THE USE OF SPARSE PLUS LOW-RANK DECOMPOSITION ON MOVING OBJECT AND CHANGE DETECTION IN VIDEOS

Lucas Arrabal Thomaz

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Sergio Lima Netto
Eduardo Antônio Barros da Silva

Rio de Janeiro
Dezembro de 2018

THE USE OF SPARSE PLUS LOW-RANK DECOMPOSITION ON MOVING OBJECT AND CHANGE DETECTION IN VIDEOS

Lucas Arrabal Thomaz

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Sergio Lima Netto, Ph.D.

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. João Baptista de Oliveira e Souza Filho, D.Sc.

Prof. Lisandro Lovisolo, D.Sc.

Prof. Hae Yong Kim, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2018

Thomaz, Lucas Arrabal

The Use of Sparse Plus Low-Rank Decomposition on Moving Object and Change Detection in Videos/Lucas Arrabal Thomaz. – Rio de Janeiro: UFRJ/COPPE, 2018.

XVIII, 117 p.: il.; 29, 7cm.

Orientadores: Sergio Lima Netto

Eduardo Antônio Barros da Silva

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 94 – 104.

1. Object Detection. 2. Signal Processing. 3. Video Processing. 4. Sparse Decomposition. 5. Change Detection. 6. Moving Camera. I. Netto, Sergio Lima *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

You're going to be alone now, and you're very bad at that. You're going to be furious and you're going to be sad, but listen to me:
Don't let this change you. No, listen.
Whatever happens next, wherever she is sending you,
I know what you're capable of. You don't be a warrior.
Promise me.
Be a Doctor.
- Clara Oswald, Doctor Who

Never be cruel, never be cowardly. And never ever eat pears!
Remember: Hate is always foolish... and love is always wise.
- The Doctor, Doctor Who

Excelsior!

Àqueles que tem medo.
Àqueles que não vêem solução.
Àqueles que fraquejam.
Àqueles que sem esperança.
Àqueles que querem desistir.
Àqueles que ainda vão vencer.

Agradecimentos

Primeiramente gostaria de agradecer a Deus, por tudo que me concedeu até hoje, tendo me permitido chegar até aqui.

Em seguida, agradeço aos meus pais, que ajudaram a formar quem sou hoje. Obrigado pela dedicação, carinho, afeto, amor e atenção. Por sempre apoiar as minhas escolhas e me motivar a buscar sempre mais. Obrigado por se alegrarem comigo, sofrerem comigo e me consolar quando tudo parecia perdido. Também agradeço por me apoiarem quando foi a hora de sair debaixo de suas asas sempre vendo sempre com orgulho meus passos e, apesar de tudo, fazendo com que a transição fosse a melhor possível. Obrigado também à minha irmã por ser a pessoa mais igual e diferente de mim com quem pude sempre dividir essa maravilhosa família e aprender muito.

Agradeço aos meus orientadores, por me apresentarem a vida acadêmica e, com isso, me fazer amá-la. Obrigado pela confiança, por sempre me incentivar a buscar mais e por ensinar, através do exemplo, como ser um bom pesquisador e professor. Agradeço também por me permitirem ao longo desses anos experimentar um pouco do outro lado co-orientando alunos e ministrando aulas, essas experiências foram cruciais para que eu decidisse a carreira que quero seguir.

Special thanks to professor Hamid Krim who was my supervisor during my research internship at North Carolina State University. This work would not be possible without your help and insights during our meetings. I'd also like to thank my colleagues at NCSU, I was very lucky to spend this year with you.

Agradeço também aos outros professores que ajudaram na minha formação e à UFRJ onde estudei tantos anos e pude crescer pessoal e profissionalmente.

Agradeço ao restante da minha família e padrinhos, sempre muito presentes e próximos que foram fundamentais para a minha formação e crescimento até aqui. Obrigado por dividirem tantos momentos comigo e celebrarem tão intensamente cada conquista nessa jornada.

Agradeço aos meus amigos, aqueles que conheci dentro da universidade e que não importando o quão longe estiverem farão sempre parte da minha vida. Obrigado por terem me ensinado tanto e por me deixarem compartilhar o pouco que sei. Dividimos muitas alegrias, frustrações e preocupações, mas no fim tudo deu certo. Mais uma vez, agradeço em especial àqueles que participaram ativamente da desse trabalho em

qualquer etapa, seja com dicas de simulação, resolvendo problemas intratáveis no meio da noite ou ajudando com ideias, revisões, simulações e derivações de equações.

Agradeço também aos amigos que a vida me deu de diversas formas. Obrigado pelas conversas, jogos, viagens e companheirismo. Por estarem sempre disponíveis para conversar ou simplemente caminhar pela cidade sem dizer nada. Agradeço por sentarem de qualquer lado de um balcão e discutirem sobre a vida tomando uma xícara de café ou por passarem um tempo comendo um hambúrguer. E principalmente obrigado por não deixar que eu me afastasse mesmo que muitas milhas distante.

Very special thanks to the friends I first met at Event Horizon Games, you were my family away from home and made my stay in NC one of the best times of my life. My time in the US would be impossible without you. From my first weekend there you welcomed me in the group and made me feel at home and sane even when I was homesick missing Brazil. Thanks for everything!

Agradeço ao CNPq e à CAPES pelo financiamento através das bolsas de estudo e de doutorado sanduíche. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.)

Por fim, agradeço a todos que contribuíram de alguma forma com esse trabalho ou comigo ao longo desses anos.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

O USO DE DECOMPOSIÇÕES EM MATRIZES ESPARSAS E DE BAIXO POSTO EM DETECÇÃO DE OBJETOS EM MOVIMENTO E DETECÇÃO DE MUDANÇAS EM VIDEOS

Lucas Arrabal Thomaz

Dezembro/2018

Orientadores: Sergio Lima Netto
Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Uma solução para a detecção de anomalias, como a identificação de mudanças e objetos em movimento em vídeos, é buscar uma representação de baixo posto para os quadros que compõem o vídeo e reconstruir um dos quadros do vídeo original através da combinação de representações de baixo posto dos outros quadros. Esta tese propõe algoritmos que projetam as estruturas de baixo posto em uma união de subespaçoes de baixa dimensão, para solucionar esse problema sendo capaz de lidar com contextos dinâmicos, como aqueles encontrados em vídeos adquiridos com câmeras em movimento, dentre outros cenários complexos.

Parte desta tese busca detectar mudanças em vídeos obtidos com câmeras móveis. Os algoritmos propostos apresentam bons resultados, enquanto removem a restrição de sincronização prévia dos vídeos. Adicionalmente, eles utilizam propriedades da estrutura dos dados para restringir o espaço de busca para um número menor de subespaços, obtendo ganhos computacionais de até 100 vezes além de 91% de verdadeiros positivos e somente 33% de falsos positivos, em experimentos utilizando a base de dados VDAO, com objetos abandonados em um cenário industrial.

Outra parte apresenta soluções para a detecção de objetos em movimento em vídeos com cenário dinâmico. As soluções propostas utilizam decomposições em matrizes de baixo posto e esparsas com projeções em uma união de subespaços, aplicando mapas de saliência para restringir as atualizações de matrizes que representam os objetos. Os métodos apresentados têm baixa incidência de falsos positivos e apresentam desempenhos comparáveis aos do estado da arte para métodos similares, obtendo 0.74 de F1 na base UCSD.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

THE USE OF SPARSE PLUS LOW-RANK DECOMPOSITION ON MOVING OBJECT AND CHANGE DETECTION IN VIDEOS

Lucas Arrabal Thomaz

December/2018

Advisors: Sergio Lima Netto
Eduardo Antônio Barros da Silva

Department: Electrical Engineering

A solution for the detecion of anomalies, such as change and moving objects in videos, is to obtain a low-rank representation of the frames that compose the video sequence and try to reconstruct a frame from the original video using a combination of the low-rank representation of the others. This thesis propose algorithms that project the low-rank structures into a low-dimensional union-of-subspaces, to solve this problem allowing the model to cope with dynamic backgrounds such as those found in videos acquired with moving cameras and other complex scenarios.

Part of the thesis covers change detection in videos acquired with moving cameras. The proposed algorithms provide good detection results, at the same time as obviate the need for previous video synchronization. They also use properties of the data representation in order to restrict the search space to the most relevant subspaces, providing computational complexity gains of up to 100 times and 91% true positive and only 33% false positive detections on experiments using the VDAO database, with abandoned objects in a cluttered industrial scenario.

Another part presents a solution to the detection of moving objects in the presence of highly dynamic backgrounds. The proposed solutions use low-rank and sparse matrix decompositions to represent the background as a union-of-subspaces, while applying saliency maps to restrict the updates of the foreground matrix. The proposed methods presents low false positive detection rate, and is shown to achieve state-of-the-art performance among similar methods, attaining 0.74 F1 score in the UCSD dataset.

Contents

List of Figures

List of Tables

List of Abbreviations

3TD	Three-term Decomposition Model, p. 84
ADMM	Alternating Direction Method of Multipliers, p. 21
ADMULT	Anomaly Detector Using Multiscales, p. 51
ALM	Augmented Lagrangian Method, p. 21
AMBER	Adapting Multi-resolution Background Extractor, p. 68
BMS	Boolean Map based Saliency model, p. 64
CDNET	changeDetection.net, p. 15
CNN	Convolutional Neural Network, p. 66
CwisarDRP	Wikes, Stonham and Aleksander Recognition Device, p. 68
DAOMC	Detection of Abandoned objects with a Moving Camera, p. 50
DSMR	Deformed Smoothness-based Manifold Ranking, p. 65
FN	False Negative, p. 52
FP	False Positive, p. 52
FoV	Field-of-View, p. 28
GBVS	Graph-Based Saliency, p. 63
GD	Gradient Descent, p. 9
GMM	Gaussian mixture model, p. 3
GOSUS	Grassmannian Online Subspace Updates with Structured-sparsity, p. 9
GPS	Global Positioning System, p. 6
IUTIS	In Unit There is Strength, p. 67

LADM	Linearized Alternating Direction Method, p. 21
Lag-PCP	Lagrangian-Stable PCP, p. 84
MAGR-RPCA	Motion-Aware Graph Regularized RPCA, p. 10
MCBS	Moving-Camera Background Subtraction, p. 51
MLP	Multi-Layer Perceptron, p. 66
NCC	Normalized Cross-Correlation, p. 7
NVD	Normalized Vector Distance, p. 52
PAWCS	Pixel-based Adaptive Word Consensus Segmenter, p. 67
PCA	Principal Component Analysis, p. 4
PCA	Robust Principal Component Analysis, p. 4
PCP	Principal Component Pursuit, p. 9
PETS	Performance Evaluation of Tracking and Surveillance, p. 15
PSA	Principal Subspace Analysis, p. 14
PSE	Probabilistic Saliency Estimation, p. 65
PTZ	Pan-Tilt-Zoom, p. 15
RMAMR	Motion-Assisted Matrix Restauration, p. 84
RMAMR	Robust Motion-Assisted Matrix Restoration, p. 62
ROI	Region-of-Interest, p. 7
RRF	Radial Reach Filter, p. 52
RanSaC	Random Sampling Consensus, p. 7
RoSuRe	Robust Subspace Recovery, p. 19
SCM-RPCA	Shape and Confidence Map-based RPCA, p. 62
SIFT	Scale-Invariant Feature Transform, p. 7
SMD	Singapore Maritime Dataset, p. 60
SNR	Signal-to-Noise Ratio, p. 22

SRPCA	Spatiotemporal Robust Principal Component Analysis, p. 10
STC-mc	Spatio-Temporal Composition for Moving-Camera Detection, p. 51
SVD	Singular Value Decomposition, p. 8
SWCD	Sliding Window-based Change Detection, p. 68
TN	True Negative, p. 52
TP	True Positive, p. 52
UCSD	University of California San Diego Background Subtraction Dataset, p. 58
UoS	Union of Subspaces, p. 19
VDAO	Video Database for Abandoned Objects, p. 15
WNN	Weightless Neural Network, p. 68
i-LIDS	Imagery Library for Intelligent Detection Systems, p. 15
mcRoSuRe-TA	mcRoSuRe-Time Alignment, p. 40
mcRoSuRe	moving camera RoSuRe, p. 23

Chapter 1

Introduction

This thesis presents an investigation on the use of low-rank plus sparse matrix decompositions to perform anomaly detection in complex video sequences. There are two main focuses in this work: first the change detection in videos acquired with moving cameras in the presence of cluttered backgrounds; second the detection of moving foreground in videos featuring complex moving backgrounds.

1.1 Anomaly Detection in Videos

In recent years there has been a great increase in the amount of video cameras that are used for surveillance and monitoring systems. Most of that increase is due to the advances in imagery technology and the decrease in the price of such equipment. Nowadays, almost every public place has at least a single fixed color camera monitoring an area with a large field-of-view. In private facilities, as factories and industrial plants, the number of such devices is indeed much larger, as the field of view of the cameras tends to be smaller, focusing in a particular spot of the place, and several cameras are usually placed in the same region. To give an idea of the increasing size of this market according to a report from Transparency Market Research from 2016 [1], the video surveillance equipment market worldwide is expected to reach U$42,810 billions by 2019.

All this surveillance infrastructure generates a huge amount of video data that has to be analyzed if one wants to extract useful information from it. Most likely humans will not be able to analyze properly all this data, since there are too many video streams and most of these systems work without interruption. Therefore, an alternative way to extract the important information from those videos in an automatic and reliable manner is of great interest.

In the fields of signal processing and computer vision the detection of anomalies in video, sequences is a well-known challenging problem. Several algorithms have been developed to extract and process the significant information from the video

stream. The automatic surveillance systems present solutions for a wide range of applications such as personal identification [2], object recognition [3, 4], moving object detection and tracking [5], and abandoned object detection [6].

The detection of video anomalies is an important subject as the anomalies can be of various kinds, therefore its application spreads to: detection of moving objects [7] (e.g. cars, people, animals), detection of abandoned objects [8] (e.g. unexpected objects in security areas, forgotten apparel in industrial plants), detection of removed objects [9] (e.g. stolen objects from museums and public areas), or even landscape change detection [10] (e.g. buildings facade change or street graffiti detection).

This section reviews of several methods in the literature that are designed to detect anomalies in videos.

When one thinks about how to detect objects using video streams, the most simple idea that comes to mind is simply to subtract consecutive frames of the video and try to detect when and where there were significant differences between those frames. That strategy, although very simple, does not work properly, as explained in [11]. Several factors as camera jitter, wind, illumination changes, weather variations, and noise create problems in the detection that may lead to false negative (when no anomaly is detected but it is present in the scene) or false positive detections (when an anomaly is detected but in fact there is none in the scene).

Instead of using the simple (and not robust) consecutive frame subtraction, most methods designed to detect video anomalies rely on some form of background modeling or background subtraction. That is, they somehow try to create a modified version of the image without anomalies that is not subject to any of the problems that were listed before and then subtract it from the frame to analyze the residue and detect whether there are anomalies in the present frame.

The following review of the methods in the literature will be divided in two segments: methods that use fixed cameras and those that perform the detection from videos with moving cameras. A thorough review of many methods designed to detect anomalies is available in [12].

1.1.1 Fixed Camera Algorithms

One of the most straightforward strategies for detecting abandoned objects using background subtraction is presented in [13]. In this work a reference background model is generated from frames that were labeled as having no objects of interest and no movement is present in the scene. Then, a buffer of several consecutive frames is created to model the dynamic background of the scene and create a current background model. A dual background subtraction is performed by using the reference background and the current one. This dual background step is responsible

for making the algorithm robust to slow illumination changes and smooth modifications of the background. Finally if there were potential object detections in the frame, a temporal consistency (the object must be detection throughout several consecutive frames) of the object is required for the system to trigger an abandoned object alarm.

A different approach is presented in [14]. In this work several reference images (where no anomalies are present) are used to create a model of background by using a Gaussian distribution in the RGB color space of every pixel. After that, a background model is composed for every pixel in every new frame of the video stream, if the pixel value does not match the background model (the verification is made by using a Mahalanobis distance), an alarm is triggered for the pixel, else if the pixel value match the background model, the model is updated to cope with the new frame statistics.

To detect moving objects in a video, the authors of [15] present a simpler version of the previous method by substituting the Gaussian model of the background by a simpler median-filtering model. This simpler method is said to reduce the amount of computation needed to perform the calculation with respect to the previous Gaussian model. Also, this method uses a classifier to judge whereas the detected anomaly is a moving object, a shadow, or a ghost (series of connected points detected as motion by the background subtraction algorithm that does not correspond to real motion of the video). By classifying the detected anomalies in these categories, this algorithm is able to be robust against noise and illumination challenges in the video stream.

In the opposite sense, by using more complex yet more robust techniques, the authors of [16] are able to detect stationary anomalies in scenes while being robust to small variations in the background (as tree branches movement, water surface reflections, and small flags in the wind). The algorithm uses a more comprehensive background model than [14] by applying a Gaussian mixture model (GMM) to generate the background model. This method allows every pixel to be modeled as a combination of several Gaussian distributions and if the new pixel matches the model, the model is updated, whereas if it does not match, an alarm is triggered for the pixel being processed. The modeling of the background with a GMM enables the method to cope with small variations of lighting and small movements in the background, making this method robust to such slight variations in the background.

Another technique that is commonly used in the context of anomaly detection of video sequences the optical flow [17] analysis. The optical flow allows one to observe the coherent motion of points between consecutive frames. In [18] this method is used to extract information concerning the presence of abandoned objects in crowded scenes. To detect the presence of the object, a background model is generated and a background subtraction is performed to detect blobs that may contain abandoned

objects. Later, as the goal of this algorithm is to detect unattended baggages, a k-nearest neighbours classifier is used to classify the anomaly as being or not a bag. In the case of positive detection the scene is analysed using optical flow to find the original owner of the baggage by following the evolution of the coherent motion of the scene.

Completely different approaches can be used to detect anomalies by classifying if something is part of the background or an entity of the scene foreground. One case of neural networks being used for this purpose is presented in [19]. In this work the neural network is trained to classify every pixel as foreground or background. This approach needs previous training and is very susceptible to illumination changes and noise in the video. A different neural-network approach has been developed with the advances on deep learning. In [20] a LeNet-5 [21] deep network is used to compare frames from two different videos and detect the presence of foreground anomalies.

A distinct way to deal with the issue of detecting some kind of anomaly in video streams is to assume that everything in the scene can be statistically predicted from previous frames. Therefore everything that is not predictable in this way can be considered an anomaly. One work that presents this kind of algorithm is detailed in [22]. The aim of the work is to detect the presence of abandoned objects for security purposes. The algorithm divides the image in blocks and try to decompose them using the previous blocks of the video: if a block cannot be decomposed using the previous ones it is considered as having an anomaly. After selecting the anomalous blocks the algorithm classifies them to check if there are any security threats in the environment. A similar approach is presented in [23] where the algorithm divides the image in blocks and detects movement in each block by performing an incremental PCA (principal component analysis) [24] decomposition over a sequence of blocks. After the motion detection, a region of interest in the frame is selected as the blocks containing movement. The blocks that form the region of interest are then processed to classify the type of anomaly using a support vector machine classifier. There are even other methods [25] that use techniques based on the robust PCA method (RPCA) [26] to not only identify similar images, as proposed in the original paper, but also to perform some sort of background subtraction and therefore detect anomalies in the video stream.

Table 1.1 summarizes the main techniques employed by each fixed camera method discussed in this section.

1.1.2 Moving Camera Algorithms

Although the use of automatic anomaly detection using fixed cameras usually yields good results, as can be seen by inspection the results reported in the reviewed works

Table 1.1: Fixed camera algorithms and the techniques employed in them.

Method's Reference	Main Technique
WAHYONO [13]	Dual background subtraction
GALLEGO [14]	Gaussian model
CUCCHIARA [15]	Media-filtering model
HASSAN [16]	Gaussian-mixture model
BHARGAVA [18]	Optical flow
MADDALENA [19]	Shallow neural network
BRAHAM [20]	LeNet deep neural network
CHANG [22]	Statistic foreground dtection
MIEZIANKO [23]	PCA
GUYON [25]	RPCA

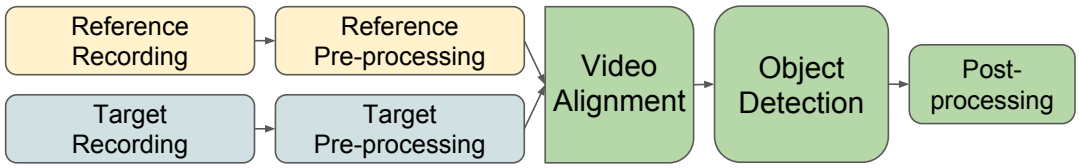


Figure 1.1: Traditional framework of abandoned object detection using moving cameras.

above, in some surveillance applications the equipment needed to perform the evaluation of all the desired characteristics of the environment may be too expensive to be attached in a single fixed position and overlook a single section of the environment. In this case a possible solution is to attach this equipment to a mobile platform that has some freedom of movement and therefore can cover a greater area.

The use of such moving cameras tends to increase due to the popularization of moving platforms (e.g. robots, cars, and drones) that perform the surveillance of large areas employing several sensors (e.g. gases, radiation, etc), and cannot be installed in fixed positions [27–30].

Most methods present a similar framework to cope with the challenge of detecting video anomalies while using the video stream from a moving camera. Even if some variations may exist between two different methods, the general outline of an anomaly detection algorithm using moving cameras consists in recording a reference video (that is certified to have no anomalies), recording a target video (where there may be anomalies), pre-processing both videos to comply with the method requirements, aligning both videos, performing the detection, and post-processing the results for better performance, as can be seen in Figure 1.1.

Video Alignment

An important difference between fixed and moving camera methods for detecting abandoned objects is the presence of two different video streams (reference and target) that are obtained at different moments. Also, due to the existence of more than one video, it is now necessary to align both sequences, as can be seen in the block-diagram in Figure 1.1.

Some works have addressed the task of aligning videos for anomaly detection. Reference [31] presents a method that aims to align videos obtained from cameras with equal intrinsic parameters by implementing a divide-and-conquer strategy to align the videos via the matching of each sequence optical flow. Also, to avoid later problems while performing the detection, the method proposes the use of a geometric registration technique with a geometric transformation between the frames using the fundamental matrix [32] of every frame from both video streams.

A different approach for video synchronization was proposed in [33], where an image descriptor is used instead of the pixel values of the frames of each video stream, resulting in less data points. This is proposed because much of the information is not relevant for the alignment of the video, therefore this method is able to perform it with much less computation. The method then performs a similarity measurement with the descriptor of every frame between both videos and align them this way. Also this method uses external sensors to check the correctness of the proposed alignment. In this case the external signal is a global positioning system (GPS) signal that is available along with the video sequences in the dataset used.

Another video alignment algorithm was recently proposed in [34]. It deals with the alignment of video sequences recorded in different times by moving cameras that follow a similar path while detecting anomalies. As in the previous work, in the dataset used to develop this method other sensors are available along with the video stream. Therefore, to speed up the alignment step, the video stream is not used to compute the alignment. Instead, an array of diverse sensors that is synchronously recorded with the video streams is used to compute a similarity measurement between the signals from reference and target video and then find the alignment between both videos.

More recently in [35] the authors propose a video alignment algorithm that uses a sequential implementation of the dynamic time-warping algorithm [36] to find time correspondences between the video stream of reference and target videos. Differently from the previous algorithms, this one takes into consideration differences in the speed of the cameras of the two videos allowing the method to align videos in which the camera stopped or even changed the direction with much better precision than the previous proposals. Similarly, this implementation is robust to challenging videos

with cluttered background and the presence of noise.

Detection Algorithms

Some solutions have been proposed in the last few years to cope with the problem of detecting anomalies while using videos recorded with moving cameras. A notable attempt to solve the moving-camera anomaly-detection problem was proposed in [6]. In this work a camera mounted on a car searches for abandoned objects on streets. To do so an algorithm similar to that in [33] was used to align the reference and target videos using the GPS signal as an external cue. The frames in this method were geometrically registered using the Random Sampling Consensus (RanSaC) algorithm [32] on Scale-Invariant Feature Transform (SIFT) descriptors [37]. Also, to detect the abandoned objects, the registered frames were compared by computing the Normalized Cross-Correlation (NCC) between the reference and target frames. Despite the method's good performance, the need for an external signal to align reference and target videos limits its usefulness.

The algorithm developed in [11, 38, 39] is able to detect abandoned objects in a heavily cluttered environment in real-time. Video synchronization is performed without the use of any external sensor other than the camera by taking advantage of the a priori knowledge of the camera's linear back-and-forth trajectory. The real-time applicability of this method makes it one of a kind. However, similarly to the method presented in [6] the algorithm's efficiency is also dependent on the correct setup of the NCC window size. Furthermore, the requirement of a specific type of camera movement to perform the video synchronization limits the algorithm applicability in the case of a more general surveillance scenario.

In another recent approach [40], a camera mounted on a train is used to detect the presence of objects across the train path. The alignment and geometric registration techniques (referred to as DeepFlow [41]) used on this method are based on the matching of features extracted with a deep convolutional neural network. This algorithm uses the location of the rails to select the region-of-interest (ROI) in the frame where the algorithm has to search for the anomalous entities, thus avoiding excessive false detections. This method has good performance in the scenario for which it was designed to operate, but has high computational cost due to the DeepFlow-based video alignment. It is also hard to generalize to other surveillance configurations.

More recently, a two-stage dictionary learning approach [42] has been proposed for the analysis of video sequences. It dispenses with the need of motion estimation, tracking or background subtraction. The resulting system considers as anomalies portions of video that are poorly represented by the dictionary. Thanks to the use of a dictionary to represent the target-video images, and unlike most of previous and

existing approaches, this algorithm requires neither temporal nor geometric video alignment. The dictionary construction, however, imposes a latency to the system that may not be always tolerable.

Yet another method that works in a cluttered environment and is able to detect anomalies in such scenarios is presented in [8, 43]. This work presents a method to detect anomalies in videos without the use of any geometric registration in advance and also requiring only that the reference and target videos are roughly aligned. To do so the algorithm uses a technique developed in [44] to transform reference and target videos into a different space and computes a similarity metric between the frames in that space to detect if there are anomalies in the target frame. To cope with the challenge of alignment and registration a hierarchical method is proposed to test some different alignments and rotations between reference and target frames.

A considerably different approach is presented in [45]. In it, a method that performs a sort background subtraction is proposed by exploring the sparsity of dense trajectory representations. In this work dense trajectory maps (e.g.: optical flow) are extracted from a moving-camera video and the matrices that compose this dense trajectories are decomposed using a singular value decomposition (SVD). By exploiting the sparsity of the SVD representations of the dense trajectories and knowing some application priors it is possible to detect the presence of independent moving objets in the videos.

In the same line of thinking as the previously presented work, the solution proposed in [46] uses sparse decomposition of matrices to detect the presence of anomalies in the video stream. This work decomposes every frame of the reference video in a low-rank representation and uses this representation to try to represent the frames from the target stream. By doing so, the algorithm is able to represent the parts of the target frames that are similar to that of the reference, and is not able to properly represent the region of the frames where there are anomalies. Therefore the anomalies would be identified as the residue of the low-rank representation of the target video frames. The method uses a technique proposed in [47, 48] to create the low-rank representation of the frames and modifies it to perform the target-frame decompositions.

With the ever growing applications of deep neural networks some solutions using those techniques have surfaced in the past few years. In [49] the authors propose a background subtraction approach based on the application of the deep learning framework of residual networks using reference and target videos. Although the method employs several image downsamplings along the network, it is able to output an detection mask with the same dimensions of the original images by employing different reconstruction methods to restore the original image resolution. Also in [50] the authors propose a method that uses deep convolutional neural networks to ex-

tract visual features from frames of the reference and target videos and determine via fully connected neural networks and random forest classifiers whether there are or there are not abandoned objects in the scene.

Table 1.2 summarizes the main techniques employed by each moving camera method discussed in this section.

Table 1.2: Moving camera algorithms and the techniques employed in them.

Method's Reference	Main Technique
KONG [6]	Normalized cross-correlation
CARVALHO [11]	Normalized cross-correlation
MUKOJIMA [40]	Deep-flow + background subtraction
Nakahata [42]	Dictionary learning
THOMAZ [8]	Subspace projection
CUI [45]	Dense trajectory background subtraction
JARDIM [46]	Low-rank plus sparse matrix decomposition
CINELLI [49]	Deep residual network background subtraction
AFONSO [50]	Deep neural network feature classification with random forests

1.1.3 Moving Object Detection in Moving Backgrounds via Matrix Decomposition

One of the most common applications of the computer vision methods is to detect moving foreground objects. Some of the state-of-the-art techniques designed to deal with this task are briefly discussed here.

The transformed Grassmannian Robust Adaptive Subspace Tracking Algorithmt (t-GRASTA) [51] obtains two matrices (low-rank background model and sparse foreground) from a set of original images and a geometric transformation (such as a rotation). In order to do so, it uses an incremental gradient descent (GD) constrained to the Grassmanian manifold of the estimated subspaces.

The Grassmannian Online Subspace Updates with Structured-sparsity (GOSUS) [52] performs the decomposition using a sequential subspace learning algorithm. It applies a structural restriction to the updates on a Grassmannian manifold based on a group-norm.

The work presented in [53] proposes a sequential RPCA algorithm that uses geometric transformations for image alignment. Unlike most methods, in [53] these transformations are not applied on the noisy input samples, but only on the recovered samples.

Translational and rotational incremental principal component pursuit (PCP) [54] is a method that aims to processes one frame at a time, avoiding the need for batch

processing and yielding a small memory footprint. It is also capable of dealing with translational and rotational jitter which makes it more robust than its predecessors.

Motion-Aware Graph Regularized RPCA (MAGR-RPCA) [55] creates a background model by using a modified version of RPCA to generate a low-rank matrix from a set of matrices. In order to do so, an optical flow algorithm is used to estimate the motion, and intra-frame and inter-frame graphs are used to preserve geometric information in the low-rank matrix estimation.

The Spatiotemporal Robust Principal Component Analysis (SRPCA) [56] proposes the use of a motion mask that separates the pixels clearly belonging to the foreground. These pixels are labeled as missing data while estimating a temporally smooth background model from the remaining data.

Comprehensive surveys about these low-rank decomposition for forground/background separation methods can be found in [57, 58] and implementations of the algorithms can be found in [59].

Table 1.3 summarizes the main techniques employed by each moving object detection method discussed in this section.

Table 1.3: Moving object detection algorithms using low-rank plus sparse decomposiontion and the techniques employed in them.

Method's Reference	Main Technique
t-GRASTA [51]	Gradient descent
GOSUS [52]	Grassimannian manifold restrictions
sequential RPCA [53]	Geometric transformations plus RPCA
PCP [54]	Principal component pursuit
MAGR-RPCA [55]	Optical-flow motion estimation
SRPCA [56]	Motion masks restrictions

1.2 Objectives and main contributions

For the first part, change detection in moving camera videos, our investigation is centered on the development of novel methods to perform the detection of anomalies using videos recorded with moving cameras. The framework of the algorithm to be developed is similar to those that employ two videos acquired at different times to perform the anomaly detection. As an inspiration for the method to be proposed, the work in [46] will be used and expanded in a way to allow better performance and to operate without the need of previous video alignment, nor the use of external cues or signals to perform the video synchronization.

For the second part, moving object detection in dynamic backgrounds, we aim to develop methods that expand the capabilities of the current state-of-the-art on

sparse plus low-rank decomposition methods. We aim to do so by allying these decompositions to the use of eye-fixation prediction methods. Such methods focus the attention of the decompositions and avoid the appearance of false positive detections. In this part the main inspiration to our proposals is the method proposed in [60], which aim to perform the same task by projecting the low-rank matrix representation into a low-dimensional subspace.

The main contributions of this work are related to the minimization of data usage in the representation to achieve essentially the same results, while performing less computations. For the moving camera case, we will also incorporate the time alignment, that is usually applied to the videos as a pre-processing step, into the algorithm, while using intrinsic properties of the decomposition matrices. For the moving object detection case we will propose to use a tri-sparse optimization scheme to obtain new batch and sequential algorithms with state-of-the art performances.

1.3 Text Organization

In order to organize the text, the following chapter structure is used: Chapter 2 presents the algorithms that were used as inspiration to the development of our methods for change detection in videos obtained with moving camera. Chapter 3 features the developed algorithms to automatically perform change detection in videos acquired with moving cameras using sparse plus low-rank matrix decomposition. Chapter 4 discusses the current state-of-the-art on moving foreground detection in the presence of dynamic background, as well as some algorithms that inspired our solution to this problem. Chapter 5 shows our contributions to the moving foreground detection in the presence of dynamic background algorithms. Finally Chapter 6 will summarize the work discussed in this thesis and propose ideas for its continuation.

1.4 Associated Publications

This thesis work for moving object and change detection was carried out by the author between March 2015 and December 2018 at Federal University of Rio de Janeiro (UFRJ) and during a research internship at North Carolina State University (NCSU) between March 2017 and February 2018. The majority of the developed work has been published in international conferences and in peer reviewed journals. Some other work that approach the same subject using different techniques was developed by the author in partnership with other colleagues and is also briefly discussed throughout this thesis. A list of the most relevant publications from the author that was developed during this period is shown bellow

Journal Publications

[i] THOMAZ, L. A., JARDIM, E., DA SILVA, A. F., et al. "Anomaly detection in moving-camera video sequences using principal subspace analysis", IEEE Transactions on Circuits and Systems I: Regular Papers, v. 65, n. 3, pp. 1003-1015, March 2018.

[ii] JARDIM,E., THOMAZ, L. A., DA SILVA, E. A. B., Netto, S. L.. "Domain-transformable sparse representation for anomaly detection in moving-camera videos," IEEE Transactions Image Processing, Under Review 2018.

