Low-rank and Sparse based Representation Methods with the Application of Moving Object Detection

by

Seyed Moein Shakeri

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

In this thesis, we study the problem of detecting moving objects from an image sequence using low-rank and sparse representation concepts. The identification of changing or moving areas in the field of view of a camera is a fundamental step in visual surveillance, smart environments, and video retrieval. Recent methods based on low-rank representation have been successfully employed for change detection; however, they still have difficulties in handling the following situations. First, the existing methods rely on a batch formulation whose computational complexity grows with the size of the input data. Secondly, they are not able to deal with significant illumination change including shadow and abrupt or discontinuous change in illumination. This thesis proposes solutions to the above two problems, with the end goal of developing a reliable low-rank and sparse decomposition to perform an efficient and accurate change detection method in such cases, especially for the moving object detection tasks.

To cope with the computational complexity of the low-rank methods for change detection, we propose a sequential solution using contiguous sparsity constraint. We formulate the problem of moving object detection under integration of online robust PCA and low-rank matrix approximation with contiguous sparse outliers. This combination enables us to extract foreground objects in the case of online and long-term continuous tasks, which cannot be achieved by the batch formulation.

To deal with discontinuous change in illumination, we first propose a robust representation of images against illumination, which can be used in classification, place recognition, and change detection applications. We then build a prior map from the invariant representation, and formulate a low-rank and invariant sparse decomposition (LISD) method by incorporating the original representations and the obtained prior maps. This joint framework empowers the accuracy of object detection by separating the sparse outliers into real changes and illumination change matrices. We also propose an iterative version of LISD (ILISD) to improve the performance of LISD by updating the prior map. Experiments on challenging benchmark datasets demonstrate the superior performance of the proposed method under complex illumination changes.

As the second solution to deal with discontinuous change in illumination and to boost the accuracy of foreground detection, we propose a robust solution based on the multilinear (tensor) data low-rank and sparse decomposition framework. In this method we first introduce a way to provide multiple invariant representations of an image as priors that can characterize the changes in the image sequence due to illumination. To deal with concurrent, two types of changes, we employ two regularization terms, one for detecting moving objects and the other for accounting for illumination changes, in a novel unified framework named tensor low-rank and invariant sparse decomposition (TLISD). Extensive experiments on challenging datasets demonstrates a remarkable ability of the proposed formulation to detect moving objects under discontinuous change in illumination.

Preface

Research for this thesis was conducted under the supervision of Dr. Hong Zhang in the Department of Computing Science, University of Alberta. Portions of this thesis were published as:

- Chapter 3: Moein Shakeri and Hong Zhang. “COROLA: A sequential solution to moving object detection using low-rank approximation”, Computer Vision and Image Understanding (CVIU), 146 (2016): 27-39.
- Chapter 4: Moein Shakeri and Hong Zhang. “Illumination invariant representation of natural images for visual place recognition.” In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 466-472. IEEE, 2016.
- Chapter 4: Moein Shakeri and Hong Zhang. “Moving object detection in time-lapse or motion trigger image sequences using low-rank and invariant sparse decomposition.”, In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5123-5131, 2017.
- Chapter 5: Moein Shakeri and Hong Zhang. “Moving object detection under discontinuous change in illumination using tensor low-rank and invariant sparse decomposition”, In Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Acknowledgements

I would like to sincerely acknowledge my supervisor, Prof. Hong Zhang for all of his support, kindness and guidance throughout my PhD program. The Ph.D. program has been an enjoyable journey for me because of his patience, continuous support and the freedom he gave me to explore my ideas. His expertise and thoughtful feedbacks over the course of this research, was invaluable.

I would like to express my gratitude to my committee members, Prof. Martin Jagersand, Prof. Nilanjan Ray, Dr. Ron Kube, and also my external examiner, Prof. Weichuan Yu, for their invaluable time and constructive comments that improved the technical quality of my work.

I also thank all the people who have helped me to improve the quality of my research by providing helpful comments and discussions including faculty members, all robotics/vision lab members, and anonymous reviewers of the research papers that I have published. This research study would not be successful without supporting grants from Alberta Innovates Technology Futures (AITF), and NSERC through its Discovery Grant and Strategic Network Grant (NCRN) programs. I would also like to thank my friends and colleagues for making this journey more pleasant and memorable.

Finally, I wish to express my deep gratitude to my parents for their never-ending love and support along the way. My sincere thank goes to my beloved wife, Homa, for her encouragement, support and love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Moving object segmentation from an image sequence or a video stream is a fundamental problem in computer vision with such applications as visual surveillance [121], traffic monitoring [24], [107]–[109], vehicle tracking [126], medical imaging [154], avian protection [111], object-based video encoding and social signal processing [130] where the accuracy of segmentation is critical to the solution of the application.

Moving object detection is a well studied field of research and many traditional methods have been proposed, which can be grouped into two major categories. Motion-based methods [29], [129] use motion information of the image pixels to separate the foreground from the background. These methods work based on the assumption that foreground objects move differently from the background. Therefore it is possible for these methods to classify pixels according to their movement characteristics. However, these methods require point tracking to identify the foreground [89], which can be difficult especially with dynamic background or noisy data [127].

Another popular category for moving object detection methods is background subtraction [5], [8], [33], [93], [104], [105], [110], [117], [135], [142], which compares the pixels of an image with a background model and considers those that differ from the background model as moving objects. These methods model the background for each pixel independently and so they are not robust against global variations such as illumination changes. Although some region-based methods [54], [58], [88], [100] have been proposed to take advantage of inter-pixel relations for identifying foreground objects from image regions, they can only obtain rough shapes of foreground objects and they are still vulnerable to significant illumination changes that arise in outdoor applications [142].

Recent years have seen the development of a new category of methods, based on low-rank and sparse decomposition, under one major assumption that images in a sequence are correlated. Methods in this category follow the basic idea from [90], where the principal component analysis (PCA) for background modeling was proposed. Extending this idea, current methods exploit the fact that the background model in an image sequence can be defined as a low-rank matrix by those pixels that are temporally correlated [21]. The last few years have witnessed fast development on low-rank and sparse representation methods and great success has been demonstrated in different computer vision applications including background modelling and detecting moving object as foreground [11].

Methods for background modelling and foreground detection attempt to decompose a matrix D of the observed image sequence into a low-rank matrix L and a sparse matrix S so as to recover background and foreground [11]. The problem can be solved by the well known robust principal component analysis (RPCA) [21], which has been widely studied.

Although recent methods based on low-rank representation have been successfully employed for change detection, they still have difficulties in handling the following situations.

- First, the existing methods rely on a batch formulation whose computational complexity grows with the size of the input data. To address this issue, we propose a contiguous outliers representation method via online low-rank approximation (COROLA) [106]. In particular, our proposed method uses the sparsity and connectedness terms as the prior information for moving objects and estimates the background model using sequential low-rank approximation with the help of online robust PCA

(OR-PCA). Combining the sparsity and connectedness terms with the sequential low-rank approximation within a single optimization framework, enables us to detect moving objects in the case of long-term continuous tasks with dynamic background or noisy data.

- Secondly, low-rank based methods are not able to deal with significant illumination change including shadow and abrupt or discontinuous change in illumination. Currently, many surveillance systems, specifically those that use security cameras and wildlife monitoring cameras, capture a scene using a motion trigger sensor or timer-lapse photography in order to detect moving objects of interest or changes over time. Since captured images by these cameras are in different time of a day with different illumination and weather conditions, their processing is challenging.

  Current solutions are vulnerable to complex illumination changes that frequently occur in practical situations and real environments, especially when the changes are discontinuous in time. In such cases, current methods are often not able to distinguish between illumination changes (including those due to shadow), and changes caused by moving objects in the scene.

  In general, outdoor illumination conditions are uncontrolled, making moving object detection a difficult and challenging problem. This is a common problem for many surveillance systems in industrial or wildlife monitoring areas in which a motion triggered camera or a time-lapse photography system is employed for detecting objects of interest over time. Fig. 1.1 shows an example of this kind of images and illustrates the problem of object detection under significant illumination changes. Due to significant and complex changes in illumination and independent changes of the moving objects between images of the sequences, detection of the moving objects is extremely challenging

  In this thesis, we investigate how this problem can be addressed, by introducing different priors incorporated into the low-rank framework. As the first solution, we propose a low-rank and invariant sparse decomposition

(LISD) method [103] to separate an input data matrix into three matrices: low-rank matrix as the estimated background, and the sparse real changes and illumination change matrices. In particular, we first propose a robust representation of images against illumination, which can be used in classification [40], visual place recognition [101], and change detection [103] applications. Then we compute a prior map from the invariant representation and propose a unified framework by incorporating the prior map and the original representations. This unified framework enables us to distinguish between foreground changes and illumination change through the optimization and can separate the sparse outliers into real changes and illumination change matrices.

As the second solution to deal with discontinuous change in illumination, we introduce a new formulation for moving object detection under the framework of low-rank tensor approximation. In particular, we first propose a method to create a set of prior maps for each image in the image sequence and treat it as a tensor. These prior maps enable us to use two regularization terms to distinguish between moving objects and illumi-

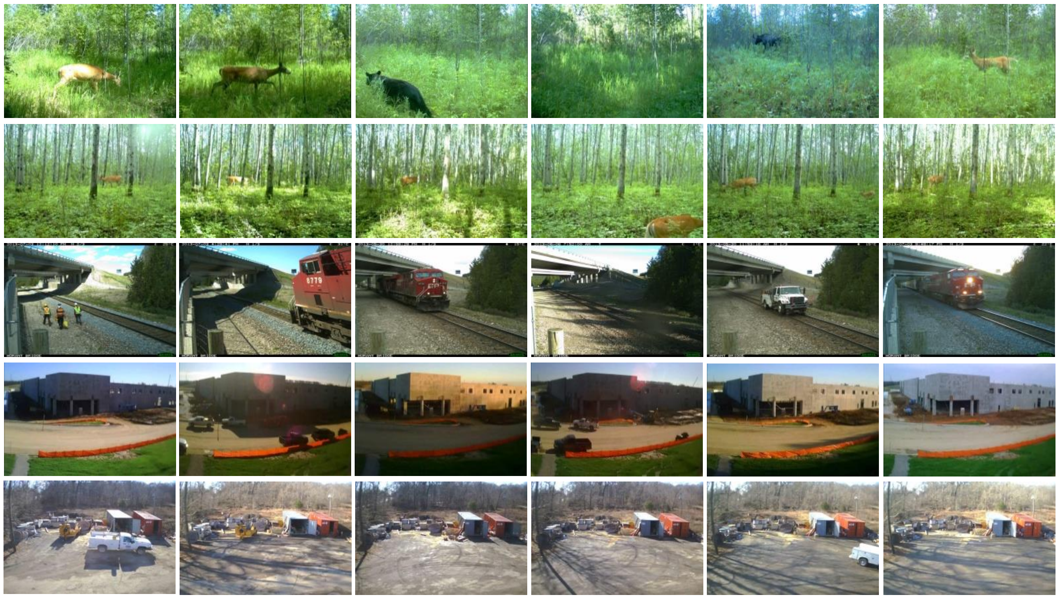


Figure 1.1: Selected images from the image sequences captured by (First three rows): a motion triggered camera for railway and wildlife monitoring, (Last two rows): a time triggered camera for industrial area monitoring.

nation changes. Then we formulate the regularization terms in a unified framework named tensor low-rank and invariant sparse decomposition (TLISD) [102].

It should be pointed out, inspired by the significant success of deep neural networks in computer vision, another group of background subtraction and moving object detection methods based on deep neural network have been proposed [3], [15], [27], [99], [144], [147]. However, these learning-based methods need supervised training with pixel-wise ground-truth masks of moving objects, which are not practical in real applications. Furthermore, these methods only learn the background variations, due to the lack of labeled data for illumination. Therefore, in the problem that we described in 1.1, these methods need to be learned with all of discontinuous variations, which is roughly impossible.

1.2 Contributions

The contributions of this thesis are as follows.

- In Chapter 3 we propose an online formulation of the low-rank approximation algorithm for foreground object detection. In particular, we use the sparsity and connectivity of moving objects as prior information in online low-rank approximation. We combine these prior information with the sequential low-rank approximation in a unified optimization framework, which enables us to detect moving objects in the case of long-term continuous task. Extensive experiments on benchmark datasets demonstrate the effectiveness and reliability of the proposed formulation in comparison with other existing online methods, especially in dynamic background scene or noisy environments.

- In Chapter 4 we propose a novel low-rank and invariant sparse decomposition, based on a new illumination regularization term to distinguish between illumination changes and real changes as moving objects. Based on our proposed method, for the first time we are able to separate discontinuous change in illumination from real changes which we call moving objects. Extensive experimental evaluations on different datasets illustrate the superior performance of our proposed method in comparison with all other existing methods.

The main contributions in this chapter are as follows.

(a) We propose a robust representation of images against illumination, which is used in visual place recognition, classification, and change detection.

(b) We introduce an illumination regularization term using the proposed illumination invariant representation in (a)

(c) We propose a joint optimization framework by incorporating the illumination regularization term and the low-rank and sparse decomposition framework.

- In Chapter 5 we propose a new low-rank tensor decomposition using group sparsity and k-support norm as two regularization terms to separate moving objects and illumination variations that undergo discontinuous changes. Our algorithm evaluation demonstrate the power of incorporating the group sparsity norm with the k-support norm, to detect moving objects accurately under discontinuous change in illumination.

  The main contributions in this chapter are as follows.

  (a) We define a method for creating multiple priors that characterize the complex variation of illumination in a video sequence

  (b) We define a unique tensor structure different from all existing methods, and encapsulate the prior maps in the tensor

  (c) We propose to use the k-support norm to capture uncorrelated illumination variations, and we evaluate the effect of this norm in comparison with L_1 norm.

  (d) We introduce a unified optimization framework by exploiting the prior maps with group sparsity and k-support norm as two regularization terms to distinguish between moving objects and illumination variations.

- Due to the lack of a comprehensive dataset with various illumination and shadow changes in a real environment, We introduce a new benchmark dataset with over 80k real images captured by industrial security cameras and wildlife monitoring systems during three years in Chapters 4 and 5.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2 we provide an overview of low-rank approximation, the background and related works to our research for moving object detection, and illumination invariant representation methods. In Chapter 3, we propose an incremental contiguous outlier representation method to deal with the batch formulations of low-rank frameworks. Then in Chapters 4 and 5 we study the problem of discontinuous change in illumination and separating them from moving objects as real changes. In particular, we propose a novel low-rank and invariant sparse decomposition using illumination regularization term and evaluate the method with extensive experimental results in Chapter 4. In Chapter 5, we propose a unique tensor by creating multiple illumination priors for images, and introduce our tensor low-rank and sparse decomposition method to separate illumination changes from moving objects. In these chapters, we also introduce a comprehensive challenging real image dataset, captured by industrial security cameras or wildlife monitoring systems. The conclusion of this thesis and directions for future works are explained in Chapter 6.

Chapter 2

Background

In Chapter 1, we explained the capability of low-rank and sparse decomposition methods in comparison with the traditional methods, to detect moving objects. Since low-rank approximation methods work based on the correlation between images in a sequence, they are more robust against global variations than the traditional methods. However, these methods still have difficulties in dealing with

- long-term continuous tasks due to the batch processing and computational cost, and
- significant illumination changes, especially for the sequences that are discontinuous in time.

In this chapter, we provide relevant background material related to our research for the above two problems. We first provide an overview of low-rank approximation in Section 2.1, and then we study the background and related works of moving object detection using low-rank based methods in Section 2.2 to deal with both computational cost and discontinuous illumination change. Since we will introduce an illumination regularization term in Chapter 4, we also review relevant illumination invariant representation methods in Section 2.3.

2.1 Low-Rank Approximation

In the recent years, low-rank matrix recovery, which effectively separates sparse outliers from corrupted observations, has been successfully applied to a variety of computer vision applications, such as face recognition [31], [84], [133], [136], image classification [39], [53], [149], images alignment [92], image restoration and denoising [64], [73], [81], [148], [156], subspace clustering [79], [112], [128], data compression [21], [56], [146] and background subtraction [10], [11], [156].

Methods in the category of background subtraction and moving object detection follow the basic idea from [90], where the principal component analysis (PCA) for background modeling was proposed. It is based on the observation that the underlying background images should be linearly correlated and the composed matrix of vectorized background images can be naturally modeled as a low-rank matrix.

Algebraically speaking, if an image is vectorized in a column and all images of a sequence are concatenated into a 2D matrix, then the columns are dependent and its low-rank approximation matrix represents the background model of the images. As a result, the background modeling problem is converted to the low-rank approximation problem.

In general, by decomposing an input matrix D of vectorized images into a low-rank matrix L and a sparse matrix S, the background and foreground objects can be recovered.

The basic form of this method is as follows.

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad s.t.\ D = L + S \tag{2.1}$$

where $\|L\|_*$ denotes the nuclear norm of matrix L - i.e., the sum of its singular values - and $\|S\|_1 = \Sigma_i |S_i|$ is the l_1-norm of S.

The overview of this decomposition is shown in Fig. 2.1. One popular method to solve (2.1) is augmented Lagrangian method (ALM) [77] as follows.

$$\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda \|S\|_1 + < Y, D - L - S > + \frac{\mu}{2} \|D - L - S\|_F^2 \tag{2.2}$$

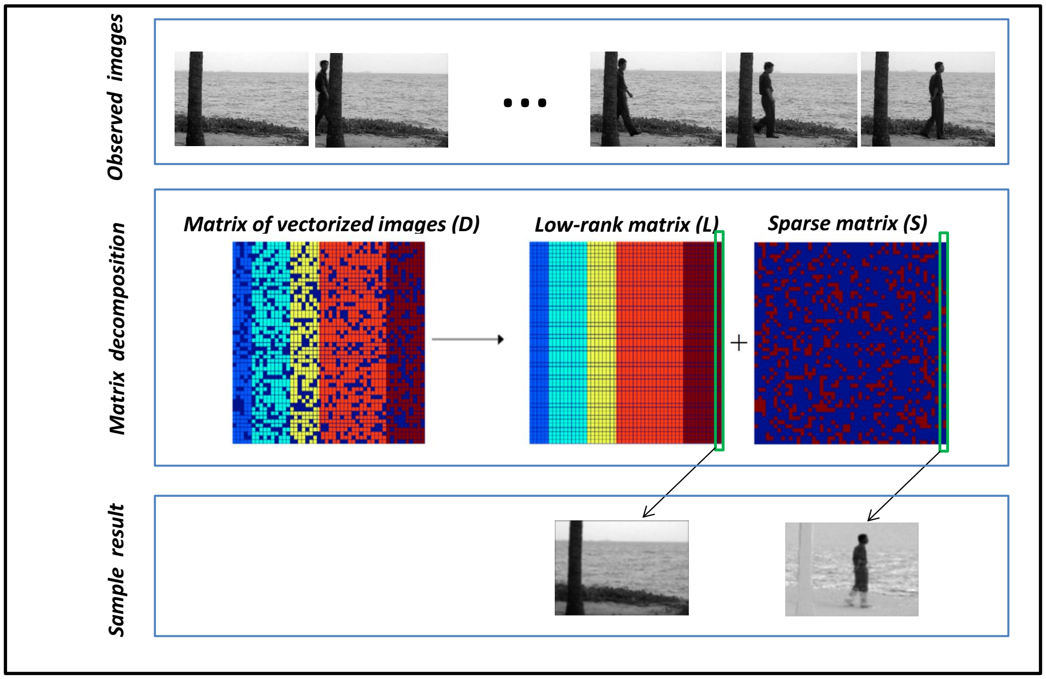


Figure 2.1: Overview of low-rank decomposition by RPCA [21]

where Y is the Lagrange multiplier, μ is a positive constant and $< \Delta_1, \Delta_2 >= tr(\Delta_1^T \Delta_2)$ is the inner product. Lin *et al.* [77] proposed two ALM algorithms to solve (2.2); namely exact and inexact ALM. Each iteration of the exact ALM to update L an S, solves the following sub-problem, which needs couple of iterations to converge.

$$(L_{k+1}^*, S_{k+1}^*) = \arg\min_{L,S} \mathcal{L}(L, S, Y_k^*, \mu_k) \tag{2.3}$$

Lin *et al.* [77] also showed updating L and S once when solving (2.3) is sufficient to converge to the optimal solution, thereby yielding the inexact ALM algorithm, which has been summarized in Algorithm 2.1. In this algorithm, SVD is singular value decomposition, and $\Omega_\varepsilon[.]$ is the soft-thresholding (shrinkage) operator, defined as follows.

$$\Omega_\varepsilon[x] = \begin{cases} x - \varepsilon \quad if \quad x > \varepsilon \\ x + \varepsilon \quad if \quad x < -\varepsilon \\ 0 \qquad otherwise \end{cases} \tag{2.4}$$

Algorithm 2.1 Inexact ALM Algorithm for Equation 2.2 [77]

```
    Input: Observation matrix $D, \lambda$.
 1: Initialize $Y$, $\mu > 0$, $\rho \geq 1$ .
 2: While not converged
            //Lines 3-4 solve $L_{k+1} = \arg\min_L \mathcal{L}(L, S_k, Y_k, \mu_k)$
 3:         $(U, \Sigma, V) = SVD(D - S_k + \mu_k^{-1} Y_k)$
 4:         $L_{k+1} = U \, \Omega_{\mu_k^{-1}}[\Sigma] \, V^T$
            //Line 5 solves $S_{k+1} = \arg\min_S \mathcal{L}(L_{k+1}, S, Y_k, \mu_k)$
 5:         $S_{k+1} = \Omega_{\lambda\mu_k^{-1}} \, [D - L_{k+1} + \mu_k^{-1} Y_k]$
 6:         Updating $Y$: $Y_{k+1} = Y_k + \mu_k \, (D - L_{k+1} - S_{k+1})$
 7:         Updating $\mu$: $\mu = \rho\mu$
 8:         $k \leftarrow k + 1$
 9: end
10: Output: $(L_k, S_k)$
```

2.1.1 Low-Rank Approximation of Tensors

Real world data are ubiquitously in multi-dimensional way such as color images, image sequences or videos. Converting them into a matrix form usually leads to the information loss. Recently, multi-way or tensor data analysis has attracted much attention and has been successfully used in many applications [42], [70], [80], [86], [150]. Formally and without loss of generality, denote a 3-way tensor by $\mathcal{D} \in R^{n_1 \times n_2 \times n_3}$. Low-rank tensor methods attempt to decompose $\mathcal{D} \in R^{n_1 \times n_2 \times n_3}$ into a low-rank tensor $\mathcal{L}$ and an additional sparse tensor $\mathcal{S}$ [45]. The overview of low-rank tensor and sparse decomposition is shown in 2.2. This decomposition is applicable in solving many computer vision problems, including moving object detection.

The main challenge for low-rank tensor estimation is the definition of tensor rank [41]. Several tensor low-rank decomposition methods have been proposed based on the definition of tensor rank such as CPrank [70], sum of nuclear norms (SNN) rank [43], [80], [113], [122], and Tucker rank [97]. Zhang *et al.* [67] proposed a tensor tubal rank based on a new tensor decomposition

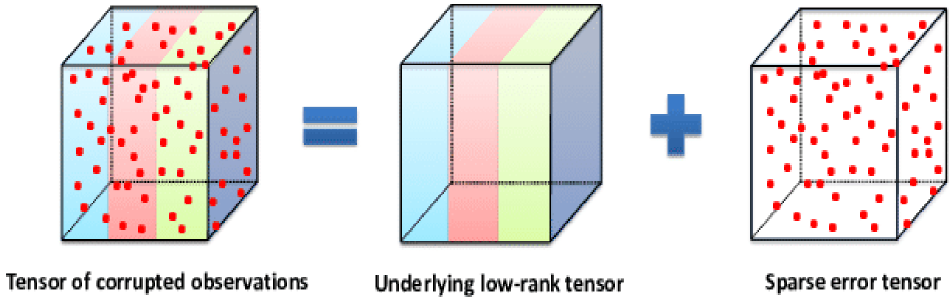


Figure 2.2: Overview of low-rank tensor decomposition [83]

scheme in [16], [68], which is referred as tensor SVD (t-SVD). The t-SVD is based on a new definition of tensor-tensor product which uses many similar properties as the matrix case.

One of the most recent methods relevant to our research is proposed by Lu *et al.* [83]. A tensor nuclear norm based on a tensor SVD (t-SVD) [68] was used to estimate the rank of tensor data and RPCA was extended from 2D to 3D to formulate the following tensor robust PCA (TRPCA):

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_* + \lambda\|\mathcal{S}\|_1 \quad s.t.\ \mathcal{D} = \mathcal{L} + \mathcal{S} \tag{2.5}$$

where $\|\mathcal{L}\|_*$ is the tensor nuclear norm of $\mathcal{L}$, i.e. the average of the nuclear norm of all the frontal slices ($\|\mathcal{L}\|_* = \frac{1}{n_3}\Sigma_{p=1}^{n_3}\|\mathcal{L}_{:,:,p}\|_*$). [83] showed that the tensor nuclear norm on tensor data can appropriately capture higher order relations in data than the nuclear norm on 2D matrices.

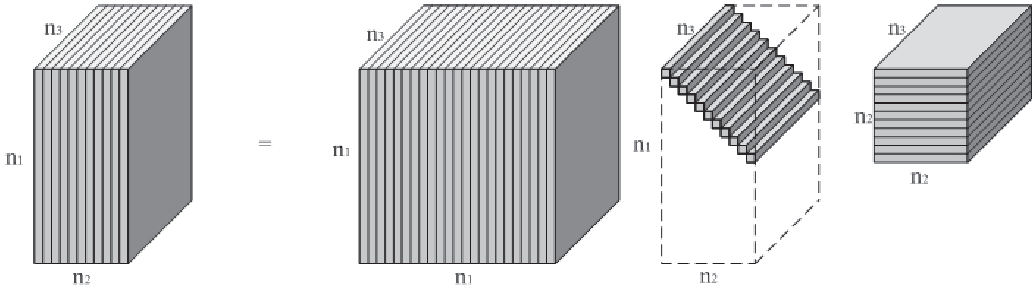


Figure 2.3: Illustration of the t-SVD of an $n1 \times n2 \times n3$ tensor [67]

2.2 Low-Rank based Methods for Moving Object Detection

In this section, we specifically review the current solution to deal with both computational cost and dynamic scenes, the two mentioned problems in Chapter 1, for moving object detection.

2.2.1 Methods to Deal with Computational Cost

Due to batch processing, the following two problems occur: *memory storage* and *time complexity.* In continuous monitoring tasks or video processing, if matrix D is built with a large number of images, memory storage will be a problem [19]. In addition, by increasing the size of the input matrix D, time complexity for the matrix decomposition is also increasing.

To address the problem of time complexity, some efficient algorithms have been proposed [96], [152], [153]. Rodrigues and Wohlberg proposed a fast PCP [96] algorithm to reduce the computation time of SVD in inexact ALM (IALM). The "Go Decomposition" (GoDec) method, proposed by Zhou *et al.* computes RPCA using bilateral random projections (BRP) [153]. Semi-Soft GoDec (SSGoDec) and Greedy SSGoDec methods [152] are extensions of GoDec to speed it up. Although these algorithms reduce the computation time of low-rank approximation, they are still not satisfactory for applications such as visual surveillance and robot navigation due to their batch formulation. In many applications, online processing is critical and batch methods are infeasible.

To overcome the limitations of batch processing methods, incremental and online robust PCA methods have been developed [9], [25], [34], [51], [52], [95], [131], [140]. He *et al.* [51] proposed Grassmannian robust adaptive subspace tracking algorithm (GRASTA), which is an incremental gradient descent algorithm on Grassmannian manifold for solving the robust PCA problem. This method incorporates the augmented Lagrangian of l_1-norm loss function into the Grassmannian optimization framework to alleviate the corruption by outliers in the subspace update at each gradient step. Following the idea of

GRASTA, He *et al.* [52] proposed transformed GRASTA (t-GRASTA), which iteratively performs incremental gradient descent constrained to the Grassmann manifold in order to simultaneously decompose a sequence of images into three parts: a low-rank subspace, foreground objects, and a transformation such as rotation or translation of the image. This method can be regarded as an extension of GRASTA and RASL [92] (robust alignment by sparse and low-rank decomposition) by computing the transformation and solving the decomposition with incremental gradient optimization framework. To improve the accuracy of online subspace updates especially for dynamic backgrounds, Xu *et al.* [140] developed an online Grassmannian subspace update algorithm with structured-sparsity (GOSUS) via an alternating direction method of multipliers (ADMM) [12].

To deal with noisy conditions and dynamic background scene, Wang *et al.* [131] proposed a probabilistic approach to robust matrix factorization (PRMF) and its online extension for sequential data to obtain improved scalability. This model is based on the empirical Bayes approach and can estimate better background model than GRASTA.

Recently, Feng *et al.* [34] proposed an online robust principal component analysis via stochastic optimization (OR-PCA), and [63] used it for foreground detection. [34] does not need to remember all the past samples and uses one sample at a time by a stochastic optimization. OR-PCA reformulates a nuclear norm objective function by decomposing it to an explicit product of two low-rank matrices, which can be solved by a stochastic optimization algorithm. We benefit from this approach in our proposed method to provide a sequential solution for moving object detection using low-rank approximation in Chapter 3.

2.2.2 Methods to Deal with Dynamic Scenes

Low-rank and sparse decomposition methods attempt to decompose an observed matrix D of vectorized images into the low-rank matrix L and the sparse matrix S, which correspond to the background and moving objects, respectively. Since moving objects are not linearly correlated in an image sequence, they are grouped into the sparse matrix, and so the accuracy of detected moving objects is highly related to the quality of the sparse matrix S. Regarding this fact, low-rank based methods can be categorized into different groups, based on different constraints on sparse matrix S. Candes *et al.* [21] used l_1-norm to constrain the sparse matrix. Following this approach, Wang *et al.* [131] proposed a probabilistic matrix factorization (PRMF) using Laplace error and Gaussian prior, which correspond to an l_1 loss and l_2 regularizer, respectively.

To improve the performance of moving object detection, some other constraints have been recently imposed on sparse matrix S using prior knowledge of spatial continuity of objects [30], [48], [49], [118]. These methods use a block sparsity as a spatio-temporal constraint to detect the foreground. For example, Guyon *et al.* [48], [49] proposed the low-rank and block sparse matrix decomposition (RPCA-LBD) using $l_{2,1}$-norm as a spatial continuity to enforce the block-sparsity of the foreground.

Although these methods are more robust than conventional RPCA in the presence of illumination changes, the block-sparsity property is unable to model sparse outliers or filter out significant illumination changes and moving shadows. Besides, in the case of time-lapse video or low frame rate image sequences, where consecutive frames are captured with a large time-interval, the position of an object in each frame is discontinuous from other frames and $l_{2,1}$-norm cannot handle the situation.

Recently, tensor data analysis has attracted much attention for background subtraction and foreground detection in frame-rate sequences, and many methods based on the idea of spatial continuity using tensor decomposition has been proposed [22], [57], [60], [75], [114]. These methods stack two dimensional images into a three dimensional data structure, using which tensor decomposition can capture moving object due to the continuity of object positions in the third dimension. Obviously, these methods still suffer from the same issue of the matrix-based methods, and are not applicable to time-lapse image sequences with discontinuous changes in both object location and illumination.

Another group of methods used the connectivity constraint on moving objects [82], [106], [132], [138], [140], [155]. Xu *et al.* [140] proposed an online subspace update method GOSUS that defines an objective function with a superpixel method to achieve sparsity of the groups. Wang *et al.* [132] proposed a full Bayesian robust matrix factorization (BRMF). They further extended it by assuming that the outliers form clusters with close within-group spatial proximity which correspond to moving objects. This is achieved by placing a first-order Markov random field (MRF) [76], and the method is referred to as Markov BRMF or MBRMF. Zhou *et al.* [155] proposed DECOLOR by assuming the moving objects are connected components and relatively small. They also incorporated these priors using MRF.

Following the connectivity constraint, [44] proposed a method by incorporating a spatial contiguity prior in the form of blocks and motion saliency map to detect moving objects. Liu *et al.* [82] improved it using a structured sparsity norm for the spatial contiguity. Liu *et al.* [82] proposed a method using a structured sparsity norm [85] based on 3×3 overlapping-patch groups. Since the foreground is usually spatially contiguous in each image, computing the maximum values of each group promotes the structural distribution of sparse outliers during the minimization. They also used a motion saliency map to distinguish the foreground object from background motion. Using this saliency map, the method is robust in the case of background motion and sudden illumination changes in the image sequence. However [82] cannot handle discontinuous change in illumination or moving shadows, especially in time-lapse videos where the foreground objects are completely stochastic as are shadow and illumination changes.

Inspired by the concept of the group sparsity structure, recently, Ebadi *et al.* [32] proposed an approach by employing structured sparsity norm in the context of tree-structured groups [69], where each group is a superpixel region. The structured sparsity norm of their optimization framework decides whether each region belongs to foreground or must be classified as background. The regions belonging to background are left-off, while the regions hinting foreground elements are divided into two smaller regions once again to form the shape of foreground during the minimization. Although [32] can detect moving objects accurately in frame-rate sequences with moderate illumination changes, [32] still vulnerable to discontinuous change in illumination, especially in time-lapse videos.

Based on the idea of spatial constraint, recently, a new approach using multi-scale structured sparsity for background subtraction has been proposed [151]. [151] explores the structured smoothness with both appearance consistency and spatial compactness in the low-rank and sparse decomposition framework. This methods assumes that the foregrounds are generally consistent in appearance and so, foreground pixels are homogeneous in the same concept of spatial region, such as the same superpixel. Then, [151] encourages this structure of spatial compactness on the foregrounds at different scales. Coarse scale with smaller number of superpixels imposes global structure of the pattern such as the whole body in a single superpixel, while fine scale captures local structure. Finally, [151] integrates the multi-scale cues into a unified structured low-rank and sparse decomposition framework to capture the structure of the foregrounds.

In general, although the low-rank framework is well-known to be robust against moderate illumination changes in frame-rate sequences, the existing methods are still not able to handle discontinuous change in illumination and shadow, especially in time-lapse sequences, where the foreground objects are completely stochastic as are shadow and illumination changes.

To deal with this problem, in Chapter 4 we first propose a robust representation of images against illumination. Then we propose a low-rank and invariant sparse decomposition method by incorporating the original and invariant representations. As a second solution to deal with discontinuous illumination change, in Chapter 5 we first propose a method to create a set of prior maps to build a tensor data structure. Then we formulate the problem of moving object detection in a unified framework named tensor low-rank and invariant sparse decomposition with the help of two regularization terms to distinguish between moving objects and illumination changes.

