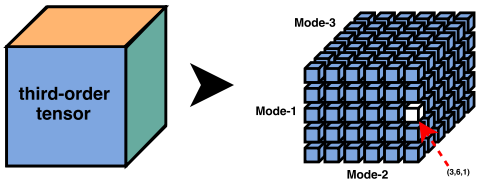
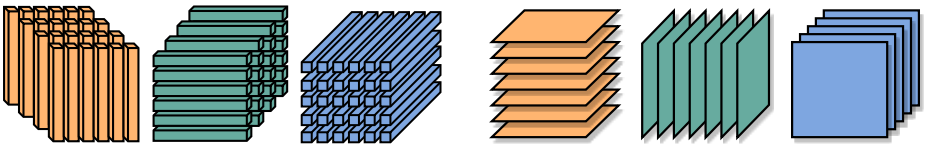


Figure C.2: Illustration of a third-order tensor $\mathcal{X} \in \mathbb{R}^{5\times6\times6}$ and its entries.



(a) Tensor fibers (b) Tensor slices

Figure C.3: Decomposing a third-order tensor into fibers and slices.

the facial image data tensor is represented by $\mathbb{R}^{7943\times28\times15\times4\times3}$, yielding a total of 7943 pixels per image.

Tensors have been widely used in mathematics and physics for decades, and they have become very popular in psychometrics and chemometrics for multi-way data analysis [99]. However, in the last few years, with the accelerated growth of higher-dimensional data sets, the use of tensors has expanded to other fields, such as neuroscience, data mining, signal/image/video processing, computer vision and machine learning, among the others [7, 45, 46, 66, 99, 101, 210].

In the next sections, we introduce the basic operations of multi-linear algebra on tensors. For a deeper discussion on tensors, their properties, their operations and their applications, the reader may refer to [3, 99].

C.1 Tensor basics

As introduced in the previous section, a tensor can be defined as a multidimensional array of data[1]. Following the usual conventions found in the literature [99, 104], a tensor is denoted by calligraphic letters, e.g. $\mathcal{X}$. The order of a tensor is the number of dimensions, also known as ways or modes, and an N-th order tensor of size $I_1 \times I_2 \times \ldots \times I_N$ is defined as $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$. Each element in tensor $\mathcal{X}$ is addressed by $\mathcal{X}_{i_1 i_2 \ldots i_n}$ representing an entry at position $(i_1, i_2, \ldots, i_n)$, where $1 \leq i_j \leq I_j$, $j = 1, \ldots, N$. Figure C.2 shows a third-order tensor of size $5 \times 6 \times 6$ and its modes. In the next sections, we present some of the most important operations that are usually done in tensors.

[1]The definition of tensors used in this thesis should not be confused with *tensor fields* used in physics and differential geometry in mathematics, such as, stress tensor, Einstein tensor, metric tensor, curvature tensor, among the others [99].

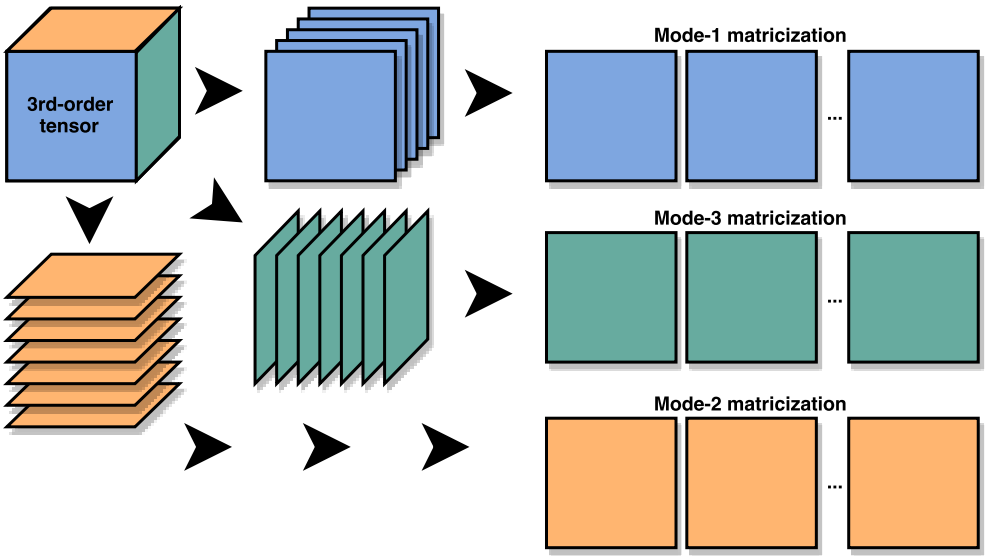


Figure C.4: Matricization of a third-order tensor into its n-mode matrices.

C.2 Fibers and slices

A tensor can be decomposed into subarrays by fixing a subset of its indices. A tensor $fiber$ can be regarded as a one-dimensional fragment (or column vector) of a tensor and it is defined by fixing every index except one. A third-order tensor $\mathcal{X}_{ijk}$ has column, row, and tube fibers denoted by $\mathcal{X}_{:jk}$, $\mathcal{X}_{i:k}$ and $\mathcal{X}_{ij:}$, as can be seen in Figure C.3 (a). A second property is the tensor $slice$. A tensor slice is a two-dimensional section of a tensor, when all but two indices are fixed, resulting in a matrix called slice. A third-order tensor $\mathcal{X}_{ijk}$ has horizontal, lateral and frontal slices indicated by $\mathcal{X}_{i::}$, $\mathcal{X}_{:j:}$ and $\mathcal{X}_{::k}$, as can be seen in Figure C.3 (b). Fibers and slices are the core of the most important operations on tensors, such as vectorization, matricization, n-mode product, among the others [99], some of which are described in the next sections.

C.3 Vectorization and matricization

In order to work with tensors, it is often convenient to represent tensors as vectors or matrices. This process is known as *vectorization* or *matricization*, and consist in reordering the elements of an N-th order tensor into a vector or a matrix, respectively [99]. For example, a $5 \times 6 \times 8$ tensor can be arranged as a vector of 240 elements or a 5×48 matrix.

Definition C.1. (Tensor vectorization). Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order tensor. The vectorized tensor, denoted by $vec(\mathcal{X})$, is a vector formed by the tensor entries, such that tensor entry $(i_1, i_2, \ldots, i_n)$ is mapped to vector entry j, where

$$j = 1 + \sum_{k=1}^{N} (i_k - 1) J_k \quad \text{and} \quad J_k = \prod_{m=1}^{k-1} I_m \tag{C.1}$$

Definition C.2. (Tensor matricization). Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ be an N-th order tensor. The n-mode tensor matricization, with $n \in \{1, 2, \ldots, N\}$, denoted by $\mathcal{X}^{[n]}$, maps the tensor element $(i_1, \ldots, i_l, \ldots, i_n)$ to matrix element (i_l, j), where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq l}}^{N} (i_k - 1) J_k \quad \text{and} \quad J_k = \prod_{\substack{m=1 \\ m \neq l}}^{k-1} I_m \tag{C.2}$$

In essence, the tensor vectorization $vec(\mathcal{X})$ is formed by stacking the entries of $\mathcal{X}$ in column-major order, while in the tensor matricization the n-mode fibers are rearranged to be the columns of the matrix $\mathbf{X}^{[n]}$. Consider the following example:

Example C.1. (Tensor vectorization and matricization) Let a third-order tensor $\mathcal{X} \in \mathbb{R}^{2 \times 3 \times 2}$ formed by the following two frontal slices

$$\mathbf{X}_{::1} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_{::2} = \begin{bmatrix} 7 & 9 & 11 \\ 8 & 10 & 12 \end{bmatrix}; \tag{C.3}$$

then, the vectorization of $\mathcal{X}$ is

$$vec(\mathcal{X}) = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12]^T \tag{C.4}$$

and the three n-mode matrices of $\mathcal{X}$ are

$$\mathbf{X}^{[1]} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 \\ 2 & 4 & 6 & 8 & 10 & 12 \end{bmatrix} \tag{C.5}$$

$$\mathbf{X}^{[2]} = \begin{bmatrix} 1 & 2 & 7 & 8 \\ 3 & 4 & 9 & 10 \\ 5 & 6 & 11 & 12 \end{bmatrix} \tag{C.6}$$

$$\mathbf{X}^{[3]} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix} \tag{C.7}$$

C.4 Other tensor operations

Similarly to vectors and matrices, addition and subtraction between two tensors are defined in a element-wise manner. However, the multiplication between tensors is more complex. For a complete treatment of tensor multiplication, please refer to Kolda and Bader [99]. In the thesis, we focus only on the n-mode product between a tensor and a vector or a matrix.