[iii] DE CARVALHO, G. H. F., THOMAZ, L. A., DA SILVA, A. F., et al. "Anomaly Detection with a Moving Camera using Multiscale Video Analisys", Multidimensional Systems and Signal Processing, pp. 1-32, February 2018.

[iv] NAKAHATA, M. T., THOMAZ, L. A., DA SILVA, A. F., et al. "Anomaly detection with a moving camera using spatio-temporal codebooks", Multidimensional Systems and Signal Processing, pp. 1-30, March 2017.

Peer Reviewed Conference Publications

[i] AFONSO, B. M., CINELLI, L. P., THOMAZ, L. A., et al. "Moving-Camera Video Surveillance in Cluttered Environments Using Deep Features". In: 2018 25th IEEE International Conference on Image Processing, pp. 2296-2300, Athens, Greece, Oct 2018.

[ii] THOMAZ, L. A., DA SILVA, A. F., DA SILVA, E. A. B., et al. "Detection of abandoned objects using robust subspace recovery with intrinsic video alignment". In: IEEE International Conference on Circuits and Systems, pp.1-4, Baltimore, USA, May 2017.

[iii] DA SILVA, A. F., THOMAZ, L. A., NETTO, S. L., et al. "Online video-based sequence synchronization for moving camera object detection". In: IEEE International Workshop on Multimedia Signal Processing, pp. 1-6, Luton, United Kingdom, October 2017.

[iv] CINELLI, L. P., THOMAZ, L. A., DA SILVA, A. F., et al. "Foreground Segmentation for Anomaly Detection in Surveillance Videos Using Deep Residual Networks". In: Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, pp. 914-918, São Pedro, Brazil, September 2017.

[v] DA SILVA, A. F., THOMAZ, L. A., DA SILVA, E. A. B., et al. "Alinhamento de sinais obtidos em trajetórias fechadas utilizando um conjunto genérico de sensores". In: Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, pp. 1-5, Santarém, Brazil, September 2016.

[vi] DA SILVA, A. F., THOMAZ, L. A., CARVALHO, G., et al. "An annotated video database for abandoned-object detection in a cluttered environment". In: International Telecommunications Symposium, pp. 1-5, São Paulo, August 2014.

Chapter 2

Review of Change Detection with Sparse Representation in Moving Camera Videos

Some of the main inspirations for the development of the present work are the algorithms that explore low dimensional structures in high dimensional spaces while analysing high dimensional data. By converting the data that lies in a high-dimensional space into less complex representations one is able to simplify the data analysis that follows.

This chapter discusses methods that are able to perform background subtraction by means of representing target videos as a combination of low-rank surrogates of the reference video. By doing so, one may detect video anomalies as the sparse residue that was not represented by the low-rank surrogate combination.

This chapter is divided as follows: Section 2.1 presents some of the most widely used databases for object detection using moving cameras; Section 2.2 presents the concept of principal subspace analysis (PSA) along with the most traditional algorithms to solve these problems; Section 2.3 presents the RoSuRe algorithm, a PSA algorithm that was one of the main inspirations to this work; finally Section 2.4 presents mcRoSuRe, a modification of the RoSuRe algorithm that will be later modified on the course of this thesis.

2.1 Moving Camera Object Detection Databases

With the goal of testing the performance of automatic anomaly detection algorithms, some databases were designed and recorded. Most of them address multiple challenges, as the tracking and recognition of personal and objects in several different scenarios which are subject to various challenging interferences.

One of the most famous databases for anomaly detection in security applications is that of the performance evaluation of tracking and surveillance (PETS) program (2000-2007) [61]. It contains videos from various scenarios with or without the presence of people and with a variable number of cameras observing the same area. The main goal of this database is to evaluate the detection of abandoned and removed objects in the scene.

Another database created specifically to test algorithms of anomaly detection in video sequences is the one called imagery library for intelligent detection systems (i-LIDS) [62]. This database contains videos of 4 different scenarios: abandoned baggage detection, parked vehicle detection, doorway surveillance, and sterile zone monitoring. Every video of the database is provided with a ground-truth marked manually with a description of every event and the location of every moving or static object.

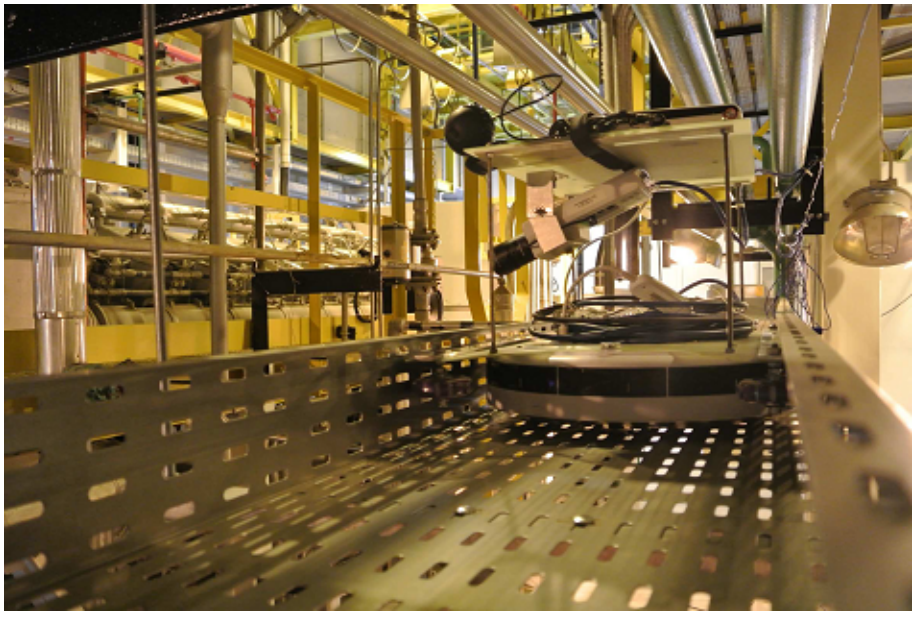
The changeDetection.net (CDNET) [63] is a database containing videos designed to test algorithms that deal with six different challenges in anomaly detection: dynamic background, camera jitter, detection of shadows, intermittent object motion, thermal anomaly detection, and a combination of several of those challenges. All videos have ground-truth labels.

Although several distinct challenges are present in the above mentioned databases, only a minor part of them has videos that contain significant camera motion. Even in cases where there are videos with such motion, it is restricted only to camera jitter or PTZ (pan-tilt-zoom) movement. Some applications that deal with anomaly detection using moving camera are designed to deal with a different kind of camera movement (e.g. translational movement).For the moving camera object detection applications a useful database is the one called video database for abandoned objects (VDAO) [64], available at [65]. This database contains over 8 hours of videos recorded in visually cluttered complex environments of industrial plants. The database videos contain several challenges as illumination variation, occlusion of objects, and camera jitter. All videos feature reference and target sequences with manually marked ground-truth labels.

The VDAO database was recorded using a camera mounted on top of a moving robotic platform that follows a linear path of about 6 m in a hanging rail at a height of approximately 2.5 m. The camera is pointed at a cluttered environment comprising several pipes and valves simulating a scene of interest inside an industrial facility. Figure 2.1 shows the experimental setup used to record the database. The database videos are separated in two groups: single and multi objects. The single object videos have only one abandoned object placed along the camera path, while the multi-object videos have at least two abandoned objects present in every frame of the video. Figure 2.2 shows the objects used in the single object videos while

Figure 2.3 shows the objects featuring the multi-object sequencies.

(a)



(b)

Figure 2.1: General setup of abandoned-object detector and database recording using a moving camera on a robotic platform. (a) rail placement (b) robotic moving platform

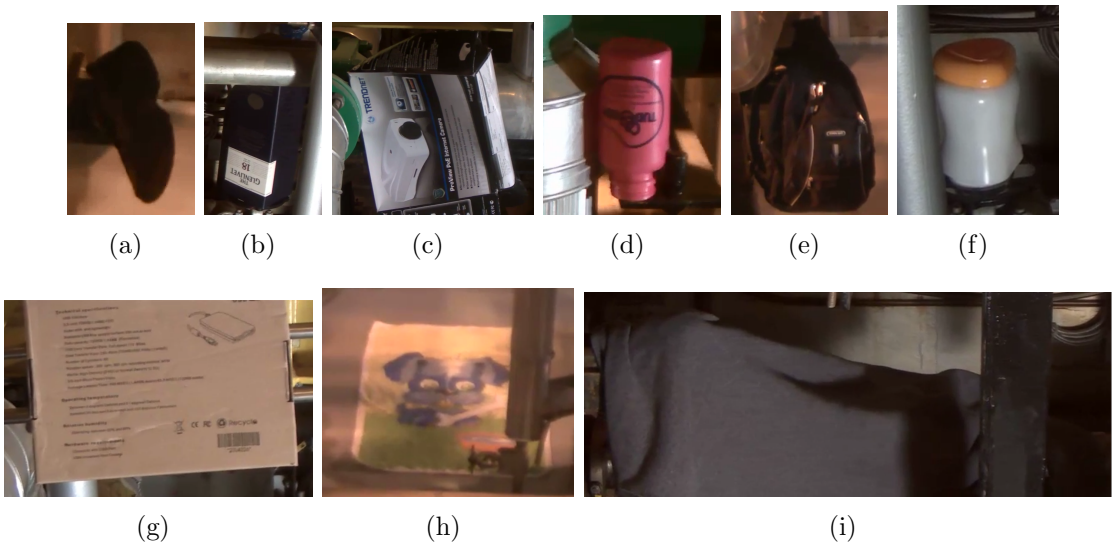


Figure 2.2: Objects used in the single object videos (scales have been changed for a better presentation): (a) shoe; (b) dark blue box; (c) camera box; (d) pink bottle; (e) black backpack; (f) white jar; (g) brown box; (h) towel; (i) black coat.

2.2 Principal Subspace Analisys

When dealing with high-dimensional data one usually wants to find a representation with a reduced dimensionality that allows the data to be analyzed and stored using less resources. A common assumption in those cases is that the data was acquired from a real-world source (e.g. a sensor or a transducer). This implies that it is most likely subjected to noise and other perturbations, which tend to be reduced in the low-dimensional model. In this section, we provide a unified framework for some of the main methods used to project high-dimensional data onto subspaces of low dimension, which is known as subspace learning or principal subspace analysis (PSA) [66].

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ be an $m \times n$ data matrix with $\mathbf{x}_i$ comprising m-dimensional observations. The projection algorithms model the data as

$$\mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{2.1}$$

where $\mathbf{L}$ is a low-rank matrix and $\mathbf{E}$ is a sparse residue matrix.

One of the most well-known and widely used algorithms for this type of analysis is the principal component analysis [24], which employs the singular value decomposition [67] to find out the orthonormal basis that supports the low-dimensional data subspace, while casting the remaining noisy components to the residue matrix. This approach is able to find the optimal subspace that minimizes the projection

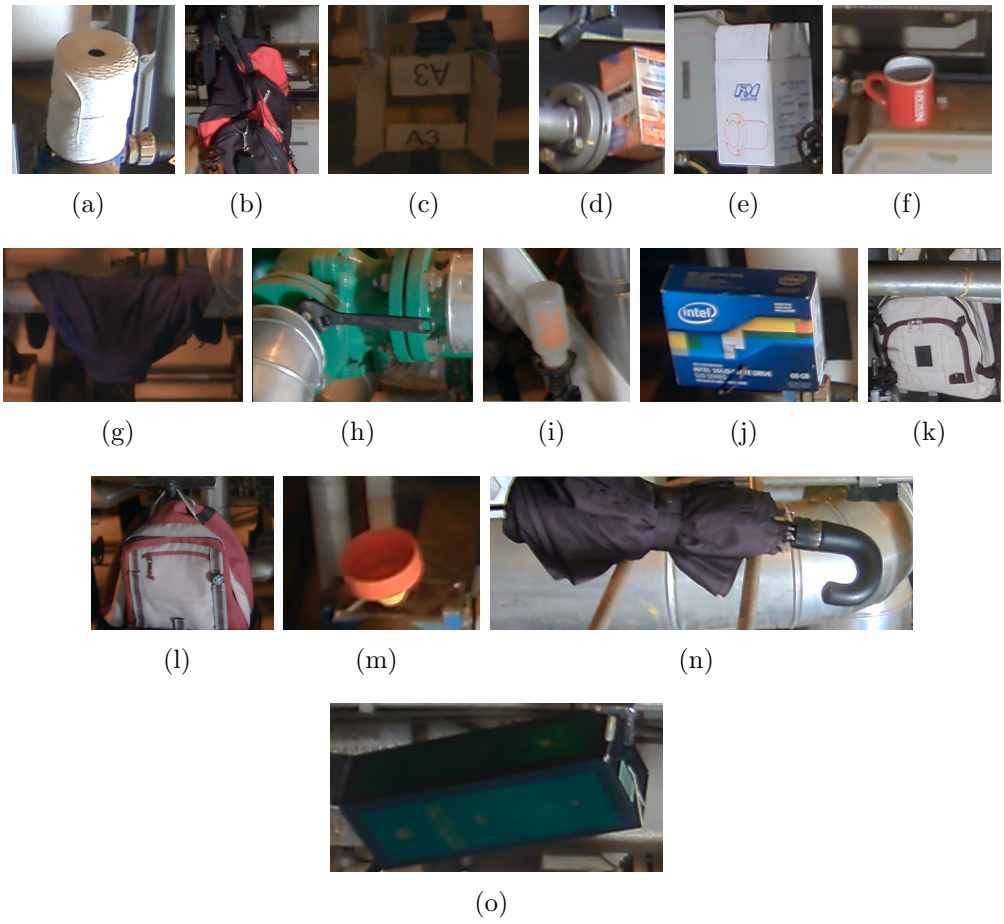


Figure 2.3: Objects used in the multi object videos (scales have been changed for a better presentation): (a) string roll; (b) bag; (c) white box; (d) lamp-bulb box; (e) spotlight box; (f) mug; (g) blue coat; (h) wrench; (i) bottle; (j) blue box; (k) backpack; (l) pink backpack; (m) bottle cap; (n) umbrella; (o) green box.

error of the columns of $\mathbf{X}$ and may be expressed as

$$\min_{\mathbf{L},\mathbf{E}} \|\mathbf{E}\|_{\mathrm{F}} \quad \text{s.t.} \quad \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathrm{rank}(\mathbf{L}) \leq r \end{cases}, \tag{2.2}$$

where $\|.\|_{\mathrm{F}}$ denotes the Frobenius [68] norm and r is the maximum rank of matrix $\mathbf{L}$. The PCA, however, is only able to cope with small corruptions in the original data, since large corruption levels modify the subspace support vectors significantly, compromising the resulting data decomposition. Also, the maximum rank of the $\mathbf{L}$ matrix must be known a priori, thus requiring some previous knowledge about the data.

The so-called robust PCA [26] is a refined version of the PCA algorithm that is able to recover a low-rank matrix $\mathbf{L}$ even when the original data matrix $\mathbf{X}$ includes outliers (heavy-tail noise). Note that formulation of RPCA assumes the rank (r) unknown, and hence an intrinsic property of the underlying model to be unveiled.

Mathematically the formulation of RPCA may be written as

$$\min_{\mathbf{L},\mathbf{E}} \ \text{rank}(\mathbf{L}) + \lambda \|\mathbf{E}\|_0 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{2.3}$$

where $\|.\|_0$ is the l_0-norm (number of non-zero entries in the matrix) and λ is a weighting parameter. Although this problem formulation is very simple and effective, it is an intractable NP-hard problem that cannot be solved for large data sizes. A relaxed version is often used [26],

$$\min_{\mathbf{L},\mathbf{E}} \ \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 \quad \text{s.t.} \ \mathbf{X} = \mathbf{L} + \mathbf{E}, \tag{2.4}$$

where $\|.\|_*$ is the nuclear norm (defined as $\|\mathbf{A}\|_* = \text{tr}(\sqrt{\mathbf{A}^{\text{H}}\mathbf{A}})$, with $\mathbf{A}^{\text{H}}$ denoting the conjugate transpose of $\mathbf{A}$) and $\|\mathbf{A}\|_1$ is the sum of the absolute values of all the entries of $\mathbf{A}$.

2.3 RoSuRe Algorithm

Both PCA and RPCA are able to project the data onto a single subspace. When the data matrix is better interpreted by the projection onto a union of subspaces of lower dimensions, however, one may consider the Robust Subspace Recovery (RoSuRe) algorithm proposed in [47, 48]. This method is able to represent data originally located in high-dimensional spaces into a union of subspaces (UoS). Unlike some of the predecessors, it is able to cope with the presence of corrupted data and still provide good results.

The formal problem formulation is the following: If one considers the union of subspaces $S = \cup_{i=1}^{J} \mathcal{S}^i$ and $\mathbf{L_1}, \mathbf{L_2}, \cdots, \mathbf{L_J}$ being matrices whose columns $\mathbf{l_j}$ are uniformly sampled from $\mathcal{S}^i$, with sufficient sampling density, every column $\mathbf{l_j}$ can be represented by a linear combination of columns $\mathbf{l_{i \neq j}}$ from the same subspace. In this formulation, one considers the UoS $\mathcal{S} = \cup_{j=1}^{J} \mathcal{S}^{(j)}$ with $\mathbf{L}$ being a matrix whose columns are uniformly sampled from $\mathcal{S}$. We group all the samples from the same subspace $\mathcal{S}^{(j)}$ into matrix $\mathbf{L}^{(j)}$ so that

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}^{(1)} & \mathbf{L}^{(2)} & \cdots & \mathbf{L}^{(J)} \end{bmatrix}. \tag{2.5}$$

With sufficient sampling density, every column $\mathbf{l}_k^{(j)}$ of $\mathbf{L}^{(j)}$ can be represented by a linear combination of the other columns $\mathbf{l}_i^{(j)}$, $i \neq k$ from the same subspace. In this case, one can say that the set of columns of $\mathbf{L}^{(j)}$ is self-representative, and it is

possible to state that

$$\mathbf{L}^{(j)} = \mathbf{L}^{(j)}\mathbf{W}^{(j)}, \tag{2.6}$$

where $\mathbf{W}^{(j)}_{i,i}$, the entry of the $\mathbf{W}^{(j)}$ matrix in the i-th column and in the i-th line, the diagonal, is equal to zero for all values of i.

Differently from the projection onto a single subspace, the projection of the data onto a UoS allows the data model to cope with a wider range of values. This happens since in a single subspace all datapoints should be obtainable as a combination of other datapoints within the subspace, while in a UoS, a given datapoint can be obtained as a combination of other lying in the same subspace but not necessarily as the combination of other points from a different subspace in the same UoS, thus allowing the representation of high dimensional structures in a sparse way. In the single subspace projection to cope with high dimensional data, the projection matrix would be nearly full rank, in opposition to the low-rank matrix obtained in the UoS projection. This flexibility allows the model to store less data and yet display a powerful representation capability.

As a result, from Eq. (2.5) one can write that $\mathbf{L} = \mathbf{LW}$, with

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{W}^{(J)} \end{bmatrix}, \tag{2.7}$$

where, from Eq. (2.6), $\mathbf{W}^{(j)}_{k,k} = 0$, $j = 1, \ldots, J$. Note that by observing the underlying structure from $\mathbf{W}$ it is possible to infer the subspace structure from $\mathbf{L}$, which is said to be blockwise low-rank as induced by $\mathbf{W}$.

Let now $\mathbf{X}$ be such that it can be represented as an element belonging to the UoS $\mathcal{S}$ added to a sparse residue $\mathbf{E}$. This is equivalent to stating that $\mathbf{X}$ can be decomposed as

$$\mathbf{X} = \mathbf{LW} + \mathbf{E}, \tag{2.8}$$

where, from Eq. (2.7), $\mathbf{W}$ is blockwise diagonal with $\mathbf{W}_{k,k} = 0$ for all k, $\mathbf{L}$ is blockwise low-rank, and $\mathbf{E}$ is sparse.

The RoSuRe method assumes sparsity both on $\mathbf{W}$ (due to its structure) and $\mathbf{E}$ (as it is considered that each $\mathbf{E}_i$ is sparse). To perform the decomposition and assure

the above constraints, the method solves the following optimization problem:

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_0 + \lambda\|\mathbf{E}\|_0, \quad \text{s.t.} \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{i,i} = 0, \ \forall i \end{cases}, \tag{2.9}$$

where $\|\cdot\|_0$ represents the number of non-zero entries on the matrix. As this is a hard non-convex optimization problem, [48] proposes solving a relaxation of it given by

$$\min_{\mathbf{W},\mathbf{E}} \|\mathbf{W}\|_1 + \lambda\|\mathbf{E}\|_1, \quad \text{s.t.} \begin{cases} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \end{cases}. \tag{2.10}$$

Due to the presence of the bilinear term composed by the simultaneous optimization of $\mathbf{W}$ and $\mathbf{E}$ this problem is clearly non-convex [69]. To deal with that by using convex optimization methods it is necessary to use some technique like the *alternating direction method of multipliers* (ADMM), where the optimization is performed by minimizing the l_1 norm of $\mathbf{W}$ and $\mathbf{E}$, alternatively, thus varying the direction of minimization between every step. This can be further improved by applying the *augmented Lagrangian method* (ALM) [69].

Particularly to this application a variation of the ADMM called *linearized alternating direction method* (LADM) [70] was used instead of the traditional one, due to its faster conversion. By using this method one can find the solution to the optimization problem by solving a dual problem instead. As stated in Theorem 4 in [71], if the objective function is lower bounded and the solution exists then the duality gap is zero, thus the dual problem can be solved to achieve the global solution via the ALM.

The optimization proposed in Eq. (2.10) can be solved with the use of Algorithm 1. In this and in the subsequent algorithms, the variable μ_k is the augmented Lagrange multiplier, ρ is the step used to update μ_k, $\eta_1 \geq \|\mathbf{L}\|_2^2$ and $\eta_2 \geq \|\hat{\mathbf{W}}\|_2^2$ are normalizing weights, $\tau_\alpha(\cdot)$ is the soft-thresholding operator for the augmented Lagrangian multiplier, defined as [72]

$$\tau_\alpha(x) = \begin{cases} x - \alpha, & x \geq \alpha \\ 0, & |x| \leq \alpha \\ x + \alpha, & x \leq -\alpha. \end{cases} \tag{2.11}$$

Further details can be found on [48].

The RoSuRe algorithm was proven [47] to work properly on synthetic and real data created by randomly sampling vectors from UoS and adding sparse corrupting

Algorithm 1 RoSuRe

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, λ, $\rho > 1$, η_1, η_2, μ_0, $\mathbf{W}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$, $\hat{\mathbf{W}}_0 = \mathbf{I}$.

while not converged **do**
$k = k + 1$
Update $\mathbf{W}$ by linearized soft-thresholding:
$\mathbf{L}_{k+1} = \mathbf{X} - \mathbf{E}_k$
$\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu_k \eta_1}} \left(\mathbf{W}_k - \frac{1}{\eta_1} \mathbf{L}_{k+1}^{\mathrm{T}} \left(\mathbf{L}_{k+1} \hat{\mathbf{W}}_k - \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
$\mathbf{W}_{k+1}^{ii} = 0$
Update $\mathbf{E}$ by linearized soft-thresholding:
$\hat{\mathbf{W}}_{k+1} = \mathbf{I} - \mathbf{W}_k$
$\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu_k \eta_2}} \left(\mathbf{E}_k + \frac{1}{\eta_2} (\mathbf{L}_{k+1} \hat{\mathbf{W}}_{k+1} - \frac{\mathbf{Y}_k}{\mu_k}) \hat{\mathbf{W}}_{k+1}^{\mathrm{T}} \right)$
Update the Lagrange multiplier Y and the augmented Lagrange multiplier μ_k
$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k \left(\mathbf{L}_{k+1} \mathbf{W}_{k+1} - \mathbf{L}_{k+1} \right)$
$\mu_{k+1} = \rho \mu_k$
end while

noise with different signal-to-noise ratios (SNR). The original paper in [47] details the results of testing with synthetic data.

The method has proven useful for other applications including computer vision ones, as it was successfully applied to face clustering and surveillance videos for background subtraction and anomaly detection. As the focus of this investigation is to use sparse decomposition methods to detect anomalies in video sequencies, the results of RoSuRe in surveillance applications are very encouraging.

In Figure 2.4 the results of the method for background subtraction in static camera application are shown. Here the background is modeled into the low-rank $\mathbf{L}$ matrix from the previous frames of the sequence and, as new objects appear, the decomposition algorithm casts them upon the residue matrix $\mathbf{E}$, separating that "sparse noise" from the background formed by the previous frames combination.

Yet more encouraging are the results of RoSuRe in moving camera scenarios. Tests were made using synthetic movement created by simulating panning in the previous surveillance footages, without considering the parallax effect. Figure 2.5 shows some experimental results of the RoSuRe algorithm that have proven to pass the proof of concept stage.

Another good feature is the structure of matrix $\mathbf{W}$ which, as seen in Figure 2.6, that $\mathbf{W}$'s structure carries information about the path of the panning camera, giving a hint that it can be used to align videos.

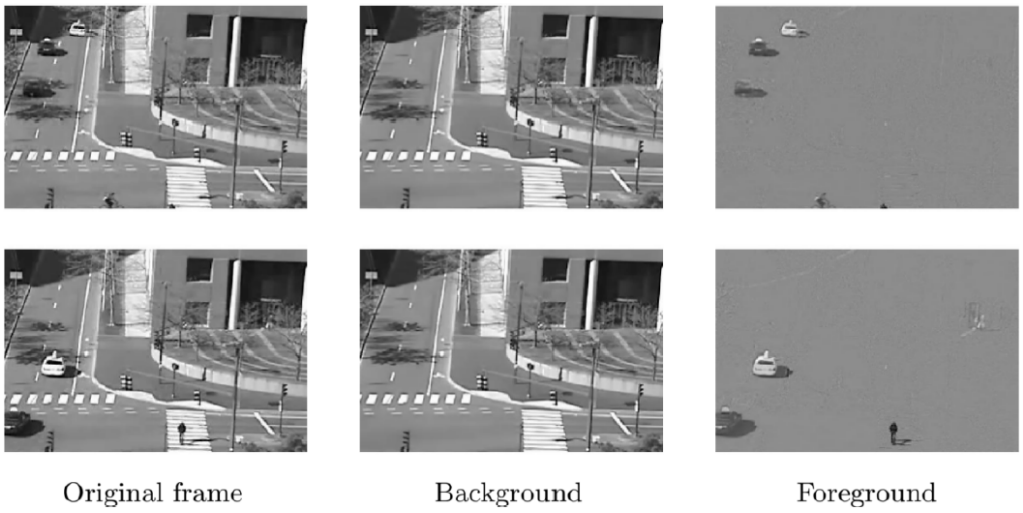


Figure 2.4: Background subtraction using RoSuRe for static camera scenario. (Figure acquired from [48])

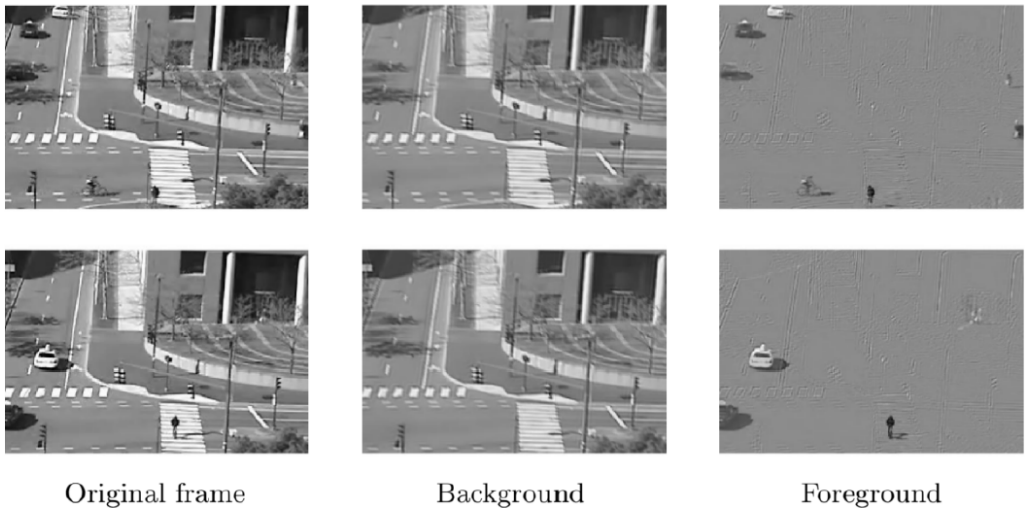
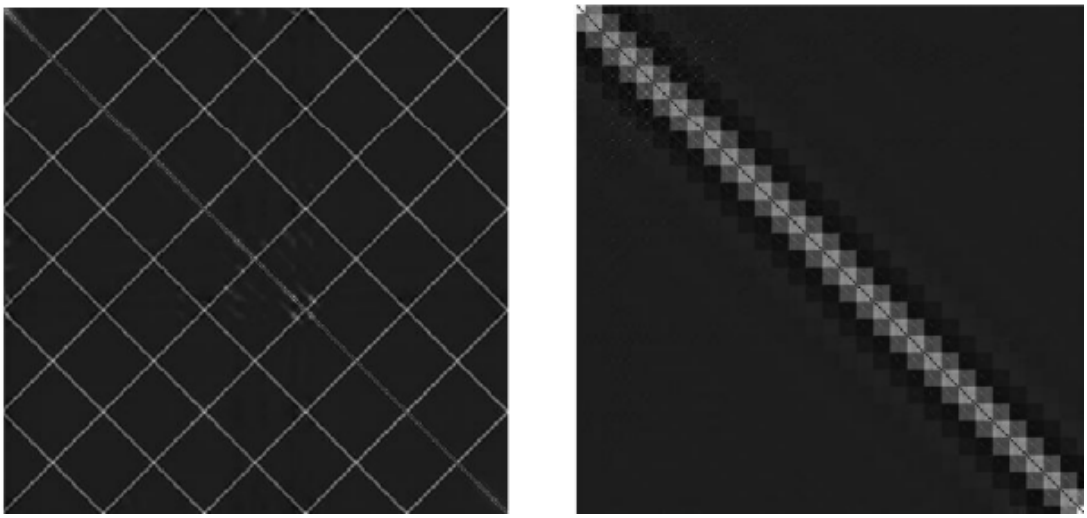


Figure 2.5: Background subtraction using RoSuRe for moving camera scenario. (Figure acquired from [48])

2.4 mcRoSuRe Algorithm

Based on the previously discussed work, reference [46] presents an anomaly detection algorithm intended to work with videos acquired with moving cameras. As stated in Chapter 1, the algorithm aims to work with detection of anomalies in video sequences from moving camera and since its goal is to find static objects in the scene, the traditional scheme that features acquisition of reference and target videos, alignment and detection, is used.

Inspired by the application of RoSuRe with the panning surveillance videos, the work [46] presents a PSA method designed to work with a slowly moving camera that has mainly translational movement. Because of its inspiration and the target application, the method is called moving camera RoSuRe (mcRoSuRe). The assumption



(a) (b)

Figure 2.6: Coefficient matrix $\mathbf{W}$. (a) without rearrangement according to the position of the camera (b) with rearrangement according to the position of the camera. (Figure acquired from [48])

of slow movement of the camera is important as it is necessary that consecutive frames of the video share approximately the same low-rank representation.

The first step of the algorithm consists in the exact application of RoSuRe in the reference video $\mathbf{X}_r$, which is decomposed as

$$\mathbf{X}_r = \mathbf{L}_r\mathbf{W}_r + \mathbf{E}_r, \tag{2.12}$$

$$\mathbf{E}_r = \mathbf{X}_r - \mathbf{L}_r, \tag{2.13}$$

where $\mathbf{L}_r$ is the low-rank[1] representation of $\mathbf{X}_r$ and $\mathbf{E}_r$ is its sparse complement. Note that Eqs. (2.12) and (2.13) imply that $\mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r$. The corresponding optimization problem then becomes

$$\min_{\mathbf{W}_r,\mathbf{E}_r} \|\mathbf{W}_r\|_1 + \lambda\|\mathbf{E}_r\|_1, \quad \text{s.t.} \begin{cases} \mathbf{X}_r = \mathbf{L}_r + \mathbf{E}_r \\ \mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r \\ \mathbf{W}_{r_{ii}} = 0, \ \forall i \end{cases}. \tag{2.14}$$

In the absence of any anomalous content, the corresponding frames in both the reference and target videos in the surveillance system depicted in Fig. 1.1 share the same low-rank representation. Therefore, one can use the low-rank representation

[1]The self-representative matrix $\mathbf{L}_r$ is guaranteed to be low-rank for a single subspace. For a UoS, as presented in this case, it is usually low-rank, but there may be cases where the construction of a specific UoS may not lead to a low-rank matrix $\mathbf{L}_r$. Nevertheless, as for making the notation of the methodology compatible with that of previous works, we will refer to $\mathbf{L}_r$ as either "low-rank" or "self-representative" matrix interchangeably.

$\mathbf{L}_r$ of $\mathbf{X}_r$ to represent the target video $\mathbf{X}_t$ such that

$$\mathbf{X}_t = \mathbf{L}_r \mathbf{W}_t + \mathbf{E}_t, \tag{2.15}$$

with $\mathbf{W}_t$ and $\mathbf{E}_t$ both being sparse matrices, to which the corresponding optimization is

$$\min_{\mathbf{W}_t, \mathbf{E}_t} \|\mathbf{W}_t\|_1 + \lambda \|\mathbf{E}_t\|_1, \quad \text{s.t.} \quad \mathbf{L}_r \mathbf{W}_t = \mathbf{X}_t - \mathbf{E}_t. \tag{2.16}$$

By modifying the original RoSuRe algorithm [48], the optimization problem in Eq. (2.16) can be solved as summarized in Algorithm 2.