2.3 Illumination Invariant Representation Methods

Dealing with illumination changes is a well-studied area in computer vision and many methods have been proposed [157], [47], [55], [116], [119], [120], [134] for different applications. The first group of these methods relies on learning classifiers on color and intensity of an image. However, these methods usually focus on shadow removal and cannot produce illumination invariant representation at different times of a day [157], [47]. The other group of methods follows the idea of [6], where the definition of "intrinsic images" was introduced, and models the process of image formation [7], [116], [119], [120], [134]. These methods decompose an image into two separate component images: one for describing reflectance of the scene and the other for explaining the variation in the illumination across the scene.

Following the above definition, an illumination invariant method attempts to remove the effects of illumination on the color and the intensity of an image. This can be achieved by deriving invariant quantities which remain unchanged under illumination changes [37], [38], [36]. One can use a transformation from RGB to an invariant 2D log-chromaticity space, and then find a special direction in a 2D chromaticity feature space to produce a grayscale image which is approximately invariant to intensity and color of scene illumination. To compute the special direction which is called invariant direction, [38] uses a camera calibration method, and [36] derives it from the image itself using entropy minimization. [36] is relatively fast and popular due to find the invariant direction from the image itself. This approach is used originally for removing shadow from single images. However, [28] used this approach to provide illumination invariant representation for localization and place recognition. Since we take benefit of this approach in our proposed methods, we summarize obtaining intrinsic images using entropy minimization as follows.

Finlayson *et al.* in [36], [37] adopt a Lambertian model of image formation with a power spectral density $E(\lambda, x, y)$, surface reflectance function $S(\lambda, x, y)$, and the spectral sensitivity of the kth camera sensor - where $k = 1, 2, 3$ for red, green and blue channels of the camera - as follows.

$$\rho_k(x,y) = \sigma(x,y) \int E(\lambda,x,y) S(\lambda,x,y) Q_k(\lambda) d\lambda \tag{2.6}$$

where $\sigma(x,y)$ is a constant factor and denotes the Lambertian shading term at a given pixel. If the camera sensitivities are Dirac delta functions, $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$ and illumination can be modeled by Plank's Law [137], then (2.6) becomes:

$$\rho_k = \sigma E(\lambda_k) S(\lambda_k) q_k, \tag{2.7}$$

Restricting illumination to Planckian, an illuminant power spectral density E can be parameterized by its color temperature T.

$$E(\lambda, T) = Jc_1\lambda^{-5}e^{-\frac{c_2}{T\lambda}} \tag{2.8}$$

where J is a variable to control the overall intensity of the light, and c_1 and c_2 are constant. Now, for removing the effect of illumination, [36] computes the two-vector chromaticity $\chi = [\chi_1, \chi_2]$ as follows.

$$\chi_1 = \frac{\rho_2}{\rho_1}, \quad \chi_2 = \frac{\rho_3}{\rho_1} \tag{2.9}$$

Substituting (2.7) and (2.8) into (2.9) and taking the logarithm $\chi^{'}$ from χ, we obtain:

$$\chi^{'} = s + \frac{1}{T}e \tag{2.10}$$

where $s_k = \lambda_k^{-5} S(\lambda_k) q_k$, and $s = [s_1, s_2]$ is a two-vector which depends on the scene surface and the camera, but is independent of illumination. e is a two-vector which is independent of the scene surface, but depends on the camera. T changes when illumination changes and $\chi^{'}$ obtained from (2.10) moves along a straight line roughly. Direction of this line belongs to e and is independent of the surface and illumination. So, to remove the effect of illumination and to determine a 1D illumination invariant representation, we can easily project the log-chromaticity vector $\chi^{'}$ onto the vector orthogonal to e as follows.

$$I' = \chi' e^{\perp}, \tag{2.11}$$

where I' is a grayscale illumination invariant image. For finding the invariant projection direction without calibration, [36] showed by quantization and

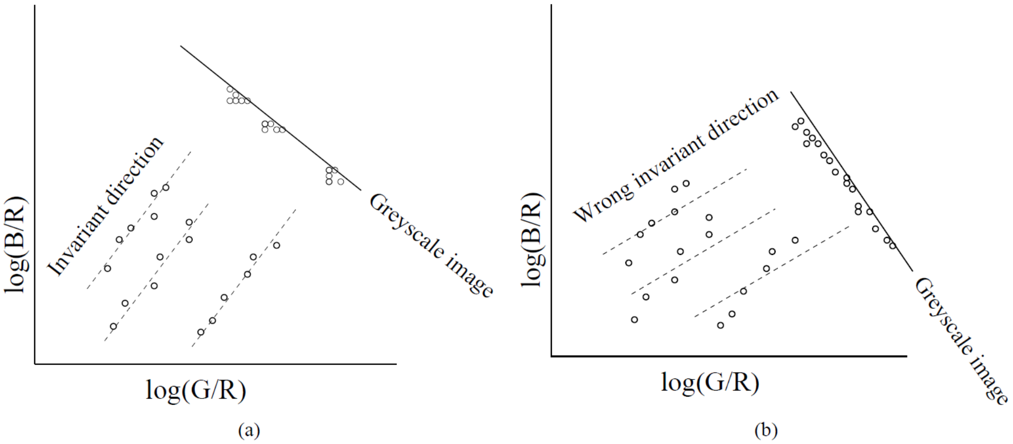


Figure 2.4: Overview of the obtained illumination invariant representation using [36], and intuition for finding best invariant direction via minimizing the entropy.

minimizing Shannon's entropy, where can estimate the best direction of vector e. Since the direction is independent of the illuminant and the surface, in the ideal case the direction should not change. The overview of this method is shown in Fig. 2.4.

Chapter 3

Contiguous Outlier Representation via Online Low-Rank Approximation

3.1 Introduction

Extracting moving objects from a video sequence and estimating the background of each individual image are fundamental issues in many practical applications, and could be a challenging task due to background variations, shadows and illumination changes. As explained in the previous chapters, many methods have been proposed to detect moving objects from image sequences. Among them, a new group of methods based on low-rank and sparse decomposition could outperform the traditional methods in terms of handling variations of a scene. Since most of them work in a batch processing form, they cannot be applied in real time application or long duration tasks. To address this issue, some online methods have been proposed; however, existing online methods fail to provide satisfactory results under challenging conditions such as dynamic background scene and noisy environments. We extensively discussed these methods in Chapter 2.2.1.

In this chapter, we offer an algorithm for the detection of moving objects named Contiguous Outliers Representation via Online Low-rank Approximation (COROLA). It solves the challenges of memory storage and time complexity of [155] with a comparable accuracy, and can provide even more accurate results in noisy environments. COROLA is also able to extract moving objects using a moving camera on a continuous basis, which cannot be achieved in general by a batch processing method especially in the case of large camera motion.

It should be underlined that recently many incremental low-rank and sparse decomposition methods for moving object detection have been proposed [61], [62], [91]. These methods use constraints based on properties of the moving objects to improve the accuracy of detection in a sequential form. Since the objects are structurally connected component and their locations are temporally correlated between frames, structured sparsity constraint [61], [62], and saliency map [91] are used to detect objects. [66] proposed a method using weighted low-rank to handle the background variations better than regular low-rank methods. [23] proposed a pan-tilt invariant method with an incremental PCP method to detect objects in the case of pan-tilt camera motion. It should be pointed out that all of these incremental methods have been proposed after our method.

3.1.1 Relation of Our Method to Other Methods

Since our COROLA method uses the sparsity and connectedness terms of DECOLOR method [155] and estimates the background model using sequential low-rank approximation with the help of OR-PCA [34], we present a summary of these two methods and in the next Section we describe our COROLA method that extends the two methods.

DECOLOR

DECOLOR is a formulation that integrates the outlier support and the estimated low-rank matrix in a single optimization problem, for joint object detection and background learning. Specifically, it works by solving the following minimization:

$$\begin{aligned} \min_{L,S} \frac{1}{2}\|\mathcal{P}_{S^\perp}(D-L)\|_F^2 + \beta_2\|S\|_1 + \gamma\|\Phi(S)\|_1 \\ s.t.\ \ rank(L) \leq r, \end{aligned} \tag{3.1}$$

where D, L, and S are the matrix of vectorized images, estimated background images, and outlier support, respectively. S in (3.1) is binary and its elements are 1 for outliers. $S^{\perp}$ is the complement of S and its elements are 1 for background pixels of the images. $\Phi(S)$ means the difference between neighboring pixels and therefore the last term of the above minimization encourages connectedness of outliers. Zhou *et al.* [155] solved the first term of (3.1) with its constraint using an alternating algorithm (SOFT-IMPUTE) [87]. They then solved the rest of the minimization problem by Markov Random Field (MRF) [76]. This two-step optimization is iterated until convergence. Although this method provides promising results, it still suffers from memory storage and time complexity problems in large datasets and, due to batch processing, it is not appropriate to operate on a continuous basis. Furthermore, in the case of a moving camera, DECOLOR only works for short video sequences with small camera motion and cannot deal with a moving camera in general.

OR-PCA

OR-PCA [34] decomposes an input matrix into low-rank and sparse matrices sequentially, processing one sample at a time and producing a solution that is equivalent to that of the batch RPCA. As a result, its computation cost is independent of the number of samples. To compute the low-rank via online optimization, OR-PCA uses an equivalent form of the nuclear norm for the matrix L where rank is upper bounded by r [94], as follows.

$$\|L\|_* = \underset{U \in R^{m\times r}, V \in R^{r\times n}}{inf} \left\{ \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 : L = UV \right\} \tag{3.2}$$

where U and V are the basis and coefficients of the low rank matrix. Using this equivalent form for low rank matrix, OR-PCA solves the following minimization problem:

$$\min_{U,V} \frac{1}{2}\|(D - UV - E)\|_F^2 + \frac{\lambda_1}{2}(\|U\|_F^2 + \|V\|_F^2) + \lambda_2\|E\|_1 \tag{3.3}$$

where E is sparse error matrix. Feng *et al.* [34] solved (3.3) in an online manner for one sample per time by two iterative updating parts. First, the coefficients and the sparse error for each new sample is updated by the previous basis. Then, the basis is updated using the new sample, updated coefficients, and sparse errors.

In the next section, extending the work of DECOLOR with the help of OR-PCA for updating U and V, we introduce a novel non-convex closed-form formulation for detection of moving objects named contiguous outliers representation via online low-rank approximation (COROLA).

3.2 Online Moving Object Detection by COROLA

In this section, we focus on online detection of moving objects for static cameras and then we show that the method can be easily extended to the case of moving cameras. We first formulate the problem of background modelling and foreground object detection and then describe in detail our COROLA algorithm, which computes the low-rank approximation and foreground detection sequentially.

3.2.1 Notations and Formulation

Let $X \in R^m$ be a vectorized image, and X_j be the j^{th} image in a sequence, expressed as a column vector of m pixels. Then, $D = [X_1, ..., X_n] \in R^{m \times n}$ is a matrix of n images and the i^{th} pixel in each observed image X is denoted as x_i. To indicate foreground for an observed image j, we use a binary indicator vector $\mathrm{s} = [s_1, s_2, ..., s_m]^T$ as the foreground support where

$$s_i = \begin{cases} 0 & \text{if } i \text{ is background} \\ 1 & \text{if } i \text{ is foreground} \end{cases} \tag{3.4}$$

and matrix $S = [\mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_n]$ shows a binary matrix of all images in D. Also, we use the function $\mathcal{P}_S(X)$ to construct a vector of at most m foreground pixels of image X. $S^{\perp}$ is the complement of S and its elements are 1 for background pixels of the images, where $\mathcal{P}_S(X) + \mathcal{P}_S^{\perp}(X) = X$. Now, let $L = UV$. The objective function in (3.1) can be rewritten as follows.

$$\begin{gathered}\min_{U,V,S} \frac{1}{2}\|\mathcal{P}_{S^{\perp}}(D-UV)\|_F^2 + \beta_2\|S\|_1 + \gamma\|\Phi(S)\|_1 \\ s.t.\ \ rank(U) = rank(V) \leq r,\end{gathered} \tag{3.5}$$

With the above notations and equations, we relax the constraints of (3.5) based on [34], and so the problem of background modelling and foreground object detection via sequential low-rank approximation and contiguous outlier representation solves the following optimization problem for each observed image.

$$\min_{U,\mathrm{v},\mathrm{s}} \frac{1}{2}\|\mathcal{P}_{S^{\perp}}(X-U\mathrm{v})\|_F^2 + \frac{\beta_1}{2}(\|U\|_F^2 + \|\mathrm{v}\|_F^2) + \beta_2\|\mathrm{s}\|_1 + \gamma\|\Phi(\mathrm{s})\|_1 \tag{3.6}$$

where $X \in R^m$ is an observed image, r is the upper bound on the rank of the basis matrix $U \in R^{m\times r}$, and $\mathrm{v} \in R^r$ is a coefficient vector. $\Phi(\mathrm{s})$ means the difference between neighboring pixels and it is computed by $\|\Phi(\mathrm{s})\|_1 = \sum\limits_{(i,k)\in\mathcal{E}} |s_i - s_k|$ and $\mathcal{E}$ is the neighborhood clique. Note that the objective function defined in (3.6) is non-convex and involves both continuous and discrete variables. Since (3.6) is our online formulation for each input image, the loss over all data would be the cumulative for each image. The first three terms try to compute the low-rank representation of input image X by first expressing it as a linear combination of the background basis U and its coefficient vector v using extraction function $\mathcal{P}_{S^{\perp}}$. The last two terms of (3.6) find continuous and small outliers to represent the foreground mask. Specifically, the fourth term imposes a sparsity constraint on the foreground mask s; i.e., the foreground pixels should be low in number. The last term imposes a connectivity constraint on mask s to account for correlation between neighboring pixels of an image. By minimizing (3.6) we can estimate the low-rank representation of an input image and detect foreground objects, concurrently. We use a two-step alternating optimization procedure by separating it to a low-rank approximation step involving U and v, and then a contiguous sparse optimization step involving s to obtain background estimation and foreground detection, alternatively. In the first step we use the same rule as OR-PCA method to update U and v. In the second step, minimization over s is conducted. In this step, we use the combination of Gaussian Mixture Model (GMM) and first order Markov Random Field (MRF) with binary labels to improve the foreground detection performance.

3.2.2 Online Low-Rank Approximation

For solving the first step of (3.6), we describe in this section our sequential method to compute the low rank background model of an image sequence and the foreground as its sparse outliers, in a way that is suitable for continuous and real time operation. In our sequential formulation, we adopt an online updating approach for optimization over U and v. Therefore (3.6) can be rewritten as:

$$\min_{U,\mathrm{v}} \frac{1}{2}\|\mathcal{P}_{S^\perp}(X - U\mathrm{v})\|_F^2 + \frac{\beta_1}{2}(\|U\|_F^2 + \|\mathrm{v}\|_F^2) \tag{3.7}$$

Initialization Step: With a small number of images at the beginning of a sequence no fewer than the rank of the background model, we initialize U with a batch method. This enables us to estimate the rank r roughly for the images in the rest of the sequence. Since this step is performed only once, the complexity of using a batch formulation is not an issue. After the initialization of U, for each input sample X, we use an incremental approach to solve (3.7) by the following two parts, repeatedly. These two parts update v, and then U for each sample to build the background model incrementally as follows:

Part 1: Because every two consecutive images in a sequence are similar, we can update coefficient vector v (or U) for the current image via background model U (or v) computed for the previous image. To update v with the fixed U, (3.7) becomes:

$$\hat{\mathrm{v}} = \underset{\mathrm{v}}{\operatorname{argmin}} \frac{1}{2}\|\hat{X} - \hat{U}\mathrm{v}\|_F^2 + \frac{\beta_1}{2}\|\mathrm{v}\|_F^2 \tag{3.8}$$

where $X \in R^m$ is the current image, $\hat{X} = \mathcal{P}_{S^\perp}(X)$, and $\hat{U} = \mathcal{P}_{S^\perp}(U)$. By fixing $\hat{U}$, (3.8) is a least squares problem and can be solved by

$$\hat{\mathrm{v}} = (\hat{U}^T\hat{U} + \beta I)^\dagger \hat{U}^T\hat{X} \tag{3.9}$$

where $(.)^\dagger$ is the Moore Penrose pseudoinverse [152].

Part 2: To update U, (3.7) can be rewritten as:

$$\min_U \frac{1}{2}\|\mathcal{P}_{S^\perp}(X - U\text{v})\|_F^2 + \frac{\beta_1}{2}\|U\|_F^2 \tag{3.10}$$

Assuming $U\text{v} = \mathcal{P}_{S^\perp}(U\text{v}) + \mathcal{P}_S(U\text{v})$, (3.10) can be rewritten as follows.

$$\begin{aligned} &\min_U \frac{1}{2}\|\mathcal{P}_{S^\perp}(X) + \mathcal{P}_S(U\text{v}) - U\text{v}\|_F^2 + \frac{\beta_1}{2}\|U\|_F^2 \\ &= \min_U \frac{1}{2}\|Z - U\text{v}\|_F^2 + \frac{\beta_1}{2}\|U\|_F^2 \end{aligned} \tag{3.11}$$

where $Z = \mathcal{P}_{S^\perp}(X) + \mathcal{P}_S(U\text{v})$. [87] showed that using this approach iteratively, the optimal U can be obtained. According to Frobenius norm properties, (3.11) can be solved by:

$$\hat{U} = \underset{U}{\operatorname{argmin}} \frac{1}{2}Tr[U(A + \beta_1 I)U^T] - Tr(U^T B) \tag{3.12}$$

where $A = \text{v}\text{v}^T$ and $B = Z\text{v}^T$. (3.12) has a solution in [34], where the basis U minimizes a cumulative loss w.r.t the previously estimated coefficients v.

3.2.3 Online Foreground Detection

Let current X and its corresponding L be X_j and L_j, respectively. Also S_j is the indicator vector $\mathbf{s}$ for the j^{th} image. Now we investigate how to compute the foreground mask $\mathbf{s}$ given the residual $E_j = X_j - L_j$ (L_j is computed in background modeling in the previous section for the j^{th} observed image). The goal now is to find the indicator vector S_j on E_j. Assuming that the foreground objects are relatively small connected components, we can model the foreground mask S_j by a Markov Random Field (MRF) [76]. Specifically, let graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices that correspond to the pixels of an image and $\mathcal{E}$ is the set of edges that connect neighboring pixels. Then, by defining an energy function of S_j

$$\beta_2 \sum_{i \in \mathcal{V}}(s_i) + \sum_{(i,k)\in\mathcal{E}} \gamma_{i,k}|s_i - s_k| \tag{3.13}$$

we can derive the foreground mask S_j. β_2 is the cost of assigning the label s_i to the i^{th} pixel. $\gamma_{i,k}$ is also the cost of assigning the labels s_i and s_k to the adjacent pixels i and k.

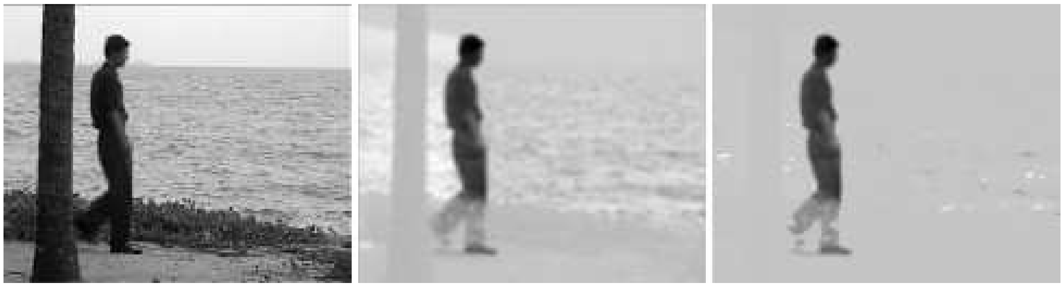


Figure 3.1: The effects of using GMM on outliers obtained from low rank approximation on noisy and dynamic background. The left figure shows an input image, and the middle and right figures show the obtained outliers E and $\hat{E}$, respectively.

The first and the second terms impose sparsity and continuity on S_j, in a way that is similar to the last two terms of (3.6) and shows that S_j can be modeled using MRF [76]. However, extracting foreground objects from E, which is combination of outliers and noise, would not be accurate especially in noisy environment like dynamic backgrounds. In most cases we need to separate reliable outliers representing true foreground from noise in estimating foreground support S_j. In most applications, noise comes from a complicated and dynamic background such as waving trees or sea waves, which should be classified as background.

Here, we describe outliers with a Gaussian model $\mathcal{N}(\mu, \sigma^2)$. Using this model of the outliers enables us to control the complexity of the background variations and also recognize true outliers in the presence of noise. In our method, adaptive Gaussian Mixture Model (GMM) [158] is used for each component of E to separate the true outliers $\hat{E}$ from noise. Fig. 3.1 shows the effect of using GMM on E for dynamic backgrounds.

Now for solving the second step of our optimization problem that extracts moving objects from outliers, (3.6) can be rewritten as the following objective function to minimize the energy over S_j via obtained outliers $\hat{E}$, similar to [155].

$$\begin{aligned}&\min_{S} \frac{1}{2}\|\mathcal{P}_{S^{\perp}}(\hat{E})\|_F^2 + \beta_2\|S_j\|_1 + \gamma\|\Phi(S_j)\|_1 \\ &= \frac{1}{2}\sum_i \hat{E}_i^2(1-s_i) + \beta_2\sum_i s_i + \gamma\|\Phi(S_j)\|_1 \\ &= \sum_i(\beta_2 - \frac{1}{2}\hat{E}_i^2)s_i + \gamma\|\Phi(S_j)\|_1 + C\end{aligned} \tag{3.14}$$

where $C = \frac{1}{2}\sum_i \hat{E}_i^2$ is a constant. (3.14) is the first order MRF with binary labels (the same as (3.13)), which can be solved using graph-cut [14], [71]. The result of (3.14) is the binary mask S_j, which indicates the foreground pixels of X_j. So far, the first iteration of (3.6) is completed and, based on mask S_j, the next iteration starts from (3.8). In our experiments, COROLA converges in approximately r iterations where r is the rank of data in the sequence. Our convergence criterion is similar to [155] and we use $(energy_{prev} - energy)/energy < 10^{-4}$, where $energy = \frac{1}{2}\|(X_j - U\mathrm{V})\|_F^2 + \beta_2\|S_j\|_1$. In this formulation, the first and the second terms show the error of background model, and the foreground object size. The algorithm is considered to have converged if the error of background model and the size of the foreground object stabilize.

3.3 Convergence and Time Complexity of COROLA

In this section, we explain the convergence criteria of COROLA. In general, our main objective function (3.6) is non-convex and we solve it by alternating between two steps. In step one for low-rank approximation, we always minimize a single lower-bounded energy function by two sub-problems to update U and V, which have closed form solutions. In the second step for outlier detection, we use MRF and its convergence has been discussed in [14]. Using these two steps, the algorithm must converge to a local minimum. Furthermore, [155] showed that this combinatorial optimization decreases the energy monotonically through iterations and can converge to acceptable results in background modeling and moving object applications.

3.3.1 Time Complexity

- The complexity of our sequential low-rank approximation by COROLA consists of contributions from two major parts. The computational complexity of the first part is $O(mr)$. The second part of the low-rank approximation in our model are $O(r^2 + mr) + O(mr^2)$. Therefore, the overall complexity of COROLA for the low-rank approximation step is $O(r^2 + mr^2)$.

- The online foreground detection part is the first order MRF with binary labels, which is solved using graph-cut. Since the labels are binary, the solution can be found in polynomial time by computing a minimum cost cut on the graph, which its complexity is bounded by $O(km^2)$ [13]. m is the number of nodes and k is the number of edges in the graph. Since we use 4-neighborhood connectivity and each edge is bidirectional, the maximum edges is $2 \times m$. Therefore, time complexity of online foreground detection part is $O(m^3)$. However, we use GMM on outliers, which boosts the true outliers and (3.14) can converge much faster than its time complexity.

Thus the overall time complexity of the optimization problem (3.6) is $O(r^2 + mr^2 + m^3)$ per iteration.

3.4 Online Moving Object Detection with a Moving Camera

In this part, we extend our moving object detection method to the case of a moving camera. As we mentioned in Section 3.1, due to the dissimilarity between the first and the last images in a sequence, a batch method is not able to deal with continuous processing using a moving camera. However, in online methods the background model evolves with time and similarity between the first and the current image is not required. In our method, we build the background model for the current image and based on a transformation function between the current and the new image, the model is transformed to be matched with the new image. Then we can update it for the new image to detect the foreground objects. Note that the background model is transformed through time. So the key in foreground detection using a moving camera is the transformation of the low-rank structure to the new input image.

Let τ_j be a transformation that maps X_{j-1} to X_j. This transformation is obtained from an affine transformation estimated from the two 2D images. We also assume $X_{j-1} = U_{j-1}\mathrm{v}_{j-1}$ and there is no changes into both last two images except for affine transformation so that $X_j = \tau \circ X_{j-1}$. For the sake of brevity, we state without proof that the following equation allows us to reconstruct the current view X_j from the background model and the registration transform τ_j.

$$X_j = \tau_j \circ X_{j-1} = (\tau_j \circ U_{j-1})\mathrm{v}_{j-1} \tag{3.15}$$

From (3.15) the transformation only changes U where $\bar{U}_{j-1} = \tau \circ U_{j-1}$. Based on the above assumptions and (3.15), $U_j = \bar{U}_{j-1}$ and $\mathrm{v}_j = \mathrm{v}_{j-1}$. After the transformation, some elements of $\bar{U}_{j-1}$, which are related to the pixels on the border of the current image, have no corresponding pixels and we have to estimate them using other pixels. Using X_j and v_{j-1} we estimate missing pixels of $\bar{U}_{j-1}$ by replacing them by the corresponding values obtained from [152], as follows.

$$\bar{U}_{j-1} = X_j \mathrm{v}_{j-1}^T (\mathrm{v}_{j-1}\mathrm{v}_{j-1}^T)^\dagger \tag{3.16}$$

Based on the experimental results, this approach can estimate missing pixels of U after transformation. In addition, the location of GMM parameters for the previous E_{j-1} should be transformed via τ to match with the current E_j. After all of these transformations, we can apply the COROLA method for a static camera to build the background model and detect the foreground objects. Fig. 3.2 shows a sample image, its computed background model and extracted moving object via COROLA, together with the intermediate results.

3.5 Experimental Results

In this section, we compare COROLA with competing algorithms in the literature. We perform two sets of experiments on synthetic data and real bench-

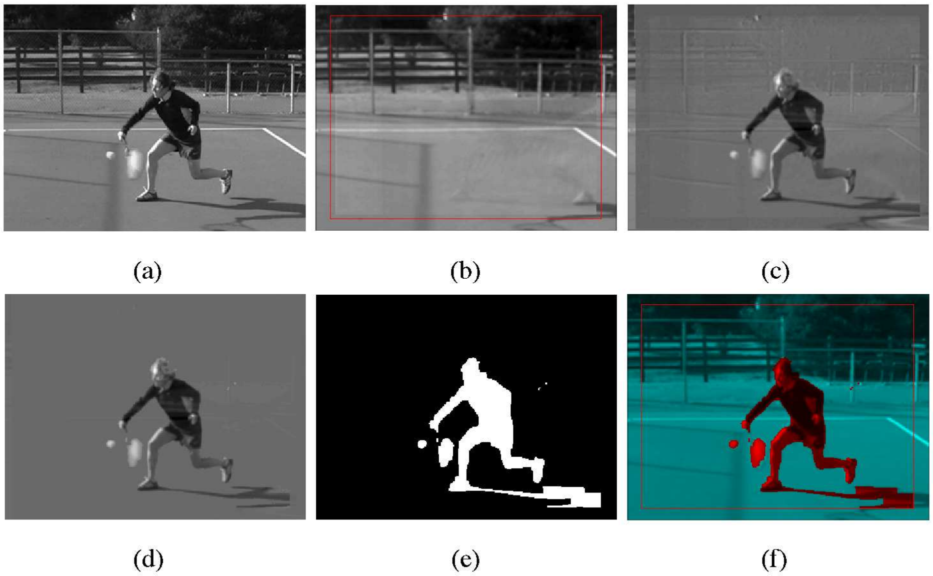


Figure 3.2: An example of COROLA for a moving camera. (a) input image from a sequence (b) background model (c) E, (d) $\hat{E}$, (e) S, and (f) extracted foreground object using mask S. Red lines show the processing area.

mark datasets and show quantitative and qualitative results. For quantitative evaluation where ground truth is available, we use pixel-level precision and recall, defined as follows:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN} \tag{3.17}$$

where TP, FP, TN, and FN are the numbers of true positives, false positives, true negatives and false negatives, in pixels, respectively. Also, instead of using precision-recall curves, we use F-measure to show the overall accuracy [18].

$$\text{F-measure} = 2\,\frac{precision \times recall}{precision + recall} \tag{3.18}$$

In all experiments $\beta_1 = 0.1$. For coefficients β_2 and γ, we use the same strategy as [155]. We set $\beta_2 = 0.02$ for each input image, and decrease β by a factor of 2 in each iteration with a lower bound of 0.002. We also set $\gamma = 5\beta_2$.

3.5.1 Synthetic Data

In this set of experiments, we use synthetic data to control noise and to show the capability of COROLA. The synthesized images are 30 × 100 pixels ($m =$

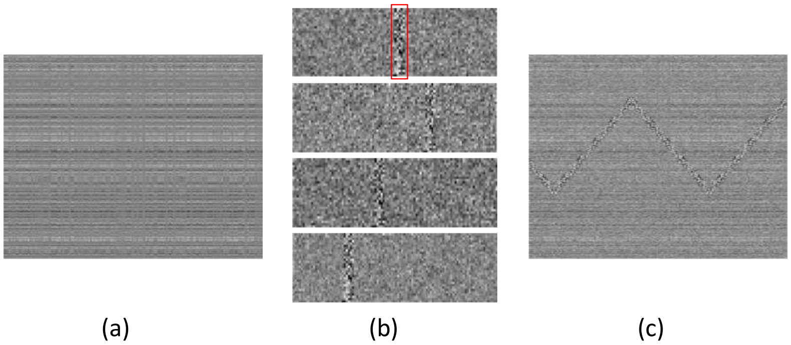


Figure 3.3: An example of synthetic data. (a) shows matrix $L \in R^{3000\times200}$, with $m = 3000$, $n = 200$, and rank $r = 5$, where $L = UV$, $U \in R^{3000\times5}$, and $V \in R^{5\times200}$. (b) shows some sample images from selected column of L, where an object is superimposed each of them. The object is represented by a red box in the first image in (b). other images show the movement of the object to left and right of the image, frequently. (c) shows a sample of generated matrix D as the input data.

3000). We use $n = 200$ images. Zhou *et. al.* [155] used the similar scheme to investigate the robustness of their method against outliers.

To visualize the results we show all images in a 2D matrix where each column shows one image of the sequence. We generate the input data D by adding a foreground to a background matrix L. For generating the foreground and background we use the same approach as DECOLOR. The background matrix $L = UV$ is generated via $U \in R^{m\times r}$ and $V \in R^{r\times n}$ with random samples from a standard normal distribution. An object with a small size is superimposed on each image in matrix L, and shifts from left to right of the images by one pixel per image, until the right border of the image. The motion direction of the object is then reversed, and the process repeats. Fig. 3.3(b) shows some selected images. The intensity of this object is independently sampled from a uniform distribution. Also, we add i.i.d Gaussian noise ϵ to D with the corresponding signal-to-noise ratio defined as

$$SNR = \sqrt{\frac{var(L)}{var(\epsilon)}} \tag{3.19}$$

Figs. 3.3(a), (b) and (c) show an example of generated L, the movement of generated foregrounds and the obtained matrix D.

We test the COROLA method and compare it with leading online methods such as GRASTA, OPRMF, ORPCA and DECOLOR, one of the best batch methods, in terms of different SNR ratios, different ranks of matrix, and different sizes of the foreground object. One sample of our experiments with different SNR ratios between COROLA and all mentioned methods is shown in Fig. 3.4. In the first row of Fig. 3.4, with $SNR = 10$, COROLA, OPRMF and DECOLOR methods have roughly the same results for extracting the foreground object, but when we increase noise in the second row ($SNR = 1$), COROLA method works better than all other methods including DECOLOR in extracting the moving object. That is mainly attributed to using GMM to compute the coefficients of outliers to separate the foreground object from noise. Tuning up the outliers coefficient via GMM enables us to separate noise and outliers especially in a noisy environment and the result becomes more and more accurate over time.

To evaluate COROLA in comparison with GRASTA, OPRMF, ORPCA, and DECOLOR methods, we tested the effects of some scene parameters such as SNR, rank of matrix D, and size of the object. The quantitative results of this comparison in terms of F-measure are provided in Fig. 3.5. The first column of Fig. 3.5 illustrates the effect of noise in all methods, when we change the SNR ratio from 8 to 1 in different ranks. The rows from top to bottom show our experiments in different ranks of 1, 3, and 5. Since one of the advantages of DECOLOR method is high accuracy of object detection with different sizes,

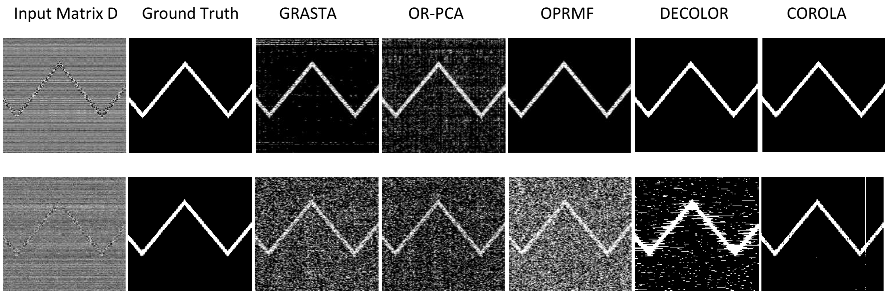


Figure 3.4: Comparison of COROLA, GRASTA, ORPCA, OPRMF and DECOLOR with different SNR ratio. The first row and the second row show the results of the methods with $SNR = 10$, and $SNR = 1$, respectively.