C.4.1 n-mode tensor vector product

The n-mode vector product of an N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{v}$. Each n-mode fiber is multiplied by the vector $\mathbf{v}$, and usually is expressed by

$$\mathcal{X} \times_n \mathbf{v} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} v_{i_n} \tag{C.8}$$

C.4.2 n-mode tensor matrix product

The n-mode matrix product of an N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$. Each n-mode fiber is multiplied by the matrix $\mathbf{U}$, and usually is expressed by

$$\mathcal{G} = \mathcal{X} \times_n \mathbf{U} \quad \Leftrightarrow \quad \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \tag{C.9}$$

C.4.3 t-product

Let $\mathcal{A}$ be an $I_1 \times I_2 \times I_3$ tensor and $\mathcal{B}$ be an $I_2 \times I_4 \times I_3$. The t-product of $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{C} = \mathcal{A} * \mathcal{B}$, is an $I_1 \times I_4 \times I_3$, defined as follows [97]:

$$\mathcal{C}_{ij:} = \sum_{k=1}^{I_2} \mathcal{A}_{ik:} \bigodot \mathcal{B}_{kj:}, \tag{C.10}$$

where $\bigodot$ denotes a circular convolution.

C.4.4 f-diagonal

An $I_1 \times I_2 \times I_3$ tensor $\mathcal{A}$ is called f-diagonal, if each frontal face of $\mathcal{A}$ is a diagonal matrix [97].

Appendix D

LRSLibrary

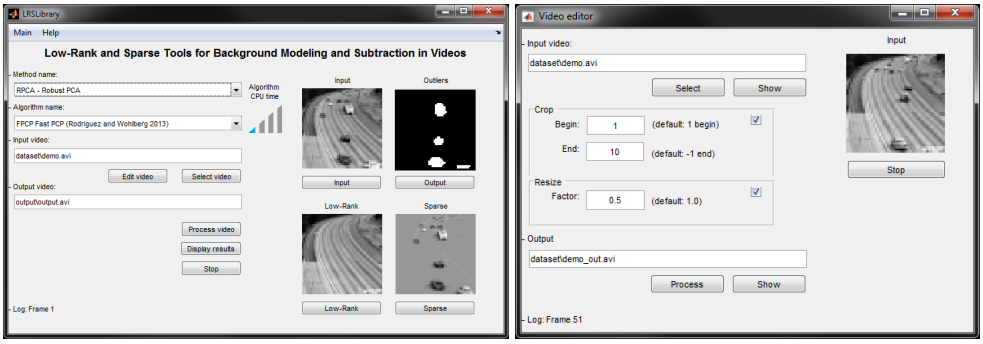


Figure D.1: LRSLibrary GUI.

The LRSLibrary [180][1] provides a collection of low-rank and sparse decomposition algorithms in MATLAB. The library was designed for background/foreground separation in videos, and it contains a total of 104 *matrix*-based and *tensor*-based algorithms. The library is also equipped with an easy-to-use graphical user interface (GUI), enabling the user to select the type of method (e.g. RPCA for Robust PCA) and its related algorithm (e.g. FPCP for Fast PCP), please see Figure D.1 (left). The library also disposes of an additional tool to resize and crop videos (Figure D.1 (right)).

The remainder of this appendix is organized as follows. First we start with the motivation behind the LRSLibrary in Section D.1. Section D.2 presents a brief overview of the algorithms available in the LRSLibrary. Section D.3 evaluates the computational cost of each algorithm and its speed. Finally, in Sections D.4 and D.5, we present a usage example of the LRSLibrary, as well as conclusions.

[1]LRSLibrary: https://github.com/andrewssobral/lrslibrary

D.1 Motivation

The main motivation behind the LRSLibrary was to build an easy-to-use framework for applying low-rank and sparse decomposition tools for the background/foreground separation problem. The library was developed to be open source and free for academic/research purpose. The LRSLibrary was crucial for all experiments conducted in the current thesis.

D.2 Algorithms

Up to the date of writing, the LRSLibrary provided 104 algorithms for B/F. An updated list of currently available algorithms can be found in the library website. The algorithms were grouped into the following categories: **RPCA** for Robust PCA, **ST** for Subspace Tracking, **MC** for Matrix Completion, **TTD** for Three-Term Decomposition, **LRR** for Low-Rank Representation, **NMF** for Non-negative Matrix Factorization, **NTF** for Non-negative Tensor Factorization, or **TD** for standard Tensor Decomposition.

D.3 Computational cost

Many efforts have been recently concentrated to develop low-computational subspace learning algorithms. In this section, an evaluation of the computational cost of the LRSLibrary algorithm's is shown in Figure D.3. It presents the averaged CPU time and the speed classification of each algorithm to decompose a 2304×51 matrix or a $48 \times 48 \times 51$ tensor. The speed classification criterion (SCC) function was defined as:

$$SCC(\bar{t}) = \begin{cases} 1 & \text{if } \bar{t} < 1 \quad \text{(very fast: represented by blue color)} \\ 2 & \text{if } 1 \leq \bar{t} < 5 \quad \text{(fast: represented by green color)} \\ 3 & \text{if } 5 \leq \bar{t} < 20 \quad \text{(medium: represented by yellow color)} \\ 4 & \text{if } 20 \leq \bar{t} < 60 \quad \text{(slow: represented by red color)} \\ 5 & \text{if } \bar{t} \geq 60 \quad \text{(very slow: represented by dark red color)} \end{cases} \tag{D.1}$$

where $\bar{t}$ is the average time (in seconds) over three successive executions. Figure D.2 presents the icons used by LRSLibrary GUI to represent the speed classification of each algorithm. The experiments were performed using an Intel Core i7-3740QM CPU 2.70GHz with 16Gb of RAM running MATLAB R2013b and Windows 7 Professional SP1 64 bits.

Figure D.2: Icons that represent the speed classification of each LRS algorithm.