Algorithm 2 Sparse representation of $\mathbf{X}$ given the low-rank representation $\mathbf{L}$

Input: $\mathbf{L}$, $\mathbf{X}$, λ,$\rho > 1$, η_1, η_2, μ_0, $\mathbf{W}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$.
while not converged **do**
  $k = k + 1$
  $\mathbf{L}'_{k+1} = \mathbf{X} - \mathbf{E}_k$
  $\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu\eta_1}} \left(\mathbf{W}_k - \frac{1}{\eta_1} \mathbf{L}^{\mathrm{T}} \left(\mathbf{L}\mathbf{W}_k - \mathbf{L}'_{k+1} + \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
  $\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu\eta_2}} \left(\mathbf{E}_k - \frac{1}{\eta_2} \left(\mathbf{L}\mathbf{W}_{k+1} - \mathbf{L}'_{k+1} + \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
  $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k \left(\mathbf{L}\mathbf{W}_{k+1} - \mathbf{L}'_{k+1} \right)$
  $\mu_{k+1} = \rho \mu_k$
end while

Solving the problem in Eq. (2.16), all the anomalous information in $\mathbf{X}_t$ that could not be represented from $\mathbf{L}_r \mathbf{W}_t$ are cast upon $\mathbf{E}_t$. Actually, there are in $\mathbf{E}_t$ other artifacts (such as high-frequency components not representable by the low-rank matrix $\mathbf{L}_r$) that are not related to the anomalies of interest. Those artifacts, however, are indeed supposed to be present in matrix $\mathbf{E}_r$. Therefore, one can remove these artifacts from $\mathbf{E}_t$ by performing an additional decomposition of this matrix using $\mathbf{E}_r$ as its low-rank component, as given by

$$\mathbf{E}_t = \mathbf{E}_r \mathbf{W}_e + \mathbf{E}_e, \tag{2.17}$$

such that the final residue matrix $\mathbf{E}_e$ contains only the anomalies of interest in the target video. To allow such representation, one has to perform the following optimization

$$\min_{\mathbf{W}_e, \mathbf{E}_e} \|\mathbf{W}_e\|_1 + \lambda \|\mathbf{E}_e\|_1, \quad \text{s.t.} \quad \mathbf{E}_r \mathbf{W}_e = \mathbf{E}_t - \mathbf{E}_e. \tag{2.18}$$

A summarized version of the complete moving-camera RoSuRe algorithm is presented in Algorithm 3. The optimizations in each line are described in detail in Algorithms 1 and 2.

Figure 2.7 shows some of the results obtained with mcRoSuRe method. From

Algorithm 3 Moving-camera RoSuRe algorithm

Require: $\mathbf{X}_r$, $\mathbf{X}_t$

$\min_{\mathbf{W}_r,\mathbf{E}_r} \|\mathbf{W}_r\|_1 + \lambda\|\mathbf{E}_r\|_1$, s.t. $\mathbf{X}_\mathrm{r} = \mathbf{L}_\mathrm{r} + \mathbf{E}_\mathrm{r}$, $\mathbf{L}_r\mathbf{W}_r = \mathbf{L}_r$, $\mathbf{W}_{r_{ii}} = 0$

$\min_{\mathbf{W}_t,\mathbf{E}_t} \|\mathbf{W}_t\|_1 + \lambda\|\mathbf{E}_t\|_1$, s.t. $\mathbf{L}_\mathrm{r}\mathbf{W}_\mathrm{t} = \mathbf{X}_\mathrm{t} - \mathbf{E}_\mathrm{t}$

$\min_{\mathbf{W}_e,\mathbf{E}_e} \|\mathbf{W}_e\|_1 + \lambda\|\mathbf{E}_e\|_1$, s.t. $\mathbf{E}_\mathrm{r}\mathbf{W}_\mathrm{e} = \mathbf{E}_\mathrm{t} - \mathbf{E}_\mathrm{e}$

this figure, one can easily see that the algorithm has a good performance when the anomalies to be detected using a moving camera are abandoned objects in visually cluttered environments.

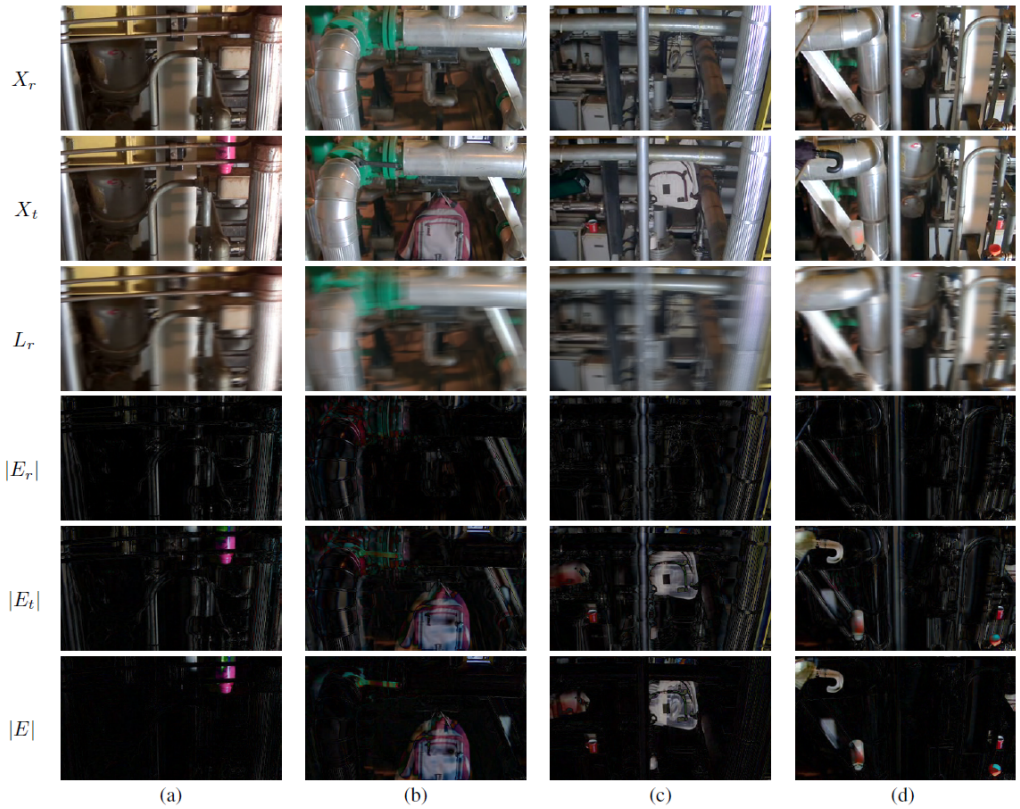


Figure 2.7: Experimental results (single frames of matrices Xr, Xt, Lr, Er, Et, and E) using the low-rank representation proposed in [46] for 4 different abandoned-object scenarios: (a) pink bottle; (b) backpack + wrench + box; (c) backpack + green box + mug + string roll; (d) umbrella + bottle + bottle cap + mug. (Figure acquired from [46])

2.5 Summary

This chapter presented a view of the current state-of-the-art on change detection algorithms using sparse representation. Some of the most well known datasets for abandoned object and change detection were presented along with the VDAO dataset, which has some unique characteristics that make it the best choice for this

thesis experiments.

A brief overview on the most well-known PSA algorithms was presented in Section 2.2 which led to the description of the RoSuRe method, that is able to project the data into a UoS, instead of a single subspace as in most PSA methods. Later, in Section 2.4 the mcRoSuRe method was presented. This method, based on the RoSuRe, is capable of extending the PSA capabilities of the RoSuRe method and allow the application of such method in the traditional framework for moving camera object detection.

In the next chapter our contributions on change detection using sparse representation using moving cameras will be presented. Our proposals extend and improve the capabilities of the mcRoSuRe method allowing it to be executed in feasible time.

Chapter 3

Contributions to Change Detection with Sparse Representation

Both algorithms discussed in Chapter 2 have shown to be very effective on their target applications that are background subtraction for static surveillance videos for RoSuRe [47, 48] and anomaly detection on videos acquired from moving cameras for mcRoSuRe [46]. Besides the important results presented in the articles that introduce each of the methods, it is possible to see that both of them have severe limitations that do not make them able to perform in the traditional anomaly-detection frameworks, achieving near real-time performances.

The obvious drawback from RoSuRe is that the method is not prepared to deal properly with many kinds of moving background nor is able to perform adequately with the traditional framework of moving camera anomaly detection that was presented in Chapter 1, since the algorithm's structure only supports a single video stream.

The mcRoSuRe was proposed to adapt the RoSuRe method to the traditional framework of moving camera anomaly detection. But as the authors affirm in the original paper [46] the method is too heavy to perform for large videos, like those usually used in the surveillance schemes. Also it still relies on external methods to assure that the target video field-of-view (FoV) is completely comprised on that of the reference video. For this reason the authors of mcRoSuRe have performed tests to assess the quality of detection of their algorithms only on small chunks of reference and target videos featuring about 70-frame target videos and 100-frame reference videos of 320×180 [46].

In this chapter several experiments that were done aiming to improve the efficiency of the methods will be presented. They are presented in the way the algo-

rithms were developed in a historical and incremental approach.

In the video alignment techniques, several attempts are made while trying to detect the best way to use the mcRoSuRe matrix $\mathbf{W}_t$ to extract the temporal information that allow reference and target videos to be synchronized without the use of any other external information, as that provided by peripheral sensors. In the preprocessing part, two different approaches will be used to transform the videos before the beginning of the algorithm while correcting some differences between both videos and allowing better correspondence between their frames. In the section related to speed-up techniques, the knowledge acquired in the video alignment part of the work will be used to reduce the reference video allowing the algorithm to work with less data, thus speeding it up by a great amount. Finally, a post-processing technique will be discussed to improve the results of the detection residue matrices.

This chapter is organized as follows: Section 3.1 presents the mcRoSuRe-TA algorithm, that was initially presented in [73], and that implements a new version of the mcRoSuRe algorithm with intrinsic video time alignment; Section 3.2 introduces further modifications to mcRoSuRe-A, originally developed in [74], that accelerate the algorithm to near real-time performance; Section 3.3 shows a computational complexity analysis for the RoSuRe algorithm family methods, showing the superior performance of the proposed algorithms; Section 3.4 presents the experimental comparison between the proposed methods and some of the state-of-the-art methods; finally Section 3.5 presents our conclusions about this chapter.

3.1 mcRoSuRe-TA

3.1.1 Video Alignment

In this section some experiments that were performed aiming to extract information from the raw videos matrix $\mathbf{X}$ and the combination matrix $\mathbf{W}$ of the mcRoSuRe method to perform the video synchronization will be described. The experiments will be presented in the following order:

- Attempt to use plain mcRoSuRe $\mathbf{W}_t$ matrix for video alignment;
- Check how the temporal downsampling of the reference impacts the ability to find correspondences between reference and target videos;
- Detection of camera direction changes using the $\mathbf{W}_t$ matrix;
- Detection of camera direction changes in target videos using the $\mathbf{W}_t$ matrix and reference videos that do not have camera direction changes;

- Localization of target videos in large reference videos using mcRoSuRe $\mathbf{W}_t$ matrix;
- Localization of target videos in large reference videos using a modified version of $\mathbf{W}_t$ matrix;

One of the first attempts during this investigation was to explore the possibility of using the plain mcRoSuRe matrix $\mathbf{W}$ as a feature to perform the alignment between reference and target videos. This idea came from the observation of the $\mathbf{W}$ matrix's structure from RoSuRe in [48] for the moving camera experiment as depicted in Figure 2.6. Some early experiments were performed using small reference and target videos of 320×180 pixels per frame. The reference videos were about 200-frames long and the target videos were 70-frames long.

For the following experiments we employed videos from the VDAO [65] database, which is described in Section 2.1. The videos feature translational camera movement and eventually direction changes.

Figure 3.1 displays the structure of the $\mathbf{W}_t$ matrix in this experiment. It is possible to see that the organization of the weights in this matrix supports the idea of using it for video alignment as the region from which each frame was extracted is clear in the observed matrix. Since the high values of in the $\mathbf{W}_t$ represent the contribution of a given frame in the reconstruction of another frame, locating which frames contribute the most to the reconstruction of the frames of the target video allows one to infer a rough alignment between reference and target video.

The second test performed was about the effects of downsampling the reference video. This investigation intended to find out if even with downsampled reference videos the target frames could be correctly reconstructed and find out how the matrix $\mathbf{W}_t$ would appear in such cases. In this experiment reference and target videos originally had a similar number of frames, around 400.

Figure 3.2 shows the $\mathbf{W}_t$ matrices that resulted from those experiments. It is possible to note that even with large downsampling rates (the reference videos were temporally downsampled with 1:1, 5:1, and 10:1 ratios) the $\mathbf{W}_t$ matrices still possess good spatial correspondences. This conclusion drawn from this experiment was a seed to a pre-processing idea proposed later.

Another experiment was performed intending to find out if it was possible to detect changes in the direction of movement of the target video. For this investigation reference and target videos of similar length were used, both reference and target videos feature changes in the direction of the camera movement. Here, the reference videos are only a few frames longer to insure that the target FoV is contained in the reference one. Also, in this test, the behaviour of the $\mathbf{W}_r$ matrix was observed.

The analysis of the $\mathbf{W}_r$ and $\mathbf{W}_t$ matrices in Figure 3.3 shows that it is possible

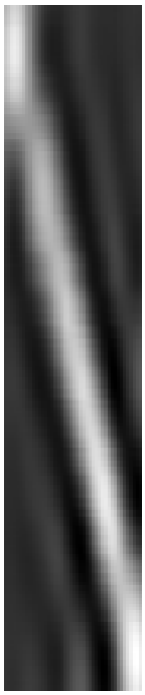
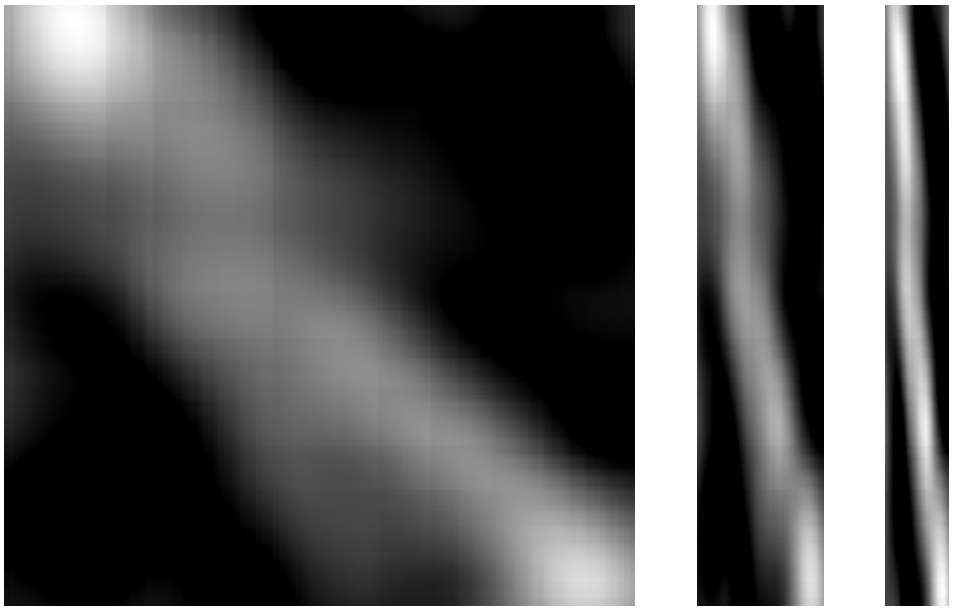


Figure 3.1: Matrix $\mathbf{W}_t$ generated with mcRoSuRe method. Brighter pixels denote higher values in the matrix. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The region from which the frames from the reference video were taken to reconstruct the target frames can be seen here.

to detect the position (in frames) where the direction change happened. In these images the "X"-like structures come from the fact that the both before and after the direction change in the video the camera covers the same path, therefore presenting similar FoV. As a consequence frames from both paths contribute for the reconstruction of frames in the other path, hence the symmetric structure in the $\mathbf{W}_t$ matric. It is clear, however, that an accurate assessment of the turning point cannot be made due to the abundance of high values on the $\mathbf{W}_t$ matrix around that position. In contrast, when observing the $\mathbf{W}_r$ matrix, the change of direction in the reference video can be detected with much greater accuracy.

Another important conclusion that could be drawn from the analysis of the $\mathbf{W}_r$ and $\mathbf{W}_t$ matrices in Figure 3.3 is that it is only necessary that the reference video has one passing in the scene, since in a single passing in the scene the camera covers completely the field-of-view, therefore we will have all the frames available to reconstruct the target video. The multiple passings of the reference video only contributes to the increase of the computational complexity in the algorithm (as it is dependent of the size of $\mathbf{X}_r$) and makes the attempt of using matrix $\mathbf{W}_t$ to align reference and target videos more challenging.

With those analyses in mind the test was repeated using reference videos that had no direction change. Figure 3.4 shows the results. The behaviour of the $\mathbf{W}_r$



(a) 1:1 (b) 5:1 (c) 10:1

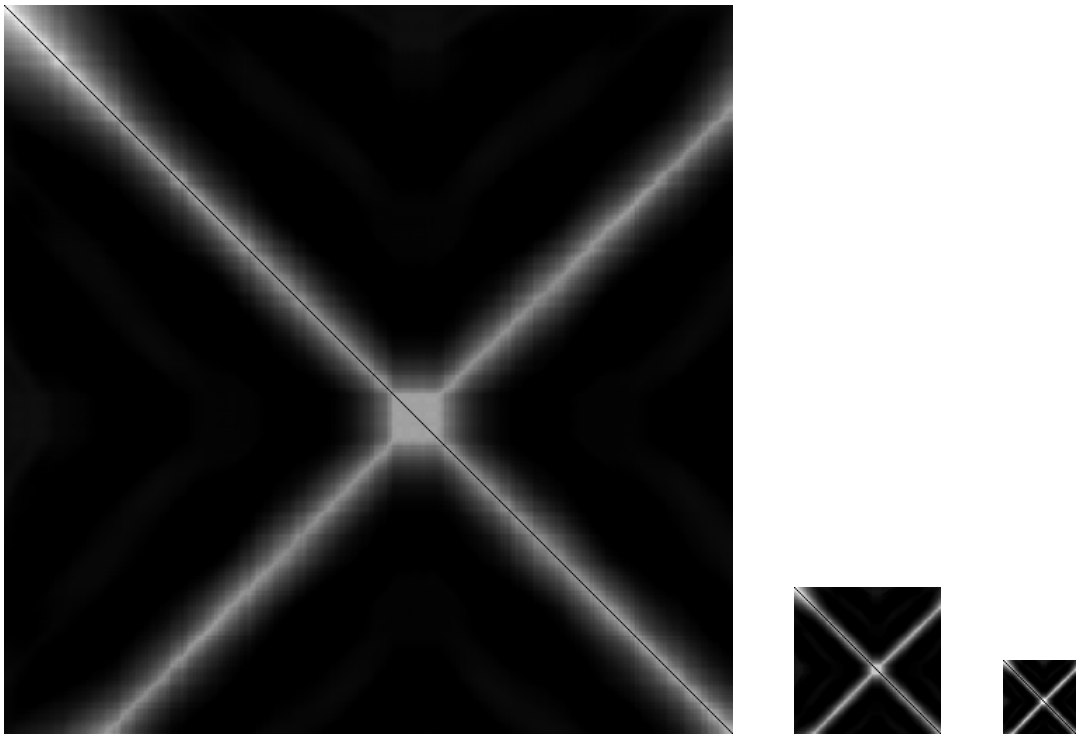
Figure 3.2: Matrices $\mathbf{W}_t$ generated with the mcRoSuRe method while downsampling the reference videos. (a)1:1 downsampling ratio. (b)5:1 downsampling ratio. (b)10:1 downsampling ratio. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The region from which the frames from the reference video were taken to reconstruct the target frames can be seen here.

and $\mathbf{W}_t$ matrices in this experiment reaffirms the conclusions presented before. The localization of the turning point in the target videos can be performed with more accuracy using this setup. On the other hand, the matrix is still not sharp enough and therefore the synchronization by inspecting matrix $\mathbf{W}_t$ is still not solved.

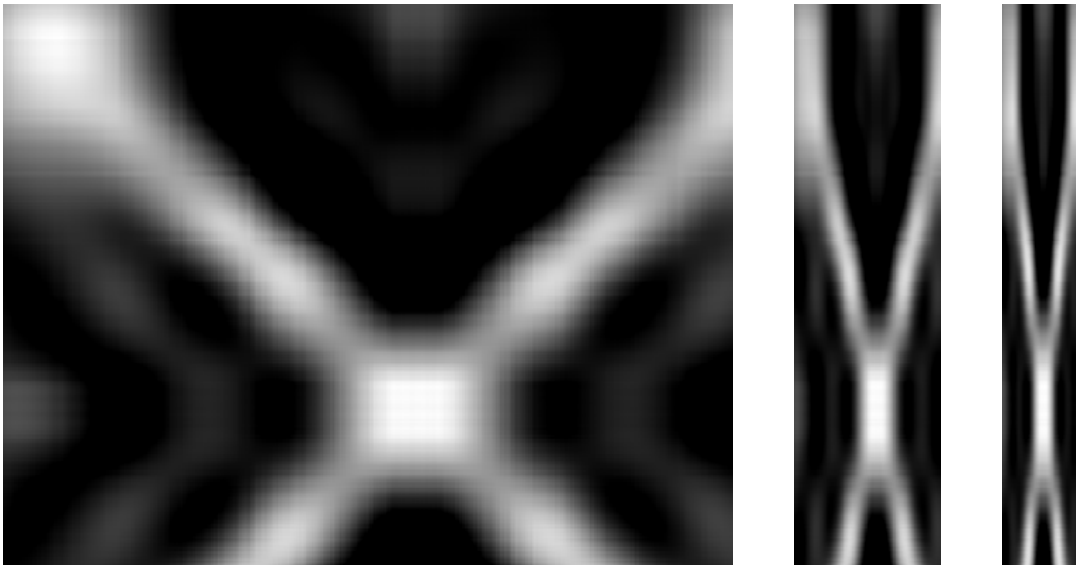
After performing several tests using reference and target videos that had no more than 400 frames new experiments were designed. Now reference videos of about 900 frames and target videos of about 100 frames were used. The goal of those tests was to detect the region in the reference videos that corresponded to the target ones. Again $\mathbf{W}_t$ matrix was to be used to extract the spatial information. The idea was to find, for every column of $\mathbf{W}_t$ matrix, its largest value and get the correspondence between reference and target videos from that. Figure 3.5 shows the results of these experiments.

Although it is possible to perceive up to some precision the region in reference video that corresponds to the target sequence, the method of selecting the columnwise maximum of $\mathbf{W}_t$ matrix fails with high probability, therefore the method is not suitable for locating the target background from the reference video.

One of the main reasons behind the failure to locate the frame correspondence between reference and target videos is that the high frequency components of the

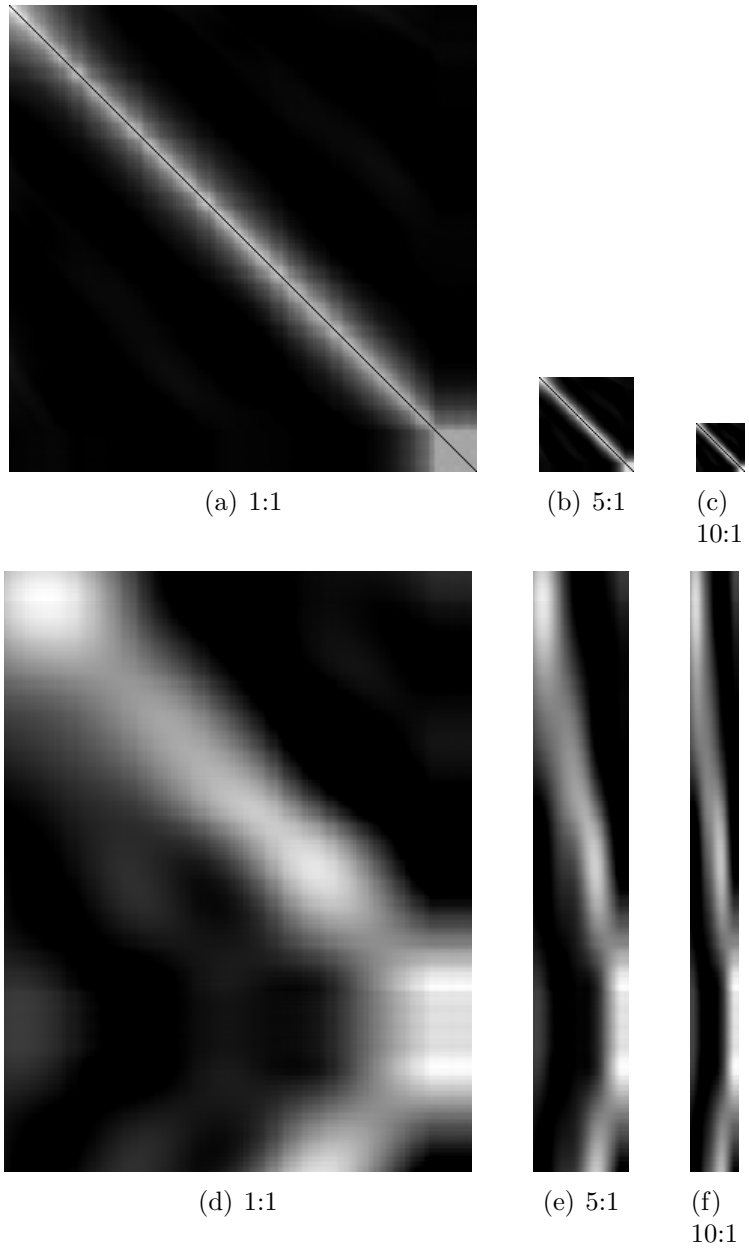


(a) 1:1 (b) 5:1 (c) 10:1



(d) 1:1 (e) 5:1 (f) 10:1

Figure 3.3: Matrices $\mathbf{W}_t$ generated with the mcRoSuRe method while downsampling the reference videos to test the localization of turning points (detection of moving direction changes). (a)$\mathbf{W}_r$ matrix 1:1 downsampling ratio. (b)$\mathbf{W}_r$ matrix 5:1 downsampling ratio. (c)$\mathbf{W}_r$ matrix 10:1 downsampling ratio. (d)$\mathbf{W}_t$ matrix 1:1 downsampling ratio. (e)$\mathbf{W}_t$ matrix 5:1 downsampling ratio. (f)$\mathbf{W}_t$ matrix 10:1 downsampling ratio. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The frames where the moving direction changes can be found in every matrix up to some precision.

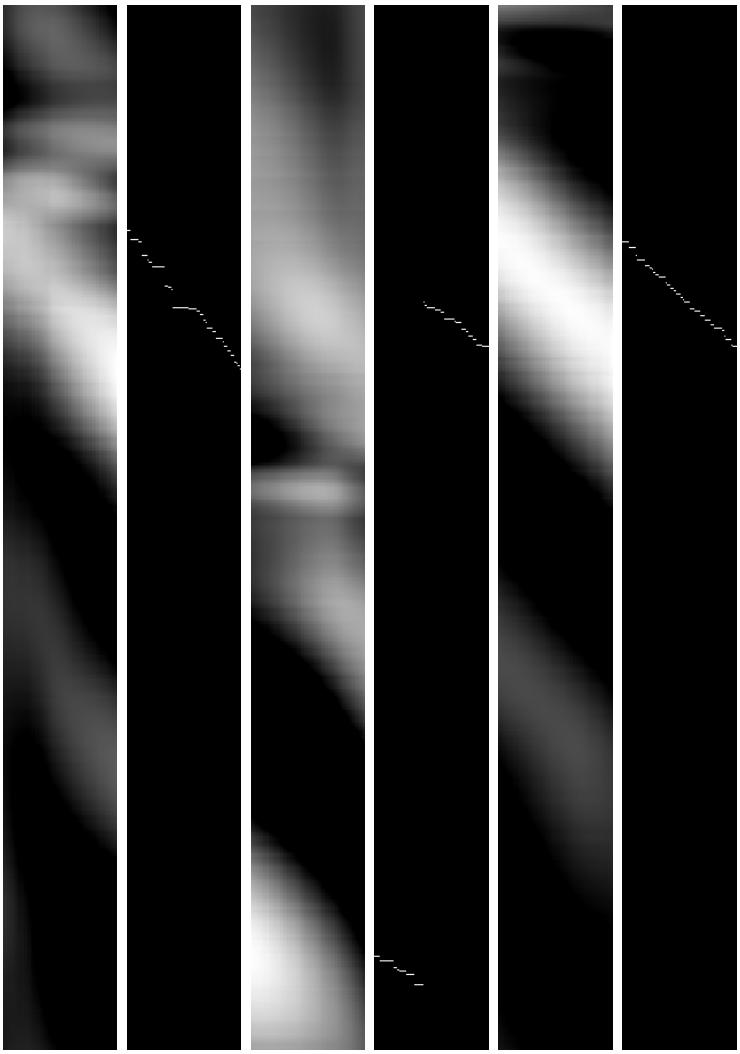


(a) 1:1 (b) 5:1 (c) 10:1

(d) 1:1 (e) 5:1 (f) 10:1

Figure 3.4: Matrices $\mathbf{W}_t$ generated with the mcRoSuRe method while downsampling the reference videos to test the localization of turning points (detection of moving direction changes). Now reference videos do not change movement direction. (a)$\mathbf{W}_r$ matrix 1:1 downsampling ratio. (b)$\mathbf{W}_r$ matrix 5:1 downsampling ratio. (c)$\mathbf{W}_r$ matrix 10:1 downsampling ratio. (d)$\mathbf{W}_t$ matrix 1:1 downsampling ratio. (e)$\mathbf{W}_t$ matrix 5:1 downsampling ratio. (f)$\mathbf{W}_t$ matrix 10:1 downsampling ratio. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The region from which the frames from the reference video were taken to reconstruct the target frames can be seen here.

reference video are important to make the precise correspondences between both videos. Since matrix $\mathbf{L}_r$, that is used to decompose the target video and is related with the distribution of $\mathbf{W}_t$, is composed only by low-frequency components of $\mathbf{X}_r$,



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

Figure 3.5: Experiment to locate target videos inside reference ones. (a),(c),(e) $\mathbf{W}_t$ matrix. (b),(d),(e) Columnwise maximum from $\mathbf{W}_t$ matrix. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. It is possible to perceive up to some precision the region in the reference video that correspond to the target. This process, however, it is not accurate, and the method of maximum location sometimes fail.

it may not be the ideal input to use in this application.

If the above assumption is correct, a possible solution is to use, in place of the mcRoSuRe $\mathbf{X}_t$ decomposition, the following proposal

$$\mathbf{X}_t = \mathbf{X}_r \mathbf{W}_t + \mathbf{E}_t. \tag{3.1}$$

In this equation the problem with high-frequency terms is solved. An example of $\mathbf{W}_t$ matrices obtained for some videos using the new decomposition proposal instead of that from mcRoSuRe is shown in Figure 3.6. It can be seen in this example, as

compared to the results shown in Figure 3.5, that the localization becomes easier and the maximum yields a better estimate of the correspondences between the frames of both reference and target videos.

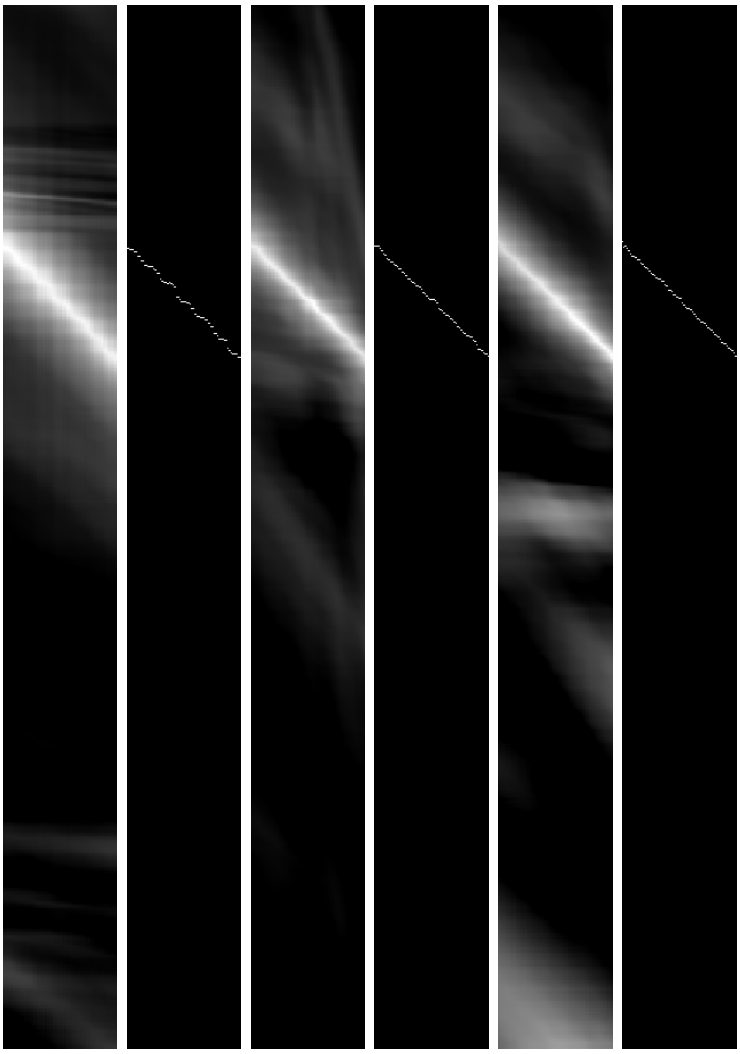


Figure 3.6: Experiment to locate target videos inside reference ones using the newly proposed decomposition of Eq. 3.1. Using the proposed decomposition instead of that from the mcRoSuRe method. (a),(c),(e) $\mathbf{W}_t$ matrix. (b),(d),(e) Columnwise maximum from $\mathbf{W}_t$ matrix. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The detection with the newly obtained $\mathbf{W}_t$ shows the improvement in the capacity of finding the correspondences between the frames of reference and target videos.