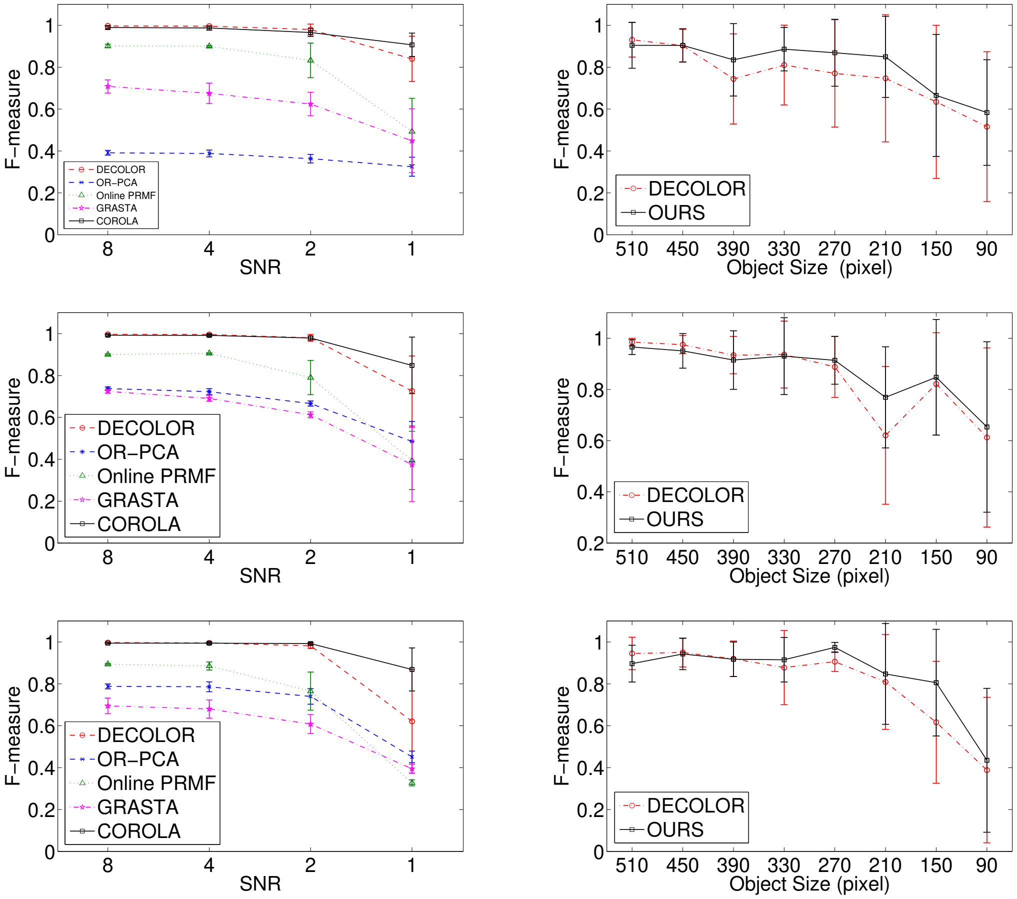


Figure 3.5: First column: the comparison in terms of F-measure between COROLA and other methods with different signal-to-noise (SNR) ratio. Second column: the comparison of F-measure between COROLA and DECOLOR with different object size. The three rows show three different ranks at 1, 3, and 5 respectively.

the second column of Fig. 3.5 shows the accuracy of COROLA in comparison with DECOLOR to extract the moving object of different sizes. This result demonstrates that the capability of our method is comparable with DECOLOR in terms of average F-measure. Although, the result of DECOLOR method is more accurate than COROLA for large objects, by reducing the size of object, COROLA generates a better result than DECOLOR even when we increase the rank of matrix D from 1 to 5.

3.5.2 Real Data

In this section, we use real benchmark datasets to conduct quantitative and qualitative evaluation of COROLA. The real datasets used are popular in moving object detection and publicly available[1], and they include "2014 Change Detection" [46], "Perception or I2R" [74], and "Wallflower" [124] image sequences. Table 3.1 provides the length and image size of the sequences used in our experiments.

Table 3.1: Details of all sequences used in our experiments for stationary camera

Dataset	Sequences	Size × #frames
I2R	Water surface	[160,128] × 48
	Fountain	[160,128] × 523
	Curtain	[160,128] × 2964
	Hall	[176,144] × 1927
	Campus	[160,128] × 372
	Escalator	[160,130] × 824
	Lobby	[160,128] × 138
	ShoppingMall	[320,256] × 433
Change Detection	Canoe	[320,240] × 1189
	Fall	[180,120] × 1500
	Fountain02	[216,144] × 720
	Overpass	[320,240] × 3000
Wallflower	Waving trees	[160,120] × 287
	ForegroundAperture	[160,120] × 489
	TimeOfDay	[160,120] × 1850

Evaluation by accuracy

Figs. 3.6, 3.7, and 3.8 show the qualitative results of COROLA for background estimation and foreground detection for all sequences of Table 3.1 from three datasets I2R, Change Detection, and Wallflower, respectively. Figs. 3.6, 3.7,

[1] https://sites.google.com/site/backgroundsubtraction/test-sequences

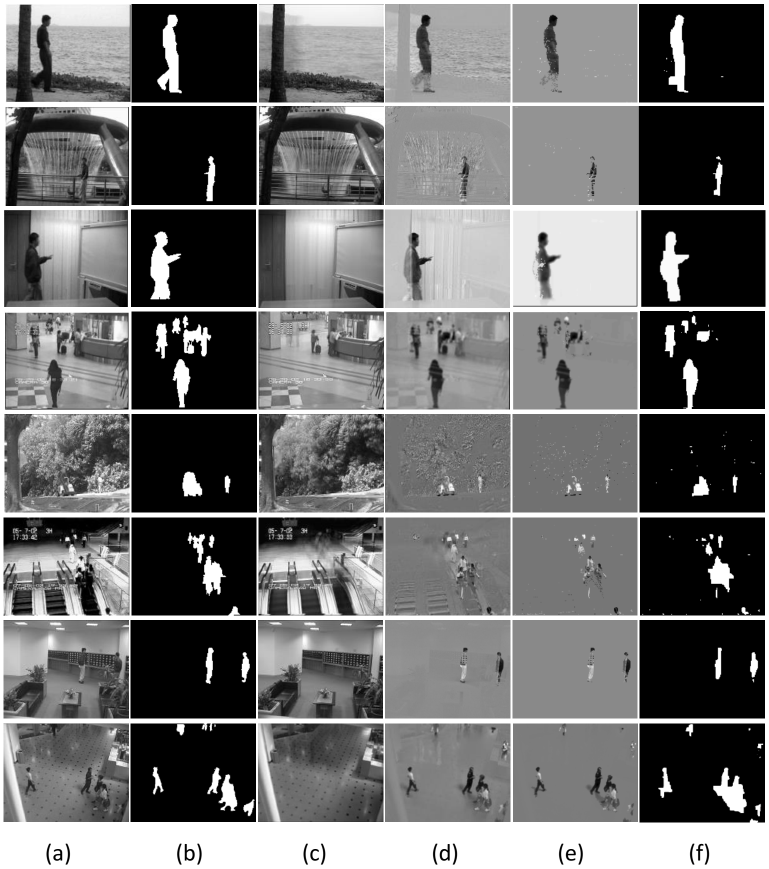


Figure 3.6: The results of COROLA on 8 sequences from I2R dataset. Columns (a) and (b) show the original query image and the ground truth (GT) for the foreground. Columns (c) and (f) show the results of COROLA for estimating the background L, and the detected foreground objects S, respectively. Columns (d) and (e) show intermediate results for outliers E, and $\hat{E}$, respectively.

and 3.8 also shows the role of GMM to separate outliers from noise. These results are shown in columns (d) and (e) as E, and $\hat{E}$, respectively. The results in Figs. 3.6, 3.7, and 3.8 demonstrate the capability of COROLA to detect moving objects and background modeling accurately. The estimated background in the first row of Fig. 3.6 has some ghost because the input image is the 23^{rd} of the sequence and the parameters have not been learned well enough to build an accurate background. In general, for short sequences the computed background model by a batch method such as DECOLOR is more accurate than COROLA because online methods need sufficient samples

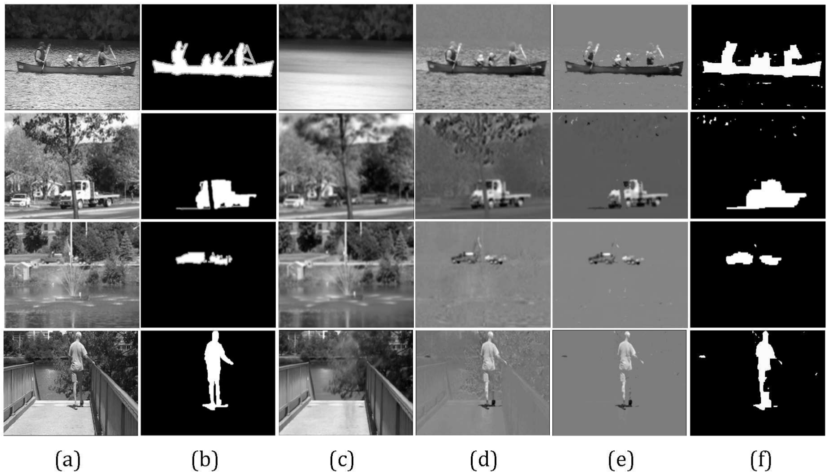


Figure 3.7: The results of COROLA on 4 sequences from Change Detection dataset. Columns (a) and (b) show the original query image and the ground truth (GT) for the foreground. Columns (c) and (f) show the results of COROLA for estimating the background L, and the detected foreground objects S, respectively. Columns (d) and (e) show intermediate results for outliers E, and $\hat{E}$, respectively.

for training to be stable. However, for long sequences COROLA can provide comparable results with batch methods.

We also compare COROLA quantitatively with competing online and batch methods. However, by tuning the parameters of GMM, which is common in online methods, COROLA can provide even more accurate results. Table 3.2 compares COROLA with MOG, GRASTA, OPRMF, and ORPCA in terms of F-measure. In most of the cases COROLA works much better than all other online methods, specifically in very noisy and dynamic scenes such as Fountain, Campus, Canoe, Fall, and Fountain02 sequences. Because in these sequences moving parts of background are often classified as foreground in other online methods. In contrast, COROLA is able to deal with the difficult background conditions. By using GMM the difference between outliers and the rest of pixels is boosted and this allows COROLA to detect intermittently moving objects better than other competing online methods.

Table 3.3 compares COROLA with IALM, FPCP, GoDec, SSGODec, APG, and DECOLOR, which are fast and accurate batch methods in the literature, in terms of F-measure. For some sequences such as Fountain, Campus, Canoe,

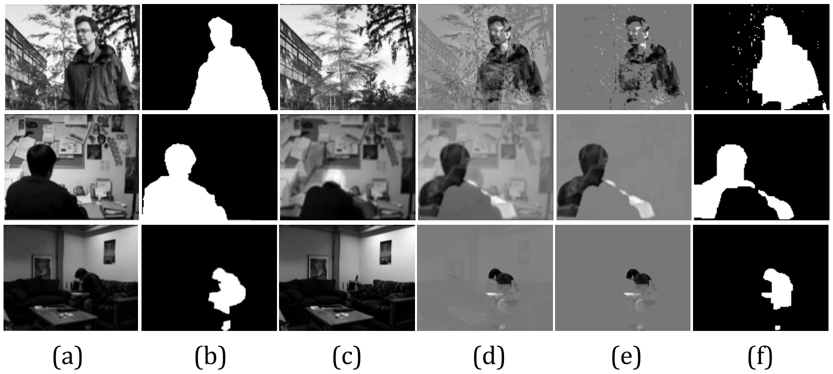


Figure 3.8: The results of COROLA on 3 sequences from Wallflower dataset. Columns (a) and (b) show the original query image and the ground truth (GT) for the foreground. Columns (c) and (f) show the results of COROLA for estimating the background L, and the detected foreground objects S, respectively. Columns (d) and (e) show intermediate results for outliers E, and $\hat{E}$, respectively.

Fountain02, Overpass, and TimeOfDay COROLA works much better than other methods. Because in some of these sequences, background is very noisy (i.e. Campus and Fountain02), the constraints of connectedness and sparseness on the subspace of images prove to be useful, which both DECOLOR and COROLA methods exploit leading to much better results than other methods. Further, in some cases the objects move very slowly (i.e. Canoe) or stop for a long time (Overpass, Fountain, and TimeOfDay) none of the competing methods can produce accurate results. In contrast, COROLA produces acceptable results for these challenging sequences for the same reasons as for the results of Table 3.2, i.e., using GMM the difference between outliers and the rest of pixels is boosted and so COROLA can detect intermittently moving objects better than other methods. In summary, Tables 3.2 and 3.3 convincingly demonstrate that our method outperforms the state-of-the-art online methods, and provides comparable results with the batch methods in terms of F-measure.

Computational Cost

COROLA is implemented in Matlab and C++. We run all experiments on a PC with a 3.4 GHz Intel i7 CPU and 16 GB RAM. To show the importance of online methods in continuous operation we compare the scalability of

Table 3.2: Comparison of F-measure score between COROLA and online methods

Sequence	MOG	GRASTA	OPRMF	ORPCA	COROLA
WaterSurface	0.4723	0.7531	0.5483	0.6426	**0.9129**
Fountain	0.7766	0.4978	0.2393	0.2870	**0.8833**
Curtain	0.7709	0.7046	0.4199	**0.8504**	0.8236
Hall	0.5802	0.7471	0.7215	0.7329	**0.7808**
Campus	0.4510	0.1885	0.1700	0.1893	**0.7177**
Escalator	0.3869	0.5474	0.5179	0.4452	**0.7858**
Lobby	0.5628	**0.8231**	0.6728	0.6336	0.8128
ShoppingMall	0.5275	0.6816	0.6621	0.5541	**0.7494**
Canoe	0.5114	0.5386	0.4400	0.5152	**0.7403**
Fall	0.5420	0.5057	0.4929	0.4030	**0.8422**
Fountain02	0.7801	0.3569	0.2926	0.4684	**0.8205**
Overpass	0.5095	0.5609	0.5105	0.6079	**0.7732**
WavingTrees	0.6639	0.7354	0.5259	0.6315	**0.8475**
ForegroundAperture	0.2601	**0.6757**	0.5628	0.6118	0.6401
TimeOfDay	0.6147	0.5645	0.5258	0.6315	**0.8291**

COROLA with DECOLOR under varying spatial resolution and the number of images.

Unlike DECOLOR, the computational cost of COROLA is independent of the number of images because the dominant cost of DECOLOR comes from the computation of SVD in each iteration. By increasing the size of the matrix D, the computation time of DECOLOR grows at least linearly with respect to the number of images. We compare the computation time of COROLA with DECOLOR after convergence of both methods in Table 3.4. In this table, the average time for processing of each frame by DECOLOR increases where it is an order of magnitude slower than COROLA for sequences longer than 1000 images.

Scalability in spatial resolution is another advantage of online method against batch processing methods. Increasing the resolution of images sig-

Table 3.3: Comparison of F-measure score between COROLA and batch methods

Sequence	IALM	FPCP	GoDec	SSGoDec	APG	DECOLOR	COROLA
WaterSurface	0.3519	0.4910	0.4304	0.4473	0.5907	0.9022	**0.9129**
Fountain	0.1633	0.1894	0.1531	0.2574	0.2641	0.2075	**0.8833**
Curtain	0.3184	0.5290	0.3706	0.4344	0.7260	**0.8700**	0.8236
Hall	0.5716	0.7295	0.7128	0.5713	0.7601	**0.8169**	0.7808
Campus	0.1660	0.1701	0.1640	0.1649	0.1979	**0.7811**	0.7177
Escalator	0.5066	0.5192	0.1316	0.5075	0.5440	**0.8205**	0.7858
Lobby	0.3213	0.7188	0.7393	0.6194	0.7286	0.6579	**0.8128**
ShoppingMall	0.6093	0.6256	0.6143	0.5880	0.7057	0.6382	**0.7494**
Canoe	0.5072	0.5169	0.5107	0.3091	0.4193	0.1603	**0.7403**
Fall	0.4112	0.4191	0.4137	0.4236	0.5232	**0.8760**	0.8422
Fountain02	0.2553	0.3066	0.2713	0.2714	0.3204	**0.8327**	0.8205
Overpass	0.5492	0.5528	0.5454	0.5517	0.5698	0.3573	**0.7732**
WavingTrees	0.5130	0.5130	0.5113	0.1829	0.7031	**0.8845**	0.8475
F-A	0.3233	0.3238	0.3238	0.6854	**0.7200**	–	0.6401
TimeOfDay	0.1523	0.2187	0.1630	0.1664	0.6808	0.4683	**0.8291**

nificantly affects DECOLOR method. Using high resolution images results in a huge matrix D so that decomposing D becomes very expensive. On the other hand, COROLA is an online method and is independent from the number of images, i.e., we do not have to deal with a large D and its computation time grows only with the image resolution.

3.5.3 Experiments on a Moving Camera

In this section, we test our method on real public sequences for moving cameras namely “Berkeley motion segmentation dataset” [125]. Table 3.5 shows the details of four sequences that we use in our experiments.

We compare our method with DECOLOR as the leading method based on low-rank approximation that can handle the problem of object detection with a moving camera in a short sequence. Fig. 3.9 shows the qualitative results of

Table 3.4: Time evaluation of COROLA with DECOLOR method

Methods	Resolution × #images	Low Rank (s)	MRF (s)	Total (s)
DECOLOR	[320 × 240] × 200	0.1036	0.0828	0.1864
	[320 × 240] × 400	0.1531	0.1297	0.2828
	[320 × 240] × 600	0.1687	0.1601	0.3279
	[320 × 240] × 800	0.2016	0.1825	0.3841
	[320 × 240] × 1000	0.3948	0.3191	0.7139
COROLA	[320 × 240] × 1000	0.0231	0.0605	0.0836

COROLA in comparison with DECOLOR method for moving object detection using a moving camera. First two experiments have been performed on short sequences “cars7”, “people1” and the results from COROLA are comparable with those from DECOLOR method. For the last two sequences “marple13” and “Tennis”, DECOLOR has a problem to align images when the last images are not similar with the first images of these sequences. This is common in continuous processing and all of batch methods have problem with this. To show the result of DECOLOR on marple13 and tennis sequences (in the last two rows of Fig. 3.9), we used last 30 images of the sequences, which have less camera motion. Since the last images in the sequence are no longer similar to the initial ones in the matrix, DECOLOR failed, as expected. In contrast, since COROLA works online and only considers the last two images it can process the last two sequences of Table 3.5 without any problems and provides acceptable results in comparison with DECOLOR.

Table 3.5: Details of all sequences used in our experiments for moving camera

Dataset	Sequences	Size × #frames
Berkeley motion segmentation	cars7	[320,240] × 24
	people1	[320,240] × 40
	tennis	[320,240] × 200
	marple13	[320,240] × 75

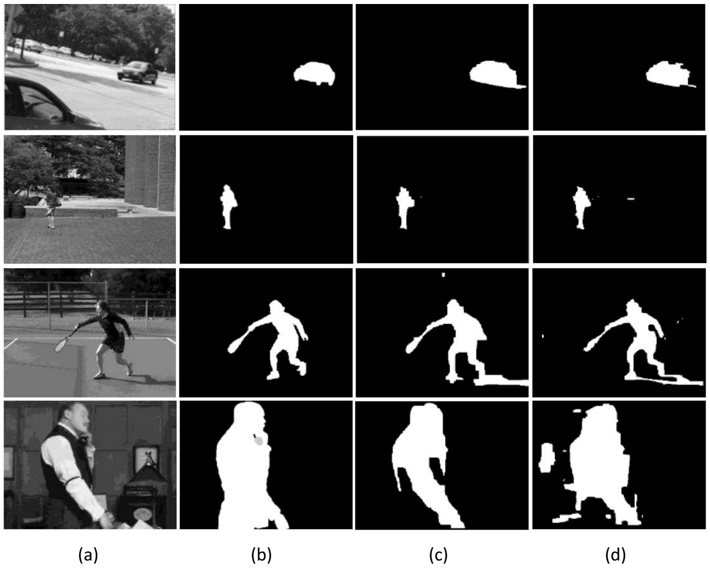


Figure 3.9: Comparison of foreground objects between DECOLOR and COROLA. columns (a) and (b) show the input image and its ground truth. columns (c) and (d) show the obtained foreground mask for DECOLOR and COROLA methods, respectively.

Table 3.6: Comparison of F-measure score between DECOLOR and COROLA

Sequence	DECOLOR	COROLA
cars7	**0.8441**	0.8296
people1	**0.9666**	0.9113
tennis	—	**0.8184**
marple13	—	**0.7943**

Table 3.6 shows the quantitative evaluation of COROLA in comparison with DECOLOR. Experiments over all four sequences show that the results of COROLA is comparable with DECOLOR but has the advantage in terms of its ability for real-time continuous processing. With more than 30 images in a sequence, DECOLOR can no longer produce a valid result due to the significant dissimilarity of the images later in the sequence from the initial ones. In contrast, our sequential method is always able to produce a valid result.

3.6 Summary

In this chapter, we proposed a novel online method named COROLA to detect moving objects in a video using the framework of low-rank matrix approximation. Our online framework works iteratively on each image of the video to extract foreground objects accurately. The key to our online formulation is to exploit the sequential nature of a continuous video of a scene where the background model does not change discontinuously and can therefore be obtained by updating the background model learned from preceding images. We have applied COROLA to the case of a moving camera. Since our method works online and is independent of the number of images, it is suitable for real-time object detection in continuous monitoring tasks. Our method overcomes the problems of batch methods in terms of memory storage, time complexity, and camera motion. Also important to the success of COROLA is using Gaussian model to separate noise from outliers, especially in the case of dynamic background. Based on our extensive experiments on synthetic data and real data sequences, we are able to establish that COROLA achieves the best performance in comparison with the state-of-the-art online methods. COROLA also outperforms most of the batch evaluated methods, and provides comparable results to DECOLOR.

Despite its satisfactory performance in all of our experiments, COROLA shares one disadvantage with DECOLOR. Since both methods have non-convex formulations, they might converge to a local minimum with results depending on initialization of parameters; however, for the case of background modeling, images are roughly similar and parameters do not change significantly. Therefore, the issue of local minimum has not affected successful object detection in our experiments. A challenge facing COROLA is severe illumination changes and this is a problem of all online and batch methods. In the next chapter, we will propose a method that can work under severe illumination changes.

Chapter 4

Low-Rank and Invariant Sparse Decomposition

4.1 Introduction

As explained in Chapter 1, existing low-rank based methods for the problem of moving object detection have two difficulties to deal with long term continuous tasks and significant change in illumination. Although our proposed method in Chapter 3, can address the first issue, it is still vulnerable to significant illumination changes that arise in certain applications, which is a problem for all existing online and batch methods. The problem is particularly challenging when the frames of the image sequence are not continuous in time as the result of the capturing process, such as in motion-triggered or time-lapse photography as explained in Chapter 1. This challenge arises primarily from the significant illumination variation among the frames of the sequence that confuses appearance change due to object motion and that due to illumination. In the rest of this thesis, we focus on this challenge and we are interested in moving object detection in applications involving time-lapse image sequences for which current methods mistakenly group moving objects and illumination changes into foreground.

As discussed in Section 2.2.2, current methods use temporal and spatial constraints on the sparse outliers, and they could improve the performance of moving object detection in frame-rate sequences. However, those are not able to distinguish between discontinuous changes caused by illumination and those caused by moving objects in the scene, especially in time-lapse image sequences due to the discontinuous change in both illumination and object location. In other words, there is no solution in the literature to detect moving objects under significant illumination change that can occur in time-lapse image sequences including shadow and abrupt or discontinuous change in illumination.

To address this issue, in this chapter, we propose a method using low-rank and sparse decomposition framework that not only decomposes the images into low-rank and sparse outliers but is also able to further separate the sparse outliers into those due to moving objects, and those due to illumination. However, separating the sparse outliers into two sparse matrices is an ill-posed problem. We address this challenge by proposing a robust representation of images against illumination, which can serve as prior information in our formulation.

Since changes due to illumination and shadow are easily lumped with moving objects and detected as the sparse outliers in the low-rank formulation, we then compute a prior map using the obtained illumination invariant representation of images to provide information about the effect of illumination. Finally, we define two penalty terms based on the prior map to decompose an image into three parts: the background model, illumination changes, and moving objects or real changes.

The key to our solution is incorporating the prior information in the low-rank approximation (LRA) framework to form our proposed low-rank and invariant sparse decomposition (LISD) method. We also propose an iterative version of LISD (ILISD) to improve the performance of LISD by updating the prior map. Since we use two representations (grayscale and illumination invariant representations), the prior map in ILISD is updated iteratively from the results of each representation that is used as a constraint in another representation.

4.2 Robust Image Representation Against Illumination

In this section we propose a method to provide an illumination invariant representation of an image, which can be used in both indoor and outdoor applications. We use this invariant image representation to obtain the prior information, which enables us to distinguish between moving objects and illumination changes. As discussed in Chapter 2.3, illumination invariant and shadow free images have been well studied and many methods have been proposed [36], [38], [47], [157]. One of the most popular and fastest methods for this task is proposed by Finlayson *et al.* [36]. This method assumes the camera sensor sensitivities are Dirac delta functions and illumination can be modeled by Planck's law. For removing the effect of illumination, [36] computes the two-vector log-chromaticity $\chi^{'}$ using red, green and blue channels. Finlayson *et al.* [36] showed that by changing illumination, $\chi^{'}$ moves along a straight line e roughly. Projecting the vector $\chi^{'}$ onto the vector orthogonal to e, which is called invariant direction, the invariant representation I is computed as follows.

$$I = \chi^{'} e^{\perp} \tag{4.1}$$

The best direction for e can be found by minimizing Shannon's entropy [36]. Fig. 4.1 shows the details of this concept. Although this method works with the mentioned assumptions for some real images, in case of significant illumination changes, specially if the assumptions do not hold, χ' necessarily does not move along a straight line. This issue causes two major problems in the invariant representation I.

P1 : First, in the process of projection onto the orthogonal direction of illumination variation, some pixels with the same log-chromaticity but from different objects are projected to the same location in the orthogonal vector, and the invariant representation removes much meaningful information about the image, especially around edges.

P2 : Secondly, χ' vectors of the same material under different illumination

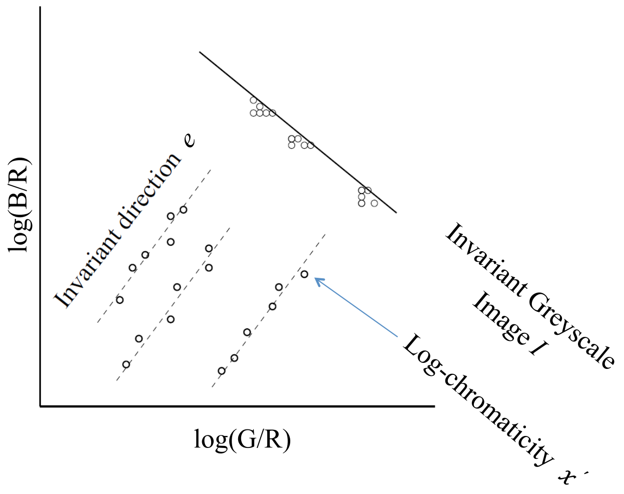


Figure 4.1: Illumination invariant representation of an image using [36]

are not projected to the same location in the orthogonal vector and therefore, the method cannot remove the effect of illumination accurately.

In the next section, we propose a solution for the first issue (P1) with some experimental results of the method. By solving the first issue, the obtained invariant representation becomes appropriate to use in our formulation for detecting moving object. Since the second issue can be solved by a low-rank decomposition, we will discuss on P2 separately, in Section 4.3.1.

4.2.1 Solution to Preserve the Structural Information of Illumination Invariant Representation

To alleviate the effect of the first issue (P1) for preserving the structural information of images, we extract invariant features $\tilde{I}$ from each image using Wiener filter [50], which has been used successfully for face recognition in [26]. Wiener filter decomposes a signal into its components from two stationary processes with different autocorrelation functions, where the Fourier transform of the autocorrelation function is the power spectral density in the frequency domain. One reason to use wiener filter is that this method retains features at

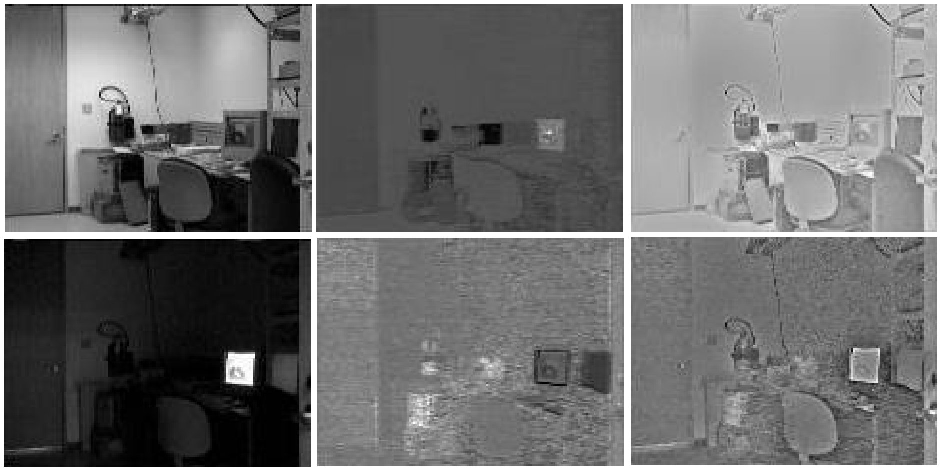


Figure 4.2: Columns from left to right: two images with extreme illumination changes, their corresponding invariant image I, and their corresponding final invariant representation I_{inv}.

every frequency [26]. We add $\tilde{I}$ to the invariant image I, which is obtained from (4.1). This final invariant representation is called I_{inv}.

Fig. 4.2 shows the effect of adding the invariant features $\tilde{I}$ to the invariant representation I. First column shows two images from one scene with the light switch on/off and the second column shows the corresponding invariant image I. Last column illustrates I_{inv}, the results of adding invariant features $\tilde{I}$ to the invariant image I. To combine $\tilde{I}$ and I, we use simple weighted averaging. All details of this process are explained as follows.

According to [6], an image can be represented as

$$I(x, y) = R(x, y)L(x, y) \tag{4.2}$$

where $I(x, y), R(x, y)$, and $L(x, y)$ are the intensity, the reflectance, and the illuminance of the pixel location (x, y), respectively. By taking the logarithm of I and transforming the model into an additive form we have:

$$f(x, y) = \nu(x, y) + \mu(x, y) \tag{4.3}$$

where $\nu = \log R$ and $\mu = \log L$. Let f, ν, and μ be drawn from three wide-sense stationary processes and assume the latter two are uncorrelated. Although the estimation of μ and ν is highly ill-posed, [26] showed that we can estimate μ

from a single image by Wiener filter with the optimal filtering setting and then produce the log reflectance by $\nu = f - \mu$. The stationary condition of natural images is only satisfied in the one-dimensional case, unfortunately, and severely violated in the two-dimensional case [26], [98]. To overcome this difficulty, since x and y directions are two dominant directions in a two dimensional power spectrum, we restrict ourselves to the one-dimensional power spectrum to filter an image in the x and y direction separately.

Let both μ and ν follow power law spectrum as follows.

$$P_\mu(\omega) \propto \omega^{-\alpha_\mu}, \quad P_\nu(\omega) \propto \omega^{-\alpha_\nu} \tag{4.4}$$

where P_μ and P_ν are the power spectral densities of μ and ν, respectively, and α_ν and α_μ are positive real numbers.

Now, consider the Wiener filter in the frequency domain:

$$F\{l\}(\omega) = \frac{P_\mu(\omega)}{P_\mu(\omega) + P_\nu(\omega)} = \frac{\lambda}{\lambda + \omega^\gamma} \tag{4.5}$$

where l is the Wiener filter in the spatial domain, λ is the ratio of power spectra P_μ and P_ν at the frequency $\omega = 1$, and $\gamma = \alpha_\mu - \alpha_\nu$. To obtain a satisfactory illumination invariant image we need to estimate γ by estimating the power spectrum of μ and ν.

Let $f_{s,t}$ denote the logarithm of the image from scene s under illumination condition t. Now, let us assume that the scene is fixed and that the illumination changes, so that the autocorrelation of the sequence $f_{s,1}, ..., f_{s,T}$ (T is the number of illumination conditions) can be approximated by the autocorrelation of the sequence $\mu_{s,1}, ..., \mu_{s,T}$ where $\nu_{s,1} = \nu_{s,2} = ... = \nu_{s,T}$ and the autocorrelation of the sequence ν should be close to 0. On the other hand, if we consider S different scenes with the same illumination, $(f_{1,t}, ..., f_{S,t})$, then we can approximate the autocorrelation of the sequence $\nu_{1,t}, ..., \nu_{S,t}$ where $\mu_{1,t} = \mu_{2,t} = ... = \mu_{S,t}$. Since the power spectrum density is the Fourier transform of autocorrelation, we can obtain an estimation for P_μ and P_ν.

Although computing P_μ and P_ν enables us to solve (4.4) directly to estimate μ, and therefore ν can also be computed, such computation needs data from different scenes and different illuminations. Training data of this kind may be available in some applications such as classification tasks, in most of the computer vision or robotics applications we do not have such information in general (for example for moving object detection or place recognition), and we would still like to evaluate (4.5) with a single image.

(4.5) can be evaluated if we can estimate γ and λ. In order to estimate γ, Torralba *et. al.* [123] showed for a large range of natural images γ is around 2. Since in most of computer vision and robotics applications all images are natural, we set $\gamma = 2$. Next, in order to estimate μ, we express (4.5) directly in the spatial domain similar to [26] as follows.

$$\lambda l[f_x] + \frac{\partial^2}{\partial x^2}(l[f_x]) = \lambda f_x \tag{4.6}$$

$$\lambda l[f_y] + \frac{\partial^2}{\partial y^2}(l[f_y]) = \lambda f_y \tag{4.7}$$

where f_x and f_y are a row and a column of the input image, respectively. (4.6) and (4.7) can be solved after discretization as follows.

$$(\lambda I + D^T D)\mu = \lambda f \tag{4.8}$$

where $f = (f_1, ..., f_n)^T$ is f_x or f_y, $\mu = (\mu_1, ..., \mu_n)^T$ is μ_x or μ_y, and D is a $(n-1) \times n$ difference matrix: $D_{i,j} = -1$ if $i = j$, $D_{i,j} = 1$ if $i = j - 1$, and $D_{i,j} = 0$ otherwise. Since we approximate γ by 2, λ could no longer be the ratio of power spectra and need to be chosen empirically [26]. For all experiments we set $\lambda = 0.5$.