Method	Algorithm ID	Algorithm Name	Speed classification	CPU Time (in sec)
RPCA	FPCP	Fast PCP (Rodriguez and Wohlberg 2013)	1	0.01
RPCA	GoDec	Go Decomposition (Zhou and Tao 2011)	1	0.01
RPCA	SSGoDec	Semi-Soft GoDec (Zhou and Tao 2011)	1	0.02
NMF	NeNMF	NMF via Nesterovs Optimal Gradient Method (Guan et al. 2012)	1	0.02
RPCA	GM	Grassmann Median (Hauberg et al. 2014)	1	0.03
RPCA	TGA	Trimmed Grassmann Average (Hauberg et al. 2014)	1	0.03
RPCA	GreGoDec	Greedy Semi-Soft GoDec Algotithm (Zhou and Tao, 2013)	1	0.03
RPCA	GA	Grassmann Average (Hauberg et al. 2014)	1	0.04
NMF	Deep-Semi-NMF	Deep Semi Non-negative Matrix Factorization (Trigeorgis et al. 2014)	1	0.04
NMF	LNMF	Spatially Localized NMF (Li et al. 2001)	1	0.05
NMF	INMF	Incremental Subspace Learning via NMF (Bucak and Gunsel, 2009)	1	0.05
RPCA	Lag-SPCP-QN	Lagrangian SPCP solved by Quasi-Newton (Aravkin et al. 2014)	1	0.06
LRR	ROSL	Robust Orthonormal Subspace Learning (Shu et al. 2014)	1	0.08
MC	LRGeomCG	Low-rank matrix completion by Riemannian optimization (Bart Vandereycken, 2013)	1	0.10
TD	Tucker-ALS	Tucker Decomposition solved by ALS	1	0.10
TD	t-SVD	Tensor SVD in Fourrier Domain (Zhang et al. 2013)	1	0.14
TD	CP-ALS	PARAFAC/CP decomposition solved by ALS	1	0.16
RPCA	IALM	Inexact ALM (Lin et al. 2009)	1	0.19
NMF	ManhNMF	Manhattan NMF (Guan et al. 2013)	1	0.20
RPCA	FW-T	SPCP solved by Frank-Wolfe method (Mu et al. 2014)	1	0.22
ST	GRASTA	Grassmannian Robust Adaptive Subspace Tracking Algorithm (He et al. 2012)	1	0.23
NMF	Semi-NMF	Semi Non-negative Matrix Factorization	1	0.24
LRR	FastLADMAP	Fast LADMAP (Lin et al. 2011)	1	0.25
RPCA	R2PCP	Riemannian Robust Principal Component Pursuit (Hintermüller and Wu, 2014)	1	0.26
NMF	NMF-MU	NMF solved by Multiplicative Updates	1	0.28
RPCA	STOC-RPCA	Online Robust PCA via Stochastic Optimization (Feng et al. 2013)	1	0.30
NMF	NMF-PG	NMF solved by Projected Gradient	1	0.30
TTD	MAMR	Motion-Assisted Matrix Restoration (Ye et al. 2015)	1	0.31
TTD	3WD	3-Way-Decomposition (Oreife) et al. 2012)	1	0.34
RPCA	LSADM	LSADM (Goldfarb et al. 2010)	1	0.35
RPCA	PSPG	Partially Smooth Proximal Gradient (Aybat et al. 2012)	1	0.35
NTF	bcuNCP	Non-negative CP Decomposition by block-coordinate update (Xu and Yin, 2012)	1	0.36
NMF	NMF-ALS	NMF solved by Alternating Least Squares	1	0.37
RPCA	AS-RPCA	Active Subspace: Towards Scalable Low-Rank Learning (Liu and Yan, 2012)	1	0.42
LRR	IALM	Inexact ALM (Lin et al. 2009)	1	0.42
LRR	LADMAP	Linearized ADM with Adaptive Penalty (Lin et al. 2011)	1	0.43
RPCA	L1F	L1 Filtering (Liu et al. 2011)	1	0.45
NMF	NMF-ALS-OBS	NMF solved by Alternating Least Squares with Optimal Brain Surgeon	1	0.48
RPCA	NSA1	Non-Smooth Augmented Lagrangian v1 (Aybat et al. 2011)	1	0.50
RPCA	NSA2	Non-Smooth Augmented Lagrangian v2 (Aybat et al. 2011)	1	0.54
MC	GROUSE	Grassmannian Rank-One Update Subspace Estimation (Balzano et al. 2010)	1	0.54
RPCA	RegL1-ALM	Low-Rank Matrix Approximation under Robust L1-Norm (Zheng et al. 2012)	1	0.58
NMF	DRMF	Direct Robust Matrix Factorization (Xiong et al. 2011)	1	0.61
TTD	RMAMR	Robust Motion-Assisted Matrix Restoration (Ye et al. 2015)	1	0.62
RPCA	IALM_LMSVDS	IALM with LMSVDS (Liu et al. 2012)	1	0.70
TTD	ADMM	Alternating Direction Method of Multipliers (Parikh and Boyd, 2014)	1	0.73
LRR	ADM	Alternating Direction Method (Lin et al. 2011)	1	0.74
RPCA	PCP	Principal Component Pursuit (Candes et al. 2009)	1	0.78
RPCA	APG_PARTIAL	Partial Accelerated Proximal Gradient (Lin et al. 2009)	1	0.92
TD	HoSVD	Higher-order Singular Value Decomposition (Tucker Decomposition)	1	0.92
RPCA	DECOLOR	Contiguous Outliers in the Low-Rank Representation (Zhou et al. 2011)	1	0.93
RPCA	APG	Accelerated Proximal Gradient (Lin et al. 2009)	2	1.03
RPCA	VBRPCA	Variational Bayesian RPCA (Babacan et al. 2011)	2	1.07
RPCA	IALM_BLWS	IALM with BLWS (Lin and Wei 2010)	2	1.10
RPCA	PRMF	Probabilistic Robust Matrix Factorization (Wang et al. 2012)	2	1.10
MC	OptSpace	A Matrix Completion Algorithm (Keshavan et al. 2009)	2	1.10
NMF	nmfLS2	Non-negative Matrix Factorization with sparse matrix (Ji and Eisenstein, 2013)	2	1.15
NMF	PNMF	Probabilistic Non-negative Matrix Factorization	2	1.21
RPCA	Lag-SPCP-SPG	Lagrangian SPCP solved by Spectral Projected Gradient (Aravkin et al. 2014)	2	1.50
RPCA	TFOCS-IC	TFOCS with inequality constraints (Becker et al. 2011)	2	1.59
RPCA	TFOCS-EC	TFOCS with equality constraints (Becker et al. 2011)	2	1.62
NMF	ENMF	Exact NMF (Gillis and Glineur, 2012)	2	1.80
NTF	betaNTF	Simple beta-NTF implementation (Antoine Liutkus, 2012)	2	2.14
RPCA	MoG-RPCA	Mixture of Gaussians RPCA (Zhao et al. 2014)	2	2.15
RPCA	EALM	Exact ALM (Lin et al. 2009)	2	2.21
TD	HoRPCA-IALM	HoRPCA solved by IALM (Goldfarb and Qin, 2013)	2	2.42
ST	GOSUS	Grassmannian Online Subspace Updates with Structured-sparsity (Xu et al. 2013)	2	2.48
LRR	EALM	Exact ALM (Lin et al. 2009)	2	2.64
TD	HoRPCA-S	HoRPCA with Singleton model solved by ADAL (Goldfarb and Qin, 2013)	2	2.92
NTF	NTD-APG	Non-negative Tucker Decomposition solved by Accelerated Proximal Gradient (Zhou et al. 2012)	2	3.31
NTF	NTD-MU	Non-negative Tucker Decomposition solved by Multiplicative Updates (Zhou et al. 2012)	2	3.50
RPCA	ADM	Alternating Direction Method (Yuan and Yang 2009)	2	3.58
NTF	bcuNTD	Non-negative Tucker Decomposition by block-coordinate update (Xu and Yin, 2012)	2	3.69
TD	RSTD	Rank Sparsity Tensor Decomposition (Yin Li 2010)	2	3.75
RPCA	RPCA	Robust Principal Component Analysis (De la Torre and Black, 2001)	2	3.84
TD	Tucker-ADAL	Tucker Decomposition solved by ADAL (Goldfarb and Qin, 2013)	2	4.45
NTF	NTD-HALS	Non-negative Tucker Decomposition solved by Hierarchical ALS (Zhou et al. 2012)	2	4.88
MC	FPC	Fixed point and Bregman iterative methods for matrix rank minimization (Ma et al. 2008)	3	8.17
RPCA	ALM	Augmented Lagrange Multiplier (Tang and Nehorai 2011)	3	8.38
ST	pROST	Robust PCA and subspace tracking from incomplete observations using L0-surrogates (Hage and Kleinsteuber, 2013)	3	8.94
MC	SVT	A singular value thresholding algorithm for matrix completion (Cai et al. 2008)	3	9.68
RPCA	flip-SPCP-max-QN	Flip-Flop version of Stable PCP-max solved by Quasi-Newton (Aravkin et al. 2014)	4	27.71
TD	CP-APR	PARAFAC/CP decomposition solved by Alternating Poisson Regression (Chi et al. 2011)	4	29.83
RPCA	DUAL	Dual RPCA (Lin et al. 2009)	4	38.41
TD	CP2	PARAFAC2 decomposition solved by ALS (Bro et al. 1999)	4	39.08
RPCA	OPRMF	Online PRMF (Wang et al. 2012)	4	39.88
RPCA	BRPCA-MD	Bayesian Robust PCA with Markov Dependency (Ding et al. 2011)	4	45.90
RPCA	BRPCA-MD-NSS	BRPCA-MD with Non-Stationary Noise (Ding et al. 2011)	4	46.31
RPCA	MBRMF	Markov BRMF (Wang and Yeung 2013)	4	47.18
TD	HoRPCA-S-NCX	HoRPCA with Singleton model solved by ADAL (non-convex) (Goldfarb and Qin, 2013)	4	48.43
RPCA	OP-RPCA	Robust PCA via Outlier Pursuit (Xu et al. 2012)	4	54.23
RPCA	flip-SPCP-sum-SPG	Flip-Flop version of Stable PCP-sum solved by Spectral Projected Gradient (Aravkin et al. 2014)	4	57.67
RPCA	SVT	Singular Value Thresholding (Cai et al. 2008)	5	168.90

Figure D.3: CPU time consumption and the speed classification of each algorithm.