After analysing those experiments the newly proposed decomposition scheme was selected as being the most suitable for the video alignment, since it presents the sharper looking $\mathbf{W}_t$ matrix, thus allowing a more precise alignment between reference and target videos.

3.1.2 Pre-processing

One of the steps of the traditional framework on moving-camera anomaly detection is the pre-processing stage. There is a variety of methods available for the pre-processing of videos. In this work some techniques were investigated intending to make the reference and target videos more similar to each other so that the algorithm would not make false detections in the case of illumination variation of even small irrelevant artifacts.

Two experiments were performed to test the best way to pre-process the video data:

- Global video illuminance normalization;
- Frame-by-frame video illuminance normalization;

The first attempt was to normalize of the variance of the whole video, using an average of every frame luminance. To do so for each entry $x_{i,j}$ in the i-th line j-th column of the $\mathbf{X}$ $m \times n$ video matrix for both reference and target video we did:

$$x_{i,j} = \frac{x_{i,j} - \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j}}{\max_{i,j} x_{i,j} - \min_{i,j} x_{i,j}}. \tag{3.2}$$

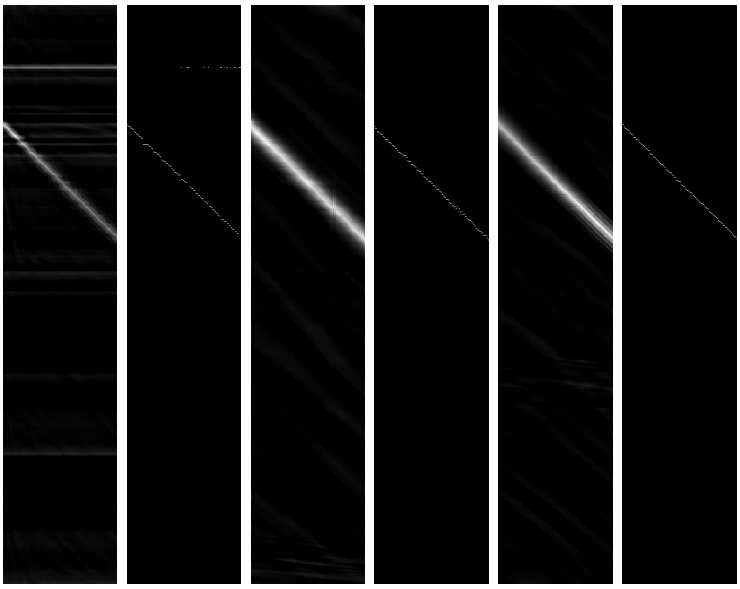
Some tests were performed to assess the validity of this approach. The results are depicted in Figure 3.7. It is possible to note in these results, that the proposed pre-processing normalization yields better results than the original ones, depicted in Figure 3.6, as the $\mathbf{W}_t$ matrix becomes more sparse and the correspondence between reference and target frames more precise.

The main reason behind the improvement obtained by the use of this method is that by normalizing the luminance of each frame one is able to correct some minor illumination differences between reference and target videos. This makes both videos more similar and therefore allows the algorithm to find better correlations between the frames of both videos.

Another version of the pre-processing technique that could be used to improve the detection results is to perform a frame-by-frame luminance normalization. In this approach for each entry $x_{i,j}$ in the i-th line j-th column of the $\mathbf{X}$ $m \times n$ video matrix for both reference and target video we did:

$$x_{i,j} = \frac{x_{i,j} - \frac{1}{n} \sum_{j=1}^{n} x_{i,j}}{\max_{j} x_{i,j} - \min_{j} x_{i,j}} \tag{3.3}$$

Figure 3.8 shows the results of the experiments concerning this pre-processing technique. In the figure it is possible to note that the $\mathbf{W}_t$ matrix becomes more sparse and thus the correspondence between frames from reference to target videos becomes sharper and more precise than the early approach. This is due to the fact



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

Figure 3.7: Experiment to locate target videos inside reference ones (using whole video luminance normalization). (a),(c),(e) $\mathbf{W}_t$ matrix. (b),(d),(f) Columnwise maximum from $\mathbf{W}_t$ matrix. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The detection with the newly obtained $\mathbf{W}_t$ shows that the correspondence between the frames from both videos becomes more precise and $\mathbf{W}_t$ becomes more sparse than the approach without pre-processing.

that the amount of light in the scene is dependant of the part of the environment. Therefore, when one uses a frame-by-frame normalization this illumination change is taken into account. This was considered the be the best normalization set-up between those considered and will, therefore, be used in the proposed algorithm.

3.1.3 Speed-up Techniques

Since the proposed algorithm uses large matrices in its computations and performs several matrices additions and multiplications it is expected that the bottleneck for the algorithm's processing speed is the dimension of the matrices used in the multiplications. In this section we aim to reduce the size of the reference matrix so that the resulting multiplications are smaller, therefore the algorithm is able to run faster.

The first step of this work looked for features to refine the $\mathbf{W}_t$ matrix. The modified approach shows a better correspondence between the target and reference video, allowing one to improve the localization of important frames from the reference video to represent the target. The following step investigated a way to use this advantage to ease the processing and allow the method to run with less computational effort.

The algorithm originally proposed in [46] uses the whole $\mathbf{X}_r$ data matrix as a

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

Figure 3.8: Experiment to locate target videos inside reference ones using the proposed frame-by-frame luminance normalization. (a)$\mathbf{W}_t$ matrix. (b)Columnwise maximum from $\mathbf{W}_t$ matrix. (c)$\mathbf{W}_t$ matrix. (d)Columnwise maximum from $\mathbf{W}_t$ matrix. (e)$\mathbf{W}_t$ matrix. (f)Columnwise maximum from $\mathbf{W}_t$ matrix. Brighter pixels denote higher values in the matrices. The horizontal dimension corresponds to the target video frames and the vertical dimension to the reference video frames. The detection with the newly obtained $\mathbf{W}_t$ shows that the correspondence between the frames from both videos becomes more precise and $\mathbf{W}_t$ becomes more sparse than with the previous pre-processing technique.

search space to obtain the frames that will be used to decompose the frames from $\mathbf{X}_t$. After analysing the $\mathbf{W}_t$ matrix one can notice that, if the reference video has more frames than the target (that is a common scenario) the algorithm is performing unnecessary work. Therefore, to allow the algorithm to process less content and yield similar results, an alignment step was proposed before the traditional algorithm.

To perform this step the decomposition presented in the previous section is to be performed as the first step of the algorithm, just after the video normalization. After the decomposition, the columnwise maximum from the $\mathbf{W}_t$ matrix is found.Then, one can select the region from the $\mathbf{X}_r$ reference matrix that shall be used in the rest of the algorithm by cropping matrix $\mathbf{X}_r$ using matrix $\mathbf{W}_t$ as a guide to form a new simpler matrix that will be called $\mathbf{X'}_r$ matrix.

In the $\mathbf{X'}_r$ matrix only the frames that correspond to the target video are present, thus there are fewer columns in $\mathbf{X'}_r$ than in $\mathbf{X}_r$, what will greatly reduce the resulting computational complexity. Figure 3.9 shows the difference between $\mathbf{W}_t$ matrices obtained with $\mathbf{X'}_r$ and $\mathbf{X}_r$. It is obvious in the figure the difference between both matrices, as the proposed $\mathbf{X'}_r$ has to find correspondences between much less frames. In the experiment depicted in the figure, the original $\mathbf{W}_t$ is a 200×5535 matrix and

the new $\mathbf{W}_t$ is 200×262 as some extra frames are kept before and after the region of interest for assurance.

After composing matrix $\mathbf{X'}_r$ and performing again the decomposition in Eq. (3.1), one obtains a matrix $\mathbf{W'}_t$ and a residual $\mathbf{E'}_t$. The resulting $\mathbf{W'}_t$ will look somewhat like Figure 3.9(b).

As in the original mcRoSuRe scheme, after performing the decomposition and obtaining the $\mathbf{E'}_t$ matrix, one then has to perform the decomposition of $\mathbf{E'}_t$ obtained from Eq. (2.15) with $\mathbf{E'}_r$ matrix (using Eq. (2.12) with $\mathbf{X'}_r$ matrix). Therefore, in the newly proposed scheme, besides the step in Eq. (3.1), three more steps are needed

$$\mathbf{X'}_r = \mathbf{L'}_r\mathbf{W'}_r + \mathbf{E'}_r, \tag{3.4}$$

$$\mathbf{X}_t = \mathbf{X'}_r\mathbf{W'}_t + \mathbf{E'}_t, \tag{3.5}$$

$$\mathbf{E'}_t = \mathbf{E'}_r\mathbf{W}_e + \mathbf{E}_e. \tag{3.6}$$

One example of the resulting residue matrix $\mathbf{E'}_t$ side by side with $\mathbf{E'}_e$ is shown in Fig. 3.10.

The proposed scheme implements a modification in the mcRoSuRe method that incorporates the alignment step in the detection. Using the proposed method obviates the need of an external temporal alignment step, usually present in the traditional moving-camera anomaly detection framework, as it performs said temporal alignment in its first step by cropping the reference video matrix $\mathbf{X_r}$ based on the inspection of the $\mathbf{W}_t$ matrix. The four step algorithm that is proposed is summarized in Algorithm 4. This method was published in [73] and is called mcRoSuRe-Time Alignment (mcRoSuRe-TA).

Algorithm 4 Object Detection Using Proposed mcRoSuRe-TA

$\min_{\mathbf{W}_t,\mathbf{E}_t} \|\mathbf{W}_t\|_1 + \lambda\|\mathbf{E}_t\|_1$, s.t. $\mathbf{X}_\mathrm{t} = \mathbf{X}_\mathrm{r}\mathbf{W}_\mathrm{t} + \mathbf{E}_\mathrm{t}$,
Crop reference frames of interest based on $\mathbf{W}_t$ matrix. Create $\mathbf{X'}_r$.
$\min_{\mathbf{W'}_r,\mathbf{E'}_r} \|\mathbf{W'}_r\|_1 + \lambda\|\mathbf{E'}_r\|_1$, s.t. $\mathbf{X'}_\mathrm{r} = \mathbf{L'}_\mathrm{r} + \mathbf{E'}_\mathrm{r}$, $\mathbf{L'}_r\mathbf{W'}_r = \mathbf{L'}_r$, $\mathbf{W'}_{r_{i,i}} = 0$
$\min_{\mathbf{W'}_t,\mathbf{E'}_t} \|\mathbf{W'}_t\|_1 + \lambda\|\mathbf{E'}_t\|_1$, s.t. $\mathbf{X'}_\mathrm{r}\mathbf{W'}_\mathrm{t} = \mathbf{X}_\mathrm{t} - \mathbf{E'}_\mathrm{t}$
$\min_{\mathbf{W}_e,\mathbf{E}_e} \|\mathbf{W}_e\|_1 + \lambda\|\mathbf{E}_e\|_1$, s.t. $\mathbf{E'}_\mathrm{r}\mathbf{W}_\mathrm{e} = \mathbf{E'}_\mathrm{t} - \mathbf{E}_\mathrm{e}$

3.1.4 Post-processing

As the pre-processing step, the post-processing has been commonly used in the traditional organization of the moving-camera anomaly detection algorithms. This step is responsible for formatting the output of the algorithm to yield the best possible results.

(a) (b)

Figure 3.9: By using $\mathbf{W}_t$ in Eq. (3.1) one can select frames of the reference video that correspond to the target video frames and create a smaller reference matrix $\mathbf{X'_r}$ that contains only the relevant reference frames for processing the target frames. Using such a smaller reference sequence saves a lot of computation. (a) Represents the $\mathbf{W}_t$ obtained using the whole reference matrix $\mathbf{X}_r$ (b) Represents the $\mathbf{W}_t$ obtained using the cropped version of the reference matrix $\mathbf{X'}_r$

(a) Residue matrix $\mathbf{E'}_t$.

(b) Residue matrix $\mathbf{E}_e$.

Figure 3.10: There are fewer undesired artifacts in the residue matrix $\mathbf{E}_e$ when compared with $\mathbf{E'}_t$.

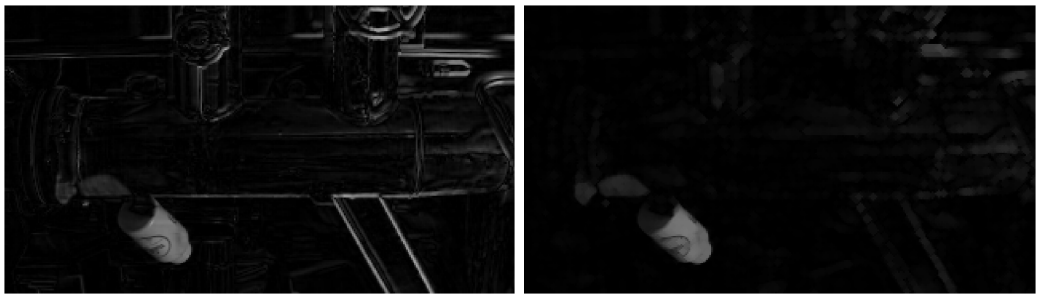
Until this point the algorithm outputs have shown good results in detecting the anomalies in the videos. A common problem has been the pose of the camera that changes from reference to target videos and make the algorithm to detect some high frequency artifacts which could not be removed by the use of the last decomposition of $\mathbf{E}'_e$.

A possible way to try to solve these problems is to apply a morphological filtering in the output of the algorithm trying to eliminate the very thin lines that are caused by the geometrical mismatching between the frames of reference and target videos. The most simple form of dealing with that task is to apply a morphological opening in the frame using a very small structuring element (possibly a disk of a few pixels).

The results of some tests involving this technique can be seen in Figure 3.11. In this figure it is possible to note that the impact of the artifacts is reduced by a great amount and the object is kept very much visible. There is, however, a drawback in the use of this technique that the object (or any anomaly) becomes more blurred. Nevertheless, this problem is not very severe since the residue image that outputs from the algorithm can be used as a detection mask after the end of the algorithm, thus a threshold can be applied to the output image to create a binary mask of detection.

3.1.5 Fast Subspaces Selection Interpretation

The original mcRoSuRe formulation does not require a precise frame-by-frame synchronization of the reference and target videos, but only that the area covered by the target video is contained within the area covered by the reference video excerpt. This is clear from the analysis of Eq. (2.12), where target-video data matrix $\mathbf{X}_t$ can be reconstructed by $\mathbf{L}_r$, the low-rank component of the reference video, up to a



(a) (b)

Figure 3.11: $\mathbf{E}_e$ matrices before and after the application of the morphological opening. It is possible to note that the thin artifacts fade away while the object remains visible.

sparse error $\mathbf{E}_t$. Using the modifications proposed in this section, which were originally introduced in the mcRoSuRe-TA paper [73], one could reduce the number of columns of $\mathbf{L}_r$ to include only those corresponding to the exact portion of the target video under analysis, great computational savings could be obtained. This is the same as saying that the UoS search space in the optimization problem described in Eq. (2.16) is restricted to a limited number of relevant subspaces.

Before mcRoSuRe-TA, the way of selecting these reference frames of interest is to observe the resulting $\mathbf{W}_t$ matrix in Eq. (2.15). This required, however, the computationally expensive implementation of the first two steps of the mcRoSuRe algorithm described in Eqs. (2.14) and (2.16), that are detailed in Algorithm 3. The proposed way to avoid this issue was to precompute $\mathbf{W}_t$ by representing the frames from the target video not as a combination of the low-rank representations of the reference frames $\mathbf{L}_r$, but as a combination of the actual reference frames $\mathbf{X}_r$. This proposition allows the construction of a version of the $\mathbf{W}_t$ matrix without the need to find the low-rank representation of the $\mathbf{X}_r$ matrix. This is, indeed, the most time costly step of the mcRoSuRe algorithm as will be shown later in the experimental results section. To perform this precomputation step one should compute the decomposition below [73]

$$\mathbf{X}_t = \mathbf{X}_r \mathbf{W}_t + \mathbf{E}_t. \tag{3.7}$$

This new added step requires solving the optimization problem defined by

$$\min_{\mathbf{W}_t, \mathbf{E}_t} \|\mathbf{W}_t\|_1 + \lambda \|\mathbf{E}_t\|_1, \text{s.t.} \quad \mathbf{X}_t = \mathbf{X}_r \mathbf{W}_t + \mathbf{E}_t, \tag{3.8}$$

whose implementation is summarized in Algorithm 5.

Selecting from $\mathbf{X}_r$ only the frames that correspond to the portion of the target video $\mathbf{X}_t$ being analyzed, one can execute the optimization steps represented by

Algorithm 5 Decomposition of $\mathbf{X}_t$ using $\mathbf{X}_r$ instead of $\mathbf{L}_r$.

Input: $\mathbf{X}'_r$, $\mathbf{X}_t$, λ, $\rho > 1$, η_1, η_2, μ_0, $\mathbf{W}_0 = \mathbf{E}_0 = \mathbf{Y}_0 = \mathbf{0}$.
while not converged **do**
$k = k + 1$
$\mathbf{X}'_{r(k+1)} = \mathbf{X}_t - \mathbf{E}_k$
$\mathbf{W}_{k+1} = \tau_{\frac{\lambda}{\mu_k \eta_1}} \left(\mathbf{W}_k - \frac{1}{\eta_1} \mathbf{X}_r^{\mathrm{T}} \left(\mathbf{X}_r \mathbf{W}_k - \mathbf{X}'_{r(k+1)} + \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
$\mathbf{E}_{k+1} = \tau_{\frac{1}{\mu_k \eta_2}} \left(\mathbf{E}_k - \frac{1}{\eta_2} \left(\mathbf{X_r} \mathbf{W}_{k+1} - \mathbf{X}'_{r(k+1)} + \frac{\mathbf{Y}_k}{\mu_k} \right) \right)$
$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \mu_k \left(\mathbf{X}_r \mathbf{W}_{k+1} - \mathbf{X}'_{r(k+1)} \right)$
$\mu_{k+1} = \rho \mu_k$
end while

Eqs. (2.14), (2.16), and (2.18), replacing $\mathbf{X}_r$ by a much smaller $\mathbf{X}'_r$ matrix, thus reducing the computational cost associated to the resulting algorithm.

3.2 mcRoSuRe-A Algorithm

As stated before, the mcRoSuRe algorithm shows great performance in the detection of abandoned objects in a cluttered environment, with a good detection performance and a reduced false-positive rate, as shown in [73]. However, the algorithm is computationally intensive and is therefore not suited for real-time applications. In fact, the computational complexity of the mcRoSuRe algorithm increases significantly with the size of the videos being analyzed (see Section 3.3 for a precise analysis). This explains the small video excerpts (70-frame long videos of 320×180-pixel frames) processed in [46]. To allow the reduction on the execution time of the algorithm, one may take advantage of some of its intrinsic properties concerning the resulting data representation. In this section new accelerating techniques that benefit from this innate representation, in addition to those introduced in the previous section and in [73], and modify the initial method are discussed.

3.2.1 Matrix Downsampling

The mcRoSuRe-TA [73] algorithm is able to reduce the amount of computation needed to perform the reconstruction of the target frames by reducing the number of subpaces where one should search for the correspondences of the reference frames. However, the proposed first alignment step of the algorithm became the more computationally complex part of the algorithm, since it is the only step to deal with large data structures that depends on the size of the reference videos, which are usually large.

In this section we propose to further reduce the computation complexity in this costly first step. To do so one should note that all the data lying in one of the

subspaces is considered to be similar, which can be understood when one accepts that a frame can be reconstructed by a low-rank representation of the reference videos. Therefore, it is expected that the error corresponding to represent a frame (column of video matrix) by another one inside the same subspace is expected to be smaller that the representation of the same frame by another, coming from a different subspace. Indeed, one could propose to use a different representation of the reference data matrix where a smaller number of frames lie in each subspace, and yet attain an equivalent alignment in the algorithm first step. This can be achieved by performing a uniform temporal downsampling of the original reference video, yielding a smaller, yet representative, reference data matrix $\mathbf{X}_r^{\text{ds}}$. If one observes the $\mathbf{W}_t$ matrices obtained by the decomposition performed by Eq. (2.16) with the original $\mathbf{X}_r$ and with $\mathbf{X}_r^{\text{ds}}$ it is possible to observe that the width of $\mathbf{W}_t$ matrix computed with $\mathbf{X}_r^{\text{ds}}$ is very much reduced in comparison with that obtained with the original $\mathbf{X}_r$ matrix. Nevertheless, the interval that relates to the frames of the target video is still clear, allowing a precise selection of the reference frames used to decompose the target video. Fig. 3.12 shows an example of the $\mathbf{W}_t$ matrices generated using the original $\mathbf{X}_r$ and decimated-in-time $\mathbf{X}_r^{\text{ds}}$.

(a) (b)

Figure 3.12: Example of resulting $\mathbf{W}_t$ matrices from Eq. (3.1) using: (a) complete reference data matrix $\mathbf{X}_r$; (b) downsampled-in-time reference matrix $\mathbf{X}_r^{\text{ds}}$.

From this figure, one can readily see the size discrepancy between the two approaches, which translates in a much reduced computational effort for the latter. Note that $\mathbf{X}'_r$ corresponds to an interval of $\mathbf{X}_r$; $\mathbf{X}_r^{\text{ds}}$ is only used to determine the limits of this interval.

3.2.2 Proposed Algorithm

With the addition of the proposed pre-processing step, that aligns the reference and target videos using a preliminary decomposition, the accelerated version of the mcRoSuRe approach becomes as summarized in Algorithm 6 and is called mcRoSuRe Accelerated (mcRoSuRe-A).

The proposed mcRoSuRe-A is similar to the previous mcRoSuRe-TA algorithm since all the steps are the same except the first one, that in the proposed framework is able to be performed in a fraction of the time, while also attaining the same results. It will be shown in the next section that the proposed modifications to the

Algorithm 6 mcRoSuRe - Accelerated

Downsample reference video to create $\mathbf{X}_r^{\mathrm{ds}}$.
$\min_{\mathbf{W}_t,\mathbf{E}_t} \|\mathbf{W}_t\|_1 + \lambda\|\mathbf{E}_t\|_1$, s.t. $\mathbf{X}_\mathrm{t} = \mathbf{X}_\mathrm{r}^{\mathrm{ds}}\mathbf{W}_\mathrm{t} + \mathbf{E}_\mathrm{t}$,
Crop reference frames of interest based on $\mathbf{W}_t$ matrix. Create $\mathbf{X'}_r$.
$\min_{\mathbf{W'}_r,\mathbf{E'}_r} \|\mathbf{W'}_r\|_1 + \lambda\|\mathbf{E'}_r\|_1$, s.t. $\mathbf{X'}_\mathrm{r} = \mathbf{L'}_\mathrm{r} + \mathbf{E'}_\mathrm{r},$, $\mathbf{L'}_r\mathbf{W'}_r = \mathbf{L'}_r$, $\mathbf{W'}_{r_{ii}} = 0$
$\min_{\mathbf{W'}_t,\mathbf{E'}_t} \|\mathbf{W'}_t\|_1 + \lambda\|\mathbf{E'}_t\|_1$, s.t. $\mathbf{X'}_\mathrm{r}\mathbf{W'}_\mathrm{t} = \mathbf{X}_\mathrm{t} - \mathbf{E'}_\mathrm{t}$
$\min_{\mathbf{W}_e,\mathbf{E}_e} \|\mathbf{W}_e\|_1 + \lambda\|\mathbf{E}_e\|_1$, s.t. $\mathbf{E'}_\mathrm{r}\mathbf{W}_\mathrm{e} = \mathbf{E'}_\mathrm{t} - \mathbf{E'}_\mathrm{e}$

original mcRoSuRe are capable of reducing the computations and yet produce the same results.

3.3 Computational Complexity Analysis

We now consider the number of arithmetic operations required to implement the different versions of the mcRoSuRe algorithm discussed in Section 2.4. For the calculation of the number of computations, the numbers of additions and multiplications were obtained from Algorithms 1, 2, and 5. For this analysis, let N_r and N_t be the numbers of $R\times C$-pixel frames in the reference and target videos, respectively, and let $P = RC$ be the number of pixels per frame, as indicated in Table 3.1.

Table 3.1: Variables related to the assessment of the computational complexity of the algorithms.

Variable	Related quantity
P	Total number of pixels in each frame
N_r	Number of columns (frames) in $\mathbf{X}_r$
N_t	Number of columns (frames) in $\mathbf{X}_t$
N_r^{ds}	Number of columns (frames) in $\mathbf{X}_r^{\mathrm{ds}}$
N'_r	Number of columns (frames) in $\mathbf{X}'_r$

The RoSuRe method, described in Algorithm 1, operates on $N_r \times P$ matrices, where each iteration requires

$$A(N_r, P) = 2N_r^2 + 5PN_r \tag{3.9}$$

additions and

$$M(N_r, P) = 4PN_r^2 + 2N_r^2 + 3PN_r + 7 \tag{3.10}$$

multiplications, which, in practice, is dominated by the $\mathcal{O}(PN_r^2)$ term [48].

The more computationally intensive mcRoSuRe algorithm, described in Algorithm 3, requires even more operations in each of its iterations, as given in Algorithm 2. In the first step, one has the RoSuRe algorithm with the associated $\mathcal{O}(PN_r^2)$ cost. The second and third mcRoSuRe steps, however, perform a distinct

optimization as given in Algorithm 2, which deals with $P \times N_r$, $P \times N_t$, and $N_t \times N_r$ matrices. With that in mind the method needs in each iteration

$$A(N_r, N_t, P) = N_r N_t + 8PN_r \tag{3.11}$$

additions and

$$M(N_r, N_t, P) = 3PN_r N_t + 3N_r N_t + 3PN_t + 7 \tag{3.12}$$

multiplications, which results in an overall cost of $\mathcal{O}(PN_r N_t)$.

The mcRoSuRe-A algorithm, introduced in Section 3.2 and described in Algorithm 6, creates an additional optimization step as summarized in Algorithm 5. Its first step considers an optimization on a downsampled reference video sequence containing $N_r^{\text{ds}} \ll N_r$ frames. Therefore the actual number of arithmetical operations for each iteration in this step is

$$A(N_r^{\text{ds}}, N_t, P) = N_r^{\text{ds}} N_t + 8PN_r^{\text{ds}} \tag{3.13}$$

additions and

$$M(N_r^{\text{ds}}, N_t, P) = 3PN_r^{\text{ds}} N_t + 3N_r^{\text{ds}} N_t + 3PN_t + 7 \tag{3.14}$$

multiplications, which is dominated by the $\mathcal{O}(PN_r^{\text{ds}} N_t)$ term. After this step, a new reference sequence is created with only $N_r' \ll N_r$ frames, corresponding to the original reference video excerpt used to reconstruct the target frames. The following steps use the same optimization described in Algorithm 2 but using $P \times N_r'$, $P \times N_t$, and $N_t \times N_r'$ matrices, requiring in the second step

$$A(N_r', P) = 2N_r'^2 + 5PN_r' \tag{3.15}$$

additions and

$$M(N_r', P) = 4PN_r'^2 + 2N_r'^2 + 3PN_r' + 7 \tag{3.16}$$

multiplications for each iteration and in the subsequent steps

$$A(N_r', P) = N_r' N_t + 8PN_r' \tag{3.17}$$

additions and

$$M(N_r', N_t, P) = 3PN_r' N_t + 3N_r' N_t + 3PN_t + 7 \tag{3.18}$$

multiplications for each iteration.

These operations lead to an overall cost of the order $\mathcal{O}(PN_r'^2)$ for the second step and $\mathcal{O}(PN_r' N_t)$ for the third and fourth ones, which once again is much smaller

than $\mathcal{O}(PN_r^2)$ and $\mathcal{O}(PN_rN_t)$ respectively, as $N_r' \ll N_r$.

A summary of the final computational complexities of the algorithms analyzed is given in Table 3.2. From the above analysis, one can infer that mcRoSuRe-TA and mcRoSuRe-A reduce drastically the resulting overall complexity when compared with mcRoSuRe, as verified quantitatively in Section 3.4.

Table 3.2: Computational Complexity per iteration of the evaluated methods (in number of multiplications).

Step	RoSuRe	mcRoSuRe	mcRoSuRe-TA	mcRoSuRe-A
1	$\mathcal{O}(PN_r^2)$	$\mathcal{O}(PN_r^2)$	$\mathcal{O}(PN_r^2)$	$\mathcal{O}(PN_r^{\mathrm{ds}}N_t)$
2	-	$\mathcal{O}(PN_rN_t)$	$\mathcal{O}(PN_r'^2)$	$\mathcal{O}(PN_r'^2)$
3	-	$\mathcal{O}(PN_rN_t)$	$\mathcal{O}(PN_r'N_t)$	$\mathcal{O}(PN_r'N_t)$
4	-	-	$\mathcal{O}(PN_r'N_t)$	$\mathcal{O}(PN_r'N_t)$

To provide numerical information on these computational complexities, we show here the figures associated with an example from Section 3.4-B. In this experimental scenario, based on a real-world application, $R = 320$, $C = 180$, yielding $P = 57600$, $N_r = 5000$ and $N_t = 200$. Using a typical value for the downsampling value one will have $N_r^{\mathrm{ds}} = 500$. Provided that the camera does not stop during the translational movement (common case in real applications) $N_r' = 210$ (the size of N_t plus a guard interval). For a list of the variables refer to Table 3.1.

With these values the mcRoSuRe method would have the following numbers of multiplications per iteration

- First step: $9.14 \cdot 10^8$ multiplications
- Second step: $1.73 \cdot 10^{11}$ multiplications
- Third step: $1.73 \cdot 10^{11}$ multiplications

while mcRoSuRe-A would have

- First step: $1.73 \cdot 10^{10}$ multiplications
- Second step: $1.02 \cdot 10^{10}$ multiplications
- Third step: $7.29 \cdot 10^9$ multiplications
- Fourth step: $7.29 \cdot 10^9$ multiplications

The gains in computation complexity in every step by using the proposed algorithm can be inferred from this example.

It would be expected that the use of sparse matrices would yield a lower number of additions and multiplications in the computational complexity analysis. However,

due to the fact that the $\mathbf{W}$ and $\mathbf{E}$ matrices are only considered sparse at the end of the algorithm we chose not to perform this analysis using the sparsity aware matrix multiplication complexity. Furthermore, since we are comparing the complexity of similar algorithms whose major changes are due to the dimensions of the matrices we considered that the analysis using the full matrix multiplication would be enough to give a figure of the complexity reduction.

3.4 Performance Evaluation

In this section, performances of the proposed algorithms are compared with algorithms for of the other mcRoSuRe family algorithms and also with the ones of other state-of-the-art anomaly detection using moving cameras.

In the first step the goal is to demonstrate that the proposed methods are less computationally complex than the mcRoSuRe while still achieving similar or better detection results.

In the second part the goal is to show the superior detection capability of the proposed algorithms in challenging scenarios involving moving cameras and complex datasets.

3.4.1 Experimental Assessment of the Proposed Algorithms

In a first experiment we compare the three versions of the mcRoSuRe algorithm: the original one summarized in Algorithm 3 (mcRoSuRe) [46] presented in Section 2.4, the one in Algorithm 4 (mcRoSuRe-TA) [73] presented in Section 3.1, and the accelerated version (mcRoSuRe-A) proposed in [74] and presented in Section 3.2 which uses a 10:1 decimated version of the reference video in the first step of the algorithm.

For comparison purposes, we evaluate the performances of these three versions when matrix $\mathbf{X}_r$ is composed by $N_r = \{5000, 1000, 200\}$ frames of a given VDAO reference video and $\mathbf{X}_t$ is comprised of $N_t = \{200, 200, 100\}$ frame excerpts, respectively, from each of the 59 single-object VDAO target videos. As for parameter initialization, we used: $\lambda = 1$, $\rho = 1.5$, $\eta_1 = 3$, $\eta_2 = 1.1\sigma_1(\mathbf{X}_r)$, and $\mu_0 = 1.25/\sigma_1(\mathbf{X}_r)$, where $\sigma_1(\mathbf{X}_r)$ is the largest singular value of input matrix $\mathbf{X_r}$, following the parameter selection of the mcRoSuRe method presented in [46].

Table 3.3 shows the time (in seconds) taken by each algorithm step when analyzing all videos in an Intel i7-3630QM with 2.4GHz and 16GB of RAM, running MATLAB ©2012b. From this table, it is noticeable how the proposed modifications accelerate the algorithm, particularly in the first step, which is the dominant one in the original mcRoSuRe version. Comparing the total running time of each algorithm, one notices how the proposed mcRoSuRe-A (using 10:1 downsampling

ratio) outperformed the other two, specially for longer video sequences where the acceleration factor becomes 2.6 with respect to the mcRoSuRe-TA and 100 with respect to mcRoSuRe.

Table 3.3: Time (in seconds) used by each step of the mcRoSuRe, mcRoSuRe-TA, and mcRoSuRe-A methods when analyzing the VDAO database with different reference/target video lengths.