Fig. 4.3 shows two sample invariant images against illumination obtained from a day image and a night image. Fig. 4.3(b) shows the obtained invariant features and Fig. 4.3(c) is the original invariant representation I. Since the obtained ν is sparse, we are not able to use it for moving object detection where the vlaue of all pixels is required. In addition, chromaticity information of the image has been removed from ν. For example the chromaticity of the tree in the second column of Fig. 4.3 is removed by Wiener filter although it has useful details. Therefore we add illumination invariant features ν to the invariant representation I. Since for dim regions of an image, the obtained invariant representation I cannot generate information from the image, we

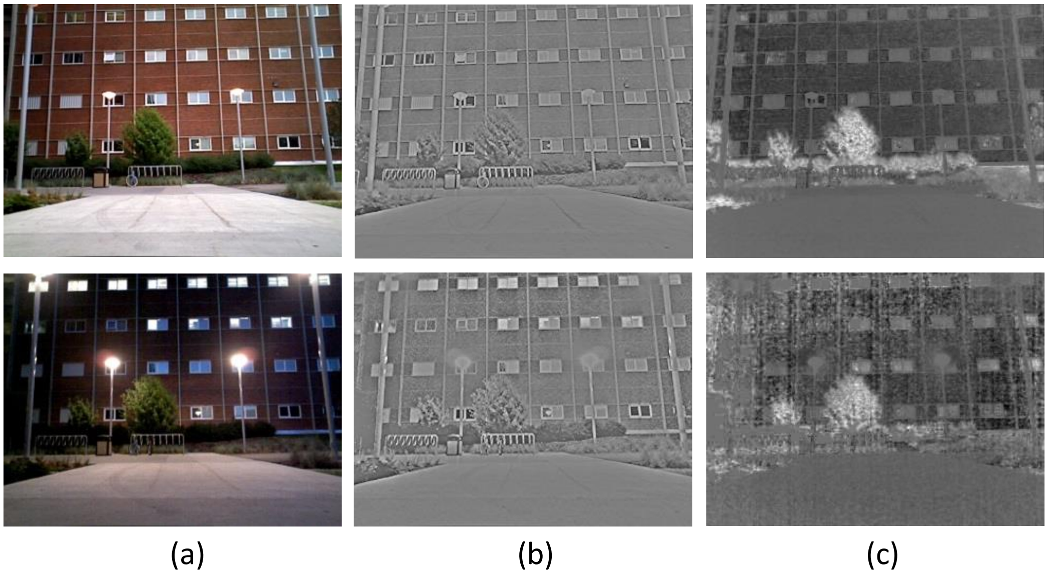


Figure 4.3: Illumination invariant images. First column shows the original images and the second column shows their illumination invariant features of images at two different times (day and night). Last column shows the obtained invariant images from [36].

normalize the original observed image X between $[0, 1]$ and use it as a mask. Then for the pixels close to zero in ν ($0 \leq |\nu| \leq 0.1$) which have not meaningful information we use the following simple equation.

$$I_{inv}(x, y) = \nu(x, y) \times (1 - X(x, y)) + I(x, y) \times X(x, y) \tag{4.9}$$

where for dim regions of an image, the effect of ν is enhanced as a result in order to build an accurate invariant image. Using I enables us to use the chromaticity of the image in the final invariant image that has been removed from ν. So, combining invariant features ν with I from the second row of Fig. 4.3 using (4.9) provides Fig. 4.4(c) which can recover details of an image through chromaticity information at the same time.

4.2.2 Implementation Details and Execution Time

To summarize, to build an illumination invariant representation of an image:

- First we compute μ_x and μ_y using (4.6) and (4.7), respectively by solving (4.8) twice for the two directions.

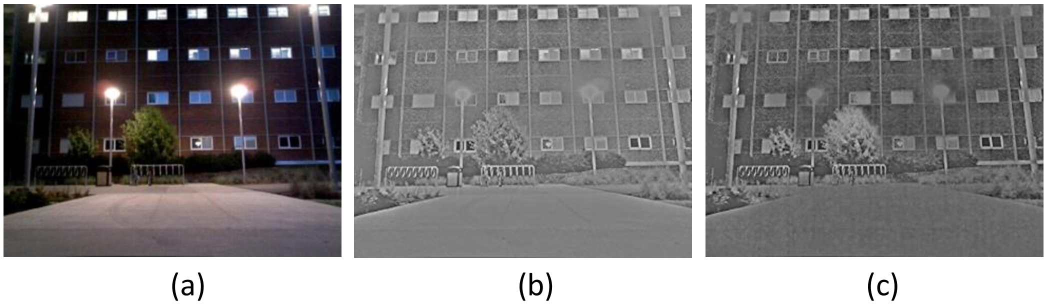


Figure 4.4: Illumination invariant images. (a) original image, (b) the obtained ν, and (c) Our proposed illumination invariant representation

- Then we obtain $\nu = \nu_x + \nu_y$ where $\nu_x = f_x - \mu_x$ and $\nu_y = f_y - \mu_y$.
- Next, we compute the invariant representation I using [36] as discussed in Chapter 2.3.
- Finally, we use (4.9) to compute the final illumination invariant representation I_{inv} of an image.

Following this procedure, the computational cost of generating the illumination invariant representation of an image is extremely low. In real time terms, we test our method in Matlab 2015a on a 3.40GHz i7 processor with 16GB RAM. The illumination invariant representation for an image with the resolution of 640 × 480 pixels can be computed in 45 ms.

4.2.3 Experimental Results

In this section, we perform a set of experiments to demonstrate the capability of the proposed method to provide a robust image representation against severe illumination changes. In particular, we evaluate the accuracy of the proposed method and compare it with [28], which uses the initial model of [36]. For the experiments, we use UACampus dataset where the images are captured by a Clearpath Husky robot on University of Alberta campus. UACampus dataset has five sequences of images of the same route taken from morning to night with different illumination and weather conditions: Sunny, Cloudy,

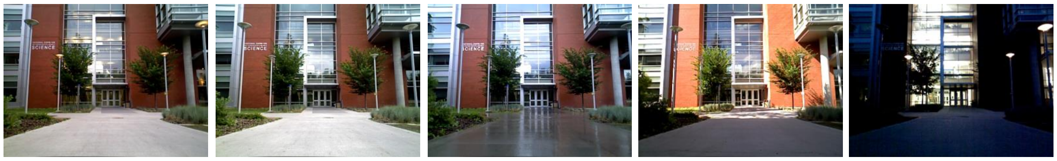


Figure 4.5: Sample images of the same place in UACampus in five different times of a day and different weather conditions. Images were captured at 6:00 am, 10:00 am, 2:20 pm, 4:40 pm and 10:15 pm from left to right.

Rainy, and Night. Fig. 4.5 shows one set of sample images from one location in the dataset.

(a) Qualitative Results

In the first set of experiments, we test our method on different image sequences captured at different times of a day and compare them with illumination invariant method presented in [28]. Figs. 4.6 and 4.7 show the results of illumination invariant images from [28] and our proposed method in the second and the third rows, respectively. Columns (a) and (b) show the results of invariant representation from [28] and our method in cloudy and sunny weather under heavy shadow in the UACampus dataset. Although the two methods produce qualitatively different outputs that are difficult to compare visually, in these two situations, the proposed method is able to remove the shadow while preserving more details of the images such as edges that could be very important in different applications (e.g., feature detectors in place recognition). The ability of our method in preserving the details is particularly obvious in Column (c) of Figs. 4.6 and 4.7 for the two images at night. This experiment is a challenging example where the source of illumination is changed from the sun to indoor lights. None of the previous methods can recover illumination invariant representations of images captured at night, but still the proposed method is feasible in this case and the results are roughly similar to the invariant representation at day hours. Although the proposed method provides satisfactory results at night, the recovered invariant image has not the same quality as images in (a) and (b) in Figs. 4.6 and 4.7. The reason is that the intensity of the original image for some regions of the night images is almost

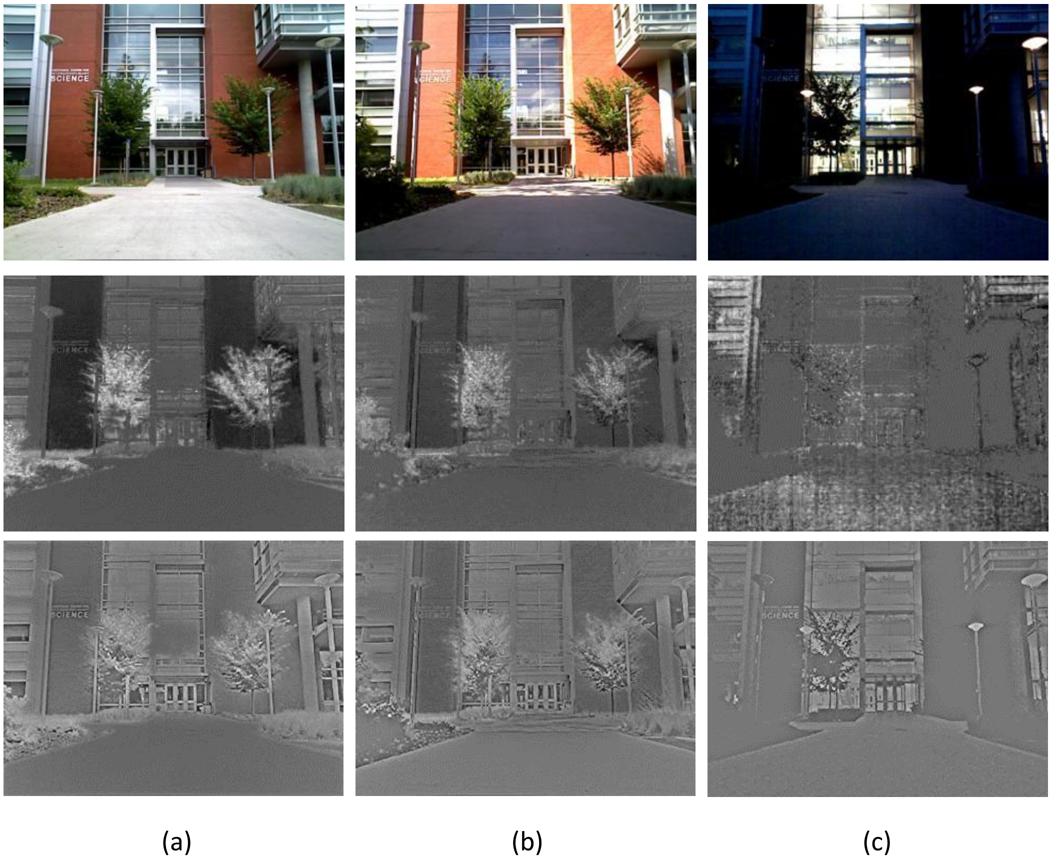


Figure 4.6: Illumination invariant representation of sample images under three different illumination conditions. (a) cloudy weather (b) sunny weather with heavy shadow and (c) dark image at night with different source of light. The second and the third rows are the results of method [28] and our illumination invariant representation, respectively.

[0;0;0] and obviously no method can produce results in such a situation.

(b) Similarity matrices

In this experiment, we use the UACampus dataset at five places. Recall that for each place, five images from different times of a day are available. One set of such images is shown in Fig. 4.5 where the last image was captured at night. We compute the zero-mean normalized cross correlation (ZNCC) [145] between images as the measure of their similarity, and the results are shown in Fig. 4.8 for different methods of constructing illumination invariant images. Fig. 4.8(a) shows the result of using original images where due to change in illumination at each place, similarity between the five images of the same place

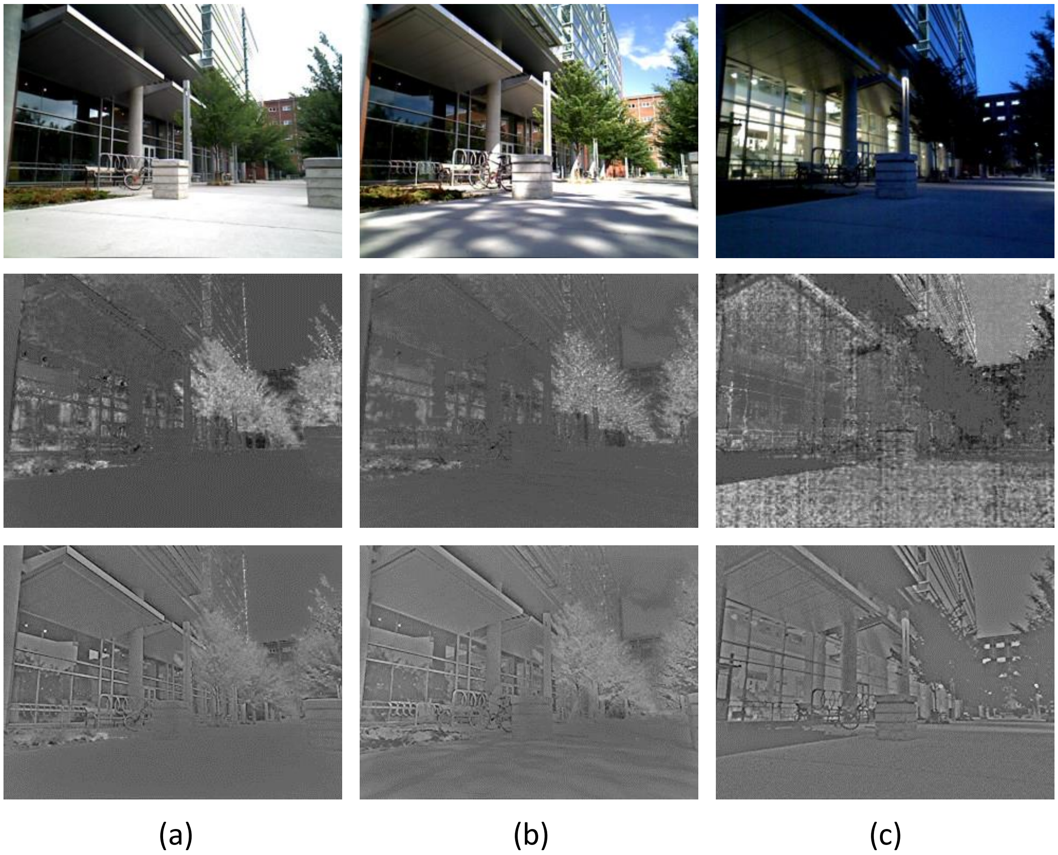


Figure 4.7: Illumination invariant representation of sample images under three different illumination conditions. (a) cloudy weather (b) sunny weather with heavy shadow and (c) dark image at night with different source of light. The second and the third rows are the results of method [28] and our illumination invariant representation, respectively.

(in a 5×5 diagonal block) can be quite low. Figs. 4.8(b), and (c) show the similarity matrices for the method in [28], and the proposed invariant image I_{inv}, respectively. By observing the diagonal blocks of these similarity matrices, it is clear that images from the same location in the proposed invariant space I_{inv} are more similar to each other and more dissimilar to images of other places compared to original images column(a) and the invariant method of [28].

4.2.4 Other Applications

In addition to the problem of change detection, we used the obtained robust image representation against illumination in more applications. Since the focus

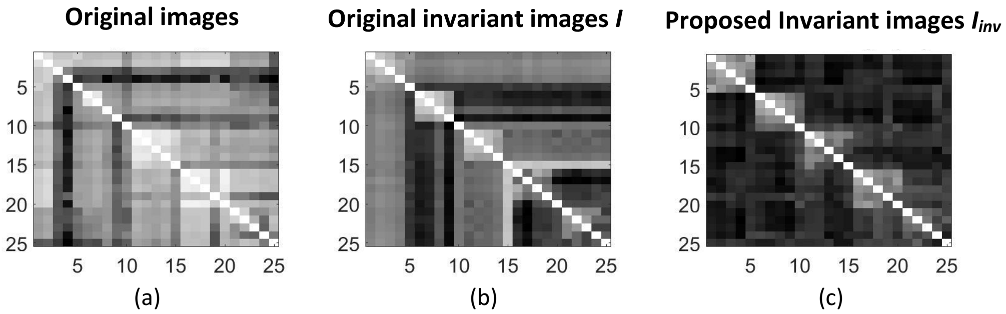


Figure 4.8: Similarity matrix for five places, each place has five images with different illumination from 6:00 am to 10:00 pm. This figure shows image similarity of (a) original images (b) invariant images I and (c) illumination invariant representation I_{inv} after adding invariant feature ν for all 25 images.

of this thesis is on the application of moving object detection using low-rank framework, we only give a brief summary of the other applications as follows.

(a) Visual place recognition: In [101] we proposed a method based on this approach for visual place recognition. Our experiments on different challenging images validate the superiority of our proposed method, for measuring similarity between images and for keypoint matching, in comparison with the existing competing methods in the place recognition application.

(b) Multi-modal face recognition: We also used the obtained invariant representation as the second representation of an image for the face recognition application [40]. We used the original and invariant representation of images into a joint dictionary learning and low-rank framework and showed that the representation of illumination invariant images combining with structured sparse low-rank representation empowers the method. Experimental results indicate that our method is robust, achieving state-of-the-art performance in the presence of illumination change.

In the rest of this chapter, we only focus on the problem of moving object detection and compute a prior map using the obtained invariant representation of images to incorporate in the LRA framework.

4.3 Low-Rank and Invariant Sparse Decomposition

As discussed in Chapter 2.2.2, all existing methods for moving object detection decompose the matrix of all observed images into the low-rank and the sparse matrices. Therefore, in the case of abrupt or discontinuous change in illumination, all the variations are grouped into the sparse matrix. In such cases, all of the existing methods fail to detect moving object. To address this problem, our proposed formulation seeks to decompose a data matrix D into a low-rank background matrix L, sparse illumination change matrix C, and sparse foreground matrix S as follows.

$$D = L + C + S \tag{4.10}$$

In (4.10), C and S are considered as outliers. Since both of them are stochastic especially in time-lapse video or low frame rate image sequences, separating them is an ill-posed problem. We address this challenge by using our proposed illumination invariant representation of an image, which serves as a prior for outliers in our formulation. This prior enables us to have a pattern for estimating C and S through the optimization as will be detailed in Section 4.3.1. Using the obtained prior map, we introduce our formulation to detect moving objects under significant illumination changes in Section 4.3.2 and in Section 4.3.3 we describe a solution to the formulation.

4.3.1 Initialization of the Prior Map

In this section we focus on obtaining the prior information, which enables us to distinguish between moving objects and illumination changes in our proposed formulation. In the case of time-lapse images, shadows and illumination changes are unstructured phenomena and most of the time they are mistakenly considered as moving objects. To distinguish between changes caused by illumination and those caused by moving objects, we construct a prior map using the illumination invariant representation, proposed in the previous section. In particular, we first address the second issue of the invariant representations as

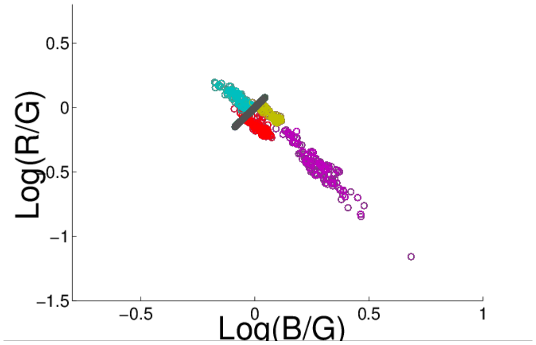


Figure 4.9: Log-chromaticity vectors of pixels from one material in different illumination condition.

discussed in Section 4.2, and then we initialize the prior map.

(a) Solution to the variations of robust representations against illumination

As discussed in Section 4.2, chromaticity vectors χ' of the same material under different illumination are not projected to the same location in the orthogonal vector and therefore, the method cannot remove the effect of illumination accurately. Fortunately, this issue would be problematic for an individual image to be accurate enough for the application of moving object detection. We realized that if we have an image sequence, corresponding pixels of the images in invariant representation are correlated to each other and therefore those pixels can be captured in a low-rank matrix.

Fig 4.9 shows the details of this concept. Four different locations but from one material are selected. Sample points with the same color show log-chromaticity of corresponding pixels from the selected locations in a sequence of images with different illumination. Assuming that the camera is fixed, the invariant direction between images is roughly similar. Black circles in Fig 4.9 show the projected pixels of the same material from all images to the average invariant directions of all images, where corresponding pixels of all images with different illumination are projected to one coordinate or are close to each other in invariant representation. In other words, the corresponding pixels of all images under different illumination are correlated. This means if we decompose the matrix of all vectorized invariant images, we can assume all illumination variations can be captured into the low-rank matrix.

(b) Generation of the prior map

To construct the prior map formally, let $D \in R^{m \times n}$ be an observed matrix (an image sequence in our problem), where each column of matrix D is a vectorized image from the sequence with m pixels, and n is the total number of images in the sequence. Then the following function convert all D_i images $i = 1, 2, .., n$ to the invariant representation D_{inv}.

$$D_{inv} = \Omega(D) \tag{4.11}$$

where $D_{inv} \in R^{m \times n}$ be a matrix of all vectorized invariant representations. We can decompose matrix D_{inv} into low-rank matrix L_{inv} and sparse matrix S_{inv} using optimization, so that all illumination variations are absorbed into the low-rank matrix L_{inv}.

$$\min_{L_{inv}, S_{inv}} \|L_{inv}\|_* + \lambda_{inv}\|S_{inv}\|_1 \qquad s.t.\, D_{inv} = L_{inv} + S_{inv} \tag{4.12}$$

To solve (4.12) we use inexact augmented Lagrangian multiplier (ALM) [77]. Optimization problem (4.12) can account for most of the illumination and shadow changes with the low-rank part and for the moving objects with the sparse part S_{inv}. Finally, we can use S_{inv} to build the prior map Φ as follows.

$$\Phi = \frac{1}{1 + e^{-\alpha(|S_{inv}| - \sigma)}} \tag{4.13}$$

where σ shows the standard deviation of corresponding pixels in D_{inv}, and α is a constant. We use the prior map Φ, to define two penalty terms in the LRA framework to extract the invariant sparse outliers as moving object, as will be explained in the next section.

4.3.2 LISD Formulation

To detect moving objects in time-lapse videos under severe illumination changes, standard low-rank method is insufficient because we need to separate illumination changes and moving shadows from real changes and both of them are sparse outliers. To do so, we define a constraint based on the prior knowledge from illumination invariant representation introduced in the previous section. In particular, real changes should be included in the subspace that is orthogonal to the illumination change subspace. Since outliers are completely independent in different frames of a low frame-rate image sequence, real changes in the ith frame should satisfy the following properties.

$$(\Phi_i^{\perp})^T|S_i| = 0, \quad \Phi_i^T|C_i| = 0 \tag{4.14}$$

where $\Phi_i^{\perp} = [1]_{m\times 1} - \Phi_i$, is the complement of Φ_i. Sparse S and C are the detected objects and illumination changes in the grayscale domain.

To formalize the prior knowledge from illumination invariant representation on the outliers, we combine the two properties of (4.14) into one function, and propose Low-rank and Invariant Sparse Decomposition (LISD) method, as follows.

$$\begin{aligned} \min_{L,S,C} \; & \|L\|_* + \lambda(\|S\|_1 + \|C\|_1) + \gamma\Psi(S, C, \Phi) \\ & s.t. \;\; D = L + S + C \end{aligned} \tag{4.15}$$

where $\|L\|_*$ is the nuclear norm, i.e. the sum of the singular values, and it approximates the rank of L. S and C are detected foreground and illumination changes, respectively. The last term in (4.15) is the geometric constraint function Φ as follows.

$$\Psi(S, C, \Phi) = \sum_i (\Phi_i^{\perp})^T|S_i| + \sum_i \Phi_i^T|C_i| \quad i = 1, ..., n \tag{4.16}$$

To make the problem more tractable, the geometric constraint Ψ can be relaxed to the penalty terms $\Sigma_i\|G_iC_i\|_F^2$, and $\Sigma_i\|G_i^{\perp}S_i\|_F^2$ so that (4.15) becomes

$$\begin{gathered}\min_{L,S,C} \|L\|_* + \lambda(\|S\|_1 + \|C\|_1) + \gamma \sum_i \left(\|G_i C_i\|_F^2 + \|G_i^{\perp} S_i\|_F^2 \right) \\ s.t.\ D = L + S + C\end{gathered} \tag{4.17}$$

where λ and γ are positive parameters and $G_i = diag[\sqrt{\Phi_{1i}}; \sqrt{\Phi_{2i}}; ...; \sqrt{\Phi_{mi}}]$. We optimize (4.17) by updating each of the variables L, S, and C in turn, iteratively until convergence. The error is computed as $\|D - L^k - S^k - C^k\|_F / \|D\|_F$. The loop stops when the error reaches the value lower than 10^{-5}.

4.3.3 Optimization Algorithm

In order to solve (4.17), we use inexact ALM method [77], and start by computing the augmented Lagrangian function $\mathcal{L}(L, S, C, Y; \mu)$, given by

$$\begin{aligned}\mathcal{L}(L, S, C, Y; \mu) = \|L\|_* + \lambda(\|S\|_1 + \|C\|_1) + \gamma \sum_i \big(\|G_i C_i\|_F^2 \\ + \|G_i^{\perp} S_i\|_F^2\big) + < Y, D - L - S - C > \\ + \frac{\mu}{2}\|D - L - S - C\|_F^2 \\ = \|L\|_* + \lambda(\|S\|_1 + \|C\|_1) - \frac{1}{2\mu}\|Y\|_F^2 \\ + h(L, S, C, Y, \mu)\end{aligned} \tag{4.18}$$

where $< A, B >= trace(A^T B)$, μ is a positive scalar, Y is a Lagrangian multiplier matrix, and $h(L, S, C, Y, \mu)$ is a quadratic function as follows.

$$h(L, S, C, Y, \mu) = \sum_i \left(\frac{\mu}{2}\|D_i - L_i - S_i - C_i + \frac{Y_i}{\mu}\|_F^2 + \gamma\|G_i C_i\|_F^2 + \gamma\|G_i^{\perp} S_i\|_F^2 \right) \tag{4.19}$$

We optimize (4.18) by updating each of the variables L, S, C, and Y in turn, iteratively until convergence, which solves the following four sub-problems:

$$\begin{cases} L^{k+1} = \arg\min_L \mathcal{L}(L^k, S^k, C^k, Y^k; \mu) \\ S^{k+1} = \arg\min_S \mathcal{L}(L^{k+1}, S^k, C^k, Y^k; \mu) \\ C^{k+1} = \arg\min_C \mathcal{L}(L^{k+1}, S^{k+1}, C^k, Y^k; \mu) \\ Y^{k+1} = Y^k + \mu(D - L^{k+1} - S^{k+1} - C^{k+1}) \end{cases} \tag{4.20}$$

Updating L^{k+1}: From (4.18), the augmented Lagrangian reduces to the following form:

$$L^{k+1} = \arg\min_{L} \|L\|_* + \frac{\mu}{2}\|L^k - (D - S^k - C^k + \frac{Y^k}{\mu})\|_F^2 \tag{4.21}$$

The subproblem (4.21) has the closed-form solution by applying the singular value thresholding algorithm [20], with the soft-thresholding shrinkage operator $\mathcal{S}_\epsilon(x)$, which is defined as

$$\mathcal{S}_\epsilon(x) = max(0, x - \epsilon) \quad x \geq 0 \quad \epsilon \geq 0 \tag{4.22}$$

Updating S^{k+1}: From (4.18), the augmented Lagrangian reduces to

$$\min_{S} \lambda\|S\|_1 + h(L, S, C, \mu) \tag{4.23}$$

Since $h(L, S, C, \mu)$ is a quadratic function, it is convenient to use the linearization technique of the LADMAP method [78] to update S^{k+1} by replacing the quadratic term h with its first order approximation, computed at iteration k and add a proximal term giving the following update.

$$\begin{aligned} S^{k+1} = \arg\min_{S} \lambda\|S\|_1 + \sum_i \Bigg(& \frac{\eta\mu}{2}\|S_i - S_i^k + [-\mu(D_i - L_i^{k+1} \\ & -S_i^k - C_i^k + \frac{Y_i^k}{\mu}) + 2\gamma(G_i^\perp)^T G_i^\perp S_i^k]/(\eta\mu)\|_F^2 \Bigg) \end{aligned} \tag{4.24}$$

Updating C^{k+1}: From (4.18), the augmented Lagrangian reduces to

$$\min_{S} \lambda\|C\|_1 + h(L, S, C, \mu) \tag{4.25}$$

Similar to (4.23) we use the LADMAP method to update C^{k+1} by giving the following update

$$\begin{aligned} C^{k+1} = \arg\min_{C} \lambda\|C\|_1 + \sum_i \Bigg(& \frac{\eta\mu}{2}\|C_i - C_i^k + [-\mu(D_i - L_i^{k+1} \\ & -S_i^{k+1} - C_i^k + \frac{Y^k}{\mu}) + 2\gamma G_i^T G_i C_i^k]/(\eta\mu)\|_F^2 \Bigg) \end{aligned} \tag{4.26}$$

The error is computed as $\|D - L^k - S^k - C^k\|_F / \|D\|_F$. The loop stops when the error reaches the value lower than 10^{-5}. All details about LISD are described in Algorithm 4.1.

Algorithm 4.1 Low-rank and Invariant Sparse Decomposition via Inexact ALM (LISD)

```
 Input: Observation matrix $D$, Parameters $\lambda$,$\gamma$,$\eta$,
 1: computing invariant representation $D_{inv}$ according to Section 4.2
 2: solve (4.12) via inexact ALM to obtain $S_{inv}$
 3: compute $\Phi$ according to (4.13)
 4: $[L, S, C] = LISD(D, \Phi, \lambda, \gamma, \mu, \eta)$

    //following lines compute $LISD$
 function $[L^k, S^k, C^k] = LISD(D, \Phi, \lambda, \gamma, \mu, \eta)$
 5: while not converged do
 6:     $(U, \Sigma, V) = svd(D - S^k - C^k + \mu^{-1}Y^k)$  //lines 2-7 solve (4.21)
 7:     $L^{k+1} = U\mathcal{S}_{(1/\mu)}(\Sigma)V^T$
 8:     Compute for all coulmns $i$                //lines 4-10 solve (4.24)
 9:         $tempS_i = S_i^k + [\mu(D_i - L_i^{k+1} - S_i^k - C_i^k + \mu^{-1}Y_i^k)$
                $-2\gamma(G_i^{\perp})^T G_i^{\perp} S_i^k]/(\eta\mu)]$
10:     $S^{k+1} = \mathcal{S}_{\lambda/(\eta\mu)}(tempS),\ tempS = [tempS_1, ..., tempS_n]$
11:     Compute for all coulmns $i$                //lines 7-8 solve (4.26)
12:         $tempC_i = C_i^k + [\mu(D_i - L_i^{k+1} - S_i^{k+1} - C_i^k + \mu^{-1}Y_i^k)$
                $-2\gamma G_i^T G_i C_i^k]/(\eta\mu)]$
13:     $C^{k+1} = \mathcal{S}_{\lambda/(\eta\mu)}(tempC),\ tempC = [tempC_1, ..., tempC_n]$
14:     $Y = Y + \mu(D - L^{k+1} - S^{k+1} - C^{k+1})$
15:     $\mu = \rho\mu$; $k = k + 1$
16: end while
 Output $L^k, S^k, C^k$
```

4.4 Iterative Version of LISD

In our proposed LISD method, we first compute a prior map from illumination invariant representation of images and then use the map to separate foreground from background and illumination changes in the grayscale representation of images. Although LISD provides satisfactory results in our experiments, still we can improve the performance by updating the prior map (4.13) iteratively.

We refer to this iterative version as Iterative LISD (ILISD). In ILISD, the first step is exactly similar to LISD where we compute the prior map in one representation and use it in another representation. Ideally, moving object in both representations should build similar prior maps. Using this assumption, we compute the second prior map Φ_{inv} from the result of LISD to use in illumination invariant representation. To do it, we rewrite (4.12) similar to (4.15) as follows.

$$\min_{L_{inv},S_{inv},C_{inv}} \|L_{inv}\|_* + \lambda_{inv}(\|S_{inv}\|_1 + \|C_{inv}\|_1) + \gamma\Psi(S_{inv}, C_{inv}, \Phi_{inv}) \\ s.t.\ \ D_{inv} = L_{inv} + S_{inv} + C_{inv} \tag{4.27}$$

where Ψ is defined the same as (4.15) but for illumination invariant representation. Φ_{inv} is computed similar to (4.13) using the obtained S from LISD method. Generally speaking in ILISD the obtained map from each representation is used into another representation in the next iteration until convergence. The convergence criterion is $\|S^{j+1} - S^j\|_F / \|S^j\|_F < 10^{-5}$. All details about ILISD are described in Algorithm 4.2.

Algorithm 4.2 Iterative Low-rank and Invariant Sparse Decomposition (ILISD)

```
Input: Observation matrix $D$, $\Phi_{inv} = [1]_{m \times n}$, Parameters $\lambda$,$\gamma$
Computing invariant representation matrix $D_{inv}$ according to Section 4.2
while not converged do
//Solve (4.27)
    $[L_{inv}^{j+1}, S_{inv}^{j+1}, C_{inv}^{j+1}] = LISD(D_{inv}, \Phi_{inv}, \lambda, \gamma, \mu_{inv})$
    compute $\Phi$ according to (4.13) using $S_{inv}^{j+1}$
//Solve (4.15)
    $[L^{j+1}, S^{j+1}, C^{j+1}] = LISD(D, \Phi, \lambda, \gamma, \mu)$
    compute $\Phi_{inv}$ according to (4.13) using $S^{j+1}$
    $j = j + 1$
end while
Output $L^j, S^j, C^j$
```

4.5 Time Complexity

In this section, we compute time complexity of the proposed method by analyzing the main three sub-problems of LISD as follows.

- For the first subproblem (4.21) to update L, the main complexity depends on SVD computations with the time complexity of $O[min(mn^2, nm^2)]$, where m and n are the dimensions of the data matrix.

- For the second and the third subproblems (4.24) and (4.26) to update S and C, we use LADMAP [78], which is an accelerated version of linearized alternating direction method (LADM) [143]. The time complexity for each of the subproblems is $O(rmn)$, where r is the rank of the data matrix. Since r is always less than m and n, $O(rmn)$ can be removed from the total time complexity of LISD.