D.4 Usage example

The LRSLibrary was designed to be easy to use. It contains several ready-to-use functions to help the user to perform B/F by low-rank and sparse representation. Listing D.4 demonstrates how to perform matrix and tensor factorization, given an input video file. The final results are stored in the **out** variable, and the function **show_results** shows the background subtraction process. Please, refer to the online version of **demo.m**[2] file for a complete overview.

```
% First run the setup script
lrs_setup; % or run('C:/lrslibrary/lrs_setup')
% Load configuration
lrs_load_conf;
% Load video file
video = load_video_file(fullfile(lrs_conf.lrs_dir,'dataset','demo.avi'));

%%--------------------------------
%% Demo: Matrix-based factorization
M = im2double(convert_video_to_2d(video));
m = video.height;
n = video.width;
p = video.nrFramesTotal;
opts.rows = m;
opts.cols = n;

% Robust PCA using FPCP algorithm
out = process_matrix('RPCA', 'FPCP', M, opts);
% Subspace Tracking using GRASTA algorithm
out = process_matrix('ST', 'GRASTA', M, opts);
% Matrix Completion using GROUSE algorithm
out = process_matrix('MC', 'GROUSE', M, opts);
%% Low Rank Recovery using FastLADMAP algorithm
out = process_matrix('LRR', 'FastLADMAP', M, opts);
% Three-Term Decomposition using 3WD algorithm
out = process_matrix('TTD', '3WD', M, opts);
% Non-negative Matrix Factorization using ManhNMF algorithm
out = process_matrix('NMF', 'ManhNMF', M, opts);

% Show results
show_out(M,out.L,out.S,out.O,p,m,n);

%%--------------------------------
%% Demo: Tensor-based factorizatin
T = tensor(im2double(convert_video_to_3d(video)));

% Non-Negative Tensor Factorization using bcuNCP algorithm
out = process_tensor('NTF', 'bcuNCP', T);
% Tensor Decomposition using Tucker-ALS algorithm
out = process_tensor('TD', 'Tucker-ALS', T);

% Show results
show_3dtensors(T,out.L,out.S,out.O);
```

[2] https://github.com/andrewssobral/lrslibrary/blob/master/demo.m

D.5 Conclusions

The LRSLibrary provides a wide variety of subspace learning algorithms that can be accessed by an easy-to-use GUI and command line functions. The library was designed to serve as a framework for detection and segmentation of moving objects using robust *matrix*-based and *tensor*-based factorization techniques. The experimental results in speed classification can further help the user to choose the best algorithm for his own experiments. We expect to continuously improve the LRSLibrary, adding new features and new subspace learning methods.

D.6 Acknowledgments

We would like to thank everyone who have contributed in some way, e.g. by making the algorithms publicly available, collaborating to the success of this library.

Appendix E

List of publications

The thesis has led to the following publications[1]:

Talks

- **2016** - **Sobral, Andrews**. "Recent advances on low-rank and sparse decomposition for moving object detection.". Workshop/atelier: Enjeux dans la détection d'objets mobiles par soustraction de fond. Reconnaissance de Formes et Intelligence Artificielle (RFIA), 2016[2].

Journal papers

- **2017** - **Sobral, Andrews**; Gong, Wenjuan; Gonzalez, Jordi; Bouwmans, Thierry; Zahzah, El-hadi. "Robust Subspace Clustering of Human Activities from 3D Skeletal Data", (*in progress*).
- **2016** - **Sobral, Andrews**; Zahzah, El-hadi. "Matrix and Tensor Completion Algorithms for Background Model Initialization: A Comparative Evaluation", In the Special Issue on Scene Background Modeling and Initialization (SBMI), Pattern Recognition Letters (PRL), 2016. [184].
- **2016** - Gong, Wenjuan; Zhang, Xuena; Gonzalez, Jordi; **Sobral, Andrews**; Bouwmans, Thierry; Tu, Changhe; Zahzah, El-hadi. "Human Pose Estimation from Monocular Images: A Comprehensive Survey", Sensors, 2016. [73].

[1] The reader can refer to https://scholar.google.fr/citations?user=0Nm0uHcAAAAJ for an updated list of publications and their citations.

[2] http://rfia2016.iut-auvergne.com/index.php/autres-evenements/detection-d-objets-mobiles-par-soustraction-de-fond

- **2016** - Bouwmans, Thierry; **Sobral, Andrews**; Javed, Sajid; Ki Jung, Soon; Zahzah, El-Hadi. "Decomposition into Low-rank plus Additive Matrices for Background/-Foreground Separation: A Review for a Comparative Evaluation with a Large-Scale Dataset", Computer Science Review, 2016. [27].

Books

- **2017** - Bouwmans, Thierry; **Sobral, Andrews**; Zahzah, El-hadi. Handbook on "Background Subtraction for Moving Object Detection: Theory and Practices", (*in progress*)[3].

Book chapters

- **2017** - **Sobral, Andrews**; Bouwmans, Thierry; Zahzah, El-hadi. "Robust Tensor Models". Chapter in the handbook "Background Subtraction for Moving Object Detection: Theory and Practices", (*in progress*).
- **2015** - **Sobral, Andrews**; Bouwmans, Thierry; Zahzah, El-hadi. "LRSLibrary: Low-Rank and Sparse tools for Background Modeling and Subtraction in Videos". Chapter in the handbook "Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing", CRC Press, Taylor and Francis Group, 2015. [180].

Conferences

- **2015** - **Sobral, Andrews**; Javed, Sajid; Ki Jung, Soon; Bouwmans, Thierry; Zahzah, El-hadi. "Online Stochastic Tensor Decomposition for Background Subtraction in Multispectral Video Sequences". ICCV Workshop on Robust Subspace Learning and Computer Vision (RSL-CV), Santiago, Chile, December, 2015. [182].
- **2015** - Javed, Sajid; Ho Oh, Seon; **Sobral, Andrews**; Bouwmans, Thierry; Ki Jung, Soon. "Background Subtraction via Superpixel-based Online Matrix Decomposition with Structured Foreground Constraints". ICCV Workshop on Robust Subspace Learning and Computer Vision (RSL-CV), Santiago, Chile, December, 2015. [90].
- **2015** - **Sobral, Andrews**; Bouwmans, Thierry; Zahzah, El-hadi. "Comparison of Matrix Completion Algorithms for Background Initialization in Videos". Scene Background Modeling and Initialization (SBMI), Workshop in conjunction with ICIAP 2015, Genova, Italy, September, 2015. [178].
- **2015** - **Sobral, Andrews**; Bouwmans, Thierry; Zahzah, El-hadi. "Double-constrained RPCA based on Saliency Maps for Foreground Detection in Automated Maritime Surveillance". Identification and Surveillance for Border Control (ISBC), International Workshop in conjunction with AVSS 2015, Karlsruhe, Germany, August, 2015. [179].

[3] https://sites.google.com/site/foregrounddetection/

- **2015** - Javed, Sajid; **Sobral, Andrews**; Bouwmans, Thierry; Ki Jung, Soon. "OR-PCA with Dynamic Feature Selection for Robust Background Subtraction". In Proceedings of the 30th ACM/SIGAPP Symposium on Applied Computing (ACM-SAC), Salamanca, Spain, 2015. [91].
- **2014** - Javed, Sajid; Ho Oh, Seon; **Sobral, Andrews**; Bouwmans, Thierry; Ki Jung, Soon. "OR-PCA with MRF for Robust Foreground Detection in Highly Dynamic Backgrounds". In the 12th Asian Conference on Computer Vision (ACCV 2014), Singapore, November, 2014. [89].
- **2014** - **Sobral, Andrews**; Baker, Christopher G.; Bouwmans, Thierry; Zahzah, Elhadi. "Incremental and Multi-feature Tensor Subspace Learning applied for Background Modeling and Subtraction". International Conference on Image Analysis and Recognition (ICIAR'2014), Vilamoura, Algarve, Portugal, October, 2014. [176].