Short Videos - $N_r = 200$ and $N_t = 100$ frames			
Step	mcRoSuRe	mcRoSuRe-TA	mcRoSuRe-A
1	44.09	17.45	9.77
2	16.70	**12.66**	13.57
3	16.15	**12.79**	12.78
4	-	14.29	**14.23**
Total	76.94	57.19	**50.34**
Medium Videos - $N_r = 1000$ and $N_t = 200$ frames			
Step	mcRoSuRe	mcRoSuRe-TA	mcRoSuRe-A
1	764.65	95.69	**29.05**
2	97.01	58.85	**58.56**
3	86.79	36.75	**35.83**
4	-	38.66	**36.35**
Total	948.46	229.97	**159.80**
Long Videos - $N_r = 5000$ and $N_t = 200$ frames			
Step	mcRoSuRe	mcRoSuRe-TA	mcRoSuRe-A
1	27333.18	537.20	**66.06**
2	529.32	**122.31**	124.43
3	477.92	50.55	**49.10**
4	-	**48.25**	52.14
Total	28340.42	758.31	**291.73**

It must be emphasized that this speed improvement occurs without hindering the system's detection capability. In fact, when one compares the outputs of both mcRoSuRe and mcRoSuRe-A methods, one readily observes that both methods have very similar (if not exactly equal) results, as depicted in Fig. 3.13. Similar results for the mcRoSuRe-TA method can be found in [73].

3.4.2 Abandoned Object Detection Algorithms Using Moving Camera

The performance of the proposed mcRoSuRe-A algorithm has been assessed by comparing it with those of some of state-of-the-art methods, such as the detection of abandoned objects with a moving camera (DAOMC) [6], the moving-camera

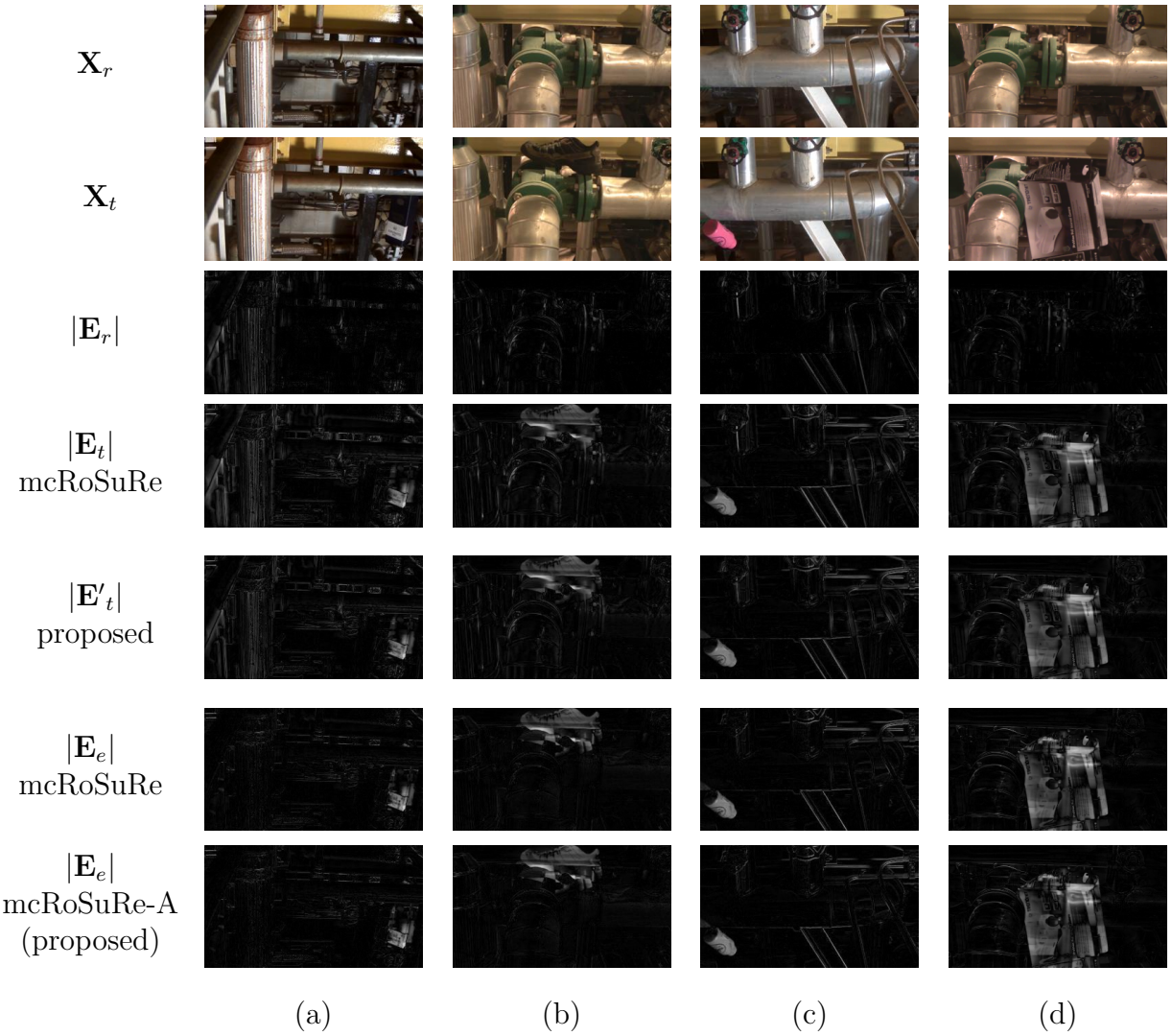


Figure 3.13: Comparative results for the mcRoSuRe and mcRoSuRe-A algorithms (single frames of matrices X_r, X_t, E_r, E_t, and E of both methods) for 4 different abandoned-object videos from VDAO [64] database: (a) blue box; (b) shoe (c) pink bottle; (d) camera box. The similar detection performance of both methods is clear from these experiments.

background subtraction (MCBS) [40], the spatio-temporal composition for moving-camera detection (STC-mc) [42], and the anomaly detector using multiscales (AD-MULT) [11]. To this end we used the annotated videos from the VDAO database. As the algorithms of [6, 40], and [42] could not be executed in a reasonable amount of time for the complete VDAO videos, only short-length 200-frame videos were employed.

The selected 200-frames video excerpts used in the experiments described in this paper are publicly available at [75]. The results of all experiments carried out with the competing methods, that are reported here, can also be found at [75].

For comparison purposes, the reference-target video synchronization was performed manually for the DAOMC algorithm. In our implementation of the DAOMC,

an NCC window small enough to detect all objects in the database was used to allow a fair comparison. For the MCBS algorithm, since the application of the method changed from a railway surveillance problem to a more general scenario, the post-processing steps that find the railway tracks were removed from the original algorithm. For the STC-mc the original author's implementation of the algorithm was used. For the ADMULT we compared with the results published in [11]. In addition, the results presented for the MCBS algorithm were obtained after the application of the optimized parameter configuration for the two similarity metrics used in the original paper [40], namely: the normalized vector distance (NVD) [76] and the radial reach filter (RRF) [77].

To obtain quantitative results for the mcRoSuRe-A algorithm, the output matrix $\mathbf{E}_e$ was post-processed with simple open and close morphological operations with 1 to 5 pixel-wide structuring elements. Also, simple binary thresholding was applied to obtain the final detection mask, its value of 0.32 was selected in a grid search from 0.25 to 0.75 using three randomly selected videos of the [75] dataset.

The performance for all methods was initially quantified with the following metrics: (i) True positive (TP) detection rate, where a TP occurs when the detection blob has at least one coincident pixel with the abandoned-object ground-truth bounding box; (ii) False positive (FP) detection rate, where an FP arises when the detection blob has all pixels outside the ground-truth bounding box; (iii) False negative (FN) detection rate, where an FN occurs when the ground-truth bounding box has no detected pixels inside it; (iv) True negative (TN) detection rate, where a TN is associated to a frame with no bounding box and no detected pixels. In addition, we consider the DIS parameter defined as

$$\mathrm{DIS} = \sqrt{(1-\mathrm{TP})^2 + \mathrm{FP}^2}, \tag{3.19}$$

which can be interpreted as the minimum distance of all operating points to the point of ideal behaviour (TP = 1 and FP = 0) in the TP×FP plane. The use of this metric allows direct comparison with the results in [42].

Even though the temporal consistency of the detections is paramount to the overall quality of the detection algorithms, we will not consider it as a metric in our experiments in order to make them compatible with previous publications in the literature.

In a first experiment, the same seven video excerpts of [42] were considered. Since those videos contain only frames with objects, only the TP and FP measurements are shown in Table 3.4 along with the distance parameter.

It is clear from the results in Table 3.4 that for those limited scenarios considered in [42] the mcRoSuRe-A method's performance is superior to most of the other

Table 3.4: Detection comparison of proposed mcRoSuRe-A method with that of STC-mc, DAOMC, MCBS, and ADMULT methods for the same seven videos extracts employed in [42].

Object	STC-mc			DAOMC			MCBS			ADMULT			mcRoSuRe-A		
	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS	TP	FP	DIS
Dark blue box 1	1.00	0.04	0.04	**1.00**	**0.00**	**0.00**	1.00	0.90	0.90	**1.00**	**0.00**	**0.00**	0.96	0.17	0.17
Towel	0.92	0.01	0.08	1.00	0.10	0.10	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**	0.99	0.47	0.47
Shoe	0.90	0.04	0.11	1.00	0.04	0.04	1.00	0.28	0.28	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
Pink bottle	0.99	0.13	0.13	1.00	1.00	1.00	1.00	0.96	0.96	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
Camera box	1.00	0.03	0.03	**1.00**	**0.00**	**0.00**	1.00	0.00	0.00	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
Dark blue box 2	0.37	0.42	0.76	1.00	1.00	1.00	1.00	0.10	0.10	**1.00**	**0.00**	**0.00**	**1.00**	**0.00**	**0.00**
White jar	0.29	0.64	0.96	1.00	0.10	0.10	1.00	0.99	0.99	**1.00**	**0.00**	**0.00**	1.00	0.75	0.75
Average	0.78	0.19	0.59	1.00	0.32	0.32	1.00	0.47	0.47	**1.00**	**0.00**	**0.00**	0.99	0.20	0.20

algorithms' and very close to that of the more recent ADMULT for all considered metrics. The average mcRoSuRe-A TP of 0.99 shows that in almost all the cases the algorithm is able to detect the presence of anomalies, with a somewhat low FP detection rate of 0.20. The DIS value of 0.20 indicates that the mcRoSuRe-A algorithm achieves a good balance between the TP and FP detections for this problem being second only to ADMULT's performance in the considered metric.

In a more extensive analysis, we considered the algorithm average performances for all 59 single-object VDAO videos, as given in Table 3.5. In these videos there are both frames with and without objects.

Table 3.5: Average detection comparison of proposed mcRoSuRe-A method with that of STC-mc, DAOMC, MCBS, and ADMULT methods for all 59 single-object videos of the VDAO database.

Method	TP	FP	TN	FN	DIS
STC-mc	0.18	0.38	0.59	0.82	0.90
DAOMC	0.83	0.43	0.54	0.17	0.46
MCBS	0.89	0.84	0.02	0.11	0.85
ADMULT	0.71	**0.28**	**0.63**	0.29	0.40
mcRoSuRe-A	**0.91**	0.33	**0.63**	**0.09**	**0.34**

By analyzing the results presented in Table 3.5 one notices that the mcRoSuRe-A method is consistently superior to the other four competing methods in every metric considered. The average mcRoSuRe-A TP detection rate is the only one above 0.90, while yielding one of the lowest average FP detection rate. Unlike most of the competing algorithms mcRoSuRe-A provides over 0.60 of TN detections. In the case of the VDAO database this is most challenging metric as even small changes in illumination and camera position can yield false detections. Finally the mcRoSuRe-A is the only one among the tested methods to have less than 0.10 average FN, providing the least amount of undetected anomalies. As a result for the complete VDAO-200 database the proposed algorithm has the lowest DIS result, being the only one below 0.40, among the tested algorithms.

In this experiment we used the parameter values tuned for the initial seven video experiment shown in Table 3.4 for all the compared methods. Since the videos in this experiment present more challenging features (as objects being occluded) and a larger variation in objects shapes and illuminations, not all methods kept their good results, notably the ADMULT that came of a DIS of 0.00 to 0.40. In contrast with most of the competing methods mcRoSuRe-A has shown to be robust to the challenges presented in this database having shown the smallest decrease in the performance when compared with the initial test results.

If one is not concerned with the identification of the anomaly position inside a given frame, but wants only to determine whether a frame presents an anomaly, a more relaxed version of the detection metrics can be used. By considering only a frame-level detection analysis, one may define a TP_{fl} (or FP_{fl}) by the presence of any detection blob in an anomalous (non-anomalous) frame and an FN_{fl} (or TN_{fl}) by the absence of a detection blob in an anomalous (non-anomalous) frame. Average results for these frame-level metrics for all four detection algorithms and for all 59 single-object videos from the VDAO database are shown in Table 3.6. In this experiment the results for the ADMULT method are not displayed since they were not available in the paper.

Table 3.6: Average detection comparison of proposed mcRoSuRe-A method with that of STC-mc, DAOMC, and MCBS methods for all 59 single-object videos of the VDAO database using frame-level metrics.

Method	TP_{fl}	FP_{fl}	TN_{fl}	FN_{fl}	DIS_{fl}
STC-mc	0.48	0.41	0.59	0.52	0.66
DAOMC	0.89	0.46	0.54	0.11	0.47
MCBS	**0.99**	0.98	0.02	**0.01**	0.98
mcRoSuRe-A	0.95	**0.37**	**0.63**	0.05	**0.37**

Table 3.6 leads to similar conclusions as Table 3.5. Since here the localization of the anomaly inside the frame is no longer an issue, then slightly misplaced detection blobs now count as a correct detection thus making the small objects more frequently detected by all methods, improving, for instance, the TP_{fl} mcRoSuRe-A measurement to 0.95. Although the TP_{fl} results for the MCBS method are superior to the ones of mcRoSuRe-A, it also yields 0.98 of FP_{fl} detection, showing it is unreliable for this type of application. On the other hand, the FP_{fl} also increased for all methods, as now only the frames where there are no anomalies count for this verification, thus making every error more relevant on the statistics.

Another test was performed using the multi-object videos from the VDAO database. Those videos are much more challenging than the single-object videos, as in this case, there are very small objects that can be hard to detect. Also, the

Table 3.7: Average detection comparison of proposed mcRoSuRe-A method with that of STC-mc, DAOMC, MCBS, and ADMULT* methods for the 9 multi-object videos of the VDAO database.

Method	TP	FP	DIS
STC-mc	0.67	0.74	0.81
DAOMC	**1.00**	0.68	0.68
MCBS	**1.00**	0.59	0.59
ADMULT*	0.71	0.42	0.51
mcRoSuRe-A	0.96	**0.25**	**0.25**

contrast of the videos is not as good as that of the single-object videos. The results of these experiments are summarized in Table 3.7.

Since in the multi-object videos each frame has at least two objects (as explained in Section 2.1), there are no TN frames. Thus, as a result, similarly to what happened with the 7-video tests, only the TP, FP, and DIS results are displayed. Unfortunately, by using the metrics that were chosen for the other experiments it is not possible to take into account the number of objects that were correctly detected in a frame with more than one object.

When analysing the results from this experiment, it is clear again that, in this more challenging scenario, mcRoSuRe-A presents more reliable results than the other compared methods. Although DAOMC and MCBS have better TP results, those two methods present much higher FP results as well, as can be seen by inspecting the DIS measurement in the last column of Table 3.7. In the original paper [11], only 3 of the 9 multi-object videos were used in this experiment.

Finally, the time performance of all the competing algorithms was compared using a computer with Intel i7-4790K with 4.0GHz and 32GB of RAM, running MATLAB ©2015a. Table 3.8 presents the total time taken by each algorithm to run the seven videos considered in Table 3.4. From these results, one can easily notice how the mcRoSuRe-A method is the fastest one, being able to run at least twice as fast as ADMULT and seven times faster than the other methods in this test.

3.5 Summary

This work presented a family of algorithms that use sparse representations for detecting anomalies in video sequences obtained from slow moving cameras. The proposed techniques project the acquired data from a reference (anomaly-free) video onto a union of subspaces, and select a small number of those subspaces that contain most of the information needed to reconstruct the target (possibly anomalous) video.

Table 3.8: Time (in seconds) used by algorithms STC-mc, DAOMC, MCBS, ADMULT, and mcRoSuRe-A methods when analyzing seven videos from the VDAO database.

	STC-mc	DAOMC	MCBS	ADMULT	mcRoSuRe-A
Dark blue box 1	433	265	50924	106	52
Towel	345	280	50403	105	38
Shoe	542	293	50427	112	38
Pink bottle	415	280	50170	121	38
Camera box	448	299	50238	115	45
Dark blue box 2	221	289	51740	114	38
White jar	248	282	49901	128	36
Average	379	284	50543	114	41

The present work has shown the efficiency of the mcRoSuRe-A method demonstrating that it is able to cope with challenging scenarios in much less processing time than the other methods in mcRoSuRe family, while attaining qualitatively similar results. Depending on the size of the videos, the method was shown to be able to run up to 2.6 times faster than mcRoSuRe-TA [73] and 100 times faster than the original mcRoSuRe [46] algorithm, placing it among the fastest methods for anomaly detection in moving-camera videos.

Extensive experiments were conducted comparing the mcRoSuRe-A detection performance with alternative state-of-the-art approaches using the challenging VDAO database. The algorithm was shown to perform well in this database attaining the best average performance in all tests, reaching an average rate of 0.91 of true positive detections and around 0.33 of false positive detection, having the best compromise among the tested methods.

Chapter 4

Review of Moving Object Detection in Dynamic Backgrounds

Another challenging application related to the detection of anomalies in videos is the detection of moving objects in the presence of moving backgrounds. The videos in this kind of scenario feature moving foreground objects int the presence of dynamic backgrounds, usually created by moving elements in the background such as moving water, clouds, trees or by slight movement of camera. The challenge arises from the fact that it is hard to differentiate the background movement from that of the foreground, leaving the classification of what an algorithm should detect in a gray area. As a general rule-of-thumb one can consider as foreground any element that last for a limited number of frames in the videos, whereas the background element are present in all the frames of the video, even if perturbed by visible or invisible elements throughout the video.

Due to its non-stationarity, many traditional solutions based on background subtraction fail to model the background, hence being unable to properly detect the objects. On the other hand, it would be possible to apply methods similar to those designed to detect static objects in videos acquired with moving cameras for such applications, since they possess superior modeling capabilities. However, the use of such algorithms tends to yield many false positive detections, since the background motion in these applications can vary in more unpredictable ways than that derived from the camera motion does.

This chapter presents a review of the state-of-the-art on moving object detection in presence of moving backgrounds, highlighting databases and algorithms designed for this application. Also, it presents some methods that were used as inspiration

to the algorithms developed in Chapter 5, as well as a few algorithms designed for saliency detection, which will be latter used in our algorithm proposal.

The chapter is organized as follows: Section 4.1 presents some databases designed to assess the performance of moving object detection algorithms; Section 4.2 presents moving object detection algorithms via constrained matrix decomposition, that will be used as inspiration to the developed algorithms in later chapters; Section 4.3 describes methods for saliency detection; Section 4.4 reviews some of the state-of-the-art moving object detection algorithms; finally, section 4.5 summarizes the discussion of this chapter.

4.1 Moving Objects with Moving Backgrounds Databases

Some databases were specifically designed to allow the comparison of moving object detection algorithms in the presence of moving backgrounds. In this section we describe three of those, that are broadly used in the literature, namely the UCSD, 2014 Change Detection.net and Singapore Maritime Dataset. Those datasets vary in level of difficulty and specificity of applications, as described below.

The University of California San Diego Background Subtraction Dataset (UCSD) [78] consists of 18 video sequences recorded in many different scenarios and featuring distinct types of moving background. The types of moving foreground objects vary among animals, pedestrians, swimmers, people practicing sports, cars, planes, motorcycles, and other examples, and the moving backgrounds are related to camera motion, waves, snow, haze, smoke, among others. The foreground objects movement is smooth and continuous, with no occlusions by the background. A manually annotated binary groundtruth mask is provided for 15 of the 18 available videos (as of November/2018). Over 70% of the frames present some foreground object. The number of frames per video varies from 30 to 246. The resolution of the videos varies, with the smallest being 232×152 pixels and the largest 468×348 pixels. Figure 4.1 displays sample frames from each video in the dataset.

The CDNET [63], that was briefly mentioned in Chapter 2, also presents moving objects in videos with moving backgrounds. The dataset is divided into 11 categories. The subset called "dynamic background" presents the videos with the desired qualities for our application. Differently from the UCSD dataset, in the CDNET dataset, the videos include occlusions, intermittent movement and also a great variability in the size of the foreground objects. The moving objects in this dataset comprise boats, cars, trucks, and pedestrians, while the moving background is mostly composed of water surfaces. A manually annotated binary groundtruth

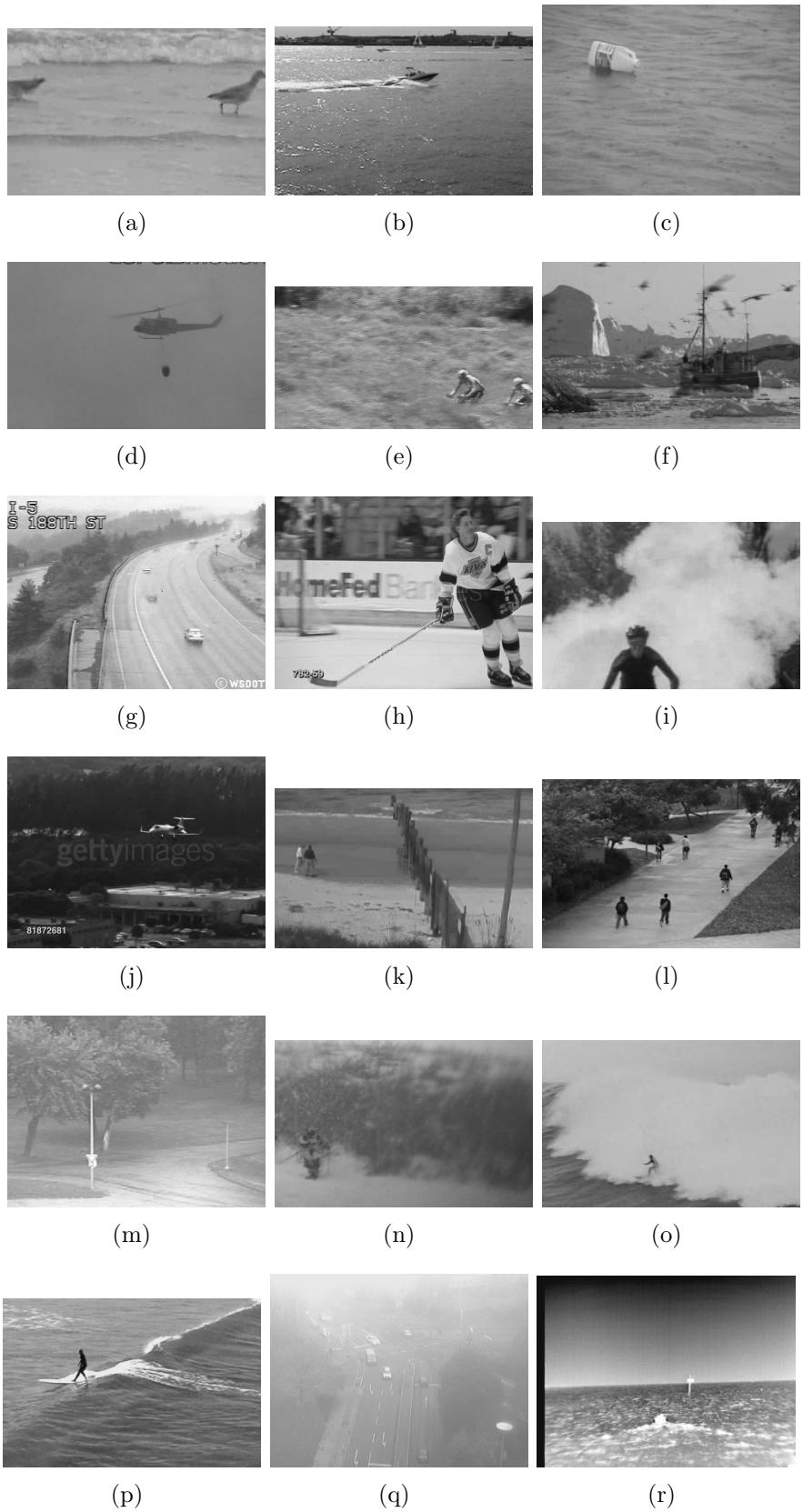
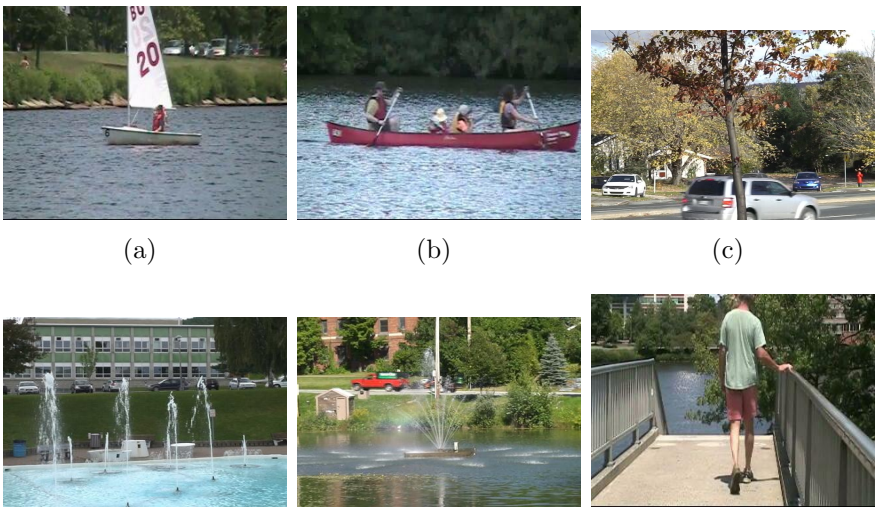


Figure 4.1: Sample frames from the UCSD dataset. (a) Bird; (b) Boats; (c) Bottle; (d) Chopper; (e) Cyclists; (f) Flock; (g) Freeway; (h) Hockey; (i) Jump; (j) Landing; (k) Ocean; (l) Peds; (m) Rain; (n) Skiing; (o) Surf; (p) Surfers; (q) Traffic; (r) Zodiac.

is available for every video in the dataset. In each video only a few frames have moving objects, while the great majority features only the moving background. The number of frames per video varies from 1184 to 7999. The frame resolution of the videos varies from 320×240 up to 70×480 pixels. Figure 4.2 displays sample frames from each video in the "dynamic background" subset of the dataset.



(a) (b) (c)

(d) (e) (f)

Figure 4.2: Sample frames from the Change Detection.net dataset. (a) Boats; (b) Canoe; (c) Fall; (d) Fountain01; (e) Fountain02; (f) Overpass;

The Singapore Maritime Dataset (SMD) [79] features a very specific set of scenarios for moving foreground detection, as the dataset displays videos from naval applications. The dataset is divided into three subsets: visible on-shore dataset, visible on-board dataset, and near-infrared on-shore dataset. The visible on-shore dataset features 40 videos from the ocean acquired from the shore. The visible on-board dataset features 11 videos from the ocean acquired from a boat. The near-infrared on-shore dataset features 30 videos from the ocean acquired from the shore, using a near-infrared spectrum camera. All videos come with groundtruth annotations of the horizon and the moving foreground objects. The frame resolution of each video is 1920×1080 pixels. To construct this dataset the videos were acquired in different weather conditions and during different times of the day. Figure 4.3 displays sample frames from each subset in the dataset.

In the experiments discussed in Chapter 5 we will use the videos from the UCSD and 2014 CDNET, since they have the most variations in scenarios and due to the fact that many other works have made their results available in such datasets. The SMD dataset was not used in our experiments as it has limited application and lack of other methods reporting their results on it.

(a) (b) (c)

Figure 4.3: Sample frames from the Singapore Maritime dataset. (a) Visible on-shore example; (b) Visible on-board example; and (c) Near-infrared on-shore example.

4.2 Constrained Matrix Decomposition Methods

Using the same concepts of subspace modeling and matrix decomposition discussed in Chapters 2 and 3, some methods in the literature discuss the viability of obtaining a more reliable detection of moving objects. In this section we discuss three methods that do so by means of constraining the objective functions using specially designed matrices imbued with some type of prior knowledge of the scene.

The three-term decomposition model (3TD) proposed in [80] introduces a three-term rank minimization problem that aims to detect moving objects in challenging applications. The method proposes to use a turbulence model to capture the structure of the background and imposes sparsity constraints to the moving foreground matrix. Thus, this method intends to solve the following problem

$$\begin{aligned} \min_{\mathbf{B},\mathbf{O},\mathbf{E}} \ & \text{Rank}(\mathbf{B}) + \tau\|\mathbf{\Pi} \odot \mathbf{O}\|_0 + \lambda\|\mathbf{E}\|_{\text{F}}^2, \\ \text{s.t.} \ & \mathbf{A} = \mathbf{B} + \mathbf{O} + \mathbf{E}, \end{aligned} \tag{4.1}$$

where τ and λ are weighting parameters, $\mathbf{O}$ is the object matrix, $\mathbf{E}$ is an undesired error matrix, $\mathbf{B}$ is the background matrix, and $\mathbf{A}$ is the video matrix (we chose to call the video matrix $\mathbf{A}$ in this application to differentiate from the reference and target videos in the previous chapters, as in the present application there is only one video). The $\odot$ symbol denotes the point-by-point matrix multiplication operator. The constraining confidence matrix $\mathbf{\Pi}$ is obtained as

$$\mathbf{\Pi} = \mathbf{1} - [vec(\mathbf{C}_1) \dots vec(\mathbf{C}_T)], \tag{4.2}$$

with $\mathbf{C}_i, i = 1, \dots, T$ being the confidence maps obtained via motion and intensity cues of the turbulence model and the operator $vec(\cdot)$ transforming a matrix into a vector by stacking all the columns of the matrix.

The use of this prior knowledge constraining confidence matrix $\mathbf{C}$ allows the method to avoid false positive detections, as it limits the possible updates of the object matrix $\mathbf{O}$.

Another method, namely the Robust Motion-Assisted Matrix Restoration (RMAMR), presented in [81], uses a different constraint matrix to perform a similar task. This method proposed to expand the RPCA formulation by adding prior information concerning the motion maps of the video, which are previously acquired and transformed into an update constraining matrix $\mathbf{\Theta}$. The problem formulation then becomes:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{F}} \|(\mathbf{B})\|_* + \lambda\|\mathbf{E}\|_1, \\ &\text{s.t. } \mathbf{\Theta} \odot \mathbf{A} = \mathbf{\Theta} \odot (\mathbf{B} + \mathbf{F}), \end{aligned} \tag{4.3}$$

where λ is a weighting parameter, $\mathbf{B}$ is the background matrix, $\mathbf{F}$ is the foreground matrix, and $\mathbf{A}$ is the video matrix. The constraining motion maps matrix $\mathbf{\Theta} \in \mathbb{R}^{mn\times k}$ (with k being the number of frames in the video with $m \times n$ frame resolution) is obtained as

$$\mathbf{\Theta}_{j,k} = \frac{1}{1 + \exp\left(\alpha\left(-\sqrt{(\mathrm{o}^{\mathrm{x}}_{\mathrm{j,k}})^2 + (\mathrm{o}^{\mathrm{y}}_{\mathrm{j,k}})^2} + \beta\right)\right)}, \tag{4.4}$$

with $o^x_{j,k}$ being an entry of $\mathbf{o}^x$, the $mn \times k$ matrix form of the horizontal motion fields obtained using the optical flow method [17] of all frames in $\mathbf{A}$, $o^y_{j,k}$ an entry of $\mathbf{o}^y$, the corresponding $mn \times k$ matrix form of the vertical motion fields, and α and β are tunable parameters.

With this constrained objective function the method is able to obtain iterative updates using the ADMM-ALM method, discussed in Chapter 2, and perform the object detection.

In [60], the authors propose to decompose a video matrix $\mathbf{A}$ into three matrices: $\mathbf{L}$ - Low-rank background model, $\mathbf{S}$ - sparse foreground, and $\mathbf{E}$ - residual detections. So using a method called Shape and Confidence Map-based RPCA (SCM-RPCA).

Inspired by the 3TD and RMAMR, the more recent SCM-RPCA goes a few steps further and proposes the simultaneous use of both restrictions, hence decomposing the data matrix as $\mathbf{A} = (\mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E})$.

In the SCM-RPCA implementation, the authors use only static information as the priors to constrain the object matrix, instead of using complex dynamic methods, such as dense motion maps, optical-flows, and turbulence maps. The SCM-RPCA proposes the use of a much simpler tool based on a visual saliency, which is explained in the next section. Although the matrix constraints do not take into account any temporal consistency, the method is still able to perform well in dynamic background scenarios, as reported in [60]. With the proposed constraining matrices the SCM-RPCA algorithm performs the background-foreground separation by decomposing

the video matrix as follows:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S},\mathbf{E}} \|\mathbf{L}\|_* + \lambda\|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \gamma\|\mathbf{E}\|_F^2, \\ \text{s.t. } \mathbf{A} = \mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E}, \end{aligned} \tag{4.5}$$

where the video contains k frames of dimensions $m \times n$, $\mathbf{\Pi} \in \mathbb{R}^{mn \times k}$ and $\mathbf{\Theta} \in [0,1]^{mn \times k}$ are the confidence map and shape constraints, respectively.