Thus, the total complexity of our proposed method, ILISD, is $O(j \times min[mn^2, nm^2])$, where j is the number of iterations in Algorithm 4.2.

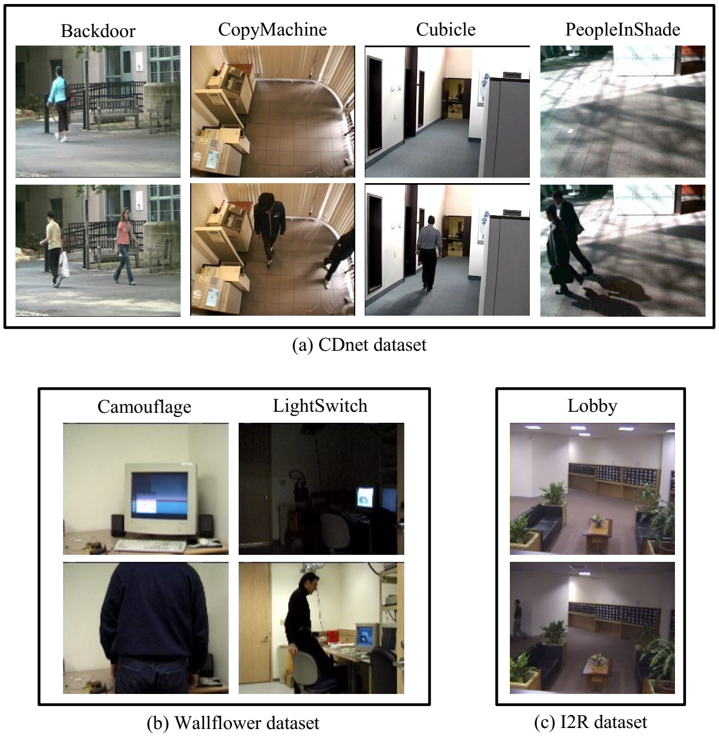
4.6 Experimental Results and Discussion

Our main application of interest is moving object detection in time-lapse videos with varying illumination. Therefore, we evaluate our method under two increasingly difficult conditions. First, we use datasets that contain moving objects and significant illumination changes or shadows but in real-time sequences with continuous object motion. Secondly, we use a challenging dataset that contains moving objects, illumination, and shadows, where images are captured via time-lapse or motion-trigger photography with large inter-image time intervals. In this case, the position of an object between two consecutive images may not be continuous. This is a common phenomenon in many long-term surveillance applications such as wildlife monitoring, as explained in Chapter 1. Since real benchmark datasets only contain the first condition, we have built a new dataset which contains the second condition and use it in this thesis. Then we perform two sets of experiments on benchmark and the newly proposed dataset.

4.6.1 Experimental Setup

Benchmark datasets: We evaluate our proposed method on selected sequences from the CDnet dataset [46], Wallflower dataset [124], and I2R [74] dataset. Since the goal of the experiments is to illustrate the ability of our method to detect real changes from illumination changes, we select sequences with varying illumination or moving shadows. From CDnet dataset four sequences are in this category depicting indoor and outdoor scenes exhibiting moderate illumination changes and moving shadows. These sequences are "Backdoor", "CopyMachine", "Cubicle", and "PeopleInShade". We also use sequences "Camouflage" and "LightSwitch" from Wallflower dataset and image sequence "Lobby" from I2R dataset, which include images with global and sudden illumination changes. Fig. 4.10 shows sample images of the mentioned sequences.

Illumination change dataset (ICD): In this section we introduce the dataset



(a) CDnet dataset

(b) Wallflower dataset

(c) I2R dataset

Figure 4.10: sample images from the selected sequences of the benchmark datasets.

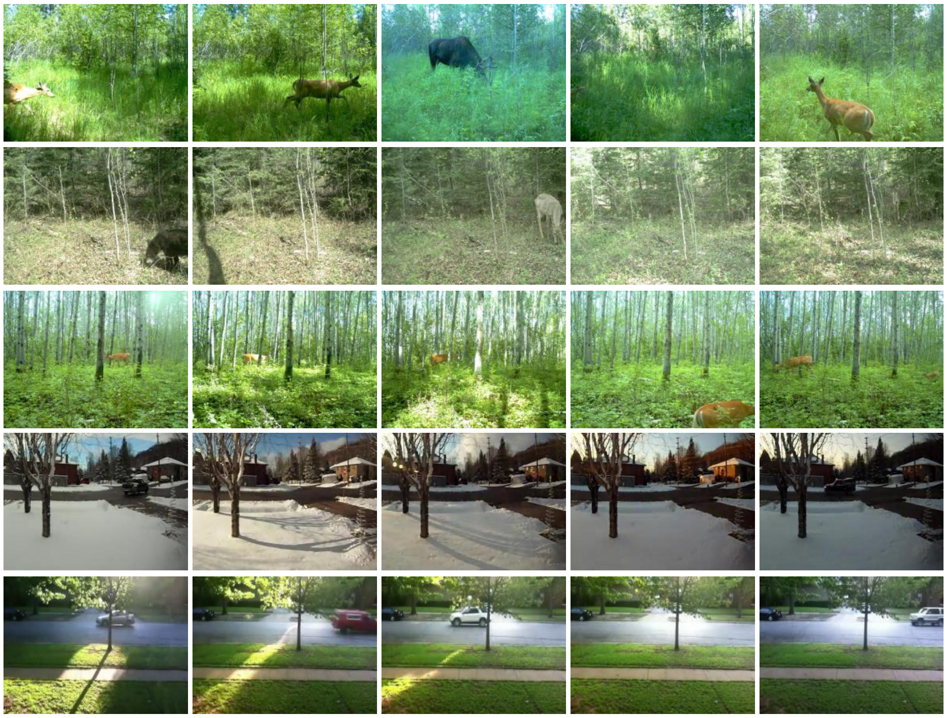


Figure 4.11: Selected images from each sequence of ICD.

that we have built, which includes five image sequences with severe illumination changes. Selected images from these sequences are shown in Fig. 4.11, and the number of images and the image size of each sequence are described in Table 4.1. Some of these images are without any object and just illumination change. The sequences of ICD are divided into two groups. The first three sequences are captured using a motion triggered camera for the wildlife monitoring application on different days. The last two sequences are taken with time-lapse photography of a large time interval to record changes of a scene that take place over time, which is common for many surveillance applications. Moving objects in the first sequence are under extreme sunlight or heavy shadow. Color of the objects in the second sequence are similar to the background or shadow and since illumination is changing, separating them is a difficult task. The third sequence shows objects with different size under varying illumination. The fourth sequence shows global illumination changes with moving shadows and the last row shows the sequence of images with moving objects while a strong moving sunbeam changes illumination of the scene.

Table 4.1: Details of all sequences of ICD.

Sequences	Size × No. of frames
Wildlife1	[508,358] × 194
Wildlife2	[508,358] × 225
Wildlife3	[508,358] × 136
WinterStreet	[460,240] × 75
MovingSunlight	[640,360] × 237

Evaluation metrics: For quantitative evaluation, we use pixel-level *precision*, *recall*, and *F-measure* as defined in Chapter 3.5.

4.6.2 Evaluation on Benchmark Datasets

In the first set of experiments we use the sequences from benchmark datasets corresponding to Section 4.6.1 to evaluate the proposed method. We compare LISD as an intermediate results of our method and ILISD with the six related RPCA algorithms, namely SemiSoft GoDec (SSGoDec) [152], PRMF [131], PCP [21], Markov BRMF [132], DECOLOR [155], and LSD [82].

Table 4.2 shows performance of LISD and ILISD in comparison with the competing methods in terms of *F-measure*. The proposed method obtains the best average *F-measure* against all the other methods, and for the all sequences our method ranked among the top two of all methods. The first four rows of Table 4.2 are from CDnet dataset and our method has superior performance. The last three rows of Table 4.2 are from "Wallflower" and "I2R" datasets. For the "Camouflage" sequence a large object comes to the scene and therefore the global illumination is changed. In this case, only DECOLOR, LSD and our method detect the foreground object relatively well. LSD uses a structured sparsity term by selecting a maximum value of outliers in a specific neighborhood for pixels in each iteration. So it can keep the connectivity of outliers and classifies the foreground better than our method. Although we can use the structured sparsity term in our formulation instead of l_1-norm, the solution becomes significantly slow. For two sequences "Camouflage" and "LightSwitch" only one frame has ground-truth and the results are based on

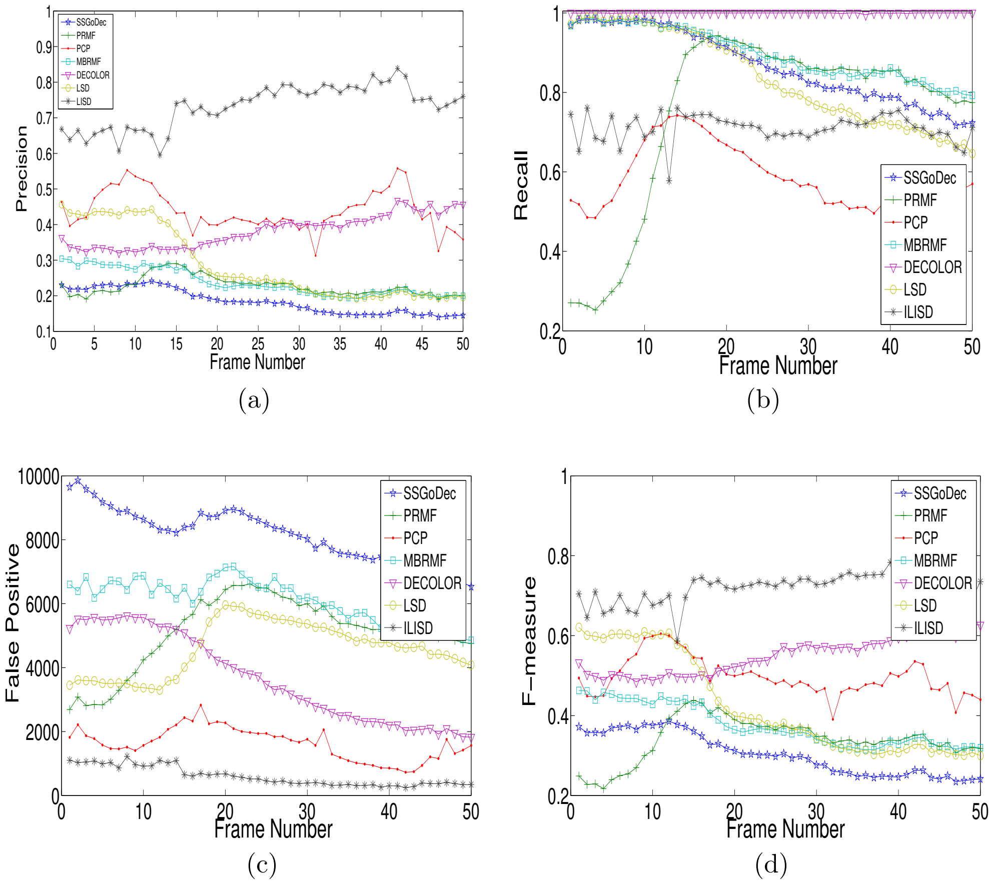
Table 4.2: Comparison of F-measure score between our proposed method and other compared methods on benchmark sequences (best F-measure: bold, second best F-measure: underline).

Sequence	SSGoDec	PRMF	Decolor	PCP	BRMF	LSD	LISD	ILISD
Backdoor	0.6611	0.7251	0.7656	0.7594	0.6291	0.7603	0.8015	**0.8150**
CopyMachine	0.5401	0.6834	0.7511	0.6798	0.3293	0.8174	0.7832	**0.8179**
Cubicle	0.3035	0.3397	0.5503	0.4978	0.3746	0.4232	**0.7201**	0.6887
PeopleInShade	0.2258	0.5163	0.5559	0.6583	0.3313	0.6168	0.6733	**0.8010**
Camouflage	0.6452	0.6048	0.8125	0.3388	0.6048	**0.9456**	0.8605	0.8663
LightSwitch	0.3804	0.2922	0.5782	**0.8375**	0.2872	0.6640	0.6904	0.7128
Lobby	0.0831	0.6256	**0.7983**	0.6240	0.3161	0.7313	0.7830	0.7849

just one frame and cannot be reliable for the whole sequence; however, our method still is in the second place. For the “lobby” sequence ground-truth is available for some selected frames, but none of them show the ground-truth while illumination is changing. In this sequence the accuracy of our method is still in the second place and the accuracy of DECOLOR is a little better than ours.

4.6.3 Evaluation of Separate Performance Metrics

We investigate the reliability of ILISD on “Cubicle” sequence from CDnet dataset. We choose “Cubicle” as a sample because size of the object is changing through the sequence and at the same time, illumination and shadows are also changing. In this sequence the size of the object is decreased as the frame number increases. Fig. 4.12 shows *precision*, *recall*, *false-positive*, and *F-measure* for 50 consecutive frames of the sequence “Cubicle” which contain a moving object (i.e. person). Fig. 4.12(a) shows the precision of the proposed method against all other methods, and as shown ILISD has highest precision and the difference with other methods is more significant when the size of the object becomes smaller because the effect of illumination changes and shadow increases. Fig. 4.12(b) illustrates that the recall of SSGoDec, MBRMF,



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

Figure 4.12: Precision, Recall, False positive and F-measure comparisons of sequence "Cubicle" from CDnet dataset.

DECOLOR, and LSD are higher than our method when the size of the object is large. The reason is that DECOLOR, MBRMF, and LSD use MRF and structural information. This helps the large object to be connected component and therefore recall increases. Although these methods provide higher recall than our method, they also produce lots of false positives in comparison with our method which is shown in Fig. 4.12(c). We also show the $F - measure$ of the results of methods on all 50 frames, that shows the reliability of the proposed method.

4.6.4 Effect of Sudden Illumination Changes

As explained in Section 4.6.2, ground-truth of the "lobby" sequence is available for some selected frames, but the ground-truth is not provided while

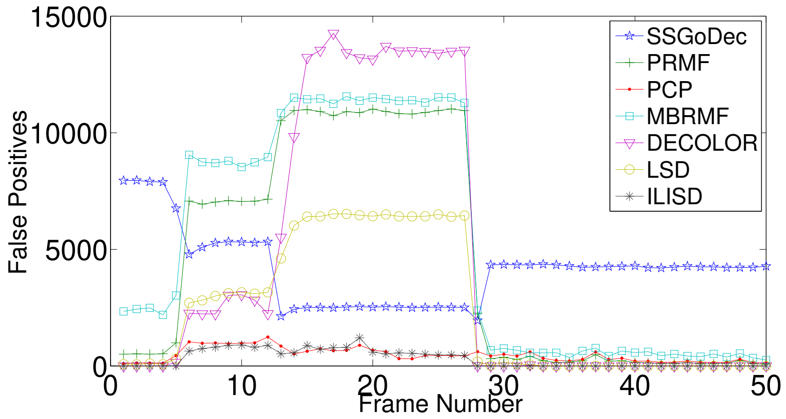


Figure 4.13: Comparison of number of false positives between the proposed method and the competing methods for sequence “Lobby” from I2R dataset.

illumination is changing. Based on Table 4.2, the accuracy of our method on this sequence is in the second place after DECOLOR. Here, we consider one more evaluation to show the reliability of our method against DECOLOR and all other competing methods on the “Lobby” sequence as a sample sequence with sudden illumination change. we select 50 consecutive frames while illumination is changing. These frames have no objects and any foreground pixel is considered as false positive. Fig. 4.13 compares the false positives of ILISD with all other methods. Fig. 4.13 clearly shows that ILISD and PCP produce much smaller number of false positives while illumination is changing. Therefore, although DECOLOR shows a little better accuracy on some specific ground-truth frames, unlike our method, it is not reliable while illumination is changing.

4.6.5 Evaluation on ICD

In the second set of experiments we evaluate our proposed method on the sequences from ICD which has the most challenging condition and compare them with competing methods. Fig. 4.14 shows the results of our method to detect objects and separating them from illumination changes. In the first row of each subfigure in Fig. 4.14, two samples per sequence are shown where real changes and illumination changes occur at the same time. The second and the third rows show the sparse outliers of C and S, respectively. Based on our experiments most of illumination changes can be classified as the outliers C and the real changes are separated into matrix S.

To show the capability of the proposed method, we compare qualitatively and quantitatively the results of our method with the results of the competing six sparse decomposition methods. We show the comparison of qualitative results on selected images of sequences “Wildlife1”, “Wildlife2”, and “Wildlife3”.

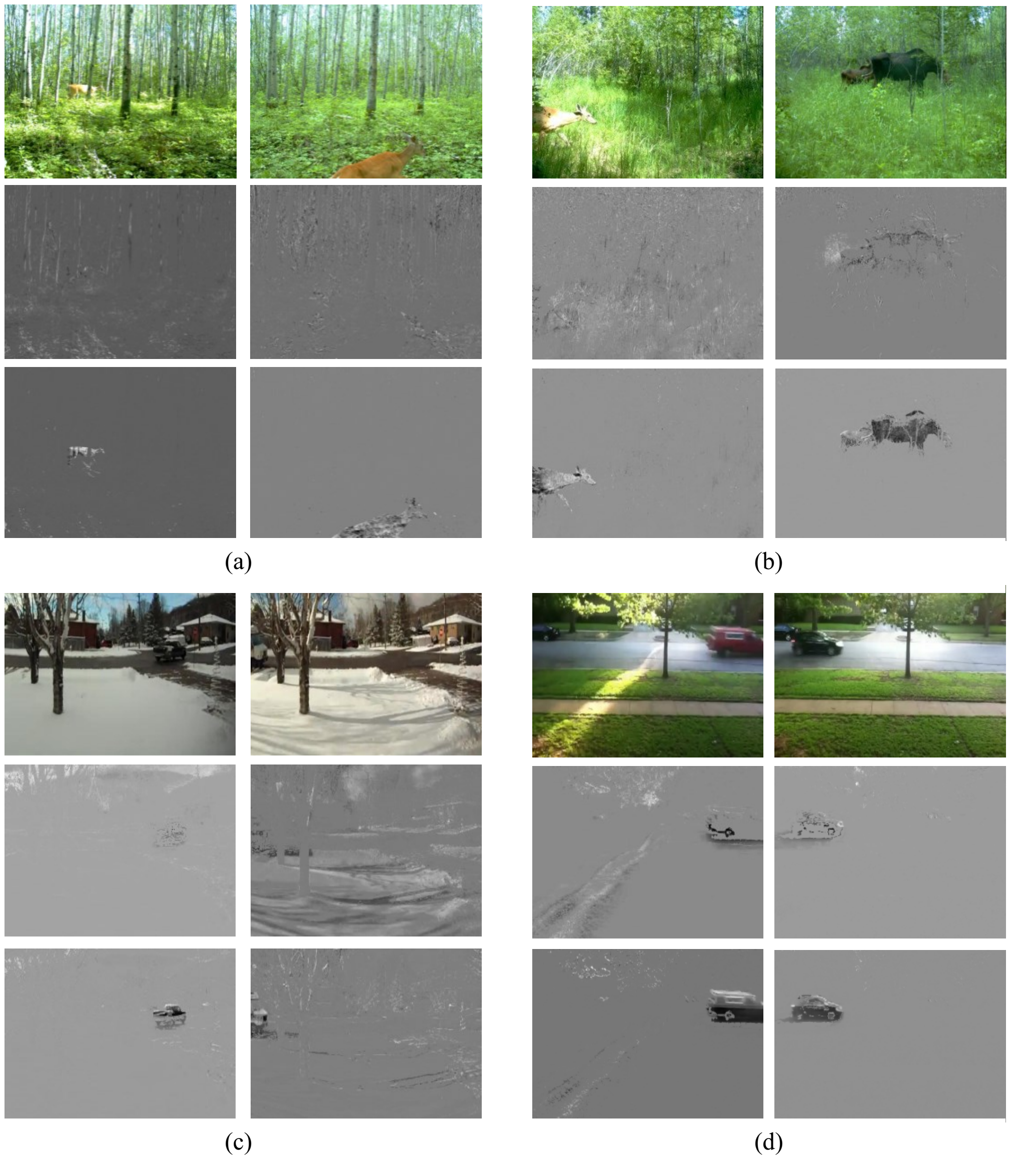


Figure 4.14: First row: two sample images with different illumination for each sequence. Second row: sparse outliers C. Third row: detected objects S.

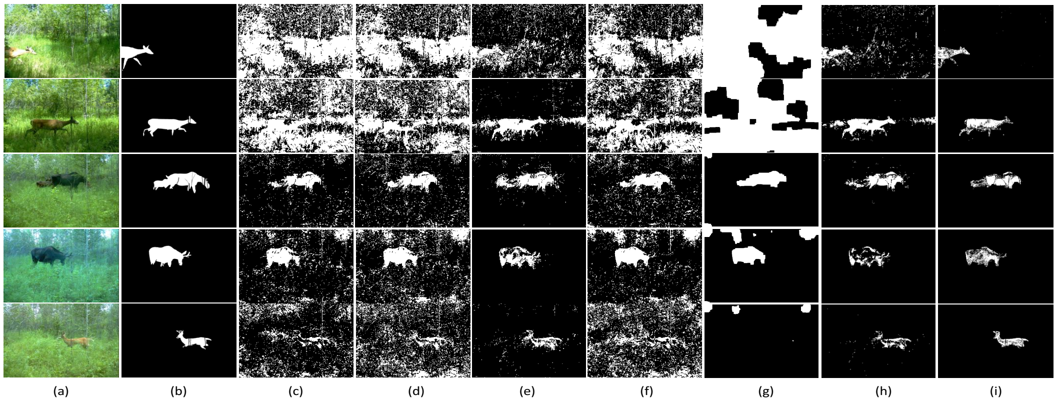


Figure 4.15: Comparison of qualitative results between our method and six competing methods. a) input, b) ground truth, c) SSGoDec, d) PRMF, e) PCP, f) MBRMF, g) DECOLOR, h) LSD, i) ILISD

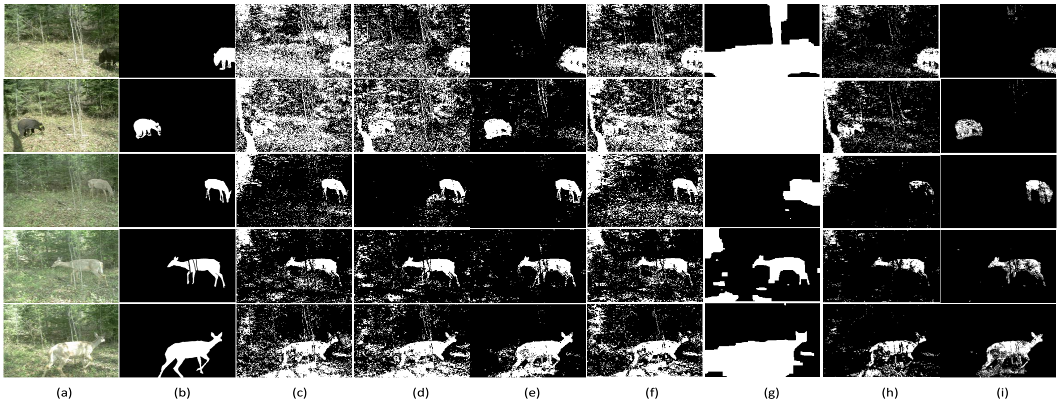


Figure 4.16: Comparison of qualitative results between our method and six competing methods. a) input, b) ground truth, c) SSGoDec, d) PRMF, e) PCP, f) MBRMF, g) DECOLOR, h) LSD, i) ILISD

Since the illumination variations in the time-lapse image sequences are significant, we show five images of the sequences in Figs. 4.15, 4.16, and 4.17 to provide a better comparison.

The first two rows of Fig. 4.15 depict heavy illumination changes, and only the results of LSD and PCP are comparable with ILISD; however, those methods still have many false positives. The third row shows the results of all methods when the illumination is relatively unchanged and the results of all methods are comparable with ILISD. In the last two rows of Fig. 4.15, although LSD has not many false positive, its recall is too low, and only PCP is comparable with our method. Fig. 4.16 demonstrates that in sequence "Wildlife2"

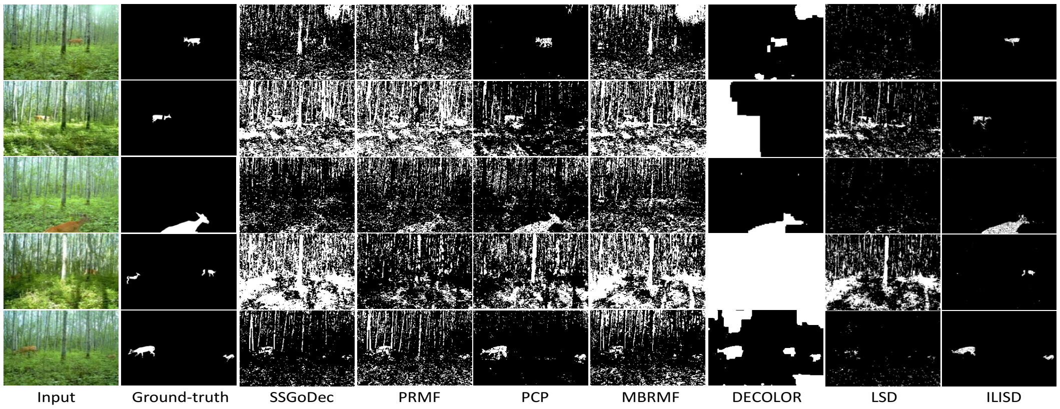


Figure 4.17: Comparison of qualitative results between our method and six competing methods on selected images of sequence "Wildlife3".

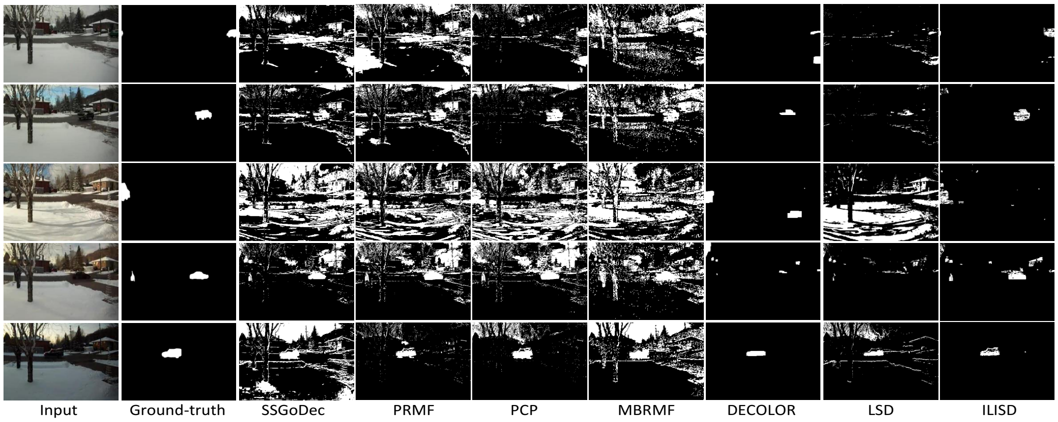


Figure 4.18: Comparison of qualitative results between our method and six competing methods on selected images of sequence "WinterStreet".

only the results of PCP is comparable with our method. In Fig. 4.17, the second and the fourth rows show heavy illumination changes, and all competing methods fail to detect objects. In the first, the third, and the last rows, where the illumination is relatively unchanged, PCP and DECOLOR can show relatively meaningful results. However, both of them are not reliable over the entire sequence.

We also compare the results of ILISD with the results of other competing methods on two more sequences "WinterStreet" and "MovingSunlight". For the first one, global illumination changes and for the second one, sunbeam is moving. The comparison results of these two sequences are shown in Figs. 4.18 and 4.19, respectively. Fig. 4.18 shows that only DECOLOR can be compara-

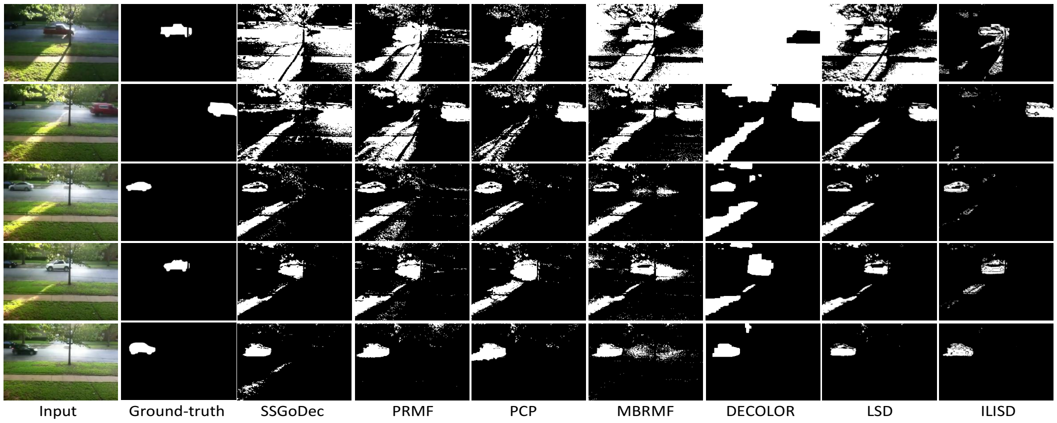


Figure 4.19: Comparison of qualitative results between our method and competing methods on selected images of sequence “MovingSunlight”

ble with our method for sequence “WinterStreet” and all other methods fail. As Fig. 4.19 shows, although all competing methods can detect the foreground in the last row the same as our method, all of them make many false positives and even cannot show meaningful results for the first four rows.

For quantitative evaluation on all sequences of ICD, Table 4.3 shows the *F-measure* of the competing methods. For each sequence we also compute standard deviations of all results that can show the reliability of each method for clear conclusions on the performance of the proposed method. The numerical results demonstrate that our method can provide better performance in handling such illumination changes than other competing methods.

4.7 Summary

In this chapter, we proposed a novel method named LISD to detect moving objects under discontinuous change in illumination such as time-lapse video sequences, using the framework of low-rank and sparse decomposition. In our proposed method, first a prior map is built based on an illumination invariant representation and then the obtained prior map is used in the proposed low-rank and invariant sparse decomposition framework to extract foreground under severe illumination changes. We also proposed an iterative version of LISD by updating the prior map in one representation and impose it as a

Table 4.3: Comparison of F-measure score between our method and other methods on ICD sequences (best F-measure: bold, second best F-measure: underline).

Sequence	Wildlife1	Wildlife2	Wildlife3	WinterStreet	MovingSunlight
SSGoDec	0.2826 ± 0.2113	0.2585 ± 0.1369	0.0753 ± 0.0722	0.1120 ± 0.0752	0.2926 ± 0.1927
PRMF	0.2586 ± 0.2000	0.4141 ± 0.2324	0.0754 ± 0.0745	0.1677 ± 0.1433	0.2925 ± 0.1732
PCP	0.5968 ± 0.2042	0.6430 ± 0.0996	0.3124 ± 0.2656	0.1766 ± 0.1021	0.3451 ± 0.1387
MBRMF	0.2679 ± 0.2117	0.2654 ± 0.1410	0.0510 ± 0.0441	0.0871 ± 0.0440	0.2426 ± 0.1403
DECOLOR	0.3409 ± 0.2834	0.3517 ± 0.2200	0.1019 ± 0.0929	0.4575 ± 0.2509	0.3466 ± 0.2590
LSD	0.6480 ± 0.1302	0.3899 ± 0.1659	0.0871 ± 0.0825	0.1604 ± 0.1086	0.3593 ± 0.2426
LISD	0.7747 ± 0.0557	0.7168 ± 0.0625	0.7318 ± 0.1269	0.6869 ± 0.0824	0.6163 ± 0.1393
ILISD	**0.8033** ± 0.0416	**0.7277** ± 0.0298	**0.7398** ± 0.1234	**0.6931** ± 0.0928	**0.6475** ± 0.1601

constraint into the LISD formulation with another representation. Based on our extensive experiments on real data sequences from public datasets, we are able to establish that LISD and ILISD achieve the best performance in comparison with all evaluated methods including the state-of-the-art methods. We also constructed novel datasets involving time-lapse sequences with significant illumination changes, which are publicly available in [59].

Despite its satisfactory performance in all our experiments, a challenge facing our proposed method is dynamic background. The reason is our proposed method uses l_1-norm for outliers without any structural constraint. In the future work chapter, we will explain an extended version of LISD that can work with dynamic background.

Chapter 5

Tensor Low-Rank and Invariant Sparse Decomposition

5.1 Introduction

In this chapter, we propose a solution to the problem of moving object detection within the tensor low-rank framework that specifically enables us to distinguish between illumination changes (including those due shadow), and changes caused by moving objects in the scene. In chapter 4 we offered a solution called LISD to effectively separate discontinuous changes due to moving objects and those due to illumination [103]. This method relies on an illumination regularization term combined with the low-rank framework to explicitly separate the sparse outliers into sparse foreground objects and illumination changes. Although this regularization term can significantly improve the performance of object detection under significant illumination changes, LISD relies on two restrictive assumptions. LISD assumes (a) the invariant representation of all images in a sequence are modeled by only one invariant direction and (b) all illumination variations are removed in the invariant representation of images, which are still vulnerable to complex illumination changes that arise in certain practical situations, and can lead to sub-optimal performance.

To address these issues, we formulate the problem in a unified framework named Tensor Low-rank and Invariant Sparse Decomposition (TLISD) [102]. Particularly, we first compute multiple prior maps as illumination invariant representations of each image to build a tensor data structure. These prior maps provide us with information about the effect of illumination in different parts of an image. Then we define two specific regularization terms using these prior maps to distinguish between moving objects and illumination changes. Finally, we introduce our TLISD formulation for moving object detection under the framework of low-rank tensor representation, which is able to decompose an image into background model, illumination changes and foreground objects. We demonstrate that the two regularization terms within our proposed method significantly improves the performance of moving object detection in the case of discontinuous changes in illumination, a problem that none of the existing methods in the literature can handle effectively.