Websites

- Andrews Sobral's homepage
  http://andrewssobral.wixsite.com/home
- Publons
  https://publons.com/author/619460/andrews-sobral#profile
- Behance.net project
  http://be.net/andrewssobral
- GitHub profile
  https://github.com/andrewssobral
- LRSLibrary - Low-Rank and Sparse tools for Background Modeling and Subtraction in Videos
  https://github.com/andrewssobral/lrslibrary
- MTT - Matlab Tensor Tools for Computer Vision
  https://github.com/andrewssobral/mtt
- IMTSL - Incremental and Multi-feature Tensor Subspace Learning
  https://github.com/andrewssobral/imtsl
- OSTD - Online Stochastic Tensor Decomposition:
  https://github.com/andrewssobral/ostd

Social networks

- Academia
  http://univ-larochelle.academia.edu/AndrewsSobral
- ResearchGate
  http://www.researchgate.net/profile/Andrews_Sobral

Bibliography

[1] An introduction to video content analysis – industry guide. 1

[2] J. Abernethy, F. Bach, T. Evgeniou, and J-P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, June 2009. 14

[3] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering*, 2009. xiii, 26, 27, 28, 112

[4] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM SIGGRAPH*, SIGGRAPH '04, pages 294–302, New York, NY, USA, 2004. ACM. 34

[5] J. K. Aggarwal and L. Xia. Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, 2014. 86

[6] M. Ahmad and S.-W. Lee. Hmm-based human action recognition using multiview image sequences. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2006. 85

[7] S. Aja-Fernndez, R. de L. Garca, D. Tao, and X. Li. *Tensors in Image Processing and Computer Vision*. Springer Publishing Company, Incorporated, 1st edition, 2009. 112

[8] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 85

[9] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science Journal*, 209(1-2):237–260, December 1998. 17

[10] A. Y. Aravkin, S. Becker, V. Cevher, and P. Olsen. A variational approach to stable principal component pursuit. *The Conference on Uncertainty in Artificial Intelligence*, 2014. 18, 62

[11] N. A. Azis, H.-j. Choi, and Y. Iraqi. Substitutive skeleton fusion for human action recognition. In *International Conference on Big Data and Smart Computing (BigComp)*, 2015. 88

[12] S. D. Babacan, S. Nakajima, and M. N. Do. Probabilistic low-rank subspace clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 24, 88

[13] C. G. Baker, K. A. Gallivan, and P. Van Dooren. Low-rank incremental methods for computing dominant singular subspaces. *Linear Algebra and its Applications*, 2012. 69

[14] A. Ball, D. Rye, F. Ramos, and M. Velonaki. Unsupervised clustering of people from skeleton data. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012. 88

[15] L. Balzano and S. J. Wright. On GROUSE and incremental SVD. In *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013. 40, 42, 69

[16] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, March 2009. 21

[17] Y. Benezeth, D. Sidibe, and J. B. Thomas. Background subtraction with multispectral video sequences. In *International Conference on Robotics and Automation (ICRA)*, 2014. 75, 80

[18] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, January 2009. 21

[19] D. D. Bloisi, L. Iocchi, and A. Pennisi. Mar - maritime activity recognition dataset, 2013. 62

[20] D. D. Bloisi, A. Pennisi, and L. Iocchi. Background modeling in the maritime domain. *Machine Vision and Applications (MVA)*, 2014. 58

[21] D. D. Bloisi, A. Pennisi, and L. Iocchi. Parallel multi-modal background modeling. *Pattern Recognition Letters*, pages –, 2016. 34

[22] A. Borji, M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *CoRR*, abs/1411.5878, 2014. 59

[23] T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, 1991. 88, 89

[24] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. In *Computer Science Review*, 2014. 4, 34, 58

[25] T. Bouwmans, L. Maddalena, and A. Petrosino. Scene background initialization: a taxonomy. *Pattern Recognition Letters*, 2017. xi, 2, 33, 34, 55

[26] T. Bouwmans, F. Porikli, B. Hferlin, and A. Vacavant. *Background Modeling and Foreground Detection for Video Surveillance*. Chapman & Hall/CRC, 1st edition, 2014. 1, 2, 3

[27] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 2016. 4, 6, 9, 11, 20, 124

[28] T. Bouwmans and E. Zahzah. Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance. In *Special Issue on Background Models Challenge, Computer Vision and Image Understanding (CVIU)*, 2014. 4, 11, 34, 58

[29] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 20

[30] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 2000. 88

[31] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications*, 2006. 69

[32] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision (ECCV)*, ECCV'10, pages 282–295, Berlin, Heidelberg, 2010. Springer-Verlag. 25

[33] V. V. Buldygin and Yu. V. Kozachenko. Sub-Gaussian random variables. *Ukrainian Mathematical Journal*, 32(6):483–489, 1980. 41

[34] J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 1978. 69

[35] S. Burer and R. DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 2003. 77

[36] P. A. Businger. Updating a singular value decomposition. *Nordisk Tidskr*, 1970. 69

[37] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010. 16, 20, 39

[38] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust Principal Component Analysis? *Journal of the ACM*, 2011. 16, 17, 18, 20, 22, 30, 34, 62

[39] E. J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 2010. 15, 39, 40

[40] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009. 15, 39

[41] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572, 2011. 16

[42] X. Chang, F. Nie, Z. Ma, Y. Yang, and X. Zhou. A convex formulation for spectral shrunk clustering. In *AAAI Conference on Artificial Intelligence*, 2015. 92

[43] J. Chen and J. Yang. Robust subspace segmentation via low-rank representation. *IEEE Transactions on Cybernetics*, 2014. 22, 24, 86, 88

[44] L. Chen, H. Wei, and J. Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 2013. 86

[45] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015. 26, 28, 29, 112

[46] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing, 2009. 112

[47] E. Cippitelli, S. Gasparrini, Ennio Gambi, and Susanna Spinsante. A human activity recognition system using skeleton data from rgbd sensors. *Journal of Computational Intelligence and Neuroscience*, 2016. 99

[48] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision (IJCV)*, 1998. 88, 89

[49] M. A. Davenport and J. Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016. 9

[50] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997. 85

[51] M. De Gregorio and M. Giordano. *Background Modeling by Weightless Neural Networks*, pages 493–501. Springer International Publishing, Cham, 2015. 34

[52] F. De La Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision - Special Issue on Computational Vision*, 54(1-3):117–142, August 2003. 34

[53] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008. 28

[54] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo. Space-time pose representation for 3D human action recognition. In A. Petrosino, L. Maddalena, and P. Pala, editors, *New Trends in Image Analysis and Processing – ICIAP 2013*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013. 87

[55] T. T. Do, Y. Chen, N. Nguyen, L. Gan, and T. D. Tran. A fast and efficient heuristic nuclear-norm algorithm for affine rank minimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3393–3396, April 2009. 15

[56] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 12

[57] A. Elgammal. *Background Subtraction: Theory and Practice*. Morgan & Claypool Publishers, 2014. 1, 2, 3

[58] E. Elhamifar, G. Sapiro, and S. S. Sastry. Dissimilarity-based sparse subset selection. *CoRR*, abs/1407.6810, 2014. 86

[59] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 22, 23, 24, 25, 86, 88, 89, 93

[60] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *CoRR*, abs/1203.1005, 2012. 86, 88

[61] N. B. Erichson, S. Voronin, S. L. Brunton, and J. N. Kutz. Randomized matrix decompositions using r. In *Journal of Statistical Software*, 2016. 41, 84, 104

[62] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. 21

[63] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, 2001. 15

[64] J. Feng, H. Xu, and S. Yan. Online robust PCA via stochastic optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2013. 76, 78

[65] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. 21

[66] E. Frolov and I. Oseledets. Tensor methods and recommender systems. *CoRR*, abs/1603.06038, 2016. 112