As explained in [60] the use of the confidence map and shape constraints reinforces the updates of the matrices focusing the attention on the most salient pixels, which are often related to the moving foreground. In order to solve this optimization problem the SCM-RPCA uses the ADMM-ALM method as in [81].

In Chapter 5 we will show the development of two object detection methods via constrained matrix decomposition that were inspired by the methods described in this section.

4.3 Saliency Detection

Saliency detection algorithms, as the one used to define the confidence maps of [60], are algorithms designed for predicting the position in an image where the human eye will most likely fixate when exposed to it. In this section we will review some of the non-biologically inspired methods to detect the salient areas of an image or video. These algorithms segment the images into blocks or superpixels and explore the relations between neighbouring blocks to find out which of those can be considered are more likely to call the human's eye attention, therefore being classified as salient. In this section we explore a brief review of some saliency detection algorithms that can be used to obtain the constraining region proposal matrices that feature an important step in the SCM-RPCA and in the our proposed algorithms that will be discussed in Chapter 5.

In [82] the authors propose a method called Graph-Based Saliency (GBVS). This algorithm proposes a three-step procedure to obtain the saliency masks. First it computes the feature maps by linear filters followed by elementary non-linearities. Second it creates the activation maps using a Markovian approach, which computes a dissimilarity measure of each node in the previous step and outputs those that are above a pre-set threshold. Finally, the activation map is normalized, and the output saliency mask is created.

The method called Geodesic Saliency is proposed by the authors of [83]. Differently from most methods, in this work the authors focus more on the background determination than the saliency. To do so they explore two common priors of background images: boundary and connectivity. Initially the method creates a weighted

graph of the image using as vertices image patches. Secondly it performs a connectivity check using a threshold based on the difference between the mean color of connected patches. Finally, the algorithm checks whether any of the salient patches belong to the image boundary; if they do, a third stage of the algorithm analyzes how likely a given boundary patch belongs to the image background.

The algorithm proposed in [84], uses a graph-based manifold ranking to obtain the saliency maps. The algorithm has two stages. In the first one it explores the boundary prior, by considering that all boundary image patches belong to the image background. Four maps are generated in this first step, by considering the boundaries of each image patch in one of the maps. The four labelled maps are then combined to form the first saliency map. In the second step the opposite procedure is performed, as a binary segmentation is applied to the patches that were labeled as foreground in the first step and the resulting foreground nodes are taken as salient queries. The nodes are then ranked based on their saliency using the ranking method proposed in [85] and the final saliency map is created.

The Boolean Map based Saliency Model (BMS) introduced in [86] implements a three-step method for saliency detection. This method explores the surroundedness cue for a robust saliency detection, that is, the method intends to find shapes that are surrounded by a continuous area of background. The first step of the method is the obtention of a Boolean map. The Boolean map is generated by randomly thresholding the input image's color maps using pre-computed intensity distributions as

$$\mathcal{B} = \text{thresh}(\phi(\mathcal{I}), \theta), \tag{4.6}$$

where θ is a preset threshold, $\phi(\mathcal{I})$ denotes a feature map of the image $\mathcal{I}$. The second step creates an activation map. This activation map ($\mathcal{M}(\mathcal{B})$) is constructed by determining the surrounded regions of the foreground in the Boolean maps. The maps are then separated into sub-activated maps that are obtained through logical AND operation between the original activation maps and both the Boolean maps and their logical complement, by doing the following operations

$$\mathcal{M}^{+}(\mathcal{B}) = \mathcal{M}(\mathcal{B}) \wedge \mathcal{B}, \tag{4.7}$$

$$\mathcal{M}^{-}(\mathcal{B}) = \mathcal{M}(\mathcal{B}) \wedge \neg\mathcal{B}, \tag{4.8}$$

where $\neg$ is the Boolean inverse (creating the logical complement), $\wedge$ is the pixel-wise Boolean conjunction operation, and $\mathcal{M}^{+}(\mathcal{B})$ and $\mathcal{M}^{-}(\mathcal{B})$ represent the selected and surrounded regions on $\mathcal{B}$ and $\neg\mathcal{B}$ respectively. The resulting sub-activation maps are normalized with the L2-norm and the result is further penalized with a morphological dilation operation, creating the $\mathcal{A}^{+}(\mathcal{B})$ relative to the sub-activation map $\mathcal{M}^{+}(\mathcal{B})$, and $\mathcal{A}^{-}(\mathcal{B})$ relative to the sub-activation map $\mathcal{M}^{-}(\mathcal{B})$. An average of

the resulting normalized maps is taken to compose the mean attention map, that is the algorithm's output which is obtained by doing

$$\mathcal{A}((B) = \frac{1}{2}\left(\frac{\mathcal{A}^{+}(\mathcal{B})}{\|\mathcal{A}^{+}(\mathcal{B})\|_2} + \frac{\mathcal{A}^{-}(\mathcal{B})}{\|\mathcal{A}^{-}(\mathcal{B})\|_2}\right). \tag{4.9}$$

The algorithm presented in [87] introduces the use of the deformed smoothness-based manifold ranking (DSMR), that uses a smoothness constraint to obtain the desired saliency map. First the input image is represented as a super-pixel graph. Secondly, a coarse saliency map is proposed using the DSMR method, using as background seeds the super-pixels from the image boundaries. An objectness map is created via an object proposal algorithm obtained from [88]. Lastly, the refined saliency map is created by integrating the information contained in the two maps by an iterative optimization refinement procedure described in [87]. The DSMR method, that is the core of the algorithm, consists of a constrained version of the manifold ranking method based on the object smoothness. The method reports to solve the issue of misclassified salient regions of the image due to low contrast with the background.

Finally, in the same line as the previous method, the work in [89] proposes the Probabilistic Saliency Estimation (PSE). This algorithm jointly optimizes saliency cues, such as boundary connectivity and smoothness constraints, to obtain the desired saliency maps. The algorithm finds a closed form solution of their cost-function based on graph-cut methods, by relaxing some of the constraints that make the original formulation an NP-hard problem. Using the resulting method the algorithm is capable of generating a robust smooth saliency map based on the original image super-pixel graph representation.

Due to its simple implementation, that is available at [90], and its reported fast performance, this method is used to generate the confidence maps of [60]. It will also be used in our implementation to obtain the attention matrices, as described in Chapter 5.

4.4 Other State-of-the-Art Methods

So far in this chapter we discussed some of the ways to solve the moving object detection in the presence of dynamic backgrounds problem via constrained matrix decomposition methods. However, many of the works in the literature approach this problem through other methods that feature completely different ideas.

This section is divided in two parts, the first one discusses supervised learning methods, that is, that learn their parameters by reinforced learning using part of the groundtruth annotation in a training step. The second part presents other methods

that do not rely on the existence of a training step to work.

4.4.1 Supervised Methods

Most of the supervised methods present deep neural network solutions to the detection problems. For instance, the work in [91] presents two deep learning approaches, namely FgSegNetS-S and FgSegNetS-M, to solve the problem of moving object detection in challenging scenarios. The authors propose an encoder-decoder approach using convolutional neural networks (CNN) and transposed convolutional neural networks to create the detection framework where each frame of the video sequence is input and a probability mask is obtained as the output of the system. The convolutional neural network model employed on the proposed systems is that of a pre-trained VGG-16 network [3]. The authors report good results in many different scenarios, stating that the proposed methods are robust to illumination changes, dynamic background, shadow occurrence, and camouflage effects.

The authors of [92] introduce the Cascade CNN method. This method aims to create an automatic groundtruth annotation system using as inputs a limited number of manually annotated frames of surveillance videos. The proposed system is composed by two connected CNNs. The first network is used to obtain a foreground probability map that is later concatenated with the original frame tensor (containing the RGB color channels) and input to the second neural network. The output of the second network is, thus, a refined foreground probability map. Both networks share the same architecture and during the training step of the method, the parameters of one network are fixed while the parameters of the other network are trained.

The method presented in [93], namely the DeepBS, uses a cascade of a CNN, plus multi-layer perceptron (MLP) [94], which are fully-connected neural network layers, and some post processing techniques to implement a real-time capable background subtraction method. The background model used as input for the CNN is obtained through a constantly updated library of background pixels as classified by an auxiliary network, namely the SubSENSE [95]. The CNN model in the DeepBS method uses only three convolutional layers and two MLP layers. The output of the MLP layers is post processed using median filters for temporal consistency.

Many other supervised learning methods have been proposed in the past few years. A survey that overviews many of those methods, including the already mentioned [49], is presented in [96].

Although the discussed methods report good results - some of them present the highest score across the results reported in the 2014 CDNET dynamic background website [63] - these methods rely on training for a specific scenario, therefore requiring individual setups for each application. An even greater problem arises from the

fact that those methods use a lot of data in the training phase, as reported in the paper that propose them, limiting their use when large amounts of annotated data are not available.

Table 4.1 summarizes the main techniques employed by each supervised learning method discussed in this section.

Table 4.1: Supervised learning algorithms and the techniques employed in them.

Method's Reference	Main Technique
FgSegNetS [91]	Encoder-decoder CNNs
Cascade CNN [92]	Connected CNN for FG and BG classification
DeepBS [93]	CNN feature extractor plus MLP classifier
HASSAN [16]	Gaussian-mixture model
CINELLI [49]	Deep Background Subtraction

4.4.2 Unsupervised Methods

The approaches employed by the unsupervised methods to solve the moving object detection problem present a much broader spectrum of methods. The big advantage of such methods, when comparing with the supervised ones, lies on the fact that these methods can perform without a training step, thus not requiring annotated groundtruth data for the parameter setups.

The work presented in [97], namely "In Unit There is Strength" (IUTIS), uses genetic programming to select the best of several change detection algorithms and composes a novel method that combines different methods strengths in a single algorithm. To perform such method the IUTIS implements a combination, using logical ANDs and ORs, of the outputs of different change detection methods, according to a specific fitness functions that is optimized on a set of benchmark dataset of video sequences. The resulting solution tree outputs the final algorithm that can be used in the detection of moving objects. A post-processing step is also created as a combination of geometrical erosion, dilation, median filter, logical OR, logical AND, and majority vote obtained in another genetic programming algorithm.

The Pixel-based Adaptive Word Consensus Segmenter (PAWCS) is proposed on [98]. This method presents a codebook approach that is updated using persistence-words to cope with background variations on long time periods. The so-called "background words" use texture and color information of the pixels to compose the background dictionary. There are two types of "background words" in the method, namely persistent and infrequent ones. The infrequent ones are constantly updated in the dictionary, while the persistent remain in the dictionary for long. The dictionary thresholds and learning rate are automatically adjusted during

the execution of the algorithm, based on the analysis of the background dynamics. A post-processing step based on simple morphological operations and median filters is applied at the output of the system to obtain the final masks.

The authors of [99] propose the Adapting Multi-resolution Background Extractor (AMBER), which was later expanded in [100]. The algorithm introduces a pixel-wise background subtraction model, where at every new frame the values of the pixel model are updated based on the "efficacy" of the values currently in the model, not by age (i.e. the number of iterations a value has remained in the model) as most methods. The resulting algorithm is designed for pixel-level parallelism which allows it to run on multi-processor hardware platforms in less time than similar algorithms. The classification of each pixel relies on template matching with the current pixels in the model: if the pixel does not match any of the current templates it is considered as foreground; on the other hand, if is matches any of the templates, it increases that template's "efficacy" and decreases that of the others. At any point if a template's "efficacy" becomes zero this template becomes inactive and a new template is created based on pixels that remained with the same values for long.

In [101] the Sliding Window-based Change Detection (SWCD) is proposed. The method performs change detection with background subtraction based on some key ideas: the background model is updated with a sliding window approach by using dynamic controllers presented in the SubSENSE paper [95], using a noise-free distance map to classify the pixels as foreground or background. Adaptive learning rates and thresholds for the distance maps are employed in the algorithm to deal with background changes and variations in the scene.

The change detection method introduced in [102], namely the CwisarDRP after the Wilkes, Stonham and Aleksander Recognition Device, presents a neural model that forgoes the initial training phase that characterize the supervised learning methods. It does so by employing a learning step after every classification step, creating a continuous learning process. Therefore, the method can perform the classification from the first frames of the video, in opposition to some methods that use the initial frames of the video to establish a background model. The core of the algorithm is the weightless neural network (WNN) called WisarD$^{\mathrm{RP}}$, whose core components are the discriminator: a layer of n-tuple neurons that map a set of n bits extracted from a binary input pattern called retina into values that are input into the network. In the learning phase the neurons have their values increased if they were stimulated by the retina or penalized if they were not. In the classification step each neuron outputs one if the stimulus was above a given threshold or zero if it was not.

The state-of-the-art methods discussed in this section are among the top-ranked unsupervised methods in the 2014 CDNET dataset. Their performances are really close to those of the supervised methods, presented in the previous section, and they

do not rely on the availability of large amounts of annotated data for training. In Chapter 5 we will compare the results of our methods with those discussed in this section for the experiments in the 2014 CDNET dateset.

Table 4.2 summarizes the main techniques employed by each unsupervised learning method discussed in this section.

Table 4.2: Unsupervised learning algorithms and the techniques employed in them.

Method's Reference	Main Technique
IUTIS [97]	Genetic Algorithm to combine methods
PAWCS [98]	Dictionary Learning
AMBER [100]	Efficacy-based background subtraction
SWCD [101]	Sliding window-based background modeling
CwisarDRP [102]	Continuously learning neural network for BG subtraction

4.5 Summary

This chapter introduced some concepts related to methods that detect moving objects in videos with moving backgrounds. We discussed some datasets that are used to assess the performance of such methods, as well as many methods that are able to perform this task. Some special attention was given to matrix decomposition methods, as they were used as inspiration to the algorithms for moving object detection we propose in the next chapter.

Chapter 5

Contributions to Moving Object Detection in Dynamic Backgrounds

Based on some ideas presented in the previous chapter and on the already extensively explored concepts of PSA, in this chapter we explore the possibility of using sparse plus low-rank matrix decomposition techniques to cope with the detection of moving objects in the presence of a moving background, as such described in Chapter 4.

The central idea of our proposed methods is to use the attention matrices (confidence map and shape constraint), obtained from the saliency detection as explained in the previous chapter, to enhance the classical structure of sparse plus low-rank matrix decompositions by restricting the updates on the sparse foreground matrix. These attention matrices work as *a priori* information that limits the possible regions where the moving foreground object can appear. The use of such constraining matrices allow the methods to separate the moving foreground detection into two sparse matrices, one responsible for the actual moving foreground and the other composed of the residue derived from poorly modeled moving background.

The proposed approach was inspired by a similar use of the saliency maps in [60], where the authors use two attention maps, namely $\mathbf{\Pi}$ - the confidence map and $\mathbf{\Theta}$ - the shape constraint, to modify the RPCA algorithm making it capable of performing background/foreground separation in the presence of dynamic background.

Two distinct algorithms were developed. First, a batch algorithm that processes the complete video data at once and uses all the frames of the video to perform the low-rank plus sparse decomposition. Second, a sequential algorithm that is able to process each frame as it arrives and outputs the detection based only on the current and previous frames.

This chapter is organized as follows: Section 5.1 describes the development of a

batch algorithm to solve the problem of moving foreground detection in the presence of moving background; Section 5.2 proposes a similar implementation of the algorithm using a sequential algorithm; Section 5.3 discusses the experimental evaluation of the proposed methods; and finally Section 5.4 presents our conclusions about the discussed topics.

5.1 Batch Algorithm

Both in Chapter 3 and in a previous work [73], we argued that the RoSuRe method is a more general and powerful tool for background/foreground separation in videos than RPCA, since it is capable of representing the background of a video as a union of subspaces instead of a single subspace. Therefore, we propose the following algorithm by applying the same saliency maps as [60] in a modified RoSuRe [48] framework.

5.1.1 Attention Matrices

In order to focus the algorithm attention to specific parts of the frame, avoiding the presence of false positive detections, a very common problem in this kind of application [80, 81], we use two types of matrices that either put different gains to different parts of the data matrices or mask it altogether with binarized thresholds. This idea was already introduced in Section 4.2 and in the SCM-RPCA paper [60].

Due to its effects we call the confidence map ($\mathbf{\Pi}$) and shape constraint ($\mathbf{\Theta}$) matrices Attention Matrices. The processes involved in obtaining them is described bellow.

If we consider a previously computed sequence of k saliency maps denoted by $M_1, M_2, \ldots, M_k$, where $M_i \in \mathbb{R}^{m \times n}$ obtained as the output of the saliency detection algorithm in [86], or some of the other saliency detection algorithms discussed in Section 4.3 to a k-frames long video, then we can obtain constraining matrices as in [60] by doing

$$\mathbf{\Pi} = [vec(norm(\mathbf{M}_1)) \ldots vec(norm(\mathbf{M}_k))], \tag{5.1}$$

$$\mathbf{\Theta} = [vec(\text{thresh}(\mathbf{M}_1)) \ldots \text{vec}(\text{thresh}(\mathbf{M}_\text{k}))], \tag{5.2}$$

where $norm(\cdot)$ is the min-max normalization, which scales every entry of $\mathbf{M}$ between 0 and 1, that is,

$$norm(\mathbf{M}_{i,j}) = \frac{\mathbf{M}_{i,j} - M_{min}}{M_{max} - M_{min}}, \tag{5.3}$$

with M_{min} and M_{max} being the minimum and maximum values of the $\mathbf{M}$ matrix, respectively. $\mathbf{M}_{i,j}$ is the element of the $\mathbf{M}$ matrix belonging to the row $i \in [1, \ldots, m]$

and column $j \in [1, \ldots, n]$, while the thresh$(\cdot)$ is defined as:

$$\text{thresh}(\mathbf{M}_{\text{i,j}}) = \begin{cases} 1 & \text{if } (0.5\mathbf{M}_{i,j})^2 < \mu \\ 0 & \text{otherwise} \end{cases}, \tag{5.4}$$

where $\mu = 0.5\eta(std(vec(\mathbf{M})))^2$, and $std(\cdot)$ denotes the standard deviation of a data vector. In this work, as in [60], η was experimentally chosen as 10.

For the experiments performed here, we obtain the attention matrices $\mathbf{\Pi}$ and $\mathbf{\Theta}$ using previously computed saliency maps obtained by applying the algorithm in [86] to every frame from the original video, which is represented as the matrix $\mathbf{A}$.

5.1.2 Tri-Sparse Decomposition

Using the rationale behind the PSA algorithms discussed in the previous chapters as inspiration, we want to obtain a decomposition of the data matrix $\mathbf{A}$, which represents the moving background with moving object video, in such a way that we can have the moving objects projected in a sparse matrix $\mathbf{S}$.

Also, due to the characteristics of the videos of interest in this application, there should be a matrix $\mathbf{L}$ which is a low-rank representation of the moving background. As in the other algorithms of the RoSuRe family, we expect the matrix $\mathbf{L}$ to be self-representative[1]. Therefore it is expected that there will also be a matrix $\mathbf{W}$ that is assumed to be sparse due to small number of subspaces that are needed to compute the representation of a frame using the low-rank surrogates. Also, due to the characteristics of the highly dynamic background of the videos, it is expected that frames that are temporally distant from each other will have different low-rank representations.

Using the video $m{\cdot}n{\times}k$ matrix $\mathbf{A}$, the $m{\cdot}n{\times}k$ low-rank background matrix $\mathbf{L}$, the $m \cdot n \times k$ sparse residue matrix $\mathbf{E}$, and the $m \cdot n \times k$ sparse foreground objects matrix $\mathbf{S}$ in combination with the attention matrices obtained in the previous section, we are able to write an initial optimization problem

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{E},\mathbf{S}} \|\mathbf{W}\|_1 + \lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2 \|\mathbf{E}\|_1, \\ &\text{s.t.} \begin{cases} \mathbf{A} = \mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \end{cases}, \end{aligned} \tag{5.5}$$

where $\mathbf{W}$ is the $k \times k$ weight matrix bearing the relations between the columns of

[1] As in the other cases the self-representative matrix $\mathbf{L}_r$ is guaranteed to be low-rank for a single subspace. For a UoS, as presented in this case, it is usually low-rank, but there may be cases where the construction of a specific UoS may not lead to a low-rank matrix $\mathbf{L}_r$. Nevertheless, as for making the notation of the methodology compatible with that of previous works, we will refer to $\mathbf{L}_r$ as either "low-rank" or "self-representative" matrix interchangeably.

the low-rank representation $\mathbf{L}$ of the data matrix $\mathbf{A}$, $\mathbf{W}_{ii}$ are the diagonal elements of the $\mathbf{W}$ matrix, $\mathbf{S}$ is the sparse matrix where foreground objects shall lie, $\mathbf{E}$ is the sparse residue matrix composed of the parts of $\mathbf{A}$ not represented by neither $\mathbf{L}$ nor $\mathbf{S}$, and λ_1 and λ_2 are weighting parameters.

To find a solution for this optimization problem, we employ the ALM and the ADMM as in [73]. This solution is detailed in the sequel. From Eq. (5.5), one has that

$$\mathbf{LW} - \mathbf{L} = \mathbf{0}, \tag{5.6}$$

and

$$\mathbf{L} = \mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}. \tag{5.7}$$

Replacing Eq. (5.6) into (5.7), we get

$$(\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E})\mathbf{W} - (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}) = \mathbf{0}, \tag{5.8}$$

which yields the following expanded cost function using the ALM-ADMM method [70]

$$\Gamma(\mathbf{W}, \mathbf{E}, \mathbf{S}, \mathbf{Y}_1, \mu_1) = \|\mathbf{W}\|_1 + \lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2 \|\mathbf{E}\|_1 + \langle \mathbf{LW} - \mathbf{L}, \mathbf{Y}_1 \rangle + \frac{\mu_1}{2} \|\mathbf{LW} - \mathbf{L}\|_F^2, \tag{5.9}$$

where $\mathbf{Y}_1$ is the augmented Lagrangian term and μ_1 is a weighting factor.

The updates for every matrix in this optimization can be shown, after some derivations (presented in the Appendix A), to be:

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \langle \mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.10}$$

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.11}$$

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \left[\lambda_2 \|\mathbf{E}\|_1 + \langle -\mathbf{E}\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_2^2 \right], \tag{5.12}$$

where $\hat{\mathbf{W}} = \mathbf{W} - \mathbf{I}$.

Using a similar development to that presented in [48], the final updates for the

$\mathbf{W}$, $\mathbf{S}$, and $\mathbf{E}$ matrices may then be written as:

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}}\left[\mathbf{W}_k + \frac{\mathbf{L}_{k+1}^{\mathrm{T}}\left(\mathbf{L}_{k+1} - \mathbf{L}_{k+1}\mathbf{W}_k - \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}}\right)}{\eta_1}\right], \tag{5.13}$$

$$\mathbf{S}_{k+1} = \tau_{\frac{\lambda_1^*}{\mu_{1(k)}\eta_2}}\left[\mathbf{S}_k + \frac{\left((\boldsymbol{\Theta}\odot\mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1} + \frac{(\boldsymbol{\Theta}\odot\mathbf{Y}_{1(k)})}{\mu_{1(k)}}\right)\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_2}\right], \tag{5.14}$$

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}}\left[\mathbf{E}_k + \frac{\frac{\mu_{1(k)}}{\mu_{2(k)}}\left(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1} + \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}}\right)\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_3}\right], \tag{5.15}$$

with, $\tau_\alpha[\cdot]$ being the soft-threshold operator defined in Eq. 2.11, and

$$\mathbf{L}_{k+1} = \mathbf{A} - \boldsymbol{\Theta}\odot\mathbf{S}_{k+1} - \mathbf{E}_{k+1}, \tag{5.16}$$

$$\mathbf{Y}_{1(k+1)} = \mathbf{Y}_{1(k)} + \mu_{1(k)}(\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}), \tag{5.17}$$

$$\mu_{1(k+1)} = \rho_1\mu_{1(k)}. \tag{5.18}$$

5.1.3 Large Residue Constraint

To assess the potential of the proposed algorithm, a few experiments were performed using a few videos from the moving background UCSD dataset [78].

The goal of these initial experiments was to detect whether the decomposition of the data matrix $\mathbf{A}$ into matrices $\mathbf{S}$, $\mathbf{L}$, and $\mathbf{E}$ was being performed in the correct manner. The second column of Figure 5.1 shows the $\mathbf{E}$ matrices, and corresponding $\mathbf{A}$ matrices for four frames using the proposed algorithm.

One can infer from the observation of the residue frames displayed in the second column of Figure 5.1 that the residue matrix $\mathbf{E}$ is too sparse and is not capturing properly the non-foreground abrupt changes that happen between neighbour frames in the data matrix $\mathbf{A}$. Indeed, the modeling of the low-rank matrix $\mathbf{L}$ and the detection of moving foreground objects in matrix $\mathbf{S}$ is subject to unwanted artifacts. To cope with this issue a modification in the optimization equation is proposed.

To guarantee that the sparse residue matrix captures the abrupt changes between neighbouring frames, therefore capturing the changes that cannot be modeled by the low-rank matrix $\mathbf{L}$ but also do not belong to the desired moving foreground, we propose to add a new constraint to the optimization function. This constraint requires that the columns of the residue matrix $\mathbf{E}$ corresponding to consecutive frames differ from each other by at least a minimum amount ϵ. This imposes a dynamical behavior on the $\mathbf{E}$ matrix, forcing it to capture some of the background

*The value of λ_1^* depends on the value of $\boldsymbol{\Pi}$ for the current point.

motion that would otherwise be represented in $\mathbf{S}$, which would cause misclassification of some background parts. The proposed new constraint is defined as follows

$$(\mathbf{E} - \mathbf{ED})\mathbf{1} \succeq \epsilon, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{0^T} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \tag{5.19}$$

where the $m \cdot n \times k$ unit matrix $\mathbf{1}$ has all entries equal to 1, and $\mathbf{0}$ is a k-dimensional null vector.

We executed again the experiment to check how much of the data matrix $\mathbf{A}$ is captured in the residue matrix $\mathbf{E}$ using the same frames as before. The new results are shown in the third column of Figure 5.1.

By inspecting Figure 5.1 one can clearly see that the newly proposed constraint makes the residue frames capture much more information from the highly dynamic background. This information was wrongly cast upon matrices $\mathbf{S}$ and $\mathbf{L}$ in the previous proposal, as explained in Section 5.1.2. The newly proposed constraint is incorporated to the algorithm from this point on.

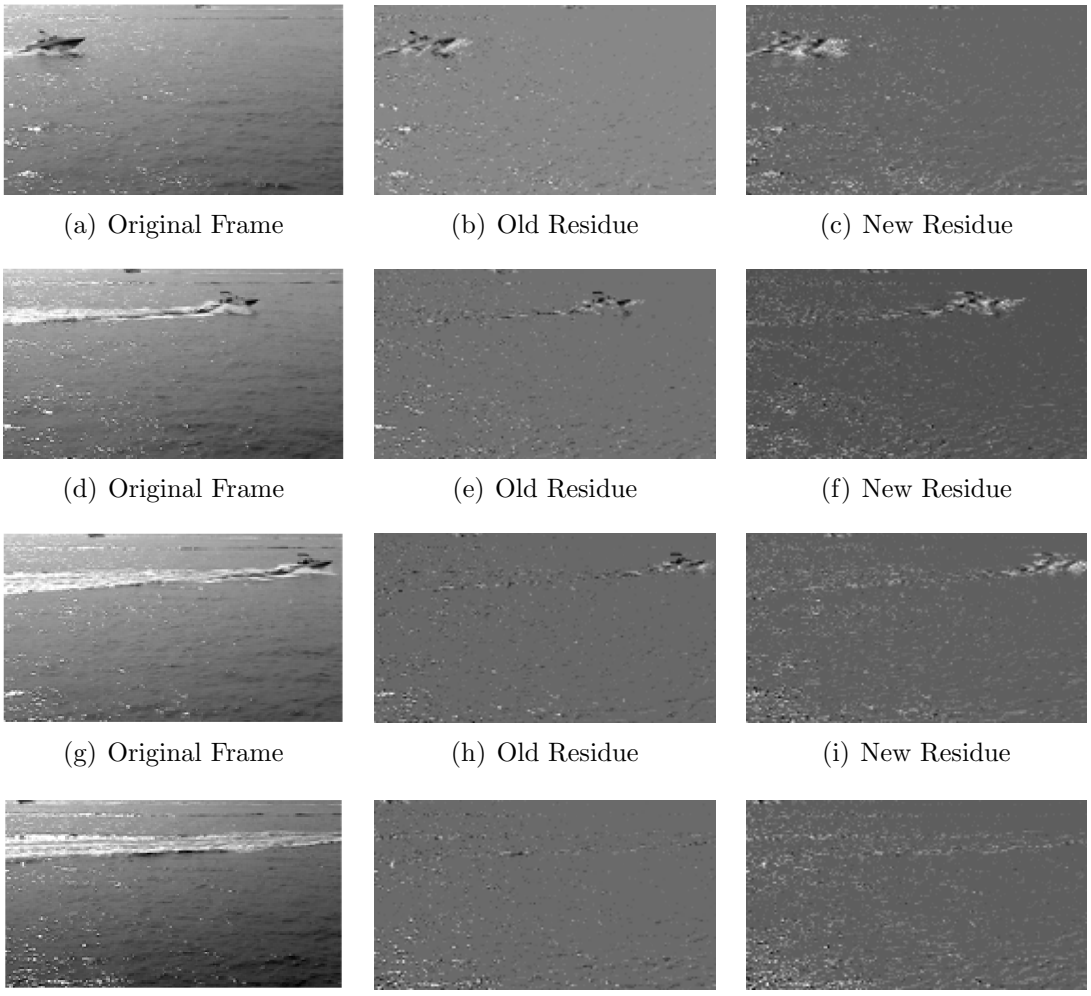
5.1.4 Proposed Algorithm

Incorporating the new residue matrix constraint, our goal becomes to solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{E},\mathbf{S}} \|\mathbf{W}\|_1 + \lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2 \|\mathbf{E}\|_1, \\ &\text{s.t.} \begin{cases} \mathbf{A} = \mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \\ (\mathbf{E} - \mathbf{ED})\mathbf{1} \succeq \epsilon, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{0^T} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \end{cases}, \end{aligned} \tag{5.20}$$

where $\mathbf{W}$ is the weight matrix bearing the relations between the columns of the low-rank representation $\mathbf{L}$ of the data matrix $\mathbf{A}$, $\mathbf{S}$ is the sparse matrix where foreground objects shall lie, and $\mathbf{E}$ is the sparse residue matrix composed of the parts of $\mathbf{A}$ not represented by neither $\mathbf{L}$ nor $\mathbf{S}$.

To find a solution for this optimization problem, we, again, employ the ALM and the ADMM as in [73], which yields the following expanded cost function using



(a) Original Frame (b) Old Residue (c) New Residue

(d) Original Frame (e) Old Residue (f) New Residue

(g) Original Frame (h) Old Residue (i) New Residue

(j) Original Frame (k) Old Residue (l) New Residue

Figure 5.1: New observation of residues from $\mathbf{E}$ matrix. The first column ((a),(d),(g), and (j)) shows the original frames from data matrix $\mathbf{A}$. The second column ((b),(e),(h), and (k)) shows the residue from the decomposition in residue matrix $\mathbf{E}$ in the original proposal. The third column ((c),(f),(i), and (l)) shows the residue from the decomposition in residue matrix $\mathbf{E}$ with the new constraint. One can observe that the new residue frames have more information than those in the previous proposal.

the ALM-ADMM method [70]

$$\begin{aligned}\Gamma(\mathbf{W},\mathbf{E},\mathbf{S},\mathbf{Y}_1,\mathbf{Y}_2,\mu_1,\mu_2) = \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi}\odot\mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1 + \\ \langle \mathbf{LW}-\mathbf{L},\mathbf{Y}_1\rangle + \langle \mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon,\mathbf{Y}_2\rangle + \\ \frac{\mu_1}{2}\|\mathbf{LW}-\mathbf{L}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_F^2, \quad (5.21)\end{aligned}$$

where $\mathbf{Y}_2$ is a Lagrangian term and mu_2 is a weighting parameter.

The updates for every matrix in this optimization can be shown, after some

derivations (presented in Appendix A), to be:

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \langle \mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.22}$$

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.23}$$

$$\begin{aligned} \mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \Big[& \lambda_2 \|\mathbf{E}\|_1 + \langle -\mathbf{E}\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)} \rangle + \\ & \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{2(k)}}{2} \|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \Big], \end{aligned} \tag{5.24}$$

Using a similar development to that presented in Section 5.1.2, the final updates for the $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{E}$ matrices may then be written as:

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}} \left[\mathbf{W}_k + \frac{\mathbf{L}_{k+1}^{\mathrm{T}} \left(\mathbf{L}_{k+1} - \mathbf{L}_{k+1}\mathbf{W}_k - \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}} \right)}{\eta_1} \right], \tag{5.25}$$

$$\mathbf{S}_{k+1} = \tau_{\frac{\lambda_1^*}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k + \frac{\left((\mathbf{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1} + \frac{(\mathbf{\Theta} \odot \mathbf{Y}_{1(k)})}{\mu_{1(k)}} \right) \hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_2} \right], \tag{5.26}$$

$$\begin{aligned} \mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}} \Bigg[& \mathbf{E}_k + \frac{\frac{\mu_{1(k)}}{\mu_{2(k)}} \left(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1} + \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}} \right) \hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_3} - \\ & \frac{\left(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon + \frac{\mathbf{Y}_{2(k)}}{\mu_{2(k)}} \right)(\mathbf{I} - \mathbf{D}^{\mathrm{T}})}{\eta_3} \Bigg]. \end{aligned} \tag{5.27}$$

where $\hat{\mathbf{W}}_k = (\mathbf{I} - \mathbf{W}_k)$, and

$$\mathbf{L}_{k+1} = \mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E}_{k+1}, \tag{5.28}$$

$$\mathbf{Y}_{1(k+1)} = \mathbf{Y}_{1(k)} + \mu_{1(k)}(\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}), \tag{5.29}$$

$$\mathbf{Y}_{2(k+1)} = \mathbf{Y}_{2(k)} + \mu_{2(k)}(\mathbf{E}_{k+1} - \mathbf{E}_{k+1}\mathbf{D}_{k+1} - \mathbf{1}\epsilon), \tag{5.30}$$

$$\mu_{1(k+1)} = \rho_1 \mu_{1(k)}, \tag{5.31}$$

$$\mu_{2(k+1)} = \rho_2 \mu_{2(k)}. \tag{5.32}$$

At the end of each iteration we modify the $\mathbf{W}$ matrix changing its diagonal entries to zero, similarly to what was done in the mcRoSuRe-A in Chapter 3.