5.2 Tensor Low-Rank and Invariant Sparse Decomposition

Recently, multi-way or tensor data analysis has attracted much attention and has been successfully used in many applications. Formally and without loss of generality, denote a 3-way tensor by $\mathcal{D} \in R^{n_1 \times n_2 \times n_3}$. Our proposed formulation seeks to decompose tensor data $\mathcal{D}$ into a low-rank tensor $\mathcal{L}$, an illumination change tensor $\mathcal{C}$, and a sparse foreground tensor $\mathcal{S}$ as follows.

$$\mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{C} \tag{5.1}$$

In (5.1), both $\mathcal{S}$ and $\mathcal{C}$ are stochastic in time-lapse image sequences due to discontinuous change in object locations and illumination changes, and separating them is an ill-posed problem. To solve this issue, we compute a set of prior maps using multiple representations of an image, which are more robust against illumination change than RGB images. These prior maps enable us to find higher order relations between the different invariant representations and the intensity images, in both space and time. These relations are exploited as the basis for separating $\mathcal{S}$ from $\mathcal{C}$ as will be detailed in Section 5.2.1. It is worth mentioning that on one hand, illumination changes are related to the material in a scene, which is invariant in all frames leading to a correlation between them. On the other hand, these changes are also related to the source of lighting, which is not necessarily correlated between frames. Consequently, illumination changes should be accounted for by both the low-rank part and the sparse part in an image decomposition. In our method, we model the highly correlated part of illumination with the low-rank tensor $\mathcal{L}$ as background, and we model the independent changes in illumination as the foreground, while recognizing that uncorrelated illumination changes are not necessarily sparse. To accomplish such illumination modeling, we propose to use a balanced norm or $k-$support norm [1], [72]. We introduce our formulation in details in Section 5.2.2, and we describe a solution to the formulation in Section 5.2.3.

5.2.1 Generation of Prior Maps and Tensor Data $\mathcal{D}$

In this section we focus on obtaining the prior information that will enable us to distinguish between moving objects and illumination changes in our proposed formulation. As explained in Chapter 4, in the case of discontinuous change in illumination, which is common in time-lapse image sequences, variation of shadows and illumination are unstructured phenomena and they are often mistakenly considered by all methods as moving objects. We addressed this problem through creating an illumination-invariant prior, which is described extensively in chapter 4.2.

In a nutshell, [36] computes the two-vector log-chromaticity χ' using red, green and blue channels and showed that with changing illumination, χ' moves along a straight line e roughly. Projecting the vector χ' onto the vector orthogonal to e, which is called invariant direction, an invariant representation $I = \chi' e^{\perp}$ can be computed. This method works well when the assumptions defined above hold true but in practice these assumptions never hold exactly, i.e., χ' does not move along a straight line. As a result, the correspond invariant representation is flawed.

In the previous chapter, we assumed that the invariant directions of the images are roughly similar and used the average of all the directions in creating the invariant representation. Then, we showed that in the case of an image sequence, corresponding pixels of the images in their invariant representations are correlated to each other and therefore this correlation among the corresponding pixels can be captured in a low-rank matrix. For the first time, this approach could solve the problem of moving object detection under discontinuous change in illumination. However, the assumption about the similarity between invariant directions is not always accurate, and can lead to sub-optimal performance as demonstrated in Fig. 5.1.

Fig. 5.1 shows an example of the variability of the illumination invariant direction in an image sequence and its impact on generating a illumination-invariant image representation. Fig. 5.1(a) shows the invariant directions of an image sequence of 200 frames while illumination changes, one direction for each image, varying mostly between -4^o and 13^o. Fig. 5.1(b) shows a selected image from the sequence, which is image 11 and corresponds to the red line in Fig. 5.1(a). The invariant direction for this image is found to be 13° while the average invariant direction of the sequence is around 5°, the direction we used in the previous chapter to create invariant representations for all images in the sequence. Fig. 5.1(c) compares the two invariant representations created with invariant directions of 5° and 13°, respectively, and Fig. 5.1(d) shows the detected foreground objects using these two different representations from the RPCA method where the use of the optimal invariant direction (13^o) produces much more desirable result than that of the sub-optimal direction (5^o). This example clearly shows the importance of the choice of the invariant direction in creating the invariant representations, and the undesirable outcome when these representations are created with a sub-optimal invariant direction.

Our idea to account for the difference in the invariant direction among the images in the sequence, is to first estimate the image-specific invariant directions for the sequence, and then use a clustering algorithm to identify the dominant directions. Subsequently, for each image, we create multiple invariant representations, one for each dominant direction, and these multiple representations serve as multiple prior maps for the image. Details of the above explanation can be detailed in the following steps.

- First, for each image in an image sequence, we use the method in [36] to determine its best invariant direction. With n_2 images in an image

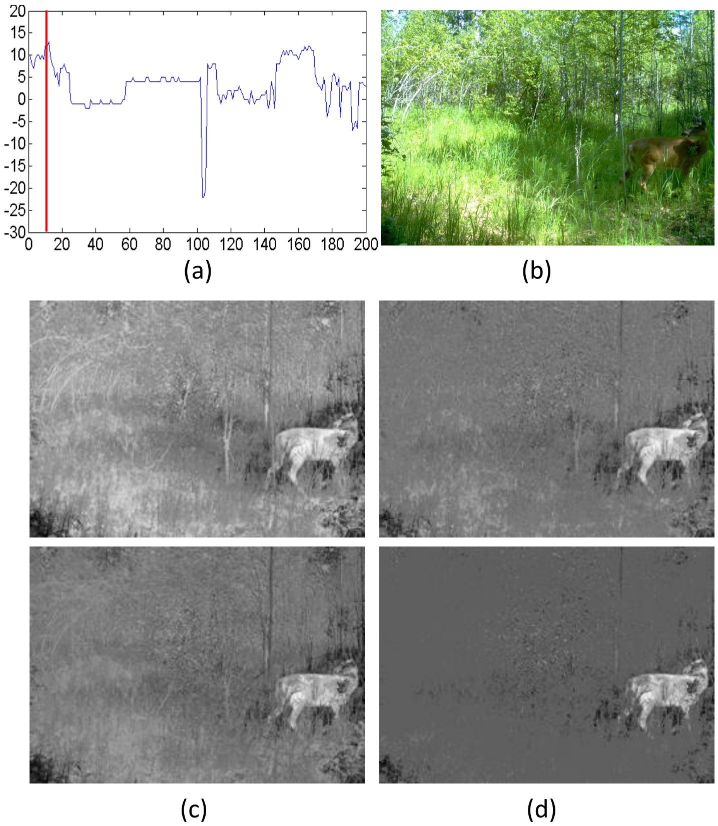


Figure 5.1: (a): Best invariant direction of each image in a sequence obtained by minimizing Shannon's entropy (with x-axis being the image index and the y-axis the angle of the invariant directions $e^{\perp}$ in degrees). (b): 11^{th} image in the sequence as shown with a red line in (a), where its best invariant direction is 13°. (c): The first and the second rows show the invariant representations of the selected image using the average direction of the sequence (5°) and its best direction (13°), respectively. (d): Obtained outliers of the invariant representations.

sequence, this results in n_2 invariant directions where $n_2 = 200$ in the example in Fig. 5.1(a).

- Second, we use k-means [2] to identify $k = 10$ clusters of the n_2 invariant directions.

- Third, we choose the centroid of a cluster as a dominant invariant direction if the cluster has support by at least 10% of the images or contains at least 20 images in the example of Fig. 5.1(a). By definition, there are no more than 10 dominant invariant directions.

Now, using the obtained dominant invariant directions, we propose a specific tensor structure $\mathcal{D}$, different from all existing tensor in the literature. To construct the tensor $\mathcal{D} \in R^{n_1 \times n_2 \times n_3}$ formally, let $\mathcal{D}(:,:,1)$ be an observed image sequence in our problem, where each column of $\mathcal{D}(:,:,1)$ is a vectorized image from the sequence with n_1 pixels, and n_2 is the number of images in the sequence. p^{th} frontal slice $\mathcal{D}(:,:,p), p = 2, ..., n_3$ is a corresponding prior map, generated with a dominant invariant direction. The constructed tensor $\mathcal{D}$ is shown in Fig. 5.2. Based on this tensor data structure, we are ready to present our new tensor low-rank and invariant sparse decomposition (TLISD) to extract the invariant sparse outliers as moving objects, as will be explained in the next section.

5.2.2 TLISD Formulation

To separate real changes due to moving objects from those due to illumination, we use multiple prior illumination-invariant maps, introduced in Section 5.2.1, as constraints on real changes and illumination changes. In particular, real changes should appear in all frontal slices. Furthermore, lateral slices are completely independent from each other in a time-lapse sequence, but the different representations in each lateral slice (see Fig. 5.2) are from one image and therefore, the locations of real changes should be exactly the same in each lateral slice. Now, based on these observations, real changes in each frame should satisfy the group sparsity constraint, which is modeled with the

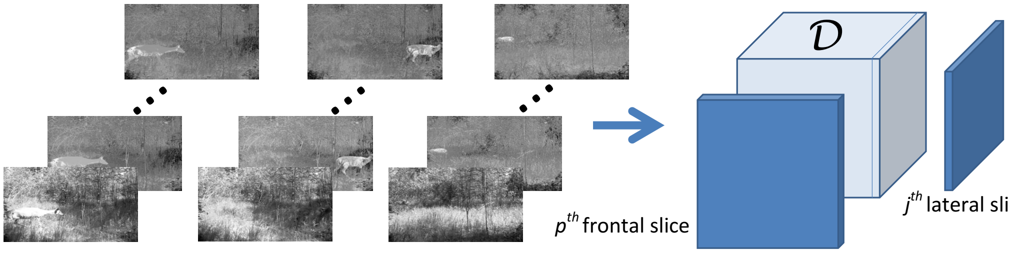


Figure 5.2: Left: sample images with their corresponding illumination invariant representations as prior maps. Right: Tensor $\mathcal{D}$. Frontal slices show p^{th} representation of the images in the sequence. Lateral slices show different representation of each image in the sequence.

minimization of the $l_{1,1,2}-$norm defined as:

$$\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\|\mathcal{S}_{i,j,:}\|_2 \tag{5.2}$$

As discussed, illumination changes in an image sequence should be accounted for by both the low-rank part and the sparse part. The highly correlated part of illumination can be modeled with the low-rank tensor $\mathcal{L}$ as background, but the independent changes in illumination are grouped as the foreground. To capture these uncorrelated illumination and shadow changes, and separate them from real changes, we recognize that they are not necessarily sparse.

Fig. 5.3 shows two sample images in a time-lapse image sequence with discontinuous change in illumination. Based on Fig. 5.3, it is easy to understand that illumination changes are on entire image and so, those uncorrelated changes are not completely sparse. In other words, discontinuous illumination changes between frames may affect all or a large part of an image, especially in time-lapse image sequences. Fig. 5.3 shows such a situation where changes occur on a large area of an image. As a result, those changes are not always completely sparse. These properties can be conveniently modeled with the $k-$support norm [1], [72], which is a balanced norm and defined as:

$$\|\mathcal{C}_{:,:,p}\|_k^{sp}=\Big(\sum_{m=1}^{k-r-1}(|c|_m^{\downarrow})^2+\frac{1}{r+1}(\sum_{m=k-r}^{d}|c|_m^{\downarrow})^2\Big)^{\frac{1}{2}} \tag{5.3}$$

where $\mathcal{C}_{:,:,p}$ and $|c|_m^{\downarrow}$ denote the p^{th} frontal slice of C and the m^{th} largest element in $|c|$, respectively. $c = vec(\mathcal{C}_{:,:,p})$ represents the vector constructed by concatenating the columns of $\mathcal{C}_{:,:,p}$ and $d = n1 \times n2$ is the dimension of the

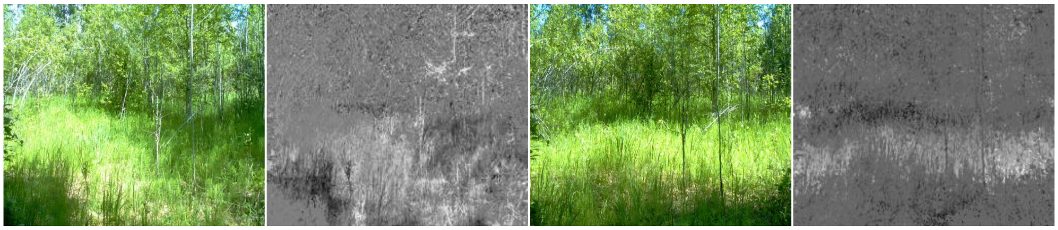


Figure 5.3: Two sample images and their corresponding illumination changes captured by our proposed method

frontal slice. Parameter $r \in \{0, 1, ..., k-1\}$ is an integer that is computed automatically by the method presented in [72]. The k-support norm has two terms: l_2-norm penalty for the large component, and l_1-norm penalty for the small components. k is a parameter of the cardinality to achieve a balance between the l_2-norm and the l_1-norm ($k = n_1$ in our experiments). The k-support norm provides an appropriate trade-off between model sparsity and algorithmic stability [139], and yields more stable solutions than the l_1-norm [72]. In this chapter we show that the k-support norm can estimate the illumination changes in an image sequence accurately. Joining of this norm and (5.2) as two constraints in one optimization framework enables us to separate real changes from illumination changes.

To summarize, we propose the Tensor Low-rank and Invariant Sparse Decomposition (TLISD) method, as follows.

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{S},\mathcal{C}} \|\mathcal{L}\|_* + \lambda_1 \|\mathcal{S}\|_{1,1,2} + \lambda_2 (\|\mathcal{C}\|_k^{sp})^2 \\ &\quad s.t. \;\; \mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{C} \end{aligned} \tag{5.4}$$

where $\|\mathcal{L}\|_*$ is the tensor nuclear norm, i.e. the average of the nuclear norm of all the frontal slices ($\|\mathcal{L}\|_* = \frac{1}{n_3} \sum_{p=1}^{n_3} \|\mathcal{L}_{:,:,p}\|_*$), and it approximates the rank of tensor $\mathcal{L}$. $\mathcal{S}$ and $\mathcal{C}$ are detected moving objects and illumination changes, respectively.

5.2.3 Optimization Algorithm

In order to solve (5.4), we use the standard inexact augmented Lagrangian method (ALM) with the augmented Lagrangian function $\mathcal{H}(\mathcal{L}, \mathcal{S}, \mathcal{C}, \mathcal{Y}; \mu)$ whose main steps are described in this section for completeness.

$$\begin{aligned} \mathcal{H}(\mathcal{L}, \mathcal{S}, \mathcal{C}, \mathcal{Y}; \mu) = \|\mathcal{L}\|_* + \lambda_1 \|\mathcal{S}\|_{1,1,2} + \lambda_2 (\|\mathcal{C}\|_k^{sp})^2 + < \mathcal{Y}, \mathcal{D} - \mathcal{L} - \mathcal{S} - \mathcal{C} > \\ + \frac{\mu}{2} \|\mathcal{D} - \mathcal{L} - \mathcal{S} - \mathcal{C}\|_F^2 \end{aligned} \tag{5.5}$$

where $\mathcal{Y}$ is a Lagrangian multiplier, μ is a positive auto-adjusted scalar, and $< A, B >= trace(A^T B)$. $\lambda_1 = 1/\sqrt{max(n_1, n_2) n_3}$ and λ_2 is a positive scalar. Since different values for λ's affect the overall accuracy of our method, We will discuss on the effect of the λ values in the experimental results. Now we solve the problem through alternately updating $\mathcal{L}, \mathcal{S}$, and $\mathcal{C}$ in each iteration to minimize $\mathcal{H}(\mathcal{L}, \mathcal{S}, \mathcal{C}, \mathcal{Y}; \mu)$ with other variables fixed until convergence as follows.

Updating L^{t+1}: From (5.5), the augmented Lagrangian reduces to the following form:

$$\min_{\mathcal{L}} \|\mathcal{L}\|_* + \frac{\mu}{2}\|\mathcal{L}^t - (\mathcal{D} - \mathcal{S}^t - \mathcal{C}^t + \frac{\mathcal{Y}^t}{\mu})\|_F^2 \tag{5.6}$$

which has a closed form solution in [83].

Updating $\mathcal{S}^{t+1}$: From (5.5), the augmented Lagrangian reduces to

$$\min_{\mathcal{S}} \lambda_1\|\mathcal{S}\|_{1,1,2} + \frac{\mu}{2}\|\mathcal{S}^t - (\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{C}^t + \frac{\mathcal{Y}^t}{\mu})\|_F^2 \tag{5.7}$$

which has a closed form solution [150].

Updating $\mathcal{C}^{t+1}$: From (5.5), the augmented Lagrangian reduces to

$$\min_{\mathcal{C}} \lambda_2(\|\mathcal{C}\|_k^{sp})^2 + \frac{\mu}{2}\|\mathcal{C}^t - (\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{S}^{t+1} + \frac{\mathcal{Y}^t}{\mu})\|_F^2 \tag{5.8}$$

The subproblem (5.8) has an efficient solution in [72].

Updating $\mathcal{Y}^{k+1}$: From (5.5),

$$\mathcal{Y}^{t+1} = \mathcal{Y}^t + \mu(\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{C}^{t+1} - \mathcal{S}^{t+1}) \tag{5.9}$$

where $\mu = min(\rho\mu, \mu_{max})$. The error is computed as $\|\mathcal{D} - \mathcal{L}^t - \mathcal{S}^t - \mathcal{C}^t\|_F / \|\mathcal{D}\|_F$. The loop stops when the error reaches the value lower than a threshold (10^{-5} in our experiments). Details of the solutions for (5.6), (5.7), and (5.8) are in Algorithms 5.1 and 5.2.

5.3 Time Complexity and Convergence Analysis of TLISD

5.3.1 Time Complexity

- In TLISD, we use sub-problems (5.6) and (5.7) to update $\mathcal{L}$ and $\mathcal{S}$, which have closed form solutions. In these two steps the main cost per-iteration

Algorithm 5.1 Tensor Low-rank and Invariant Sparse Decomposition (TLISD)

```
  Input: Tensor data $\mathcal{D}$, Parameters $\lambda_1 = 1/\sqrt{max(n_1, n_2)n_3}$,
    $\lambda_2 = 0.03$, $k = n_1$, $\rho = 1.2$, $\mu = 10^{-3}$
 1: while not converged do
 2:     $\mathcal{L}^{t+1} = prox\text{-}tnn(\mathcal{D} - \mathcal{S}^t - \mathcal{C}^t + \mu^{-1}\mathcal{Y}^t)$   //solves (8) in the paper
 3:     $\mathcal{S}_{temp} = \mathcal{L}^{t+1} + \mathcal{C}^t - \mathcal{D} + \mu^{-1}\mathcal{Y}^t$
 4:     for each row $i$ and lateral slice $j$   //lines 3-5 solve (9) in the paper
 5:       $\mathcal{S}^{t+1}(i,j,:) = \left(1 - \frac{\lambda_1}{\mu\|\mathcal{S}_{temp}(i,j,:)\|_F}\right)_+ \mathcal{S}_{temp}(i,j,:)$
 6:     $\mathcal{C}_{temp} = \mathcal{L}^{t+1} + \mathcal{S}^{t+1} - \mathcal{D} + \mu^{-1}\mathcal{Y}^t$
 7:     for each frontal slice $p$          //lines 6-8 solve (10) in the paper
 8:       $\mathcal{C}^{t+1}(:,:,p) = ksp(\mathcal{C}_{temp}(:,:,p), k, \mu^{-1}\lambda_2)$   //Algorithm 2
 9:     $\mathcal{Y} = \mathcal{Y} + \mu(\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{S}^{t+1} - \mathcal{C}^{t+1})$
10:     $\mu = \rho\mu$; $t = t + 1$
11: end while
  Output $\mathcal{L}^t, \mathcal{S}^t, \mathcal{C}^t$

  function prox-tnn($\mathcal{A}$)
12: $\mathcal{M} \leftarrow$ fft($\mathcal{A}, [\,], 3$)
13: for $i = 1 : n_3$
14:    $[U, S, V] = SVD(\mathcal{M}(:,:,i))$
15:    $\hat{\mathcal{U}}(:,:,i) = U$;  $\hat{\mathcal{S}}(:,:,i) = S$;  $\hat{\mathcal{V}}(:,:,i) = V$
16:    Updating t-rank using soft thresholding operator $\overline{\mathcal{S}}_{(1/\mu)}$ //Similar to [16]
17: End for
18: $\mathcal{U} \leftarrow$ ifft($\hat{\mathcal{U}}(:, 1 : t\text{-}rank, :), [\,], 3$);
    $\Sigma \leftarrow$ ifft($\hat{\mathcal{S}}(1 : t\text{-}rank, 1 : t\text{-}rank, :), [\,], 3$);
    $\mathcal{V} \leftarrow$ ifft($\hat{\mathcal{V}}(:, 1 : t\text{-}rank, :), [\,], 3$);
19: for $i = 1 : n_3$
20:    $\mathcal{X}(:,:,i) = (\mathcal{U}(:,:,i)\Sigma(:,:,i))\mathcal{V}^T(:,:,i)$
21: End for
22: return $\mathcal{X}$
```

lies in the update of $\mathcal{L}_{t+1}$, by computing tensor SVD (t-SVD). [83] showed that t-SVD can be efficiently computed based on the matrix SVD in Fourier domain, which requires computing FFT [17] and n_3 SVDs of $n_1 \times n_2$ matrices. Thus, time complexity of the first two steps per-iteration is $O(n_1 n_2 n_3 \log n_3 +$

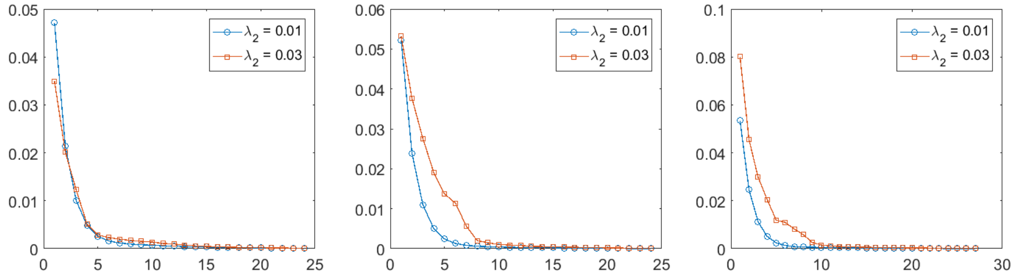
Algorithm 5.2 Solving $k-$support norm [72]

```
function ksp($W, k, \gamma$)
$\beta = 1/\gamma$, $\nu = vec(W)$    where $\nu \in R^d$, $d = n_1 \times n_2$ //size of each frontal slice
$z = |\nu|^{\downarrow}, z_0 = +\infty, z_{d+1} = -\infty$
for $r = k - 1 : 0$
  Obtain $l$ by BinarySearch(z,k-r,d)
  $T_{r,l} = \sum_{i=k-r}^{l} z_i$
  If $\frac{1}{\beta+1} z_{k-r-1} > \frac{T_{r,l}}{l-k+r+1+\beta(r+1)} \geq \frac{1}{\beta+1} z_{k-r}$
    break;
  End If
End for
For $i = 1 : d$
  calculate $q_i = \begin{cases} \frac{\beta}{\beta+1} z_i & \text{if } i = 1, ..., k - r - 1 \\ z_i - \frac{\sum_{i=k-r}^{l} z_i}{l-k+r+1+\beta(r+1)} & \text{if } i = k - r, ..., l \\ 0 & \text{if } i = l + 1, ..., d \end{cases}$
  $w_i = sign(\nu_i) q_i$
End for
Output : $W$

function BinarySearch($z, low, high$)
If $z_{low} = 0$
   return $l = low$
End If
While $low < high - 1$
  $mid = \ulcorner \frac{low+high}{2} \urcorner$        //$\ulcorner x \urcorner$ represents the smallest integer which is larger than $x$
  If $z_{mid} > \frac{\sum_{i=k-r}^{mid} z_i}{mid-k+r+1+\beta(r+1)}$
    $low = mid$
  Else
    $high = mid - 1$
  End If
End While
return $l = low$
```

$n_{(1)} n_{(2)}^2 n_3$), where $n_{(1)} = max(n_1, n_2)$ and $n_{(2)} = min(n_1, n_2)$ [83].

- To update $\mathcal{C}_{t+1}$, we use an efficient solution based on binary search where



(a) (b) (c)

Figure 5.4: Convergence of TLISD on sequence Wildlife3 with (a) $\lambda_1 = 0.001$, (b) $\lambda_1 = 0.003$, and (c) $\lambda_1 = 0.005$.

the time complexity is reduced to $O((n_1n_2 + k)\log(n_1n_2))$ for each frontal slice per-iteration [72].

Therefore, the total time complexity of the optimization problem (5.4) is $O(n_1n_2n_3\log n_3 + n_{(1)}n_{(2)}^2n_3 + (n_1n_2 + k)n_3\log(n_1n_2))$. We also evaluate the running time of TLISD, that will be discussed in the experimental section.

5.3.2 Convergence Analysis

Although our proposed TLISD in (5.4) is a non-convex formulation, the convergence of each sub-problem is guaranteed. For updating sub-problems (5.6) and (5.7), we use inexact ALM as demonstrated in Algorithm 5.1. The convergence of inexact ALM with at most two blocks has been well studied and a proof to demonstrate its convergence property can be found in [77]. [1] showed k-support norm is a convex relaxation of the matrix sparsity combined with the l_2-norm penalty, and so the convergence of Sub-problem (5.8) is guaranteed.

In addition, we demonstrate the convergence properties of our algorithm in practice. To verify the convergence of TLISD, we examine TLISD on “Wildlife3” sequence with different values of λ_1 and λ_2. Fig. 5.4 shows the convergence curves of the proposed TLISD on the sequence. It can be observed that TLISD efficiently converges, and after few iterations the value of objective function becomes stable.

5.4 Experimental Results and Discussion

In this section, we provide an experimental evaluation of our proposed method, TLISD. We first evaluate the effect of each term in (5.4) and their λ coefficients. Then, we evaluate TLISD on benchmark frame-rate image sequences or those that are captured via time-lapse or motion-triggered photography. We also extend our introduced illumination change dataset (ICD), which includes more than 80k images captured by industrial security cameras and wildlife monitoring systems during three years, and evaluate our method on this extended dataset.

5.4.1 Experiment Setup

- ***Existing datasets***: We evaluate our TLISD method on eleven selected sequences from the CDnet dataset [46], Wallflower dataset [124], I2R dataset [74], and our proposed ICD [103], introduced in the previous chapter, which include illumination change and moving shadows. All the sequences are described in Chapter 4.6.1.

- ***Extended Illumination Change (EIC) dataset:*** Due to the lack of a comprehensive dataset with various illumination and shadow changes in a real environment, and since ICD only includes five sequences, we have created more sequences and introduce extended illumination change dataset (EIC) with around 80k images in 15 sequences, captured via available surveillance systems in wildlife and industrial applications. Particularly, ten sequences are captured via wildlife monitoring systems, and five sequences from industrial applications, with three railway sequences and two construction site sequences. All sequences of this dataset and image size of each sequence are shown in Fig. 5.5 and Table 5.1, respectively. To provide a consistent names for this dataset with ICD, where has three wildlife sequences, we started the names of wildlife sequences with “Wildlife4”. We evaluate our method on these sequences and compare them with the results of competing methods.

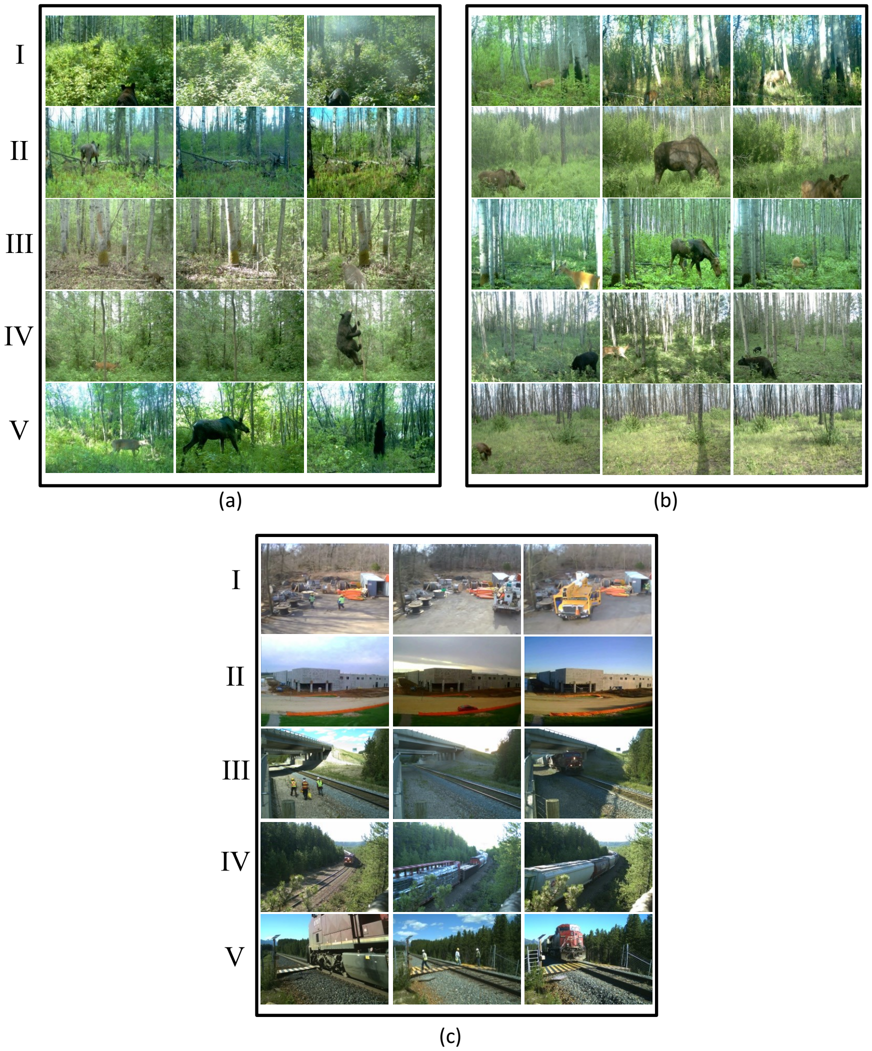


Figure 5.5: Three selected images from each sequence of EIC dataset, captured via surveillance systems in wildlife and industrial applications. Rows in (a) and (b) show 10 wildlife sequences. Rows in (c) show 5 sequences from industrial applications including construction sites and railways sequences

Table 5.1: Name and image size of EIC sequences correspond to the rows in Fig. 5.5

	Sequence	Image size		Sequence	Image size		Sequence	Image size
Fig. 5.5(a)	I. Wildlife4	[358,508]	Fig. 5.5(b)	I. Wildlife9	[358,508]	Fig. 5.5(c)	I. Industrial area1	[350,450]
	II. Wildlife5	[358,508]		II. Wildlife10	[358,508]		II. Industrial area2	[350,450]
	III. Wildlife6	[358,508]		III. Wildlife11	[358,508]		III. Railway1	[350,450]
	IV. Wildlife7	[358,508]		IV. Wildlife12	[358,508]		IV. Railway2	[350,450]
	V. Wildlife8	[358,508]		V. Wildlife13	[358,508]		V. Railway3	[350,450]

- ***Evaluation metric***: For quantitative evaluation, pixel-level F-measure $= 2\frac{recall \times precision}{recall + precision}$ is used. We also compare the different methods in execution time in seconds.