[67] C. Gao, N. Wang, Q. Yu, and Z. Zhang. A feasible nonconvex relaxation approach to feature selection. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 356–361. AAAI Press, 2011. 21

[68] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698, December 2009. 21

[69] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points – online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory*. 2015. 21

[70] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, July 1995. 21

[71] J. Goes, T. Zhang, R. Arora, and G. Lerman. Robust Stochastic Principal Component Analysis. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014. 76

[72] D. Goldfarb and Z. T. Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 2014. 26, 27, 29, 30, 43, 77, 80, 81

[73] W. Gong, X. Zhang, J. Gonzalez, A. Sobral, T. Bouwmans, C. Tu, and E. Zahzah. Human pose estimation from monocular images: A comprehensive survey. *Sensors*, 16(12), 2016. 6, 85, 123

[74] C. A. G. Gonzalez, O. Absil, P.-A. Absil, M. Van Droogenbroeck, D. Mawet, and J. Surdej. Low-rank plus sparse decomposition for exoplanet detection in direct-imaging adi sequences - the llsg algorithm. *A&A*, 589:A54, 2016. 11

[75] L. Grasedyck, M. Kluge, and S. Krämer. Variants of alternating least squares tensor completion in the Tensor-Train format. *SIAM Journal on Scientific Computing*, 37(5):A2424–A2450, 2015. 44

[76] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. 2013. 26, 43, 44

[77] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, June 2014. 21

[78] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011. 41, 84, 104

[79] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 2013. 85

[80] J. He, L. Balzano, and A. Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1568–1575, 2012. 40, 71

[81] X. He, D. Cai, and P. Niyogi. Tensor subspace analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2005. 67

[82] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *ACM*, 60(6):45:1–45:39, 2013. 28, 44

[83] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *International Journal of Computer Vision (IJCV)*, 2011. 26, 68

[84] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang. A new low-rank tensor model for video completion. *CoRR*, abs/1509.02027, 2015. 43

[85] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(9):2117–2130, Sept 2013. 21

[86] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 427–435. JMLR Workshop and Conference Proceedings, 2013. 21

[87] P. Jain, R. Meka, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems (NIPS)*. 2010. 40, 42

[88] P. Jain and S. Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1431–1439. 2014. 43

[89] S. Javed, S. Ho Oh, A. Sobral, T. Bouwmans, and S. Ki Jung. OR-PCA with MRF for robust foreground detection in highly dynamic backgrounds. In *Asian Conference on Computer Vision (ACCV)*, 2014. 76, 125

[90] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung. Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 930–938, Dec 2015. 76, 124

[91] S. Javed, A. Sobral, T. Bouwmans, and S. Ki Jung. OR-PCA with dynamic feature selection for robust background subtraction. In *ACM Symposium on Applied Computing*, 2015. 76, 125

[92] C. Jin, S. M. Kakade, and P. Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*. 2016. 21

[93] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision (ICCV)*, 2001. 24, 86

[94] Z. Kang, C. Peng, and Q. Cheng. Robust PCA via nonconvex rank approximation. *IEEE International Conference on Data Mining (ICDM)*, pages 211–220, 2015. 21, 22

[95] H. Kasai and B. Mishra. Low-rank tensor completion: a Riemannian manifold preconditioning approach. *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 1012—1021, 2016. 44

[96] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *The Journal of Machine Learning Research*, 2010. 41, 42

[97] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641 – 658, 2011. 31, 44, 115

[98] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis. Dance analysis using multiple kinect sensors. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014. 88

[99] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM REVIEW*, 51(3):455–500, 2009. xiii, 26, 27, 28, 29, 43, 77, 80, 81, 111, 112, 113, 114

[100] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2013. 43

[101] P. M. Kroonenberg. *Applied multiway data analysis*. Wiley series in probability and statistics. Hoboken, N.J. Wiley-Interscience, 2008. 112

[102] A. H. S. Lai and N. H. C. Yung. A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 241–244, 1998. 33

[103] I. Laptev and T. Lindeberg. Space-time interest points. In *International Conference on Computer Vision (ICCV)*, 2003. 85

[104] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, March 2000. 27, 28, 111, 112

[105] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r1 ,r2 ,. . .,rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 2000. 26, 27, 43

[106] B. Laugraud, S. Piérard, and M. Van Droogenbroeck. Labgen: A method based on motion detection for generating the background of a scene. *Pattern Recognition Letters*, pages –, 2016. 34

[107] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, may 2015. 21

[108] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, oct 1999. 40

[109] G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *CoRR*, abs/1406.6145, 2016. 21

[110] A. Levy and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 2000. 69

[111] C.-G. Li and R. Vidal. Structured sparse subspace clustering: A unified optimization framework. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 24, 86

[112] X. Li, W. Hu, Z. Zhang, and X. Zhang. Robust foreground segmentation based on two effective background models. In *ACM International Conference in Multimedia Retrieval*, 2008. 26, 68

[113] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Robust visual tracking based on incremental tensor subspace learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. 68

[114] X. R. Li, K. Zhang, and T. Jiang. Minimum entropy clustering and applications to gene expression analysis. In *IEEE Computational Systems Bioinformatics Conference (CSB)*, 2004. 88

[115] Y. Li, J. Yan, Y. Zhou, and J. Yang. Optimum subspace learning and error correction for tensors. In *European Conference on Computer Vision*, 2010. 29, 30

[116] X. Lin and G. Wei. Generalized non-convex non-smooth sparse and low rank minimization using proximal average. *Neurocomputing*, 174, Part B:1116 – 1124, 2016. 21

[117] Z. Lin. A review on low-rank models in data analysis. *Big Data and Information Analytics*, 1(2/3):139–161, 2016. 9, 20, 21

[118] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Mathematical Programming*, 2010. 42, 60, 71

[119] Z. Lin, R. Liu, and H. Li. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2):287–325, May 2015. 21

[120] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 612–620, 2011. 21

[121] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. xiii, 22, 23, 86, 88, 89, 93

[122] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 35, pages 208–220, 2013. 30, 43

[123] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015. 111

[124] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009. 15

[125] Z. Liu, F. Zhou, X. Chen, X. Bai, and C. Sun. Iterative infrared ship target segmentation based on multiple features. *Pattern Recognition*, 2014. 58

[126] L. Lo Presti and M. La Cascia. 3D skeleton-based human action classification: A survey. *Pattern Recognition*, 2015. 86

[127] B. Lois and N. Vaswani. Online matrix completion and online robust PCA. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1826–1830, 2015. 40

[128] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5249–5257, June 2016. 29, 31

[129] C. Lu, Z. Lin, and S. Yan. Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. *IEEE Transactions on Image Processing*, 24(2):646–654, Feb 2015. 21

[130] C. Lu, J. Tang, S. Yan, and Z. Lin. Generalized nonconvex nonsmooth low-rank minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4130–4137, June 2014. 21

[131] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin. Generalized singular value thresholding. In *AAAI Conference on Artificial Intelligence*, 2015. 21

[132] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 2011. 26, 43

[133] Q. Lyu, Z. Lin, Y. She, and C. Zhang. A comparison of typical ℓ_p minimization algorithms. *Neurocomputing*, 119:413 – 424, 2013. Intelligent Processing Techniques for Semantic-based Image and Video Retrieval. 21

[134] L. Maddalena and A. Petrosino. The SOBS algorithm: What are the limits? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26, June 2012. 34

[135] L. Maddalena and A. Petrosino. Background model initialization for static cameras. In *Background Modeling and Foreground Detection for Video Surveillance*. CRC Press, Taylor and Francis Group, 2014. 2, 33, 34

[136] L. Maddalena and A. Petrosino. Towards benchmarking scene background initialization. *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops, Lecture Notes in Computer Science*, 9281:469–476, 2015. 34, 35, 45

[137] V. Mahadevan and N. Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 62, 63

[138] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning (ICML)*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM. 10

[139] J. Melenchón and E. Martínez. Efficiently downdating, composing and splitting singular value decompositions preserving the mean information. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, Lecture Notes in Computer Science. Springer, 2007. 69