*The value of λ_1^* depends on the value of $\mathbf{\Pi}$ for the current point.

5.2 Proposed Sequential Algorithm

The algorithm proposed in the previous section works by exploring the full potential of the matrix reconstruction, by using any frame from the low-rank representation matrix $\mathbf{L}$ to represent other frames. As will be shown later, by inspecting the generated structure matrix $\mathbf{W}$, most of the frames used to perform the reconstruction come from frames within a small temporal vicinity of the target frame, although this is not true in all cases. Moreover, in the proposed implementation, very frequently, a causal relation is not respected, meaning that posterior frames are used to reconstruct a previous frame.

In this section our goal is to obtain a modified version of the proposed algorithm that works in a sequential and incremental way, being therefore suitable for online and maybe even real-time applications. Our implementation is inspired on that of [103], that was later used in [104] to obtain a sequential version of the RoSuRe algorithm [48].

5.2.1 Proposed Algorithm

To obtain the new algorithm we start from the same optimization problem stated in Equation 5.20.

$$\min_{\mathbf{W},\mathbf{E},\mathbf{S}} \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi}\odot\mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1,$$
$$\text{s.t.} \begin{cases} \mathbf{A} = \mathbf{L} + \mathbf{\Theta}\odot\mathbf{S} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0,\ \forall i \\ (\mathbf{E} - \mathbf{ED})\mathbf{1} \succeq \epsilon, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{0^T} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \end{cases}. \tag{5.33}$$

At this point, instead of the results presented in Eq. (5.21), we employ a different new expanded function without the presence of the dual functions. This new expanded function features only the quadratic penalty term from the previously used ALM expansion. The newly proposed expanded cost function is obtained through the use of incremental subgradient-proximal methods described in details in [103] due to its strong convergence guarantees and stability, as explained in the referred article, since in our sequential implementation we will use few iterations to minimize the cost function. Therefore we have

$$\Gamma(\mathbf{W},\mathbf{E},\mathbf{S},\mu_1,\mu_2) = \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi}\odot\mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1 + \frac{\mu_1}{2}\|\mathbf{LW}-\mathbf{L}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_F^2. \tag{5.34}$$

The updates for every matrix in this optimization then becomes

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.35}$$

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{5.36}$$

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \left[\lambda_2 \|\mathbf{E}\|_1 + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{2(k)}}{2} \|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \right], \tag{5.37}$$

Using a similar development to that presented in Section 5.1.2, the final updates for the $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{E}$ matrices may then be written as:

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}} \left[\mathbf{W}_k + \frac{\mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1} - \mathbf{L}_{k+1}\mathbf{W}_k)}{\eta_1} \right], \tag{5.38}$$

$$\mathbf{S}_{k+1} = \tau_{\frac{\lambda_1^*}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k + \frac{(\mathbf{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_2} \right], \tag{5.39}$$

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}} \left[\mathbf{E}_k + \frac{\frac{\mu_{1(k)}}{\mu_{2(k)}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_3} - \frac{(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}})}{\eta_3} \right]. \tag{5.40}$$

where $\hat{\mathbf{W}}_k = (\mathbf{I} - \mathbf{W}_k)$, and

$$\mathbf{L}_{k+1} = \mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E}_{k+1}, \tag{5.41}$$

$$\mu_{1(k+1)} = \rho_1 \mu_{1(k)}, \tag{5.42}$$

$$\mu_{2(k+1)} = \rho_2 \mu_{2(k)}. \tag{5.43}$$

At the end of each iteration we modify the $\mathbf{W}$ matrix changing its diagonal entries to zero, similarly to what was done in the batch algorithm and in the mcRoSuRe-A in Chapter 3.

With those updates at hand, one is able to perform few iterations (even a single one can be used) of the algorithm for every new frame arriving. That is, every time a new column is added to the data matrix $\mathbf{A}$, we should update matrices $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{E}$ using Eqs. (5.38), (5.39), and (5.40), respectively.

*The value of λ_1^* depends on the value of $\mathbf{\Pi}$ for the current point.

5.2.2 Initialization and Multiple Iterations

To analyze how well this sequential version of the method was performing the reconstruction of frames using the previously obtained frames, a small experiment was performed. In this initial setup, we initialized the **A** matrix with 20 frames of the video and executed the algorithm for a total of 75 frames. The **W** matrix was initialized with random numbers obtained from a uniform distribution between zero and one. Figure 5.2 shows the resulting **W** matrix.

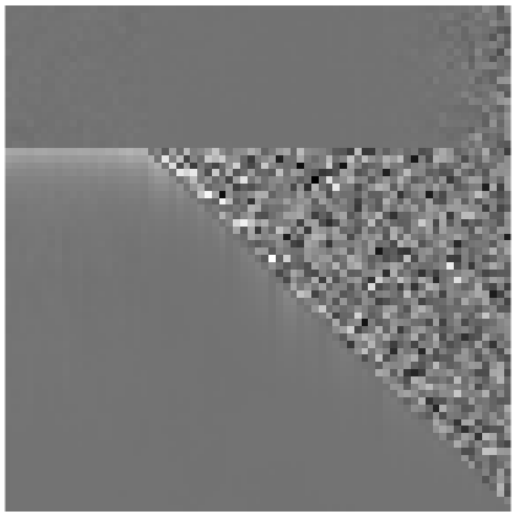


Figure 5.2: Resulting **W** matrix with random initialization. It is clear from the image that **W** matrix is far from being sparse and do not reflect the desired structure.

One can readily see by inspecting Figure 5.2 that the **W** matrix structure is far from what one would expect and also not sparse. This happens because with few iterations it is hard for the algorithm to overcome the initial trend of the matrix structure. Another way of initializing the **W** matrix was attempted, this time using a matrix whose entries were all ones as initialization for this matrix. The resulting **W** matrix is displayed in Figure 5.3.

The improvement in the sparsity of **W** with the initialization procedure shown in Figure 5.3 is easy to perceive. Also, the structure of the matrix is much closer to what one could expect with a previous knowledge of PSA algorithms such as the RoSuRe.

A second experiment was performed to determine the effect on **W** of the execution of multiple iterations of the updates from Eqs. (5.38), (5.39),and (5.40) every time a new column was added to data matrix **A**. Three setups were tested, first with 3 iterations per new frame, second with 10 iterations per new frame, and finally with 100 iterations per new frame. Again we initialized the **A** matrix with 20 frames of the video and executed the algorithm for a total of 75 frames, and used the initialization of the **W** matrix with all-ones entries. The resultant matrices can

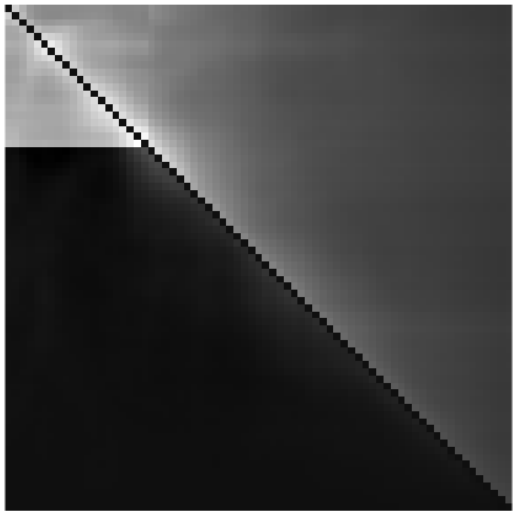


Figure 5.3: Resulting **W** matrix with initialized with all-one entries. It is clear from the image that **W** matrix is much sparser than with the previous initialization.

be seen in Figure 5.4.

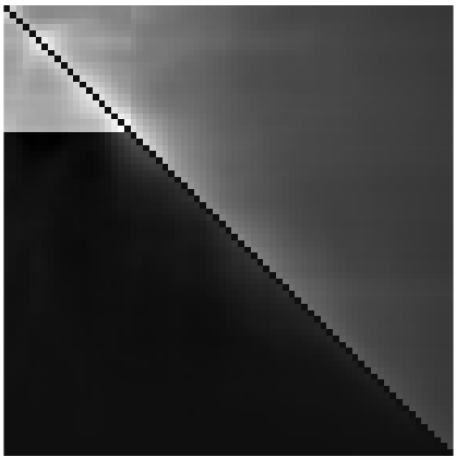
Observing Figure 5.4 it is easy to perceive the strong impact that the number of iterations per frame has on the sparsity and structure of the **W** matrix. The resulting matrix after 10 iterations for every frame already resembles that of the RoSuRe experiments presented in Chapter 2 and in [48], while after 100 iterations per frame the final **W** resembles an ideal block-diagonal matrix. An intermediate number of iterations per frame should be adopted in the algorithm for better results.

5.3 Performance Evaluation

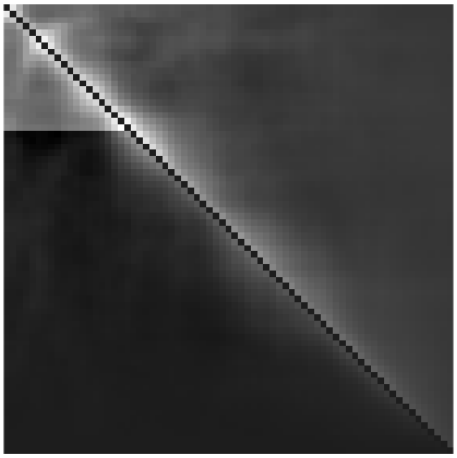
To substantiate the potential of our proposed algorithms, we consider a background-foreground separation in some challenging scenarios such as those with highly dynamic backgrounds, comparing its performance with that of some of the state of the art methods.

We performed two sets of experiments using different databases described in Section 4.1. The first set of experiments aims to compare the performance of the proposed methods in a widely known database and do so by comparing its performance against that of state-of-the-art moving object detection algorithms. The second aims to observe the potential of the proposed methods when compared with the SCM-RPCA [60] and other matrix decomposition algorithms in the same conditions presented in the original work that introduced the SCM-RPCA method.

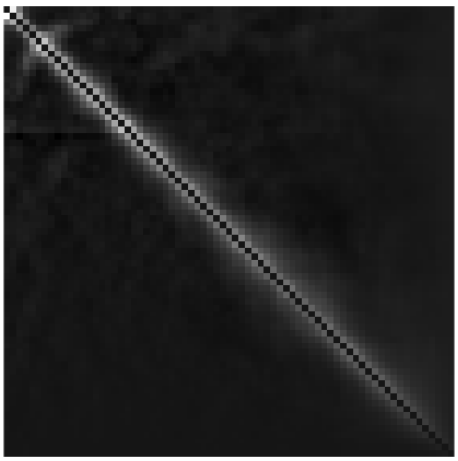
In all experiments we chose the commonly used Precision (Pr), Recall (Re) and F-Measure (F1) metrics to assess the results. Differently from the experiments in Chapter 3 where the DIS metric was employed since the F1 was not obtainable due



(a) 3 iterations per frame



(b) 10 iterations per frame



(c) 100 iterations per frame

Figure 5.4: Resulting **W** matrices using multiple iterations per frame. In (a) we used three iterations per frame, in (b) we used 10 iterations per frame, and in (c) we used 100 iterations per frame. It is possible to see that the number of iterations has a strong impact in the sparsity of the matrix with (c) being very close to the ideally pictured **W** matrix.

to the dataset characteristics. The definition of those metrics are:

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP+FP}},\ \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP+FN}},\ \mathrm{F1} = 2 \times \frac{\mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr+Re}}, \tag{5.44}$$

where TP is the number of foreground pixels correctly classified, FP is the number of background pixels wrongly classified as foreground, and FN is the number of foreground pixels wrongly classified as background.

The first experiment videos are part of the 2014 Change Detection.NET [63], dynamic background. Those videos feature occlusions, hard-to-model moving backgrounds and larger number of frames per videos. Among the two datasets that we used in our experiments, this is the one with the most total frames, even if the total number of videos is smaller (6 against the 15 videos from UCSD).

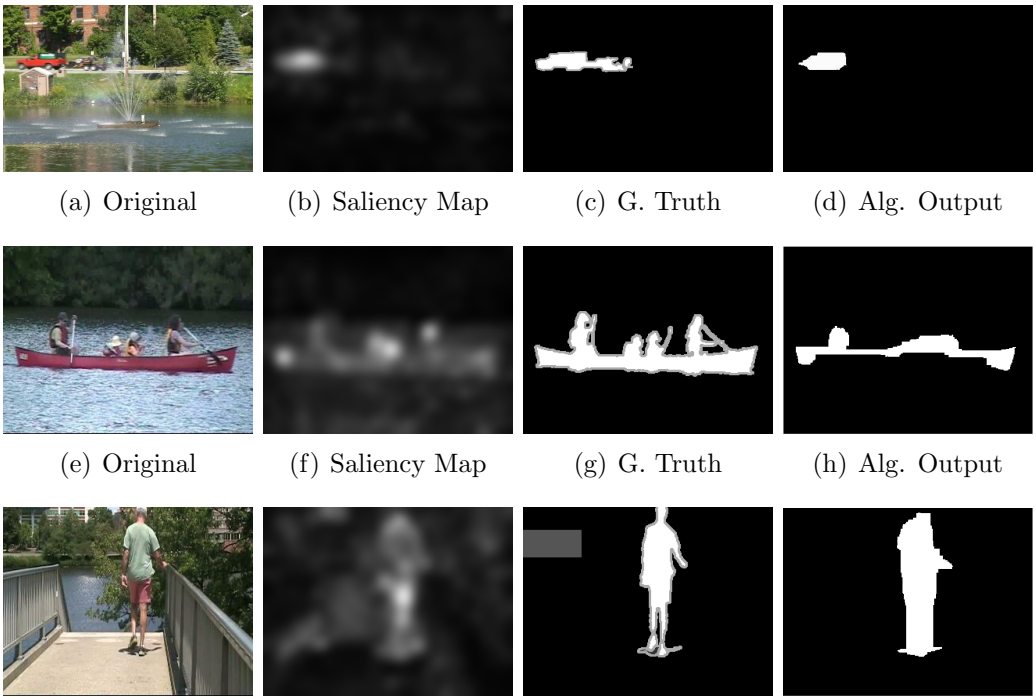
Due to the large number of frames in each video, some modifications were implemented in the proposed batch method. Since our batch method constructs a data matrix $\mathbf{A}$ with all the video frames, with a large number of frames in the videos (some are almost 8000 frames long) the original data matrix becomes excessively large, therefore making it impossible for the algorithm to run without exceeding the memory limitations of the available computers. To be able to cope with this amount of data we divided the video in 400-frames long sequences and performed the algorithm using only those frames. At the end, after processing every 400-frame sequences the $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{E}$ matrices were concatenated to form again the data structures with the same size as the original one.

In this first experiment we compared the outputs of the detection masks generated by the proposed batch algorithm and that of some state-of-the-art moving object detection methods discussed in Chapter 4, namely IUTIS-3 [97], PAWCS [98], AMBER [100], SWCD [101], and CwisarDRP [102].

Figure 5.5 shows some detection examples for the proposed algorithm in the videos os the Change Detection.net database. It is possible to see, by observing those frames, that the method presents encouraging results in such complex dataset, being able to detect the moving objects, while presenting few false positive detections, as desired.

Table 5.1 shows the average detection results of the proposed batch method, as well as some other unsupervised state-of-the-art methods on foreground-background separation, discussed in Section 4.4 for the 2014 Change Detection.net, dynamic background dataset.

The results shown in Table 5.1 make it clear that the proposed method is not working well for the Change Detection.net dataset. However, it is possible to see that the method present a Precision score close to that of some of the other compared state-of-the-art methods. This is due to the fact that the Precision score is highly



(a) Original (b) Saliency Map (c) G. Truth (d) Alg. Output

(e) Original (f) Saliency Map (g) G. Truth (h) Alg. Output

(i) Original (j) Saliency Map (k) G. Truth (l) Alg. Output

Figure 5.5: First column ((a),(e),(i)) shows the original images, second column ((b),(f),(j)) the saliency maps used to compute the focus matrices, third column ((c),(g),(k)) shows the ground truths for the background-foreground separation, and column ((d),(h),(l)) shows the proposed algorithm output. The proposed algorithm displays few false positive detections in those sample frames from the Change Detection.net.

influenceable by the number of false positive detections, which is low in our method due to the constraining attention matrices. On the other hand, the large number of false negative detections lowers the Recall and F1 scores resulting in this poor performance.

We estimate that the excess false negative detections come from the fact that the saliency detection method employed [86] is not able to focus the algorithm attention in the right regions for such complex videos. Figure 5.6 shows some cases where the saliency maps are not able to help the foreground segmentation.

The second set of experiments features a different dataset and other methods of comparison. This set compares the performance of the proposed methods with other matrix decomposition methods. The methods chosen in our comparative evaluation for the second set of experiments are the original solution for the RPCA in [26], the Lagrangian-Stable PCP (Lag-PCP), originally introduced in [105], Robust Motion-Assisted Matrix Restauration (RMAMR) presented in [81], the three-term decomposition model (3TD) proposed in [80], and the more recent SCM-RPCA [60]. In this set of experiments we chose to use the same saliency detection method as before to keep the comparison with the SCM-RPCA fair.

Table 5.1: Performance experiments using the dynamic background portion of the 2014 Change Detection.net [63].

	Avg. Re	Avg. Pr	Avg. F1
IUTIS-3 [97]	0.878	0.924	0.896
PAWCS [98]	0.887	0.903	0.894
AMBER [100]	0.912	0.799	0.834
SWCD [101]	0.869	0.863	0.863
CwisarDRP [102]	0.814	0.850	0.827
Proposed Batch Algorithm	0.323	0.706	0.355

We evaluate the performance of the selected algorithms using the UCSD background subtraction dataset [78], comprising 18 videos with highly dynamic and moving background scenes, as described in Section 4.1.

A similar experiment involving these methods and metrics has already been presented in [60]. For consistency with those experiments, and ease of comparison, we choose to use the same four videos from the UCSD background subtraction dataset in our tests. The four videos selected from the UCSD data set for experimentation, feature a maritime environment with moving water as part of the background.

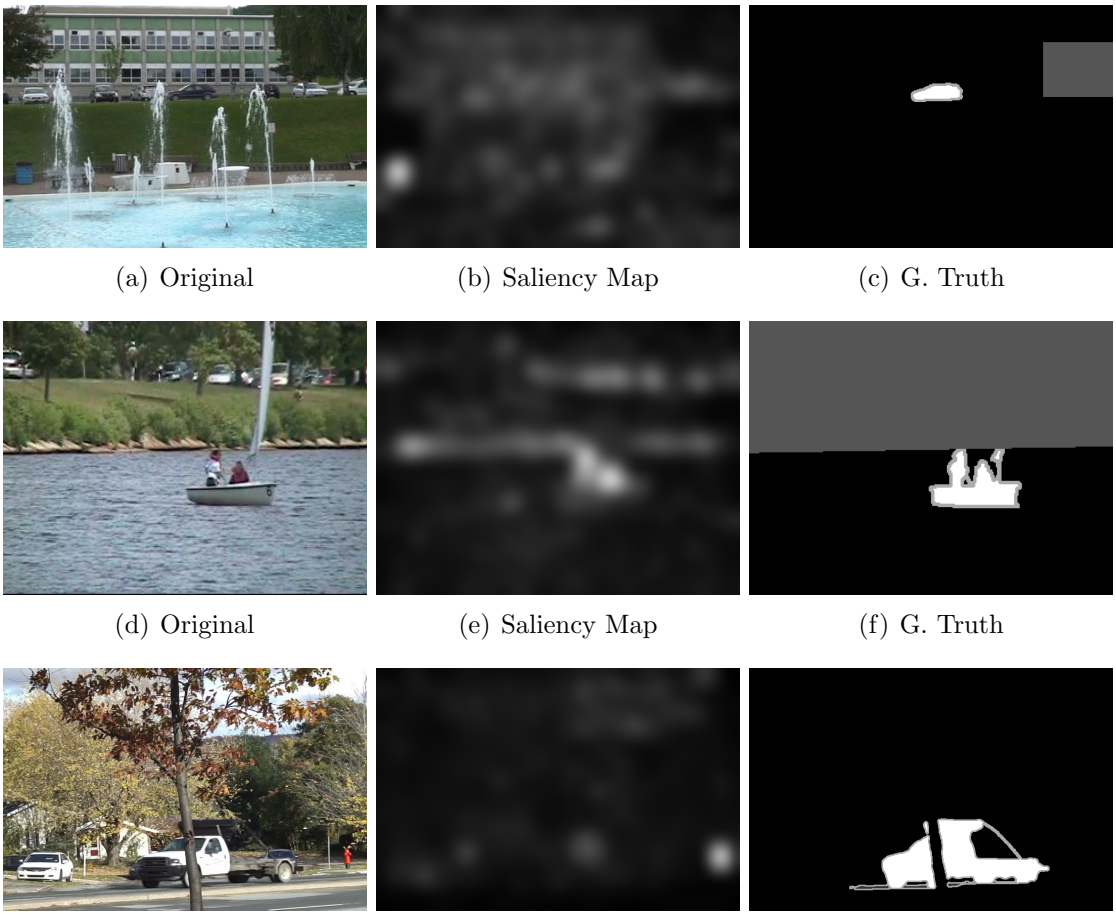
Table 5.2: Performance experiments using four videos from the UCSD dataset [78].

	Birds			**Surfers**			**Boats**			**Ocean**			**Rank**
	Re	Pr	F1	Re	Pr	F1	Re	Pr	F1	Re	Pr	F1	Avg. F1
RPCA	0.842	0.094	0.170	0.754	0.075	0.137	0.814	0.100	0.178	0.748	0.115	0.200	0.171
Lag-SPCP	0.413	0.322	0.362	0.244	0.282	0.261	0.405	0.215	0.281	0.484	0.313	0.380	0.321
RMAMR	0.823	0.229	0.358	0.775	0.248	0.376	0.816	0.230	0.359	0.777	0.175	0.286	0.345
3TD	0.586	0.604	0.595	0.538	0.405	0.462	0.673	0.473	0.556	0.563	0.337	0.442	0.509
SCM-RPCA	0.573	0.638	0.604	0.518	0.565	0.541	0.663	0.550	0.602	0.457	0.544	0.497	0.561
Proposed Sequential Algorithm	0.620	0.463	0.530	0.447	0.466	0.456	0.353	0.499	0.414	0.313	0.484	0.381	0.445
Proposed Batch Algorithm	0.674	0.981	**0.799**	0.829	0.958	**0.889**	0.877	0.946	**0.910**	0.514	0.923	**0.660**	**0.815**

Table 5.2 shows the results of an preliminary experiment considering only 4 videos from UCSD background subtraction dataset, whose results for some of the compared methods are reported in [60]. From these results, one can notice the best average F1 performance for the proposed batch method, as well as the fact that for each video individually the proposed batch method had the best F1 results. We can also observe the performance of the sequential method with 1 iteration per frame in this experiment. Although the results are not as impressive as those of the batch method, which was already expected, they are consistently superior to the other compared methods.

Figure 5.7 shows examples of detection in some frames of the four videos used in the first experiment, using the UCSD dataset. It is possible to notice that, in most cases, the detection output by our proposed batch algorithm is very close to the ground truth with few erroneous detections.

A second, more comprehensive test was performed by applying the best three methods of the experiment shown in Table 5.2 (the ones proposed here and the



(a) Original (b) Saliency Map (c) G. Truth

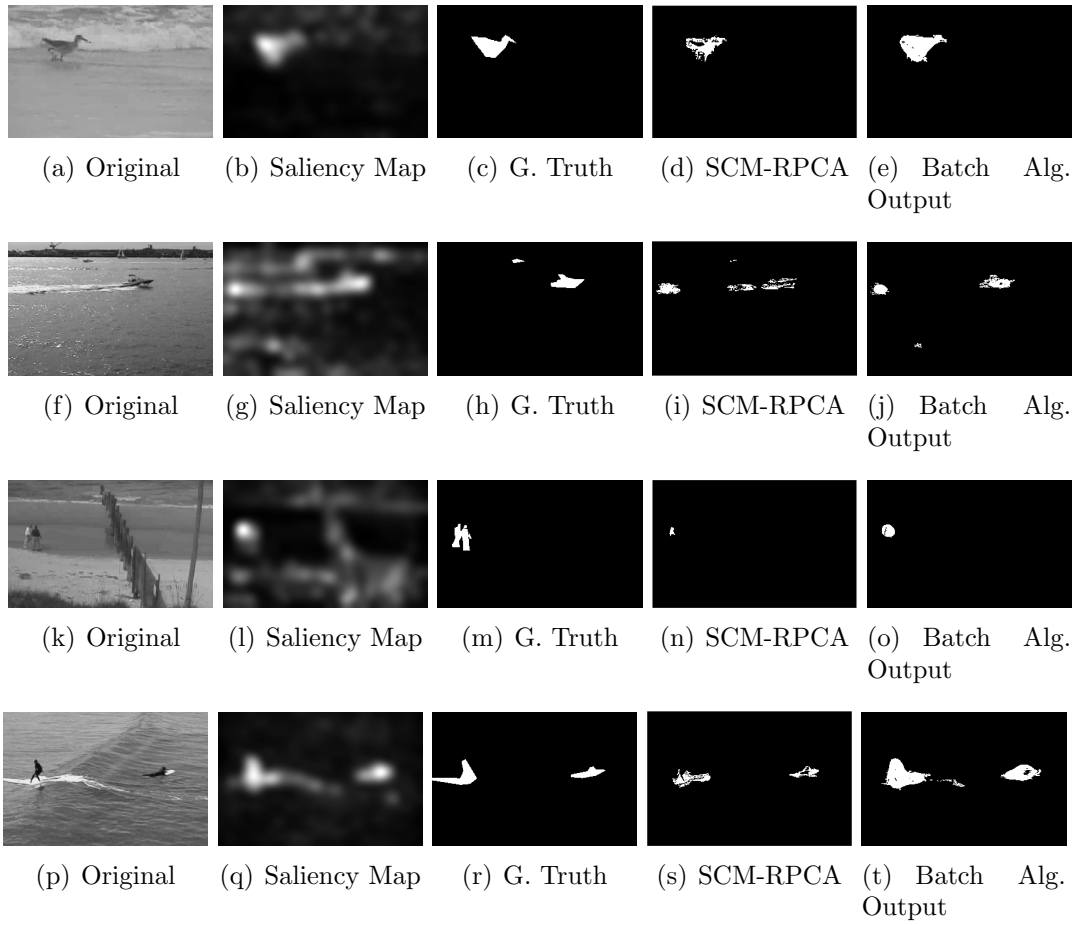
(d) Original (e) Saliency Map (f) G. Truth

(g) Original (h) Saliency Map (i) G. Truth

Figure 5.6: First column ((a),(d),(g)) shows the original images, second column (b),(e),(h) the saliency maps used to compute the focus matrices, and third column ((c),(f),(i)) shows the ground truths for the background-foreground separation. One can see that the saliency maps are misleading in those examples, making the algorithm have bad results for this database.

SCM-RPCA) to all 15 of the 18 videos in the UCSD dataset that had their ground truths available. Table 5.3 shows the average results for the three methods using the same metrics presented before. One can readily notice the superior results of the proposed batch method, demonstrating the improvement of using the union-of-subspaces technique in the background-foreground separation. It is important to notice that the proposed sequential (1 iteration per frame) algorithm also had a superior result when compared with the SCM-RPCA, although, once again, still bellow the batch version results.

Table 5.4 shows an efficiency comparative study considering the execution times of the three algorithms from the last experiment on all videos tested in the last experiment. These results were obtained using a computer with an Intel i7-7700HQ CPU @ 2.80 GHz, with 32GB RAM, and running Matlab 2017a. One can notice that the execution times of the proposed algorithm are from 5 to 6 times larger than



(a) Original (b) Saliency Map (c) G. Truth (d) SCM-RPCA (e) Batch Alg. Output

(f) Original (g) Saliency Map (h) G. Truth (i) SCM-RPCA (j) Batch Alg. Output

(k) Original (l) Saliency Map (m) G. Truth (n) SCM-RPCA (o) Batch Alg. Output

(p) Original (q) Saliency Map (r) G. Truth (s) SCM-RPCA (t) Batch Alg. Output

Figure 5.7: The first Column ((a),(f),(k),(p)) shows the original images, the second column ((b),(g),(l),(q)) the saliency maps used to compute the focus matrices, the third column ((c),(h),(m),(r)) shows the ground truths for the background-foreground separation, the fourth column ((d),(i),(n),(s)) shows the output of SCM-RPCA, and the last column ((e),(j),(o),(t)) shows the proposed batch algorithm output.

Table 5.3: Performance experiments using the complete UCSD dataset [78].

	Avg. Re	Avg. Pr	Avg. F1
SCM-RPCA	0.263	0.916	0.371
Proposed Sequential Algorithm	0.495	0.545	0.502
Proposed Batch Algorithm	**0.634**	**0.917**	**0.735**

the ones of the SCM-RPCA. This is so because in the proposed batch algorithm the optimization was set up with smaller update steps when compared to the one of SCM-RPCA. We did so on account of the nature of the algorithm that requires the projection of the data in multiple subspaces. As a consequence, the number of iterations to reach convergence is higher in the proposed method, leading to longer processing times. However, the simpler sequential algorithm has a time performance comparable to that of the SCM-RPCA, due to its small time per iterations and the fact that the method executes only 1 iteration per frame.

Table 5.4: Average time in seconds taken by each algorithm to run the videos of the UCSD dataset [78].

	SMC-RPCA	Proposed Batch	Proposed Sequential
Avg. Time per Video	7.86	42.17*	**7.84**
Avg. Time per Iteration	0.28	**0.21**	**0.21**

*In most cases the method reached the 200 iteration limit.

In all our experiments, the output of the proposed algorithm was post processed by applying a hard threshold and performing a morphological area open operation [106] to remove small false positive detections. In addition, detections that lasted less than two frames were removed as outliers. It is important to note that in the results shown in Table 5.3, the SCM-RPCA algorithm output was subjected to the same type of post processing. The selected parameter setup for the proposed methods in the second set of experiments is shown in Table 5.5, and was obtained experimentally using a grid search in one of the videos os the UCSD dataset. The setup and implementation of the SCM-RPCA was obtained from the original method author.

Table 5.5: Parameter setup for the proposed algorithms

	λ_1	λ_2	μ_1	μ_2	ϵ
Proposed Sequential Algorithm	10^3	10^4	0.5	8.9	0.01
Proposed Batch Algorithm	10^7	10^7	10^7	10^7	0.01

5.4 Summary

In this chapter we introduced two novel approaches to solve the challenging background-foreground separation in the presence of moving backgrounds by using the restrictions of the attention matrices (shape constraint and confidence map) that help the updates on the foreground detection matrix around the area surrounding the region of interest. This, as a result, reduces the false positive rate.

In addition to the use of the focus matrices, the method proposed in this paper applies a powerful technique capable of more generally capturing the background part of the video as a union-of-subspaces, in contrast to the single subspace projection used by most methods.

The proposed method was compared with some of the state-of-the-art matrix decompositon methods using the test videos from UCSD moving background dataset and with other moving object detection methods using the dynamic background videos from the 2014 Change Detection.net, which includes many videos with very complex highly dynamic backgrounds and moving foregrounds.

The results of the experiments in the Change Detection.net dataset show that the proposed batch algorithm has few false positive detections, while having a fair amount of false negative detections, leading to a low precision score. This is most likely due to the results of the saliency detection method employed in the algorithm pre-processing.

The results of the experiments in the UCSD dataset show that the proposed methods exhibit superior performance in terms of correct detection when compared to other similar methods in the literature. Both batch and sequential algorithms display superior precision, recall and F1 metric.

Although the results are not homogeneous in all experimented datasets it is evident that the proposed methods have a strong potential in the field of moving foreground detection with moving background. The advantages of using a UoS to project the low-rank representation of the video is evidenced in the performed experiments and the use of the attention matrices allow the algorithm to avoid false positive detections, which are a common problem in this type of algorithms.

Chapter 6

Conclusions and Future work

This chapter presents a discussion on the main results obtained in this thesis, as well as a summary of the main contributions of the algorithms that were proposed throughout the previous chapters. It also discusses a few ideas that could be further explored to follow up the presented results and improve the proposed methods.

6.1 Conclusions

This thesis discusses the use of sparse plus low-rank matrix decomposition to deal with the problem of moving object and change detection in videos. Throughout the previous chapters we discussed some of the main open challenges in the literature and presented new algorithms to solve them.

The foundation of all the algorithms discussed in this thesis was the representation of a frame of the video by a combination of a low-rank model of other frames either from a reference videos, that represents the unperturbed state of the environment, or from the same video. Differently from most sparse plus low-rank decomposition methods, the proposed algorithms are based on the RoSuRe method, that projects the original set of frames into a union of subspaces, instead of projecting them into a single subspace. This allows the proposed algorithm to be able to cope with more complex scenarios and represent a more wide range of images, therefore making it possible to employ such algorithms in challenging scenarios such as videos from moving cameras and videos with moving cluttered background.