5.4.2 Algorithm Evaluation: The effect of term $\mathcal{C}$

In the first set of experiments, we evaluate the effect of term $\mathcal{C}$ in TLISD when we set different values for λ_1, in comparison with TLISD without term $\mathcal{C}$, where (5.4) becomes

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_* + \lambda_1 \|\mathcal{S}\|_{1,1,2} \quad s.t. \;\; \mathcal{D} = \mathcal{L} + \mathcal{S} \tag{5.10}$$

Fig. 5.6(a) shows (5.10) can achieve around 70% accuracy with a well-tuned $\lambda_1 = 0.002$. Although the result shows the importance of multiple priors and the effect of group sparsity on them, the accuracy of (5.10) is still far below the accuracy of proposed TLISD by at least 10%, even with a well-tuned λ_1. Fig. 5.6(a) also shows that adding term $\mathcal{C}$ and $k - support$ norm increases the robustness of our algorithm against tuning λ_1. In fact, in (5.10) all illumination variations would be assigned to either of $\mathcal{L}$ or $\mathcal{S}$. In this case, those variations should be assigned to the background ($\mathcal{L}$); however, they do not actually belong to background (e.g. moving shadows). As a result, the rank would be increased to absorb these changes into $\mathcal{L}$ and naturally some parts of the moving objects $\mathcal{S}$ would be also absorbed into the background. Fig. 5.6(b) supports the conclusion and shows the obtained rank through the iterations of the optimization. Between iterations 15 and 20, the rank of our method without term $\mathcal{C}$ significantly increases to absorb all variations into $\mathcal{L}$, and to complete the conclusion, Fig. 5.6(f) shows that around the same iterations, the residual error of the method without term $\mathcal{C}$ is significantly reduced. This means, illumination variations and shadow changes must grouped into either of $\mathcal{L}$ or $\mathcal{S}$, for (5.10) to converge. Estimated rank in Fig 5.6(c) shows the proof of this concept. Obviously, with a very small λ_1, the estimated rank of $\mathcal{L}$ for (5.10) is small and all illumination variations are easily lumped with moving objects in $\mathcal{S}$. This causes less accuracy and sometimes even cannot

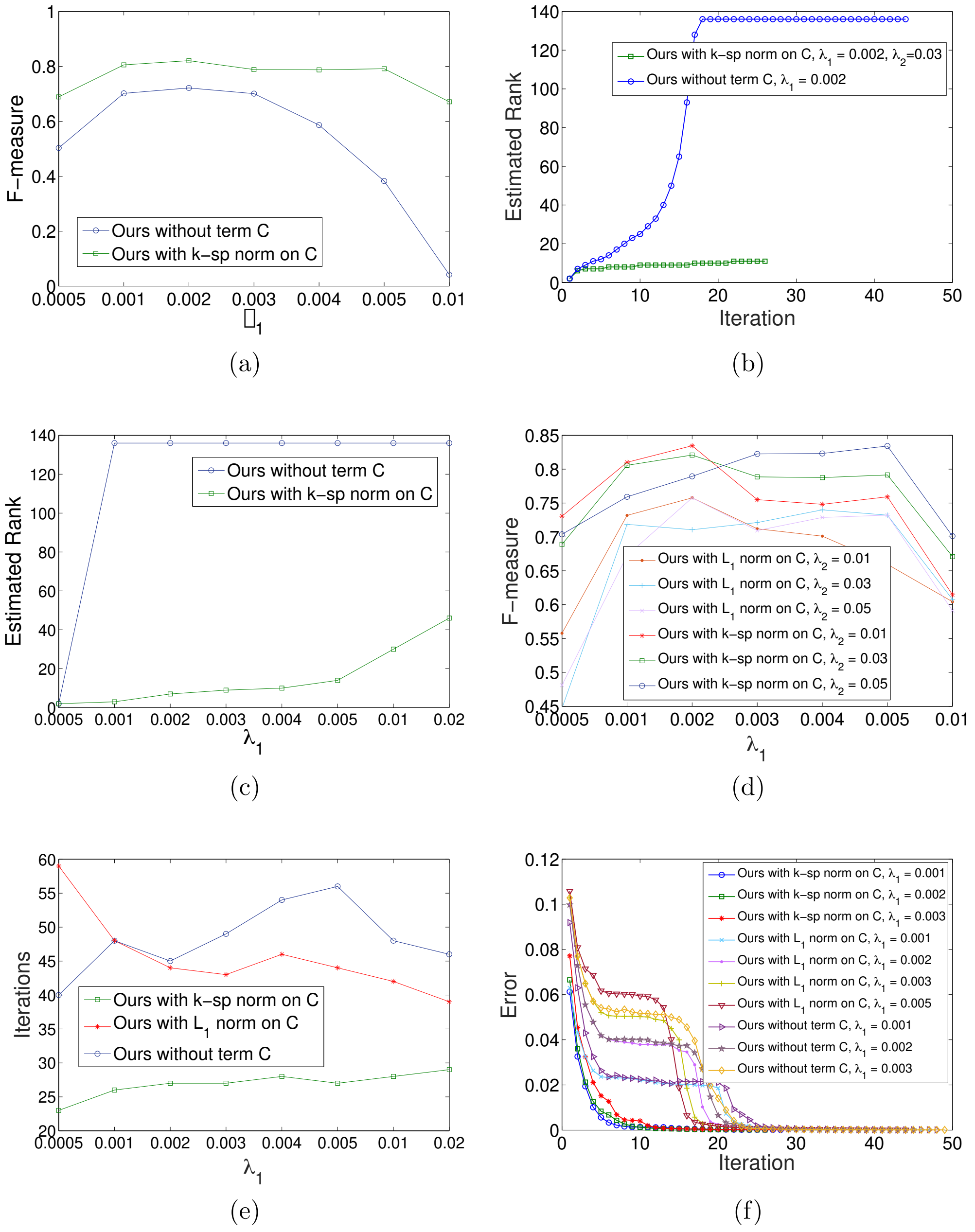


Figure 5.6: Self evaluation of TLISD. (a) Average F-measure with different values for λ_1 on all ICD sequences between TLISD and (5.10), (b) Estimated rank of TLISD and (5.10) through iterations on sequence “Wildlife3”, (c) Estimated rank of sequence “Wildlife3” with different values for λ_1, (d) Average F-measure with different values for λ_1 and λ_2 on all ICD sequences between TLISD and (5.11), (e) Average number of iterations to converge TLISD, (5.10) and (5.11) on all ICD sequences, (f) Convergence curves of minimization error for TLISD, (5.10) and (5.11) on sequence “Wildlife3”.

provide meaningful results. In contrast, TLISD can estimate a balanced rank and classify illumination variations into term $\mathcal{C}$ with $k - support$ norm on it instead of increasing the rank to absorb them into $\mathcal{L}$.

To justify the use of $k - support$ norm on $\mathcal{C}$ in TLISD, we also compare the method with the other potential term on $\mathcal{C}$, which is l_1-norm to absorb outliers, i.e., define (5.4) as

$$\min_{\mathcal{L},\mathcal{S},\mathcal{C}} \|\mathcal{L}\|_* + \lambda_1 \|\mathcal{S}\|_{1,1,2} + \lambda_2 \|\mathcal{C}\|_1 \quad s.t. \, \mathcal{D} = \mathcal{L} + \mathcal{S} + \mathcal{C} \tag{5.11}$$

For this experiment, we evaluate our method with both l_1 and $k - support$ norms on $\mathcal{C}$ under different values of λ_1 and λ_2. Fig. 5.6(d) illustrates the accuracy of our method with either of regularizers. Although l_1-norm can increase the accuracy and robustness of the moving object detection in comparison with (5.10) that we showed in Fig. 5.6(a), the obtained accuracy is still less than TLISD. In addition, the number of iterations to converge, for both (5.10) and (5.11) is much more than that of in TLISD. Fig. 5.6(e) shows the average number of iterations for all three possible methods with different setup for λ_1 on all ICD sequences. For both TLISD and (5.11), $\lambda_2 = 0.03$, which produces robust results over different values of λ_1(refer to Fig. 5.6(d)). As discussed in Section 5.2, illumination changes are not necessarily sparse and can be found throughout an image. Therefore, l_1-norm is not a suitable regularizer to capture illumination changes. In such cases, the same issue as (5.10) happens when the optimizer increases the rank to minimize the residual error. Fig. 5.6(f) shows the error of all three methods through iterations. For (5.11), the same pattern as (5.10) is seen to decrease the error while the rank increases through optimization.

5.4.3 Evaluation on Benchmark Sequences

In this section we evaluate our method on the eleven benchmark sequences described in Section 5.4.1. Fig. 5.7 shows the qualitative results of TLISD on "Cubile" and "Backdoor". The second and the third columns of Figs. 5.7(a) and (b) illustrate the first frontal slice of $\mathcal{C}$ and $\mathcal{S}$, corresponding to illu-

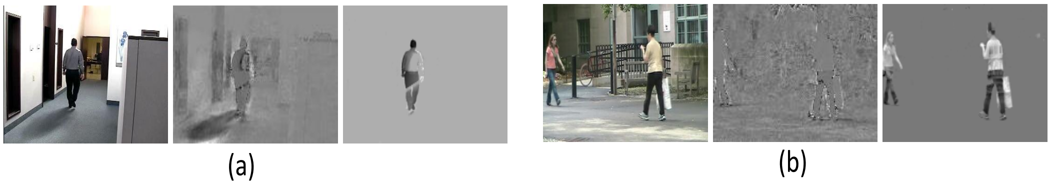


Figure 5.7: Columns from left to right show sample image, illumination changes, and detected moving objects for (a) cubicle and (b) backdoor sequences

mination changes and moving objects, respectively. The high-quality of our detection result $\mathcal{S}$ is clearly visible.

Figs. 5.8(a)and (b) show qualitative results of our method on two sample sequences of ICD, which has the most challenging conditions in terms of illumination changes. To appreciate the significant variations of illumination we show two images from each sequence. The second and the third rows of each sub-figure show the first frontal slice of $\mathcal{C}$ and $\mathcal{S}$, respectively. The results show the proposed method can accurately separate the changes caused by illumination and shadows from real changes.

We then compare TLISD quantitatively with two online and eight related

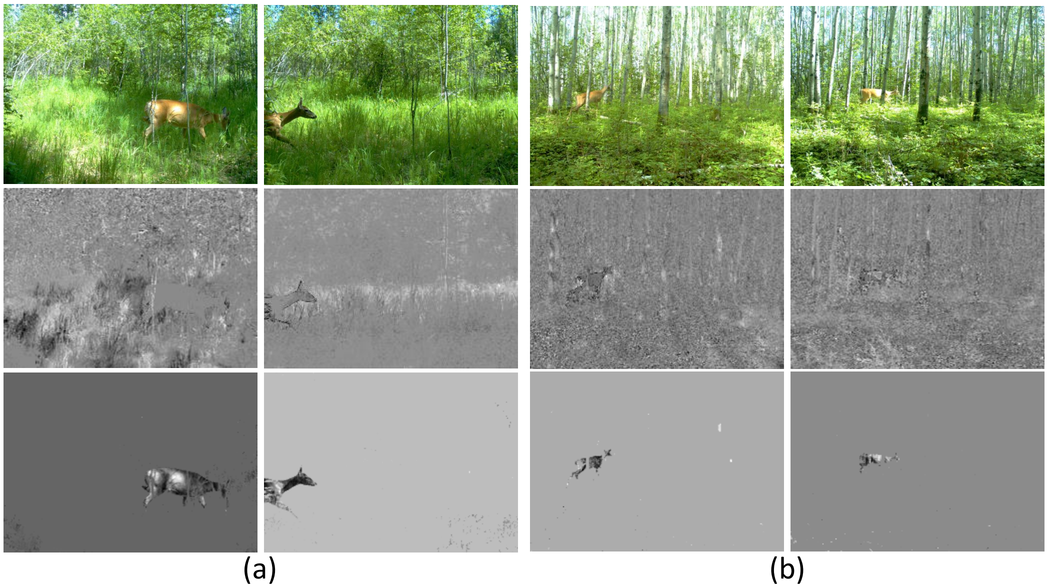


Figure 5.8: First row: two sample images from (a) Wildlife1, (b) Wildlife3 sequences. Second row: illumination changes obtained from the first frontal slice of $\mathcal{C}$. Third row: detected objects from the first frontal slice $\mathcal{S}$.

RPCA batch methods. From online methods we select GMM [158] as a baseline method and GRASTA [51] as an online method that uses the framework of low-rank and sparse decomposition. Also among batch methods, we select SS-GoDec [153], PRMF [131], PCP [21], Markov BRMF [132], DECOLOR [155], LSD [82], ILISD [103], and TRPCA [83]. For all the competing methods we use their original settings through LRS Library [115], which resulted in the best performance. For quantitative evaluation of RPCA-related methods, a threshold criterion is required to get the binary foreground mask. Similarly, we adopt the same threshold strategy as in [115] to obtain the binary mask O.

$$O_{i,j} = \begin{cases} 1 \quad if \quad S_{i,j}^2 > \sigma^2, \\ 0 \qquad otherwise \end{cases} \tag{5.12}$$

where $S = \mathcal{S}_{:,:,1}$ is the first frontal slice of $\mathcal{S}$, and σ is the standard deviation of all pixels in S. In TLISD, we set $\lambda_1 = 1/\sqrt{max(n_1, n_2)n_3}$ and $\lambda_2 = 0.03$. The value for λ_1 is similar to TRPCA, to have a fair comparison between TLISD and TRPCA.

Table 5.2 shows the performance of TLISD in comparison with the competing methods in terms of F-measure on frame-rate benchmark sequences. For all the sequences TLISD ranked among the top two of all methods, and achieves the best average F-measure in comparison with all other methods. Although DECOLOR, PCP, LSD, and ILISD work relatively well, Only ILISD method, which is proposed in the previous chapter, is comparable with TLISD due to the use of illumination regularization terms in ILISD.

Table 5.3 illustrates the performance of TLISD in comparison with the competing methods in terms of F-measure on time-lapse ICD sequences. In this experiment, almost all existing methods fail to detect moving objects under discontinuous illumination change, and only ILISD is comparable with or TLISD method. This evaluation shows the effectiveness of multiple prior maps and $k - support$ norm as two regularization terms for separating moving objects from sudden or significant illumination changes, and boosting the overall performance of object detection in comparison with ILISD.

Fig. 5.9 shows qualitative comparison of our method with all batch methods of Table 5.3. Most methods failed to detect moving objects under sig-

Table 5.2: Comparison of F-measure score between our proposed method and other compared methods on benchmark frame-rate sequences (best F-measure: bold, second best F-measure: underline)

Sequence	Backdoor	CopyMachine	Cubicle	PeopleInShade	LightSwitch	Lobby
GMM	0.6512	0.5298	0.3410	0.3305	0.4946	0.3441
GRASTA	0.6822	0.6490	0.4113	0.5288	0.5631	0.6727
SSGoDec	0.6611	0.5401	0.3035	0.2258	0.3804	0.0831
PRMF	0.7251	0.6834	0.3397	0.5163	0.2922	0.6256
DECOLOR	0.7656	0.7511	0.5503	0.5559	0.5782	0.7983
PCP	0.7594	0.6798	0.4978	0.6583	**0.8375**	0.6240
BRMF	0.6291	0.3293	0.3746	0.3313	0.2872	0.3161
LSD	0.7603	0.8174	0.4233	0.6168	0.6640	0.7313
ILISD	0.8150	0.8179	0.6887	**0.8010**	0.7128	0.7849
TRPCA	0.7022	0.6805	0.5329	0.5683	0.6924	0.6383
TLISD	**0.8276**	**0.8445**	**0.7350**	0.7961	0.7429	**0.8012**

Table 5.3: Comparison of F-measure score between our proposed method and other compared methods on ICD sequences (best F-measure: bold, second best F-measure: underline)

Sequence	Wildlife1	Wildlife2	Wildlife3	WinterStreet	MovingSunlight
GMM	0.2374	0.2880	0.0635	0.1183	0.0717
GRASTA	0.3147	0.3814	0.2235	0.2276	0.1714
SSGoDec	0.2912	0.2430	0.0951	0.1215	0.2824
PRMF	0.2718	0.3991	0.07012	0.2108	0.2932
DECOLOR	0.3401	0.3634	0.1202	0.4490	0.3699
PCP	0.5855	0.6542	0.3003	0.1938	0.3445
BRMF	0.2743	0.2812	0.0735	0.0872	0.2408
LSD	0.6471	0.3790	0.0871	0.1604	0.3593
ILISD	0.8033	0.7277	0.7398	0.6931	0.6475
TRPCA	0.4382	0.3926	0.2854	0.2721	0.3018
TLISD	**0.8862**	**0.8065**	**0.8010**	**0.7092**	**0.7122**

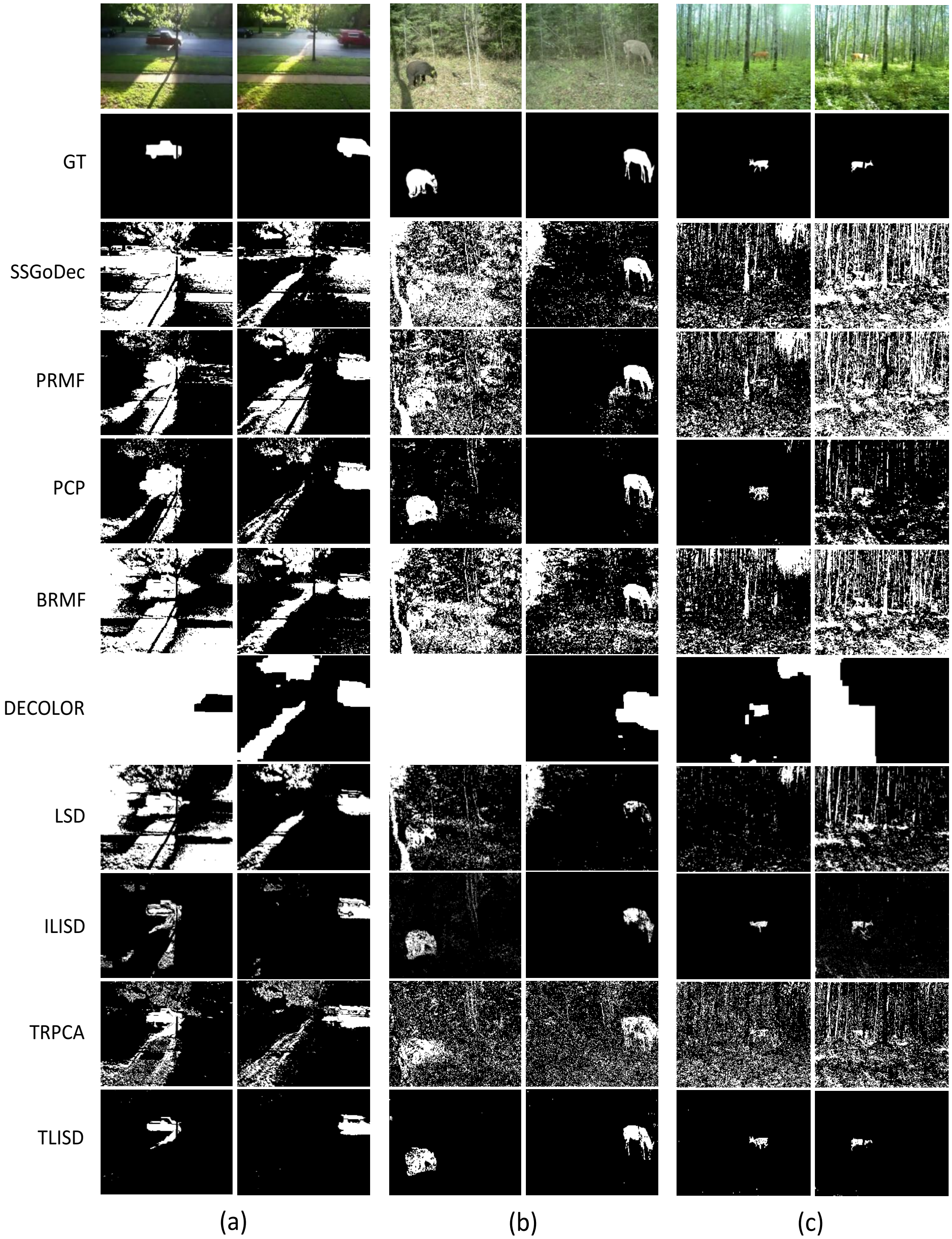


Figure 5.9: Comparison of qualitative results between our method (TLISD) and eight rpca-related methods on two selected images of sequences (a) "MovingSunLight", (b) "Wildlife2", and (c) "Wildlife3"

nificant illumination changes, and only the results of ILISD are comparable with TLISD. However, based on this qualitative results and Tables 5.2 and 5.3, TLISD outperforms all competing methods by a clear performance margin.

5.4.4 Discussion on Failure Cases of ILISD

Based on Tables 5.2 and 5.3, since ILISD is the the only method with comparable results to our new method, we examine TLISD and ILISD qualitatively and show three failure cases of ILISD which TLISD can successfully handle. For these three cases, Fig. 5.10 shows the quality of detected objects from ILISD and TLISD in the second and the third columns respectively. In these cases, due to use of an inaccurate prior map and the same norm for both illumination and real changes, ILISD generates false positive detections. Since TLISD uses multiple prior maps and two different norms for separating real changes from illumination changes, it can correctly classify those false positive pixels into $\mathcal{C}$ as illumination changes. Furthermore, in the second row of Fig. 5.10(d), some of the foreground pixels are mistakenly classified as illumination changes due to inaccurate prior map where TLISD can classify them correctly as moving object. Figs. 5.10(d) and (e) show the corresponding changes detected as

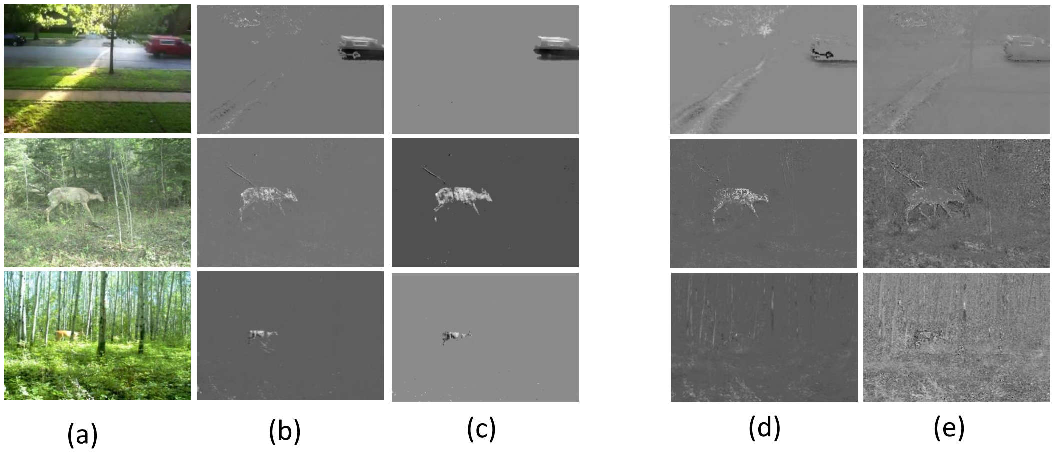


Figure 5.10: Detected objects from ILISD and TLISD in (b) and (c), and their corresponding illumination changes captured in $\mathcal{C}$ in (d) and (e), respectively. Rows from top to bottom shows MovingSunight, Wildlife2, and Wildlife3 sequences.

illumination changes, captured in $\mathcal{C}$.

5.4.5 Evaluation of TLISD on EIC Dataset

In this section, we evaluate TLISD on the introduced EIC dataset. Three selected sequences of EIC are shown in Fig. 5.11, which includes one chal-

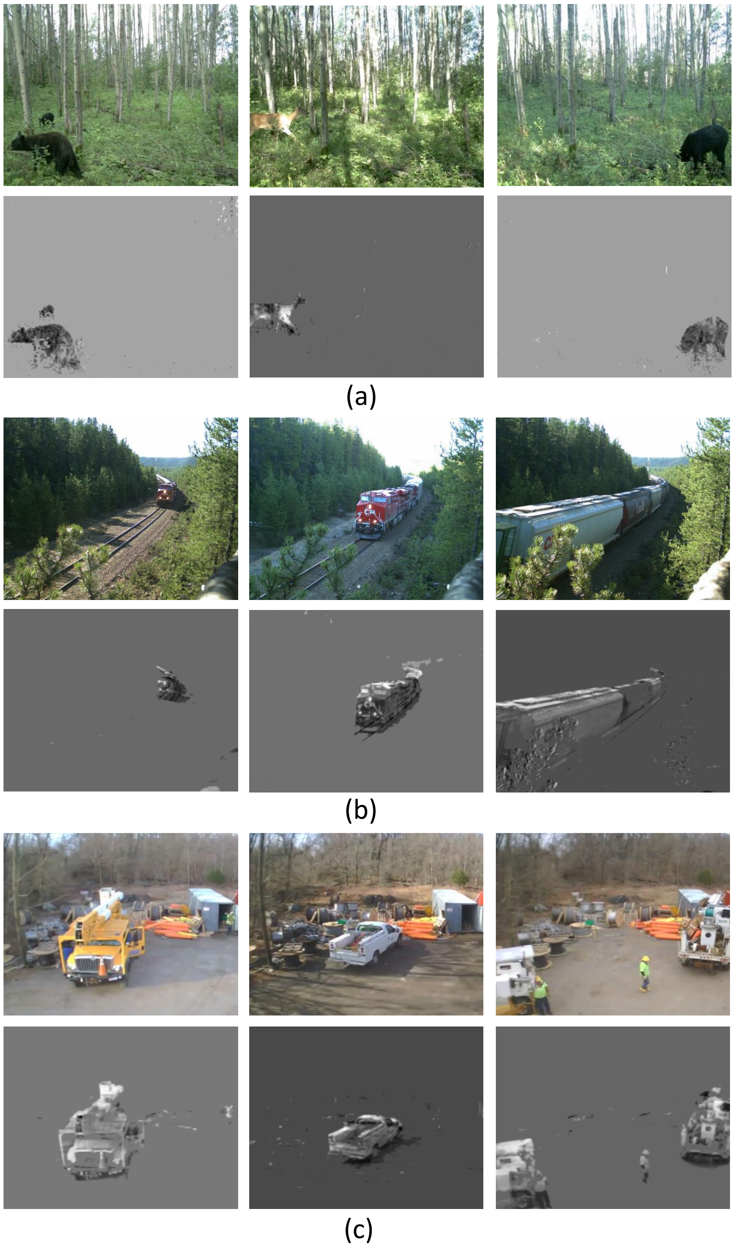


Figure 5.11: First row of each subfigure: Three selected images of sequences (a) wildlife, (b) railway, and (c) construction sites. Second row of each subfigure: Detected moving objects using TLISD.

lenging sequence from wildlife and two sequences from industrial applications in construction site and railway monitoring system. To understand the significant variations of illumination and shadow, we show three images from each sequence in Figs. 5.11(a), (b), and (c). Second row of each subfigure shows the detected objects using TLISD. The results demonstrate the proposed method can accurately detect real changes (moving objects) under dis-

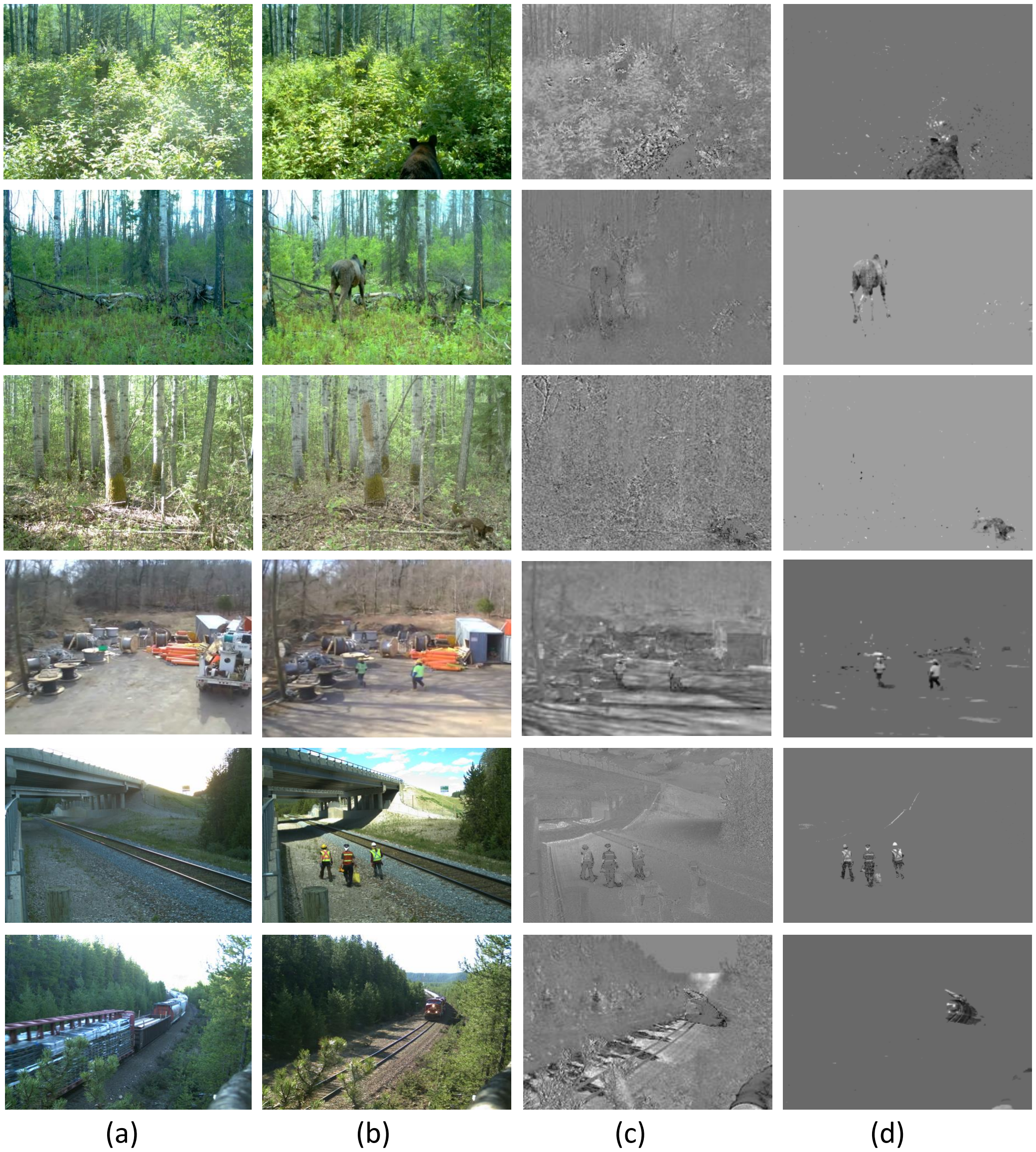


Figure 5.12: (Columns (a) and (b): two selected images of each sequence, (c) and (d): illumination changes captured in $\mathcal{C}$, and detected objects of images in (b), respectively

continuous change in illumination.

To examine the capability of TLISD for separating real changes from illumination changes, six sample sequences of EIC are shown in Fig. 5.12. To appreciate the significant variations of illumination and shadow, we show two images from each sequence in Figs. 5.12(a) and (b). Columns (c) and (d) show the first frontal slices of $\mathcal{C}$ and $\mathcal{S}$ obtained by TLISD for the images in column (b), in order to capture illumination changes and to detect moving objects. Table 5.4 shows the capability of TLISD in comparison with the four best competitive methods (based on Table 5.2) in terms of F-measure, where TLISD can outperform the other methods by a clear performance margin. Fig. 5.13 also compares TLISD with ILISD (the second best method in Table. 5.4) qualitatively. This qualitative comparison shows that one prior map only is not always sufficient for removing the effect of illumination variations and shadow. As discussed in Section 5.2.1, due to the variation in the invariant direction for images in a sequence, in some conditions separating illumination changes and shadows from real changes is roughly impossible and selecting multiple prior maps is essential.

For the sake of completeness, we show qualitative results obtained from the rest of EIC wildlife sequences. Fig. 5.14 shows one sample image from the sequences of “Wildlife7” to “Wildlife13” from EIC dataset. The second and the third columns of Fig. 5.14 illustrate the results of our method obtained from the first frontal slice of C, and S, corresponding to illumination changes

Table 5.4: Comparison of F-measure score between our proposed method and other compared methods on EIC dataset

Sequence	Wildlife4	Wildlife5	Wildlife6	Railway1	Railway2	Industrial area1
PCP	0.4150	0.4016	0.3092	0.3634	0.4086	0.2869
DECOLOR	0.3475	0.2010	0.2604	0.2853	0.3021	0.3242
ILISD	0.6020	0.6104	0.6170	0.5983	0.5414	0.5626
TRPCA	0.2934	0.3082	0.2855	0.3447	0.2805	0.2914
TLISD	**0.7508**	**0.8049**	**0.7522**	**0.7241**	**0.7116**	**0.7035**

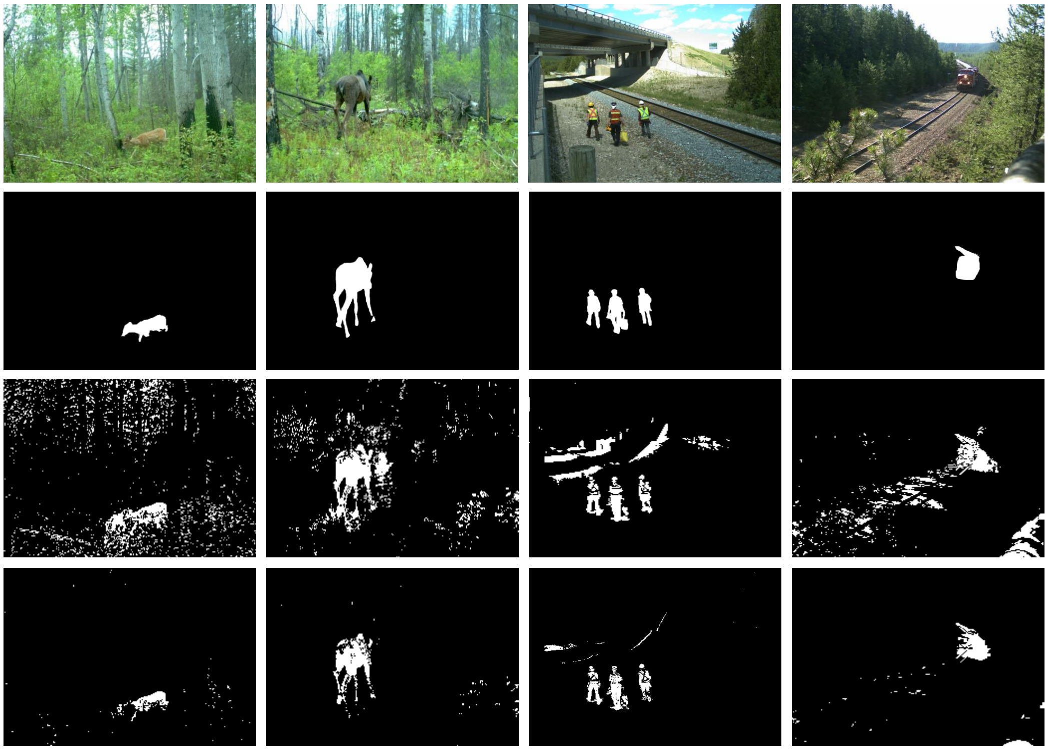


Figure 5.13: Comparison of qualitative results between TLISD and ILISD on four sequences of EIC dataset. Top to bottom: Sample Image, Ground Truth, ILISD, and TLISD

and moving objects, respectively.