[140] D. Meng and F. D. L. Torre. Robust matrix factorization with unknown noise. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1337–1344, Dec 2013. 11

[141] A. Mirzal. Similarity-based matrix completion algorithm for latent semantic indexing. In *IEEE International Conference on Control System, Computing and Engineering*, pages 79–84, Nov 2013. 14

[142] M. Mørup and L. K. Hansen. Automatic relevance determination for multi-way models. *Journal of Chemometrics*, 2009. 80

[143] M. Müller. *Information Retrieval for Music and Motion*. 2007. 90

[144] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 1957. 93

[145] Y. Nesterov. A method of solving a convex programming problem with convergence rate o(1/sqr(k)). *Soviet Mathematics Doklady*, 27:372–376, 1983. 21

[146] P. Netrapalli, N. Uma Naresh, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust PCA. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1107–1115. Curran Associates, Inc., 2014. 21, 22

[147] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances on Neural Information Processing Systems (NIPS)*, 2001. 89, 92

[148] T. Ngo and Y. Saad. Scaled gradients on Grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems (NIPS)*. 2012. 42

[149] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1583–1590, Nov 2011. xiii, 25

[150] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(6):1187–1200, June 2014. 25

[151] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):831–843, 2000. 34

[152] O. Oreifej, Xin Li, and M. Shah. Simultaneous video stabilization and moving object detection in turbulence. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. 11, 59, 60, 62

[153] I. V. Oseledets. Tensor-Train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. 44

[154] M. Oszust and M. Wysocki. Recognition of signed expressions observed by kinect sensor. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2013. 88

[155] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014. 15

[156] D. Park, C. Caramanis, and S. Sanghavi. Greedy subspace clustering. In *Advances on Neural Information Processing Systems (NIPS)*, 2014. 86

[157] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter*, 2004. 88

[158] V. M. Patel. Sparse representations, compressive sensing and dictionaries for pattern recognition. In *Asian Conference on Pattern Recognition (ACPR)*, 2011. 10

[159] V. M. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse subspace clustering. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 23, 93

[160] V. M. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse and low-rank subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 2015. 24

[161] H. Pazhoumand-Dar, C.-P. Lam, and M. Masek. Joint movement similarities for robust 3D action recognition using skeletal data. *Journal of Visual Communication and Image Representation*, 2015. 87

[162] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 2010. 85

[163] G. Ramirez-Alonso, J. A. Ramirez-Quintana, and M. I. Chacon-Murguia. Temporal weighted learning model for background estimation with an automatic re-initialization stage and adaptive parameters update. *Pattern Recognition Letters*, pages –, 2017. 34

[164] S. R. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 24, 25, 86

[165] B. Recht, M. Fazel, and P. A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 2010. 14, 15, 20, 77

[166] V. Reddy, C. Sanderson, and B. C. Lovell. A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. *Journal on Image and Video Processing - Special issue on advanced video-based surveillance*, 2011:1:1–1:14, January 2011. 34

[167] J. DM Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *International Conference on Machine Learning (ICML)*, 2005. 77

[168] G. Roffo. Report: Feature selection techniques for classification. *CoRR*, abs/1607.01327, 2016. 99, 104

[169] D. A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision (IJCV)*, 2008. 69

[170] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2004. 85

[171] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional Sift descriptor and its application to action recognition. In *ACM International Conference on Multimedia*, 2007. 85

[172] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013. 91, 99

[173] S. H. Shaikh, K. Saeed, and N. Chaki. *Moving Object Detection Using Background Subtraction*. Springer International Publishing, 2014. 1

[174] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 86, 87, 90, 99

[175] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012. 86

[176] A. Sobral, C. G. Baker, T. Bouwmans, and E. Zahzah. Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction. In *International Conference on Image Analysis and Recognition (ICIAR'2014)*, Vilamoura, Algarve, Portugal, 2014. 6, 26, 67, 125

[177] A. Sobral, T. Bouwmans, and E. Zahzah. LRS Library: low-rank and sparse tools for background modeling and subtraction in videos. In *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor and Francis Group. 84

[178] A. Sobral, T. Bouwmans, and E. Zahzah. *Comparison of matrix completion algorithms for background initialization in videos*, volume 9281. 2015. 5, 33, 34, 124

[179] A. Sobral, T. Bouwmans, and E. Zahzah. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2015. 5, 57, 124

[180] A. Sobral, T. Bouwmans, and E. Zahzah. LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos. In Taylor CRC Press and Francis Group, editors, *Handbook on "Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing"*. 2016. 5, 33, 117, 124

[181] A. Sobral, W. Gong, J. Gonzalez, T. Bouwmans, and E. Zahzah. Robust subspace clustering of human activities from 3D skeletal data. 2017. 6, 85

[182] A. Sobral, S. Javed, S.K. Jung, T. Bouwmans, and E. Zahzah. Online stochastic tensor decomposition for background subtraction in multispectral video sequences. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015. 6, 26, 29, 75, 124

[183] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding (CVIU), Special Issue on Background Models Challenge (BMC)*, 2014. 4, 58

[184] A. Sobral and E. Zahzah. Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. *Pattern Recognition Letters*, 2016. 5, 33, 34, 123

[185] M. Soltanolkotabi, Eh. Elhamifar, and E. J. Candès. Robust subspace clustering. *CoRR*, abs/1301.2603, 2013. 23, 86, 88

[186] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252 Vol. 2, August 1999. 34

[187] J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere. *CoRR*, abs/1504.06785, 2015. 21

[188] J. Sun, Q. Qu, and J. Wright. When are nonconvex problems not scary? In *Advances in Neural Information Processing Systems (NIPS)*. 2015. 21

[189] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2008. 68

[190] Q. Sun, S. Xiang, and J. Ye. Robust principal component analysis via capped norms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 311–319, New York, NY, USA, 2013. ACM. 21

[191] A. Tagliasacchi. Skeletal representations and applications. *CoRR*, abs/1301.6809, 2013. 90

[192] H. Tan, B. Cheng, J. Feng, G. Feng, W. Wang, and Y.-J. Zhang. Low-n-rank tensor recovery based on multi-linear augmented Lagrange multiplier method. *Neurocomputing*, 119:144 – 152, 2013. 26, 30

[193] G. Tang and A. Nehorai. Lower bounds on the mean-squared error of low-rank matrix reconstruction. *IEEE Transactions on Signal Processing*, 2011. 40

[194] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297, 2016. 111

[195] L. Tao and R. Vidal. Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015. 87

[196] P. D. Tao and L. T. H. An. Convex analysis approach to d.c. programming: theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289—-355, 1997. 22

[197] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 86

[198] S. Tierney, Y. Guo, and J. Gao. Segmentation of subspaces in sequential data. *CoRR*, abs/1504.04090, 2015. 86

[199] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computing*, 1999. 88

[200] K-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 2010. 15

[201] G. Tomasi and R. Bro. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75(2):163 – 180, 2005. 29

[202] G. Tomasi and R. Bro. A comparison of algorithms for fitting the PARAFAC model. *Computational Statistics & Data Analysis*, 50(7):1700 – 1734, 2006. 29

[203] R. Tomioka and T. Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1331–1339, 2013. 45

[204] L. Tran, C. Navasca, and J. Luo. Video detection anomaly via low-rank and sparse decompositions. In *Western New York Image Processing Workshop (WNYIPW)*, 2012. 26, 29, 30

[205] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007. 25

[206] J. Trzasko and A. Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization. *IEEE Transactions on Medical Imaging*, 28(1):106–121, Jan 2009. 21

[207] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre. A benchmark dataset for foreground/background extraction. *Background Models Challenge (BMC) at Asian Conf. on Computer Vision (ACCV), LNCS*, 2012. 71

[208] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 2008. 93

[209] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 2013. 41, 42

[210] M. A. O. Vasilescu. A multilinear (tensor) algebraic framework for computer graphics, computer vision, and machine learning. xv, 27, 111, 112

[211] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision (ECCV)*, pages 447–460, 2002. 27, 67, 111

[212] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 87, 90, 91, 99

[213] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 2011. 86, 88, 89

[214] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters (PRL)*, 2014. 24, 86, 88, 89, 93

[215] R. Vidal, Yi Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 25, 88

[216] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and GPCA. *International Journal of Computer Vision (IJCV)*, 2008. 24, 25, 86

[217] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *International Conference on Data Engineerin*, 2002. 87

[218] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007. 89

[219] H. Wang and N. Ahuja. Facial expression decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, 2003. 67

[220] H. Wang and D. Suter. *A Novel Robust Statistical Method for Background Initialization and Visual Surveillance*, pages 328–337. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 34

[221] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 90

[222] S. Wang, D. Liu, and Z. Zhang. Nonconvex relaxation approaches to robust matrix recovery. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1764–1770. AAAI Press, 2013. 21, 23, 24, 86

[223] Y. Wang, W.f Yin, and J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. Technical report, 2016. 21

[224] Z. Wang, M. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye. Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM Journal on Scientific Computing*, 2015. 42

[225] D. Weinland, R. Ronfard, and E. Boyer. Motion history volumes for free viewpoint action recognition. In *IEEE International Workshop on Modeling People and Human Interaction*, 2005. 85

[226] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding (CVIU)*, 2011. 85

[227] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 2012. 40, 42

[228] R. Witten and E. Candès. Randomized algorithms for low-rank matrix factorizations: Sharp performance bounds. *Algorithmica*, 2015. 41, 84, 104

[229] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 2080–2088. Curran Associates, Inc., 2009. 16, 17, 18

[230] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012. 87, 91, 99

[231] S. Xiang, X. Shen, and J. Ye. Efficient nonconvex sparse group feature selection via continuous and discrete optimization. *Artificial Intelligence*, 224:28 – 50, 2015. 21

[232] H. Xiao, Y. Liu, and M. Zhang. Fast ℓ_1-minimization algorithm for robust background subtraction. *EURASIP Journal on Image and Video Processing*, 2016(1):45, 2016. 10

[233] J. Xu, K. Xu, K. Chen, and J. Ruan. Reweighted sparse subspace clustering. *Computer Vision and Image Understanding (CVIU)*, 2015. 24, 86, 93

[234] Y. Xu, J. Dong, B. Zhang, and D. Xu. Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Transactions on Intelligence Technology*, 1(1):43 – 60, 2016. 4

[235] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems and Imaging*, 9(2):601–624, 2015. 43

[236] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal Imaging Sciences*, 6(3):1758–1789, 2013. 29, 43

[237] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 2012. 41, 42

[238] C. Yang, D. Robinson, and R. Vidal. Sparse subspace clustering with missing entries. In *International Conference on Machine Learning*, 2015. 24, 86

[239] J. Yang, X. Sun, X. Ye, and K. Li. Background extraction from video sequences via motion-assisted matrix completion. In *IEEE International Conference on Image Processing (ICIP)*, 2014. 59

[240] X. Yang and Y. Tian. Effective 3D action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 2014. 87

[241] X. Ye, J. Yang, X. Sun, K. Li, C. Hou, and Y. Wang. Foreground-background separation from video clips via motion-assisted matrix restoration. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2015. 42, 59, 60, 62

[242] X. Yi, D. Park, Y. Chen, and C. Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*. 2016. 21, 22

[243] T. Yokota, Q. Zhao, C. Li, and A. Cichocki. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016. 29, 43

[244] S. You. A non-convex admm heuristic for a low-rank matrix approximation. Technical report, 2014. 21

[245] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010. 21

[246] H. Zhang and O. Yoshie. Improving human activity recognition using subspace clustering. In *International Conference on Machine Learning and Cybernetics (ICMLC)*, 2012. 88

[247] J. Zhang and S. Sclaroff. Saliency detection: a boolean map approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 59

[248] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: a boolean map approach. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014. 59

[249] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, March 2010. 21

[250] Ying Zhang. Restricted low-rank approximation via ADMM. *CoRR*, abs/1512.01748, 2015. 21

[251] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3842–3849, 2014. 31, 43, 44

[252] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1751–1763, 2015. 29, 43

[253] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. I. Amari. Bayesian robust tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems*, 2016. 30, 80, 81

[254] S. Zhou, N. X. Vinh, J. Bailey, Y. Jia, and I. Davidson. Accelerating online CP decompositions for higher order tensors. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1375–1384, New York, NY, USA, 2016. ACM. 29

[255] T. Zhou, W. Bian, and D. Tao. Divide-and-conquer anchoring for near-separable nonnegative matrix factorization and completion in high dimensions. In *IEEE International Conference on Data Mining*, 2013. 41

[256] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *International Conference on Machine Learning (ICML)*, 2011. 41, 42, 83

[257] T. Zhou and D. Tao. Bilateral random projections. In *IEEE International Symposium on Information Theory*, 2012. 41, 83

[258] T. Zhou and D. Tao. Greedy bilateral sketch, completion & smoothing. In *International Conference on Artificial Intelligence and Statistics*, 2013. 41, 42

[259] X. Zhou, C. Yang, H. Zhao, and W. Yu. Low-rank modeling and its applications in image analysis. *ACM Computing Surveys*, 2014. 9, 39, 45

[260] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma. Stable Principal Component Pursuit. In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010. 11, 18, 19, 57

[261] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3D action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013. 99

[262] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 28–31 Vol.2, Aug 2004. 34

[263] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773 – 780, 2006. 34

Détection d'objets mobiles dans des vidéos par décomposition en rang faible et parcimonieuse: de matrices à tenseurs

Résumé :

Dans ce manuscrit de thèse, nous introduisons les avancées récentes sur la décomposition en matrices (et tenseurs) de rang faible et parcimonieuse ainsi que les contributions pour faire fâce aux principaux problèmes dans ce domaine. Nous présentons d'abord un aperçu des méthodes matricielles et tensorielles les plus récentes ainsi que ses applications sur la modélisation d'arrière-plan et la segmentation du premier plan. Ensuite, nous abordons le problème de l'initialisation du modèle de fond comme un processus de reconstruction à partir de données manquantes ou corrompues. Une nouvelle méthodologie est présentée montrant un potentiel intéressant pour l'initialisation de la modélisation du fond dans le cadre de VSI. Par la suite, nous proposons une version « double contrainte » de l'ACP robuste pour améliorer la détection de premier plan en milieu marin dans des applications de vidéo-surveillance automatisés. Nous avons aussi développé deux algorithmes incrémentaux basés sur tenseurs afin d'effectuer une séparation entre le fond et le premier plan à partir de données multidimensionnelles. Ces deux travaux abordent le problème de la décomposition de rang faible et parcimonieuse sur des tenseurs. A la fin, nous présentons un travail particulier réalisé en conjonction avec le Centre de Vision Informatique (CVC) de l'Université Autonome de Barcelone (UAB).

Mots clés: détection d'objets mobiles, soustraction de fond, ACP robuste, décomposition en rang faible et parcimonieuse.

Robust low-rank and sparse decomposition for moving object detection: from matrices to tensors

Summary:

This thesis introduces the recent advances on decomposition into low-rank plus sparse matrices and tensors, as well as the main contributions to face the principal issues in moving object detection. First, we present an overview of the state-of-the-art methods for low-rank and sparse decomposition, as well as their application to background modeling and foreground segmentation tasks. Next, we address the problem of background model initialization as a reconstruction process from missing/corrupted data. A novel methodology is presented showing an attractive potential for background modeling initialization in video surveillance. Subsequently, we propose a double-constrained version of robust principal component analysis to improve the foreground detection in maritime environments for automated video-surveillance applications. The algorithm makes use of double constraints extracted from spatial saliency maps to enhance object foreground detection in dynamic scenes. We also developed two incremental tensor-based algorithms in order to perform background/foreground separation from multidimensional streaming data. These works address the problem of low-rank and sparse decomposition on tensors. Finally, we present a particular work realized in conjunction with the Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB).

Keywords: moving object detection, background/foreground separation, low-rank and sparse representation, matrix decomposition, tensor factorization.

Laboratoire L3i - Informatique, Image, Interaction
Laboratoire MIA - Mathématiques, Image et Applications

Faculté des Sciences et Technologies, Université de La Rochelle, Avenue Michel Crépeau

17042 La Rochelle - Cedex 01 - France