5.4.6 Execution Time of TLISD

Based on Tables 5.2 and 5.4, since ILISD is the only method with comparable results to our new method, we examine both ILISD and TLISD methods in terms of computation time. Table. 5.5 compares the execution time of both methods on seven sequences. Regarding the computation time of the proposed method, our tensor-based method needs more time than ILISD [103] for each iteration, which is normal due to use of the tensor structure. However, the number of iterations in TLISD is less than that of ILISD. Fig. 5.15 shows the number of iterations to converge for both ILISD and TLISD methods. As explained in Chapter 4.4, ILISD has two independent optimization formulae: one for providing a prior map and the other for separating moving objects from illumination changes, and they have independent numbers of iterations

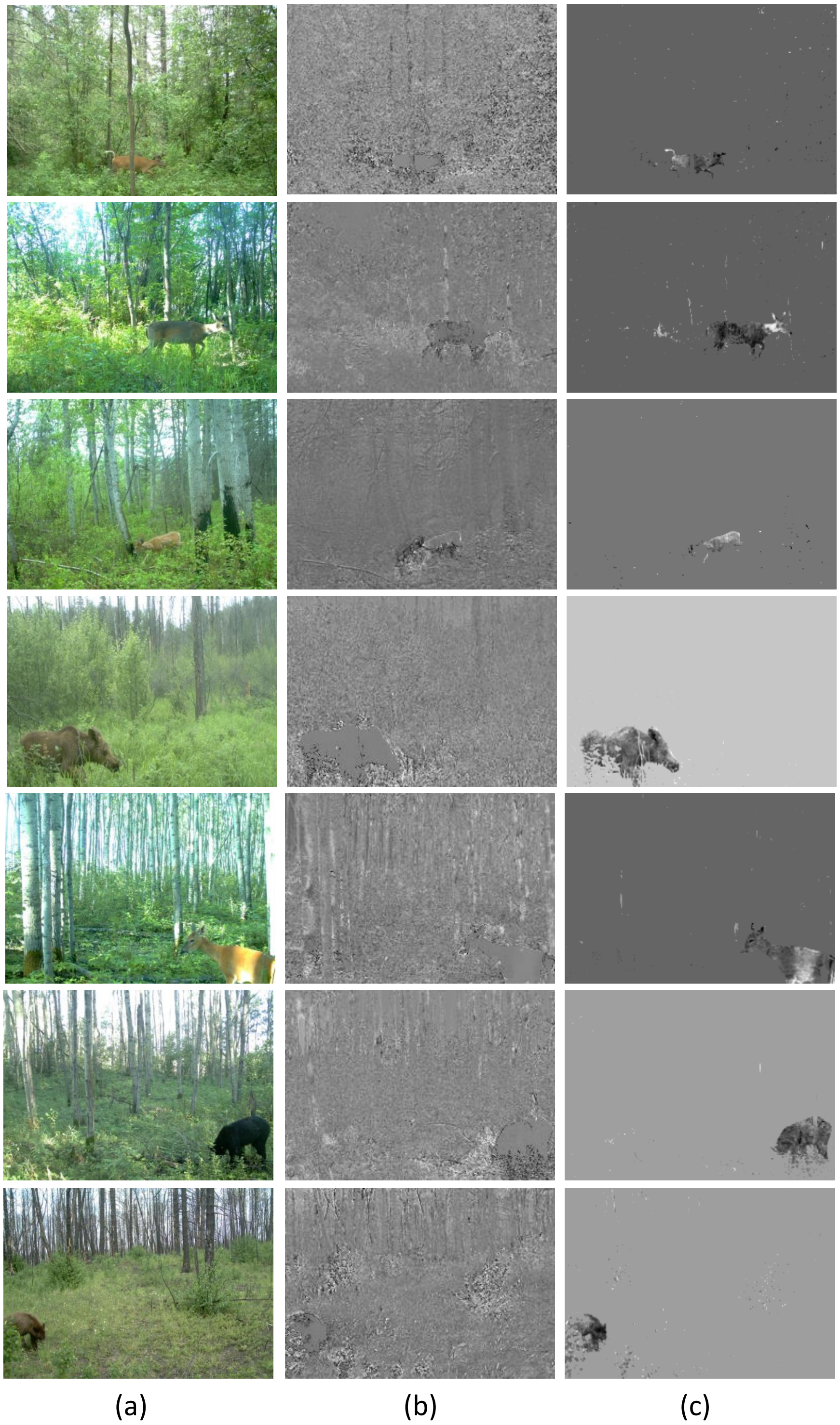


Figure 5.14: Qualitative results of our method (TLISD) on seven wildlife sequences captured by a motion-triggered camera. (a) sample image (b) corresponding illumination changes (c) detected moving objects.

Table 5.5: Comparison of execution time (in sec.) per image between TLISD and ILISD on seven sample sequences

Sequence	Backdoor	Lobby	Cubicle	Wildlife1	Wildlife2	Wildlife3	MovingSunlight
ILISD	0.49	0.53	0.74	1.24	1.33	1.18	2.2
TLISD	0.98	2.38	1.79	2.52	4.26	4.08	5.16

to converge. After convergence, the optimized values are interchangeably used in an outer loop, and hence the total number of iterations is much more than that of our method which involves one optimization formula. As discussed in Section 5.3, the dominant time in our method is SVD decomposition for frontal slices, which are independent from each other, and so can be solved in parallel on a GPU to speed up the computation. Therefore, the total time of TLISD is at least comparable with ILISD and can be even faster due to the fewer number of iterations using GPU.

5.5 Summary

In this chapter, we proposed a novel method based on tensor low-rank and invariant sparse decomposition to detect moving objects under discontinuous changes in illumination, which frequently happen in video surveillance applica-

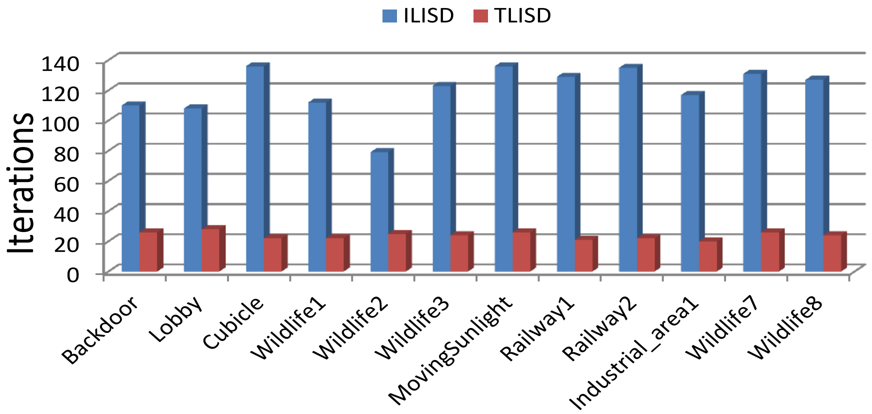


Figure 5.15: Number of iterations to converge ILISD and TLISD methods on twelve sequences

tions. In our proposed method, first we compute a set of illumination invariant representations for each image as prior maps, which provide us with cues for extracting moving objects. Then we model illumination changes in an image sequence using a k-support norm and derive a new formulation to effectively capture illumination changes and separate them from detected foregrounds.

As explained in Chapter 1, many surveillance systems, especially security and wildlife monitoring cameras, use motion triggered sensors and capture image sequences with significant illumination changes. Our proposed method can solve the problem with a performance that is superior to the state-of-the-art solutions. Our method is also able to extract natural outdoor illumination as labeled data for learning-based methods, which can be an effective alternative to optimization based methods such as ours, but with a sequential formulation, to detect illumination changes and moving objects from image sequences.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

According to the low-rank and sparse representation theory, linear correlation among the images of a sequence has great power to represent the background model of the sequence. Regarding this fact, different background subtraction and moving object detection methods using low-rank and sparse decomposition have been proposed. In this thesis, we presented a comprehensive study of low-rank and sparse representation based methods, with two special interests in a) solving the problem of moving object detection in a sequential manner with a connectivity constraint on moving objects, and b) solving the problem of moving object detection under discontinuous change in illumination which is a challenging task in computer vision. We also extensively explored the capability of our proposed algorithms, which are summarized as follows.

First, we proposed a sequential framework, namely contiguous outliers representation via online low-rank approximation (COROLA), to detect moving objects and learn the background model at the same time. Since many of the existing methods using low-rank and sparse decomposition work in a batch manner, they are not being applied in real time and long-term continuous tasks. Considering this challenge, we proposed our method, in which we borrow a continuity constraint from batch methods, and solve the problem with the help of online robust PCA. To handle local variations of the background as well as global variations, we also use GMM on the obtained sparse outliers. The experimental results reveal that proposed COROLA method performs well to detect moving objects in comparison with the other sequential methods.

Second, we proposed a method that can deal with discontinuous illumination change and shadow to detect moving object. As discussed in Chapter 1, many surveillance systems in industrial or wildlife monitoring areas use a motion triggered camera or a time-lapse photography system for monitoring the areas and provide time-lapse image sequences. Due to significant and complex changes in illumination and independent changes of the moving objects between images of the sequences, detection of the moving objects is extremely challenging. In such cases the problem of moving object detection cannot be solved by existing methods and almost all of them fail. To address this challenge, we first introduced an illumination regularization term, by proposing a new prior map obtained by illumination invariant representation of images. Then, we proposed a low-rank and invariant sparse decomposition method using the prior map to detect moving objects under significant illumination changes. We also proposed an iterative version of LISD by updating the prior map in one representation and impose it as a constraint into the LISD formulation with another representation. Experiments on challenging benchmark datasets demonstrate the superior performance of our proposed method under complex illumination changes where all other existing methods fail.

Third, we proposed a novel formulation that can capture illumination variations and can separate them from moving objects in an image sequence. In this method, we first showed that only one prior map is not sufficient for modeling illumination variations, and then proposed a way to build a set of prior maps from each single image. Then we defined a specific tensor structure using the prior maps and the original grayscale images. Finally, we proposed a new formulation based on low-rank tensor decomposition using group sparsity and k-support norm as two regularization terms to separate moving objects and illumination variations. This formulation solves the problem of moving object detection with a great improvement in separating moving objects from illumination variations that undergo discontinuous changes.

Finally, through this thesis we introduced a new benchmark dataset for the problem of moving object detection in real applications. As discussed, many surveillance systems use images which are captured by a motion triggered camera or with a time-lapse photography system. Therefore, current benchmarks are not suitable for evaluating the solutions for these systems. In this thesis, we first introduce an illumination change dataset (ICD), with five challenging sequences, and then we extend it as an extended illumination change (EIC) dataset, with fifteen more sequences captured from industrial and wildlife areas.

6.2 Limitations and Future Directions

There are several potential future research directions that can be explored to build upon the contributions of this thesis. We describe some of them as follows.

- In Chapter 3, we proposed COROLA method, which can be potentially improved in two ways. First, in COROLA we already use GMM on outliers, which is not a part of our optimization framework. This becomes more interesting if we integrate it into the optimization formulation rather than using it separately. Secondly, in COROLA we use a simple fixed affine transformation, which is not involved in the optimization and may not be accurate enough for complicated scenes. In the case of image alignment using low-rank framework, many methods used the transformation parameter into the minimization framework and showed promising results for image alignments [92], [155]. We are interested in exploring this approach in the case of moving cameras.

- Our proposed LISD method has been implemented and evaluated on challenging datasets in Chapter 4. Although the proposed method shows satisfactory results, it can be improved as follows.

  Structured LISD: In LISD we simply use l_1-norm to constrain the sparse matrices and to detect moving objects. The l_1-norm treats each entry (pixel) independently and does not consider the spatial connection of the foreground sparse pixels. While in many practical scenarios, foreground objects usually have the structural properties of spatial contiguity. To take advantage of this prior knowledge, we consider a structured sparsity-inducing norm which involves overlapping groups of variables, inspired by recent advances in structured sparsity [65], [85]. This structured sparsity norm is defined as follows.

$$\Gamma(S) = \sum_{i=1}^{n} \sum_{g \in \mathcal{G}} \|s_g^i\|_\infty \tag{6.1}$$

where $S \in R^{m \times n}$, and the i^{th} column $S^i \in R^m$ in S has m variables with indices $\{1, ..., m\}$. These indices can be partitioned into overlapping groups, and each group $g \in \mathcal{G}$ contains a subset of these indices. We define 3×3 overlapping-patch groups similar to [85], and each group overlaps 6 pixels with its neighbors. $\|.\|_\infty$ denotes the maximum value of the pixels in a group.

The foreground is usually spatially contiguous and assumed to occupy a portion of the scene. Therefore, it will be highly appropriate to model the foreground using the structured sparsity norm, because it can reflect the spatial distribution of nonzero variables and thus promote the structural distribution of sparse outliers during the minimization [82]. To formalize the structured sparsity on the outlier S, we propose the structured LISD method by replacing $\|S\|_1$ with $\Gamma(S)$ and so (5.4) can be converted to (6.2), as follows.

$$\begin{gathered} \min_{L,S,C} \|L\|_* + \lambda(\Gamma(S) + \|C\|_1) + \gamma\Psi(S, C, \Phi) \\ s.t. \;\; D = L + S + C \end{gathered} \tag{6.2}$$

Solving (6.2) enables us to detect moving objects and real changes as spatial contiguous objects and separate them from illumination changes through optimization.

Sequential LISD: One limitation of LISD is that the method works in a batch optimization framework. As discussed in Chapter 3, batch methods suffer from large image sequences and are not able to apply on real time applications. To deal with this issue, we are interested in solving our LISD method in a sequential manner. LISD has two main steps. First we need to obtain illumination invariant representation of images and then solve the optimization framework. The illumination invariant representation of each image is computed sequentially for each image independently. The low-rank and sparse decomposition part of LISD also can be solved sequentially with the help of OR-PCA method [35]. Also, the structured sparsity term $\Gamma(S)$ is a cumulative function over all images and works for each image independently. Therefore, we are interested in investigating a sequential structured LISD method with the help of OR-PCA, which enables us to use it in an online surveillance system under sudden and discontinuous illumination changes.

- Our proposed TLISD method in Chapter 5 also can be improved using structured sparsity. Recently, methods using spatio-temporal structured sparsity have been proposed to detect moving objects in the frame-rate image sequences [62]. Since moving objects are spatially connected components and their locations are temporally correlated in the frame-rate image sequences, spatio-temporal constraint improves the quality of detection, even in noisy environments. However, in the case of time-lapse video sequences, which are of interest in this thesis, the spatio-temporal constraint does not work in the form of matrix decomposition due to discontinuous change in object location. In our tensor-based method with the proposed tensor structure, we have useful information about the location of objects in each lateral slice, which can help us to take advantage of spatio-temporal structured sparsity constraint. Since each lateral slice in the proposed tensor $\mathcal{D}$ includes the original and all invariant representations of one image, all moving objects should be highly correlated in the third dimension. Therefore, we can treat each lateral slice of tensor $\mathcal{D}$ as a 2D matrix and so the spatio-temporal structured sparsity is perfectly fit on it to detect moving objects.

- Inspired by the significant success of deep neural networks in computer vision, recently, many background subtraction and moving object detection methods based on deep neural network have been proposed [3], [15], [27], [99], [144], [147]. However, these learning-based methods need supervised training with pixel-wise ground-truth masks of moving objects, which are not practical in real applications. Due to this fact, unsupervised neural networks like autoencoders are used to tackle the problem of moving object detection [4], [141]. Although [4] proposed a method following our LISD method to handle illumination variation in frame-rate sequences, it still suffers from discontinuous changes in illumination.

In general, outdoor illumination variation labeled data is not available, and therefore all methods only learn the background and its variations. So, in the case of discontinuous change in illumination, they still are not able to distinguish between changes caused by illumination and those caused by moving objects in the scene.

Since our TLISD method is able to extract natural outdoor illumination, we can use them as labeled data for learning-based methods to learn outdoor illumination variations, which can be applied as priors in unsupervised neural network. This approach may be an effective alternative to optimization based methods such as ours, but with a sequential formulation, to detect illumination changes and moving objects from image sequences.

References

[1] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the k-support norm," in *Advances in Neural Information Processing Systems*, 2012, pp. 1457–1465. 80, 84, 89

[2] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. 82

[3] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, pp. 635–649, 2018. 5, 111

[4] F. Bahri, M. Shakeri, and N. Ray, "Online illumination invariant moving object detection by generative neural network," *arXiv preprint arXiv:1808.01066*, 2018. 112

[5] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, 2011. 1

[6] H. Barrow and J Tenenbaum, "Recovering intrinsic scene characteristics," *Comput. Vis. Syst., A Hanson & E. Riseman (Eds.)*, pp. 3–26, 1978. 18, 49

[7] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, IEEE, vol. 2, 2001, pp. 383–390. 18

[8] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11, pp. 31–66, 2014. 1

[9] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust pca in image and video processing," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1427–1457, 2018. 13

[10] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. 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*, vol. 23, pp. 1–71, 2017. 9

[11] T. Bouwmans and E. H. Zahzah, “Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance,” *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014. 2, 9

[12] S. Boyd, “Alternating direction method of multipliers,” in *Talk at NIPS workshop on optimization and machine learning*, 2011. 14

[13] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 1124–1137, 2004. 30

[14] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001. 29

[15] M. Braham and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks,” in *IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava 23-25 May 2016*, IEEE, 2016, pp. 1–4. 5, 111

[16] K. Braman, “Third-order tensors as linear operators on a space of matrices,” *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010. 12

[17] E. O. Brigham and E. O. Brigham, *The fast Fourier transform and its applications*. prentice Hall Englewood Cliffs, NJ, 1988, vol. 448. 87

[18] S. Brutzer, B. Höferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 1937–1944. 32

[19] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, “Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2488–2495. 13

[20] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. 63

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011. 2, 9, 10, 15, 69, 96

[22] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, and Z. Xu, “Total variation regularized tensor rpca for background subtraction from compressive measurements,” *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4075–4090, 2016. 15

[23] G. Chau and P. Rodriguez, “Panning and jitter invariant incremental principal component pursuit for video background modeling,” in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2017, pp. 1844–1852. 22

[24] B.-H. Chen and S.-C. Huang, “An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks,” *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 837–847, 2014. 1

[25] C. Chen, J. Cai, W. Lin, and G. Shi, “Incremental low-rank and sparse decomposition for compressing videos captured by fixed cameras,” *Journal of Visual Communication and Image Representation*, vol. 26, pp. 338–348, 2015. 13

[26] L.-H. Chen, Y.-H. Yang, C.-S. Chen, and M.-Y. Cheng, “Illumination invariant feature extraction based on natural images statistics—taking face images as an example,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 681–688. 48–51

[27] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, “Pixel-wise deep sequence learning for moving object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 5, 111

[28] P. Corke, R. Paul, W. Churchill, and P. Newman, “Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 2085–2092. 18, 53–56

[29] D. Cremers and S. Soatto, “Motion competition: A variational approach to piecewise parametric motion segmentation,” *International Journal of Computer Vision*, vol. 62, no. 3, pp. 249–265, 2005. 1

[30] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas, “Background subtraction using low rank and group sparsity constraints,” in *European Conference on Computer Vision (ECCV)*, Springer, 2012, pp. 612–625. 15

[31] Z. Ding and Y. Fu, “Robust multi-view subspace learning through dual low-rank decompositions.,” in *AAAI*, 2016, pp. 1181–1187. 9

[32] S. E. Ebadi and E. Izquierdo, “Foreground segmentation with tree-structured sparse rpca,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 9, pp. 2273–2280, 2018. 16, 17

[33] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *European conference on computer vision*, Springer, 2000, pp. 751–767. 1

[34] J. Feng, H. Xu, and S. Yan, “Online robust pca via stochastic optimization,” in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 404–412. 13, 14, 22, 23, 25, 27

[35] J. Feng, H. Xu, and S. Yan, “Online robust pca via stochastic optimization,” in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 404–412. 111

[36] G. D. Finlayson, M. S. Drew, and C. Lu, “Entropy minimization for shadow removal,” *International Journal of Computer Vision (IJCV)*, vol. 85, no. 1, pp. 35–57, 2009. 18–20, 47, 48, 52, 53, 80

[37] G. D. Finlayson, M. S. Drew, and C. Lu, “Intrinsic images by entropy minimization,” in *European conference on computer vision*, Springer, 2004, pp. 582–595. 18

[38] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, “On the removal of shadows from images,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 59–68, 2006. 18, 47

[39] H. Foroughi, M. Shakeri, N. Ray, and H. Zhang, “Joint feature selection with low-rank dictionary learning.,” in *BMVC*, 2015, pp. 97–1. 9

[40] H. Foroughi, M. Shakeri, N. Ray, and H. Zhang, “Face recognition using multi-modal low-rank dictionary learning,” in *Image Processing (ICIP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1082–1086. 4, 57

[41] S. Friedland and V. Tammali, “Low-rank approximation of tensors,” in *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, Springer, 2015, pp. 377–411. 11

[42] Y. Fu and W. Dong, “3d magnetic resonance image denoising using low-rank tensor approximation,” *Neurocomputing*, vol. 195, pp. 30–39, 2016. 11

[43] S. Gandy, B. Recht, and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” *Inverse Problems*, vol. 27, no. 2, p. 025 010, 2011. 11

[44] Z. Gao, L.-F. Cheong, and Y.-X. Wang, “Block-sparse rpca for salient motion detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 10, pp. 1975–1987, 2014. 16

[45] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014. 11

[46] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changedetection. net: A new change detection benchmark dataset,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on*, IEEE, 2012, pp. 1–8. 36, 67, 90

[47] R. Guo, Q. Dai, and D. Hoiem, “Single-image shadow detection and removal using paired regions,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 2033–2040. 18, 47

[48] C. Guyon, T. Bouwmans, and E.-H. Zahzah, “Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint,” in *Asian Conference on Computer Vision*, Springer, 2012, pp. 315–320. 15

[49] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, IEEE, 2012, pp. 1225–1228. 15

[50] M. H. Hayes, *Statistical digital signal processing and modeling.* John Wiley and Sons, 2009. 48

[51] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 1568–1575. 13, 96

[52] J. He, D. Zhang, L. Balzano, and T. Tao, "Iterative grassmannian optimization for robust image alignment," *Image and Vision Computing*, vol. 32, no. 10, pp. 800–813, 2014. 13, 14

[53] Z. He, L. Liu, S. Zhou, and Y. Shen, "Learning group-based sparse and low-rank representation for hyperspectral image classification," *Pattern Recognition*, vol. 60, pp. 1041–1056, 2016. 9

[54] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 657–662, 2006. 2

[55] S. Hosseinzadeh, M. Shakeri, and H. Zhang, "Fast shadow detection from a single image using a patched convolutional neural network," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3124–3129. 18

[56] J. Hou, L.-P. Chau, N. Magnenat-Thalmann, and Y. He, "Sparse low-rank matrix approximation for data compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1043–1054, 2017. 9

[57] W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 2, pp. 724–737, 2017. 15

[58] G.-H. Huang and C.-R. Huang, "Binary invariant cross color descriptor using galaxy sampling," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, 2012, pp. 2610–2613. 2

[59] *Illumination change dataset*, `http://webdocs.cs.ualberta.ca/~shakeri/pub.htm`. 77

[60] S. Javed, T. Bouwmans, and S. K. Jung, "Sbmi-ltd: Stationary background model initialization based on low-rank tensor decomposition," in *Proceedings of the Symposium on Applied Computing*, ACM, 2017, pp. 195–200. 15

[61] S. Javed, S. Ho Oh, A. Sobral, T. Bouwmans, and S. Ki Jung, "Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 90–98. 22

[62] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. K. Jung, "Moving object detection in complex scene using spatiotemporal structured-sparse rpca," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 1007–1022, 2019. 22, 111

[63] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung, "Or-pca with mrf for robust foreground detection in highly dynamic backgrounds," in *Asian Conference on Computer Vision*, Springer, 2014, pp. 284–299. 14

[64] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," 2010. 9

[65] K. Jia, T.-H. Chan, and Y. Ma, "Robust and practical face recognition via structured sparsity," in *European conference on computer vision*, Springer, 2012, pp. 331–344. 110

[66] K. KAUST, "A batch-incremental video background estimation model using weighted low-rank approximation of matrices," 2017. 22

[67] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013. 11, 12

[68] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011. 12

[69] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity.," in *ICML*, vol. 2, 2010, p. 1. 16

[70] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009. 11

[71] V. Kolmogorov and R. Zabih, "What energy functions can be minimizedvia graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 147–159, 2004. 29

[72] H. Lai, Y. Pan, C. Lu, Y. Tang, and S. Yan, "Efficient k-support matrix pursuit," in *European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 617–631. 80, 84–86, 88, 89

[73] J. Li, X. Chen, D. Zou, B. Gao, and W. Teng, "Conformal and low-rank sparse representation for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 235–243. 9

[74] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 11, pp. 1459–1472, 2004. 36, 67, 90

[75] P. Li, J. Feng, X. Jin, L. Zhang, X. Xu, and S. Yan, "Online robust low-rank tensor modeling for streaming data analysis," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–15, 2018. 15

[76] S. Z. Li, *Markov random field modeling in image analysis*. Springer Science and Business Media, 2009. 16, 23, 27, 28

[77] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010. 9–11, 60, 62, 89

[78] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in neural information processing systems*, 2011, pp. 612–620. 63, 66

[79] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 171–184, 2013. 9

[80] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2013. 11

[81] X. Liu, Z. Yang, J. Wang, J. Liu, K. Zhang, and W. Hu, "Patch-based denoising method using low-rank technique and targeted database for optical coherence tomography image," *Journal of Medical Imaging*, vol. 4, no. 1, p. 014 002, 2017. 9

[82] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Transactions on Image Processing (TIP)*, vol. 24, no. 8, pp. 2502–2514, 2015. 16, 69, 96, 110

[83] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5249–5257. 12, 86–88, 96

[84] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2586–2593. 9

[85] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski, "Network flow algorithms for structured sparsity," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 1558–1566. 16, 110

[86] C. D. Martin, R. Shafer, and B. LaRue, “An order-p tensor factorization with applications in imaging,” *SIAM Journal on Scientific Computing*, vol. 35, no. 1, A474–A490, 2013. 11

[87] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010. 23, 27

[88] Y. Nonaka, A. Shimada, H. Nagahara, and R.-i. Taniguchi, “Evaluation report of integrated background modeling based on spatio-temporal features,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, IEEE, 2012, pp. 9–14. 2

[89] P. Ochs and T. Brox, “Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions,” 2011. 1

[90] 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 (PAMI)*, vol. 22, no. 8, pp. 831–843, 2000. 2, 9

[91] Y. Pang, L. Ye, X. Li, and J. Pan, “Incremental learning with saliency map for moving object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 640–651, 2018. 22

[92] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 11, pp. 2233–2246, 2012. 9, 14, 109

[93] M. Piccardi, “Background subtraction techniques: A review,” in *Systems, man and cybernetics, 2004 IEEE international conference on*, IEEE, vol. 4, 2004, pp. 3099–3104. 1

[94] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010. 23

[95] P. Rodriguez and B. Wohlberg, “Incremental principal component pursuit for video background modeling,” *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2016. 13

[96] P. Rodriguez and B. Wohlberg, “Fast principal component pursuit via alternating minimization,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, IEEE, 2013, pp. 69–73. 13

[97] B. Romera-Paredes and M. Pontil, “A new convex relaxation for tensor completion,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2967–2975. 11

[98] D. L. Ruderman, “The statistics of natural images,” *Network: computation in neural systems*, vol. 5, no. 4, pp. 517–548, 1994. 50

[99] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multimedia Tools and Applications*, pp. 1–19, 2017. 5, 111

[100] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, IEEE, vol. 2, 2003, pp. II–II. 2

[101] M Shakeri and H. Zhang, "Illumination invariant representation of natural images for visual place recognition," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 466–472. 4, 57

[102] M. Shakeri and H. Zhang, "Moving object detection under discontinuous change in illumination using tensor low-rank and invariant sparse decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 78

[103] M. Shakeri and H. Zhang, "Moving object detection in time-lapse or motion trigger image sequences using low-rank and invariant sparse decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, 2017, pp. 5123–5131. 4, 78, 90, 96, 103

[104] M. Shakeri and H. Zhang, "Detection of small moving objects using a moving camera," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 2777–2782. 1

[105] M. Shakeri and H. Zhang, "Cooperative targeting: Detection and tracking of small objects with a dual camera system," in *Field and Service Robotics*, Springer, 2015, pp. 351–364. 1

[106] M. Shakeri and H. Zhang, "Corola: A sequential solution to moving object detection using low-rank approximation," *Computer Vision and Image Understanding*, vol. 146, pp. 27–39, 2016. 2, 16

[107] M. Shakeri and H. Deldari, "Fuzzy cellular background subtraction for urban traffic applications," *World Applied Sciences Journal*, vol. 5, 2008. 1

[108] M. Shakeri, H. Deldari, H. Foroughi, A. Saberi, and A. Naseri, "A novel fuzzy background subtraction method based on cellular automata for urban traffic applications," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, IEEE, 2008, pp. 899–902. 1

[109] M. Shakeri, H. Deldari, A. Rezvanian, and H. Foroughi, "A novel fuzzy method to traffic light control based on unidirectional selective cellular automata for urban traffic," in *2008 11th International Conference on Computer and Information Technology*, IEEE, 2008, pp. 300–305. 1

[110] M. Shakeri and H. Zhang, “Object detection using a moving camera under sudden illumination change,” in *Proceedings of the 32nd Chinese Control Conference*, IEEE, 2013, pp. 4001–4006. 1

[111] M. Shakeri and H. Zhang, “Real-time bird detection based on background subtraction,” in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, IEEE, 2012, pp. 4507–4510. 1

[112] J. Shen, P. Li, and H. Xu, “Online low-rank subspace clustering by basis dictionary pursuit,” in *International Conference on Machine Learning*, 2016, pp. 622–631. 9

[113] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens, “Learning with tensors: A framework based on convex optimization and spectral regularization,” *Machine Learning*, vol. 94, no. 3, pp. 303–351, 2014. 11

[114] A. Sobral, C. Baker, T. Bouwmans, and E.-h. Zahzah, “Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction,” in *International Conference Image Analysis and Recognition*, Springer, 2014, pp. 94–103. 15

[115] A. Sobral, T. Bouwmans, and E.-h. Zahzah, “Lrslibrary: 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. 96

[116] I. Stainvas and D. Lowe, “A generative model for separating illumination and reflectance from images,” *Journal of Machine Learning Research*, vol. 4, no. Dec, pp. 1499–1519, 2003. 18

[117] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *cvpr*, IEEE, 1999, p. 2246. 1

[118] G. Tang and A. Nehorai, “Robust principal component analysis based on low-rank and block-sparse matrix decomposition,” in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, IEEE, 2011, pp. 1–5. 15

[119] L. Tao, R. Tompkins, and V. K. Asari, “An illuminance-reflectance model for nonlinear enhancement of color images,” in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, IEEE, 2005, pp. 159–159. 18

[120] M. F. Tappen, W. T. Freeman, and E. H. Adelson, “Recovering intrinsic images from a single image,” in *Advances in neural information processing systems*, 2003, pp. 1367–1374. 18

[121] Y. Tian, A. Senior, and M. Lu, “Robust and efficient foreground analysis in complex surveillance videos,” *Machine vision and applications*, vol. 23, no. 5, pp. 967–983, 2012. 1

[122] R. Tomioka, K. Hayashi, and H. Kashima, “Estimation of low-rank tensors via convex optimization,” *arXiv preprint arXiv:1010.0789*, 2010. 11

[123] A. Torralba and A. Oliva, “Statistics of natural image categories,” *Network: computation in neural systems*, vol. 14, no. 3, pp. 391–412, 2003. 51

[124] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on (ICCV)*, IEEE, vol. 1, 1999, pp. 255–261. 36, 67, 90

[125] R. Tron and R. Vidal, “A benchmark for the comparison of 3-d motion segmentation algorithms,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, IEEE, 2007, pp. 1–8. 41

[126] L. Unzueta, M. Nieto, A. Cortés, J. Barandiaran, O. Otaegui, and P. Sánchez, “Adaptive multicue background subtraction for robust vehicle counting and classification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 527–540, 2012. 1

[127] R. Vidal, “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011. 1

[128] R. Vidal and P. Favaro, “Low rank subspace clustering (lrsc),” *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014. 9

[129] R. Vidal and Y. Ma, “A unified algebraic approach to 2-d and 3-d motion segmentation,” in *European Conference on Computer Vision*, Springer, 2004, pp. 1–15. 1

[130] A. Vinciarelli, M. Pantic, and H. Bourlard, “Social signal processing: Survey of an emerging domain,” *Image and vision computing*, vol. 27, no. 12, pp. 1743–1759, 2009. 1

[131] N. Wang, T. Yao, J. Wang, and D. Yeung, “A probabilistic approach to robust matrix factorization,” in *European Conference on Computer Vision (ECCV)*, Springer, 2012, pp. 126–139. 13–15, 69, 96

[132] N. Wang and D. Yeung, “Bayesian robust matrix factorization for image and video processing,” in *International Conference on Computer Vision (ICCV)*, 2013, pp. 1785–1792. 16, 69, 96

[133] Y.-C. F. Wang, C.-P. Wei, and C.-F. Chen, “Low-rank matrix recovery with structural incoherence for robust face recognition,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2618–2625. 9

[134] Y. Weiss, “Deriving intrinsic images from image sequences,” in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, IEEE, vol. 2, 2001, pp. 68–75. 18

[135] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 780–785, 1997. 1

[136] C. Y. Wu and J. J. Ding, "Occluded face recognition using low-rank regression with generalized gradient direction," *Pattern Recognition*, vol. 80, pp. 256–268, 2018. 9

[137] G. Wyszecki and W. S. Stiles, *Color science*. Wiley New York, 1982, vol. 8. 19

[138] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused lasso foreground modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4676–4684. 16

[139] H. Xu, C. Caramanis, and S. Mannor, "Sparse algorithms are not stable: A no-free-lunch theorem," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 34, no. 1, pp. 187–193, 2012. 85

[140] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh, "Gosus: Grassmannian online subspace updates with structured-sparsity," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3376–3383. 13, 14, 16

[141] P. Xu, M. Ye, X. Li, Q. Liu, Y. Yang, and J. Ding, "Dynamic background learning through deep auto-encoder networks," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 107–116. 112

[142] 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*, vol. 1, no. 1, pp. 43–60, 2016. 1, 2

[143] J. Yang and X. Yuan, "Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization," *Mathematics of computation*, vol. 82, no. 281, pp. 301–329, 2013. 66

[144] L. Yang, J. Li, Y. Luo, Y. Zhao, H. Cheng, and J. Li, "Deep background modeling using fully convolutional network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 254–262, 2018. 5, 111

[145] J.-C. Yoo and T. H. Han, "Fast normalized cross-correlation," *Circuits, systems and signal processing*, vol. 28, no. 6, p. 819, 2009. 55

[146] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 67–76. 9

[147] D. Zeng and M. Zhu, "Background subtraction using multiscale fully convolutional network," *IEEE Access*, vol. 6, pp. 16 010–16 021, 2018. 5, 111

[148] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Low-rank decomposition-based restoration of compressed images via adaptive noise estimation," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 9, pp. 4158–4171, 2016. 9

[149] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 676–683. 9

[150] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3842–3849. 11, 86

[151] A. Zheng, T. Zou, Y. Zhao, B. Jiang, J. Tang, and C. Li, "Background subtraction with multi-scale structured low-rank and sparse factorization," *Neurocomputing*, vol. 328, pp. 113–121, 2019. 17

[152] T Zhou and D Tao, "Greedy bilateral sketch, completion and smoothing for large-scale matrix completion, robust pca and low-rank approximation," *AISTATS 2013*, 2013. 13, 27, 31, 69

[153] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *International conference on machine learning (ICML)*, Omnipress, 2011. 13, 96

[154] X. Zhou, C. Yang, and W. Yu, "Automatic mitral leaflet tracking in echocardiography by outlier detection in the low-rank representation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 972–979. 1

[155] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 3, pp. 597–610, 2013. 16, 21–23, 28, 29, 32, 33

[156] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 36, 2015. 9

[157] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen, "Learning to recognize shadows in monochromatic natural images," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 223–230. 18, 47

[158] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, IEEE, vol. 2, 2004, pp. 28–31. 28, 96