The thesis is divided in two main topics, namely moving-camera videos change detection and moving object detection in the presence of moving background. For each of those topics, two algorithms were proposed, as part of solution.

For the moving-camera videos change detection the first proposed algorithm is called mcRoSuRe-TA. This method implements upgrades to the mcRoSuRe method, allowing it to perform without the need of a previous temporal alignment between reference and target videos. This modification comes from the observations that the

structure matrix $\mathbf{W}_t$ carries information that may be used to find the correspondences between reference and target frames in the moving camera object detection framework. In order to obtain such matrix and make the time alignment possible, an initial step is introduced in the algorithm to find the exact region of the reference video that corresponds to the target one. After that, a new reference data matrix is created using only the frames from the reference videos that lay on the region that corresponds to the target. The second algorithm, namely the mcRoSuRe-A, further modifies the mcRoSuRe-TA aiming to accelerate the matrix computations and allow the algorithm to get closer to real-time performance. To do so, we downsample the reference data matrix in the first step of the algorithm to reduce the number of computations needed to obtain the time-aligned data matrix. The use of such technique allows the algorithm to perform up to 100 times faster than the original mcRoSuRe algorithm with at least the same detection scores.

Extensive experiments were performed comparing the proposed algorithms with some of the state-of-the-art methods in moving-camera object detection. The results from those experiments show that the proposed methods achieve better detection results in very complex scenarios such as those present in the VDAO dataset.

In the second part, the moving object detection in the presence of moving background, we proposed two distinct algorithms that perform the moving foreground detection. The first one, namely the batch algorithm, performs the decomposition of the data matrix into three different matrices, namely the sparse foreground matrix, the residue matrix, and the low-rank background representation matrix. To perform this decomposition the algorithm uses a pair of matrices called the attention matrices, which are obtained via a saliency detection method, performing a human eye fixation prediction in each frame of the original video. The second algorithm, namely the sequential one, is meant to perform the same task as the previous one, but instead of using all the frames of the video at once, it computes the decomposition each time a new frame appears, therefore updates the previous results based on the newly available information. Extensive experiments were performed in two different datasets to assess the performance of the proposed algorithms. While comparing the algorithm with state-of-the-art moving object detection methods using the CDNET dataset, we observed that it is highly dependent on the performance of the saliency method employed to obtain the attention matrices, thus it is important to obtain those with a method that performs well given the video context one wants to use. In a different set of experiments using the UCSD database, the proposed algorithm had the best performance among the algorithms that perform the sparse plus low-rank decomposition, showing results that were twice as better as the previous state-of-the-art algorithms in this category, indicating the great potential of such algorithms. The experiments also revealed that the sequential implementa-

tion of the proposed method is able to obtain results superior to those of similar algorithms while executing only one iteration per frame.

6.2 Future Work

The line of work developed here is very promising, as the obtained results are of great quality and matches the achievements of several state-of-the-art methods. In this section, the main ideas for the continuity of this line of work will be presented. For the sake of the organization, this section will be separated by the type of contribution they relate to.

6.2.1 Change Detection with Sparse Representation

Observing the behaviour of the algorithm, some modifications on the form of dealing with the data can be proposed. Since most of the database that was used to test this method (VDAO) is composed of videos whose camera movement is predominantly translational, one can try to use a different data representation to form some kind of data matrix that does not deal directly with pixels nor frames, but columns of frames.

In translational moving-camera videos, if the camera jitter and parallax effect are not to be considered, in a sequence of frames, most of the columns will remain the same, except those of the border of the frame that will either bear new information or disappear. This reasoning could lead to the adoption of a structure that uses these columns as the smaller representation and then tries to build the target video frames as a combination of the reference video columns, in such scenario. That could lead to an optimization where only a minimum number of columns has to be kept to represent the whole video. This reduction on the amount of stored data can help the algorithm to run faster and more efficiently, as it will have a smaller yet representative search-space.

6.2.2 Moving Object Detection in Moving Cluttered Backgrounds

The performed experiments point to the fact that the proposed algorithms yield a small amount of false positive detections in the results. But they also showed that depending on how complex are the backgrounds more attention has to be given to the obtention of the attention matrices, meaning that we should explore the use of different saliency detection methods to obtain higher quality attention matrices, and therefore avoid excessive false negative detections.

Another interesting point that should also receive some attention is to modify the proposed sequential algorithm to make its results closer to those obtained with the batch one. Although its performance is already superior to some of the other sparse plus low-rank representation methods there is a gap between the performance of the sequential and batch algorithms, that can, perhaps, be reduced with modifications in its cost function.

Bibliography

[1] RESEARCH, T. M. *Video Surveillance and VSaaS Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2016 - 2024*. Technical report, TMS Analysis, Albany, USA, April 2016.

[2] HOSSAIN, E., CHETTY, G. "Person identification in surveillance video using gait biometric cues". In: *International Conference on Fuzzy Systems and Knowledge Discovery*, Sichuan, China, 2012.

[3] SIMONYAN, K., ZISSERMAN, A. "Very deep convolutional networks for large-scale image recognition", *Computing Research Repository*, v. abs/1409.1556, September 2014. Available at: <`http://arxiv.org/abs/1409.1556`>.

[4] HE, K., ZHANG, X., REN, S., et al. "Deep Residual Learning for Image Recognition", *Computing Research Repository*, v. abs/1512.03385, December 2015. Available at: <`http://arxiv.org/abs/1512.03385`>.

[5] SINGH, P., DEEPAK, B., SETHI, T., et al. "Real-time object detection and Tracking using color feature and motion". In: *IEEE International Conference on Communications and Signal Processing*, pp. 1236–1241, Melmaruvathur, India, April 2015.

[6] KONG, H., AUDIBERT, J.-Y., PONCE, J. "Detecting abandoned objects with a moving camera", *IEEE Transactions on Image Processing*, v. 19, n. 8, pp. 2201–2210, August 2010.

[7] SINCAN, O. M., AJABSHIR, V. B., KELES, H. Y., et al. "Moving object detection by a mounted moving camera". In: *IEEE International Conference on Computer as a Tool*, pp. 1–6, Salamanca, Spain, November 2015.

[8] THOMAZ, L. A., DA SILVA, A. F., DA SILVA, E. A. B., et al. "Abandoned object detection using operator-space pursuit". In: *IEEE International Conference on Image Processing*, pp. 1980–1984, Quebec, Canada, September 2015.

[9] TIAN, Y., FERIS, R. S., LIU, H., et al. "Robust detection of abandoned and removed objects in complex surveillance videos", *IEEE Transactions on Systems, Man, and Cybernetics*, v. 41, n. 5, pp. 565–576, 2011.

[10] TANEJA, A., BALLAN, L., POLLEFEYS, M. "Geometric change detection in urban environments using images", *IEEE transactions on pattern analysis and machine intelligence*, v. 37, n. 11, pp. 2193–2206, 2015.

[11] DE CARVALHO, G. H. F., THOMAZ, L. A., DA SILVA, A. F., et al. "Anomaly Detection with a Moving Camera using Multiscale Video Analisys", *Multidimensional Systems and Signal Processing*, pp. 1 – 32, February 2018.

[12] CUEVAS, C., MARTÍNEZ, R., GARCÍA, N. "Detection of stationary foreground objects: A survey", *Computer Vision and Image Understanding*, v. 152, pp. 41–57, November 2016.

[13] WAHYONO, FILONENKO, A., JO, K.-H. "Detecting abandoned objects in crowded scenes of surveillance videos using adaptive dual background model". In: *International Carnahan Conference on Security Technology*, pp. 224–227, Lexington, USA, June 2015.

[14] GALLEGO, J., PARDAS, M., LANDABASO, J.-L. "Segmentation and tracking of static and moving objects in video surveillance scenarios". In: *IEEE International Conference on Image Processing*, pp. 2716–2719, San Diego, USA, October 2008.

[15] CUCCHIARA, R., GRANAA, C., PICCARDI, M., et al. "Detecting moving objects, ghosts, and shadows in video streams", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, n. 10, pp. 1337–1342, October 2003.

[16] HASSAN, W., BIRCH, P., MITRA, B., et al. "Detecting moving objects, ghosts, and shadows in video streams", *IET Computer Vision*, v. 7, n. 1, pp. 1–8, February 2013.

[17] SUN, D., ROTH, S., BLACK, M. J. "A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them", *International Journal of Computer Vision*, v. 106, n. 2, pp. 115–137, January 2014.

[18] BHARGAVA, M., CHEN, C.-C., RYOO, M. S., et al. "Detection of abandoned objects in crowded environments". In: *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 271–276, Genoa, Italy, September 2007.

[19] MADDALENA, L., PETROSINO, A. "Stopped object detection by learning foreground model in videos", *IEEE Transactions on Neural Networks and Learning Systems*, v. 24, n. 5, pp. 723–735, February 2013.

[20] BRAHAM, M., DROOGENBROECK, M. V. "Deep background subtraction with scene-specific convolutional neural networks". In: *IEEE International Conference on Systems, Signals and Image Processing*, pp. 1–4, Bratislava, Slovakia, May 2016.

[21] LECUN, Y., BOTTOU, L., BENGIO, Y., et al. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, November 1998.

[22] CHANG, L., ZHAO, H., ZHAI, S., et al. "Robust abandoned object detection and analysis based on online learning". In: *IEEE International Conference on Robotics and Biomimetics*, pp. 940–945, Shenzen, China, December 2013.

[23] MIEZIANKO, R., POKRAJAC, D. "Detecting and recognizing abandoned objects in crowded environments". In: *International Conference on Computer Vision Systems*, pp. 241–250, Santorini, Greece, May 2008.

[24] JOLLIFFE, I. T. *Principal Component Analysis*. New York, USA, Springer, 2002.

[25] GUYON, C., BOUWMANS, T., HADI ZAHZAH, E. "Robust Principal Component Analysis for Background Subtraction: Systematic Evaluation and Comparative Analysis". In: *Part F: Robotics*, InTech, chap. 12, France, 2012.

[26] E. J. CANDÈS, X. LI, Y. M., WRIGHT, J. "Robust principal component analysis?" *Journal of ACM*, v. 58, n. 3, pp. 1–37, May 2011.

[27] BENGEL, M., PFEIFFER, K., GRAF, B., et al. "Mobile robots for offshore inspection and manipulation". In: *IEEE/RSJ International Conference on Intelligent Robost and Systems*, pp. 3317–3322, Saint Louis, USA, October 2009.

[28] KYRKJEBØ, E., TRANSETH, P. L. A. "A robotic concept for remote inspection and maintenance on oil platforms". In: *International Conference on Ocean, Offshore and Arctic Engineering*, pp. 667–674, Honolulu, USA, June 2009.

[29] NREC/CMU. "Sensabot: A safe and cost-effective inspection solution", *Journal of Petroleum Technology*, v. 64, pp. 32–34, 2012.

[30] GALASSI, M., RØYRØY, A., DE CARVALHO, G. P., et al. "DORIS - A MOBILE ROBOT FOR INSPECTION AND MONITORING OF OFFSHORE FACILITIES". In: *Congresso Brasileiro de Automática*, pp. 3174–3181, Belo Horizonte, Brazil, September 2014.

[31] SERAT, J., DIEGO, F., LUMBRERAS, F., et al. "Alignment of videos recorded from moving vehicles". In: *International Conference on Image Analysis and Processing*, pp. 512–517, Modena, Italy, September 2007.

[32] HARTLEY, R., ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[33] DIEGO, F., PONSA, D., SERRAT, J., et al. "Video alignment for change detection", *IEEE Transactions on Image Processing*, v. 20, pp. 1858–1869, July 2011.

[34] DA SILVA, A. F., THOMAZ, L. A., DA SILVA, E. A. B., et al. "Alinhamento de sinais obtidos em trajetórias fechadas utilizando um conjunto genérico de sensores". In: *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 1–5, Santarém, Brazil, September 2016.

[35] DA SILVA, A. F., THOMAZ, L. A., NETTO, S. L., et al. "Online video-based sequence synchronization for moving camera object detection". In: *IEEE International Workshop on Multimedia Signal Processing*, pp. 1 – 6, Luton, United Kingdom, October 2017.

[36] DIXON, S. "Live tracking of musical performances using on–line time warping". In: *International Conference on Digital Audio Effects*, 2005.

[37] LOWE, D. G. "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, v. 60, n. 2, pp. 31–110, July 2004.

[38] CARVALHO, G., DE OLIVEIRA, J. F. L., DA SILVA, E. A. B., et al. "Um sistema de monitoramento para detecção de objetos em tempo real empregando câmera em movimento". In: *Proceedings of Simpósio Brasileiro de Telecomunicações*, pp. 1–5, Fortaleza, Brazil, September 2013.

[39] CARVALHO, G. *Automatic detection of abandoned objects with a moving camera using multiscale video analysis*. PHD Thesis, COPPE - Universidade Federal do Rio de Janeiro, Rio de Janeiro - Brasil, 2015.

[40] MUKOJIMA, H., DEGUCHI, D., KAWANISHI, Y., et al. "Moving camera background-subtraction for obstacle detection on railway tracks". In: *IEEE International Conference on Image Processing*, pp. 3967–3971, Phoenix, USA, September 2016.

[41] WEINZAEPFEL, P., REVAUD, J., HARCHAOUI, Z., et al. "DeepFlow: Large Displacement Optical Flow with Deep Matching". In: *IEEE International Conference on Computer Vision*, pp. 1385–1392, Sydney, Australia, December 2013.

[42] NAKAHATA, M. T., THOMAZ, L. A., DA SILVA, A. F., et al. "Anomaly detection with a moving camera using spatio-temporal codebooks", *Multidimensional Systems and Signal Processing*, pp. 1–30, March 2017.

[43] THOMAZ, L. A. *Abandoned object detection using operator-space pursuit*. Master Dissertation, COPPE - Universidade Federal do Rio de Janeiro, Rio de Janeiro - Brasil, 2015.

[44] BIAN, X., KRIM, H. "Optimal operator space pursuit: A framework for video sequence data analysis". In: *Computer Vision*, v. 7725, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 760–769, 2013.

[45] CUI, X., HUANG, J., ZHANG, S., et al. "Background subtraction using low rank and group sparsity constraints". In: *European Conference on Computer Vision*, pp. 612–625, Florence, Italy, October 2012.

[46] JARDIM, E., BIAN, X., DA SILVA, E. A. B., et al. "On the detection of abandoned objects with a moving camera using robust subspace recovery and sparse representation". In: *IEEE International Conference on Acoustics, International Conference on Speech and Signal Processing*, pp. 1295–1299, South Brisbane, Australia, April 2015.

[47] BIAN, X., KRIM, H. "Robust subspace recovery via dual sparsity pursuit", *Computing Research Repository*, v. abs/1403.8067, March 2014. Available at: <http://arxiv.org/abs/1403.8067>.

[48] BIAN, X., KRIM, H. "Bi-sparsity pursuit for robust subspace recovery". In: *IEEE International Conference on Image Processing*, pp. 3535–3539, Quebec, Canada, September 2015.

[49] CINELLI, L. P., THOMAZ, L. A., DA SILVA, A. F., et al. "Foreground Segmentation for Anomaly Detection in Surveillance Videos Using Deep Residual Networks". In: *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 914–918, São Pedro, Brazil, Sept. 2017.

[50] AFONSO, B. M., CINELLI, L. P., THOMAZ, L. A., et al. "Moving-Camera Video Surveillance in Cluttered Environments Using Deep Features". In: *IEEE International Conference on Image Processing*, pp. 2296–2300, Athens, Greece, Oct 2018.

[51] HE, J., ZHANG, D., BALZANO, L., et al. "Iterative Grassmannian Optimization for Robust Image Alignment", *Computing Research Repository*, v. abs/1306.0404, June 2013. Available at: <http://arxiv.org/abs/1306.0404>.

[52] XU, J., ITHAPU, V. K., MUKHERJEE, L., et al. "GOSUS: Grassmannian Online Subspace Updates with Structured-sparsity". In: *International Conference on Computer Vision*, pp. 3376–3383, Sydney, Australia, December 2013.

[53] SONG, W., ZHU, J., LI, Y., et al. "Image Alignment by Online Robust PCA via Stochastic Gradient Descent", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 26, n. 7, pp. 1241–1250, July 2016.

[54] RODRÍGUEZ, P., WOHLBERG, B. "Translational and rotational jitter invariant incremental principal component pursuit for video background modeling". In: *IEEE International Conference on Image Processing*, pp. 537–541, Phoenix, USA, September 2015.

[55] JAVED, S., JUNG, S. K., MAHMOOD, A., et al. "Motion-Aware Graph Regularized RPCA for background modeling of complex scenes". In: *International Conference on Pattern Recognition*, pp. 120–125, December 2016.

[56] JAVED, S., MAHMOOD, A., BOUWMANS, T., et al. "Spatiotemporal Low-rank Modeling for Complex Scene Background Initialization", *IEEE Transactions on Circuits and Systems for Video Technology*, v. PP, n. 99, pp. 1–1, November 2016.

[57] ZHOU, X., YANG, C., ZHAO, H., et al. "Low-Rank Modeling and Its Applications in Image Analysis", *Computing Research Repository*, v. abs/1401.3409, October 2014. Available at: <http://arxiv.org/abs/1401.3409>.

[58] BOUWMANS, T., SOBRAL, A., JAVED, S., et al. "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*, v. 23, pp. 1 – 71, 2017.

[59] SOBRAL, A., BOUWMANS, T., HADI ZAHZAH, E. "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, Taylor and Francis Group.

[60] SOBRAL, A., BOUWMANS, T., HADI ZAHZAH, E. "Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, Karlsruhe, Germany, August 2015.

[61] "PETS: The performance evaluation of tracking and surveillance". [Online], 2000-2007. Available at `http://www.cvg.rdg.ac.uk/`.

[62] ILIDS TEAM. "Imagery library for intelligent detection systems (i-lids); A Standard for Testing Video Based Detection Systems". In: *International Carnahan Conference on Security Technology*, pp. 75–80, Lexington, USA, October 2006.

[63] GOYETTE, N., JODOIN, P.-M., PORIKLI, F., et al. "Changedetection.net: A new change detection benchmark dataset". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, Rhode Island , USA, June 2012.

[64] DA SILVA, A. F., THOMAZ, L. A., CARVALHO, G., et al. "An annotated video database for abandoned-object detection in a cluttered environment". In: *International Telecommunications Symposium*, pp. 1–5, São Paulo, Brazil, August 2014.

[65] "VDAO - Video Database of Abandoned Objects in a Cluttered Industrial Environment". [Online], 2014. Available at `http://www.smt.ufrj.br/~tvdigital/database/objects`.

[66] YAN, W.-Y. "On principal subspace analysis", *Journal of the Franklin Institute*, v. 335, n. 4, pp. 707 – 718, 1998.

[67] STRANG, G. *Linear algebra and its applications*. 4th ed. Belmont, CA, Thomson, Brooks/Cole, 2006.

[68] HORN, R. A., JOHNSON, C. R. *Matrix Analysis*. Cambridge University Press, 1990.

[69] BOYD, S., VANDENBERGHE, L. *Convex optimization*. New York, USA, Cambridge University Press, 2004.

[70] LIN, Z., LIU, R., SU, Z. "Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation". In: *Advances in Neural Information Processing Systems 24*, Curran Associates, Inc., pp. 612–620, 2011.

[71] ROCKAFELLAR, R. T. "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming", *SIAM Journal on Control*, v. 12, n. 2, pp. 268–285, July 1974.

[72] PARIK, N., BOYD, S. "Proximal Algorithms", *Foundations and Trends in Optimization*, v. 1, n. 3, pp. 127–239, January 2014.

[73] THOMAZ, L. A., DA SILVA, A. F., DA SILVA, E. A. B., et al. "Detection of abandoned objects using robust subspace recovery with intrinsic video alignment". In: *IEEE International Conference on Circuits and Systems*, pp. 1–4, Baltimore, USA, May 2017.

[74] THOMAZ, L. A., JARDIM, E., DA SILVA, A. F., et al. "Anomaly detection in moving-camera video sequences using principal subspace analysis", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 65, n. 3, pp. 1003–1015, March 2018.

[75] "200-frame Excerpts form VDAO database". [Online], 2017. Available at `http://www02.smt.ufrj.br/~tvdigital/database/research/`.

[76] MATSUYAMA, T., OHYA, T., HABE, H. "Background subtraction for non-stationary scene". In: *Asian Conference on Computer Vision*, pp. 662–667, Taipei, Taiwan, August 2000.

[77] SATOH, Y., TANAHASHI, H., WANG, C., et al. "Robust Event Detection by Radial Reach Filter (RRF)". In: *International Conference on Pattern Recognition*, v. 2, pp. 623–626, Québec, Canada, August 2002.

[78] MAHADEVAN, V., VASCONCELOS, N. "Spatiotemporal Saliency in Dynamic Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 1, pp. 171–177, May 2010.

[79] PRASAD, D. K., RAJAN, D., RACHMAWATI, L., et al. "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey", *IEEE Transactions on Intelligent Transportation Systems*, v. 18, n. 8, pp. 1993–2016, August 2017. Available at: <`http://arxiv.org/abs/1811.05255v1`>.

[80] OREIFEJ, O., LI, X., SHAH, M. "Simultaneous Video Stabilization and Moving Object Detection in Turbulence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 2, pp. 450–462, February 2013.

[81] YE, X., YANG, J., SUN, X., et al. "Foreground-Background Separation From Video Clips via Motion-Assisted Matrix Restoration", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 25, n. 11, pp. 1721–1734, November 2015.

[82] HAREL, J., KOCH, C., PERONA, P. "Graph-Based Visual Saliency". In: *Advances in Neural Information Processing Systems 19*, MIT Press, pp. 545–552, 2007.

[83] WEI, Y., WEN, F., ZHU, W., et al. "Geodesic Saliency Using Background Priors". In: *European Conference on Computer Vision*, pp. 29–42, Florence, Italy, 2012.

[84] YANG, C., ZHANG, L., LU, H., et al. "Saliency Detection via Graph-Based Manifold Ranking". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3173, June 2013.

[85] ZHOU, D., WESTON, J., GRETTON, A., et al. "Ranking on Data Manifolds". In: Thrun, S., Saul, L. K., Schölkopf, B. (Eds.), *Advances in Neural Information Processing Systems 16*, MIT Press, pp. 169–176, 2004.

[86] ZHANG, J., SCLAROFF, S. "Exploiting Surroundedness for Saliency Detection: A Boolean Map Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 5, pp. 889–902, May 2016.

[87] WU, X., MA, X., ZHANG, J., et al. "Salient Object Detection Via Deformed Smoothness Constraint". In: *IEEE International Conference on Image Processing*, pp. 2815–2819, Athens,Greece, Oct 2018.

[88] HUANG, F., QI, J., LU, H., et al. "Salient Object Detection via Multiple Instance Learning", *IEEE Transactions on Image Processing*, v. 26, n. 4, pp. 1911–1922, April 2017.

[89] AYTEKIN, C., IOSIFIDIS, A., GABBOUJ, M. "Probabilistic saliency estimation", *Pattern Recognition*, v. 74, pp. 359 – 372, 2018.

[90] "BMS - Boolean Map based Saliency modelt". [Online], 2016. Available at `http://cs-people.bu.edu/jmzhang/BMS/BMS.html`.

[91] LIM, L. A., KELES, H. Y. "Foreground segmentation using convolutional neural networks for multiscale feature encoding", *Pattern Recognition Letters*, v. 112, pp. 256 – 262, 2018.

[92] WANG, Y., LUO, Z., JODOIN, P.-M. "Interactive deep learning method for segmenting moving objects", *Pattern Recognition Letters*, v. 96, pp. 66 – 75, 2017.

[93] BABAEE, M., DINH, D. T., RIGOLL, G. "A deep convolutional neural network for video sequence background subtraction", *Pattern Recognition*, v. 76, pp. 635 – 649, 2018.

[94] ROSENBLATT, F. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books, 1962.

[95] ST-CHARLES, P.-L., BILODEAU, G.-A., BERGEVIN, R. "Subsense: a universal change detection method with local adaptive sensitivity", *IEEE Transactions on Image Processing*, v. 24, n. 1, pp. 359 – 373, January 2015.

[96] BOUWMANS, T., JAVED, S., SULTANA, M., et al. "Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation", *Neural Networks*, v. Submitted, November 2018. Available at: <`http://arxiv.org/abs/1811.05255v1`>.

[97] BIANCO, S., CIOCCA, G., SCHETTINI, R. "Combination of Video Change Detection Algorithms by Genetic Programming", *IEEE Transactions on Evolutionary Computation*, v. 21, n. 6, pp. 914 – 928, March 2017.

[98] ST-CHARLES, P.-L., BILODEAU, G.-A., BERGEVIN, R. "A Self-Adjusting Approach to Change Detection Based on Background Word Consensus". In: *IEEE Winter Conference on Applications of Computer Vision*, pp. 990 – 997, Hawaii,USA, January 2015.

[99] WANG, B., DUDEK, P. "AMBER: Adapting multi-resolution background extractor". In: *IEEE International Conference on Image Processing*, pp. 3417–3421, Melbourne, Australia, Sept 2013.

[100] WANG, B., DUDEK, P. "A Fast Self-tuning Background Subtraction Algorithm". In: *IEEE Workshop on Change Detection*, pp. 401 – 404, Columbus, USA, June 2014.

[101] IŞIK, A., ÖZKAN, K., GÜNAL, S., et al. "SWCD: a sliding window and self-regulated learning-based background updating method for change detection in videos", *Journal of Electronic Imaging*, v. 27, n. 2, pp. 1 – 11, March 2018.

[102] GREGORIO, M. D., GIORDANO, M. "WiSARDRP for change detection in video sequences". In: *European Symposium on Artificial Neural Networks*, pp. 453–458, Bruges, Belgium, April 2017.

[103] BERTSEKAS, D. P. "Incremental proximal methods for large scale convex optimization", *Mathematical Programming*, v. 129, n. 2, pp. 129–163, Jun 2011.

[104] PANAHI, A., BIAN, X., KRIM, H., et al. "Robust Subspace Clustering by Bi-Sparsity Pursuit: Guarantees and Sequential Algorithm". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1302–1311, Lake Tahoe, USA, March 2018.

[105] ARAVKIN, A., BECKER, S., CEVHER, V., et al. "A variational approach to stable principal component pursuit". In: *Conference on Uncertainty in Artificial Intelligence*, pp. 32–41, Quebec, Canada, July 2014.

[106] SOILLE, P. *Morphological Image Analysis: Principles and Applications*. 2nd ed. Secaucus, NJ, USA, Springer-Verlag, 2003.

[107] LUENBERGER, D. G., YE, Y. *Linear and Nonlinear Programming*. New York, USA, Springer, 2008.

Appendix A

Mathematical Derivation of Moving Object Detection Algorithm

In this appendix we will show the derivation of the batch and sequential algorithms introduced in Chapter 5.

A.1 Batch Algorithm Derivation

For the batch algorithm the goal is to find the updates that allow one to minimize the following function

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{E},\mathbf{S}} \|\mathbf{W}\|_1 + \lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2 \|\mathbf{E}\|_1, \\ &\text{s.t.} \begin{cases} \mathbf{A} = \mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \\ (\mathbf{E} - \mathbf{ED})\mathbf{1} \succeq \epsilon, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{0^T} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \end{cases}, \end{aligned} \tag{A.1}$$

where $\mathbf{W}$ is the weight matrix bearing the relations between the columns of the low-rank representation $\mathbf{L}$ of the data matrix $\mathbf{A}$, $\mathbf{S}$ is the sparse matrix where foreground objects shall lie, and $\mathbf{E}$ is the sparse residue matrix composed of the parts of $\mathbf{A}$ not represented by neither $\mathbf{L}$ nor $\mathbf{S}$. The matrix $\mathbf{1}$ of dimensions $m \times n$ has all entries equal to 1.

To find a solution for this optimization problem, we will employ the ALM and the ADMM, as in [73]. This solution is detailed in the sequel. From Eq. (A.1), one

has that

$$\mathbf{LW} - \mathbf{L} = \mathbf{0}, \tag{A.2}$$

and

$$\mathbf{L} = \mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}. \tag{A.3}$$

Replacing Eq. (A.2) into (A.3), we get

$$(\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E})\mathbf{W} - (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}) = \mathbf{0}, \tag{A.4}$$

which yields the following expanded cost function using the ALM-ADMM method [70]

$$\begin{aligned}\Gamma(\mathbf{W}, \mathbf{E}, \mathbf{S}, \mathbf{Y}_1, \mathbf{Y}_2, \mu_1, \mu_2) = \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1 + \\ \langle \mathbf{LW} - \mathbf{L}, \mathbf{Y}_1\rangle + \langle \mathbf{E} - \mathbf{ED} - \mathbf{1}\epsilon, \mathbf{Y}_2\rangle + \\ \frac{\mu_1}{2}\|\mathbf{LW} - \mathbf{L}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{E} - \mathbf{ED} - \mathbf{1}\epsilon\|_F^2.\end{aligned} \tag{A.5}$$

To obtain the update function for $\mathbf{W}_{k+1}$ we need to write the augmented cost function with the terms that depend on $\mathbf{W}_k$

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \langle \mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}, \mathbf{Y}_{1(k)}\rangle + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2\right], \tag{A.6}$$

Linearizing the function around the $\mathbf{W}_k$ operation point

$$\begin{aligned}\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \Big[\|\mathbf{W}\|_1 + \langle(-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)}\rangle + \\ \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{1(k)}\eta_1}{2}\|\mathbf{W} - \mathbf{W}_k\|_2^2\Big]\end{aligned} \tag{A.7}$$

with $\hat{\mathbf{W}} = \mathbf{LW} - \mathbf{L}$. We can use the quadratic approximation the function of $\mathbf{W}$ as $F(\mathbf{W}) \approx F(\mathbf{W}_k) + \langle \mathbf{W} - \mathbf{W}_k, \bar{\nabla}F(\mathbf{W}_k)\rangle$ according to [105], with

$$F(\mathbf{W}) = \text{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{L}_{\mathrm{k}+1})\right] + \frac{\mu_{1(\mathrm{k})}}{2}\|\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{L}_{\mathrm{k}+1}\|_2^2. \tag{A.8}$$

In this formulation we have

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \frac{\mu_{1(k)}\eta_1}{2}\left\|\mathbf{W} - \left(\mathbf{W}_k - \frac{1}{\mu_{1(k)}\eta_1}\bar{\nabla}F(\mathbf{W}_k)\right)\right\|_2^2\right]. \tag{A.9}$$

We can, therefore, write it as [107]

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}}\left[\mathbf{W}_k - \frac{\bar{\nabla}F(\mathbf{W}_k)}{\mu_{1(k)}\eta_2}\right]. \tag{A.10}$$

Which means that, to find the final update for $\mathbf{W}_{k+1}$ we only need to find $\bar{\nabla}F(\mathbf{W}_k)$. To do so we will re-write $F(\mathbf{W})$, so that

$$F(\mathbf{W}) = \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{L}_{\mathrm{k}+1})\right] + \frac{\mu_{1(\mathrm{k})}}{2}\|\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{L}_{\mathrm{k}}\|_2^2 \tag{A.11}$$

$$= \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \frac{\mu_{1(\mathrm{k})}}{2}\langle\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}, \mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}\rangle \tag{A.12}$$

$$= \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \frac{\mu_{1(\mathrm{k})}}{2}\langle\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}, \mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}\rangle, \tag{A.13}$$

here we can expand the righthand side of the equation fixing one of the $\mathbf{W}$ at $\mathbf{W}_k$ and letting the other as it stands. By doing so with both the original $\mathbf{W}$, and since $\hat{\mathbf{W}} = \mathbf{L}\mathbf{W} - \mathbf{L}$, we get

$$F(\mathbf{W}) = \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \frac{\mu_{1(k)}}{2} \times 2\langle\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}, \mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\rangle \tag{A.14}$$

$$= \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \mu_{1(k)}\mathrm{Tr}\left[(\mathbf{L}_{\mathrm{k}+1}\mathbf{W} - \mathbf{L}_{\mathrm{k}+1})^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1})\right] \tag{A.15}$$

$$= \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \mu_{1(k)}\mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1}) - \mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1})\right] \tag{A.16}$$

$$= \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{1(\mathrm{k})}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}\right] + \mu_{1(k)}\mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1})\right] - \mu_{1(\mathrm{k})}\mathrm{Tr}\left[\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1})\right]. \tag{A.17}$$

Since we are going to take the derivative with respect to $\mathbf{W}$, we can simplify all the terms of $F(\mathbf{W})$ that do not depend on $\mathbf{S}$, so that

$$F(\mathbf{W}) = \mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}\mathbf{Y}_{1(\mathrm{k})}\right] + \mu_{1(\mathrm{k})}\mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}+1}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k}+1}\hat{\mathbf{W}}_{\mathrm{k}+1})\right] \tag{A.18}$$

$$= \langle\mathbf{L}_{k+1}^{\mathrm{T}}\mathbf{Y}_{1(k)}, \mathbf{W}\rangle + \mu_{1(k)}\langle\mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}), \mathbf{W}\rangle. \tag{A.19}$$

Which leads to

$$\bar{\nabla}F(\mathbf{W}) = \mathbf{L}_{k+1}^{\mathrm{T}}\mathbf{Y}_{1(k)} + \mu_{1(k)}\mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}), \tag{A.20}$$

and

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}}\left[\mathbf{W}_k + \frac{\mathbf{L}_{k+1}^{\mathrm{T}}\left(\mathbf{L}_{k+1} - \mathbf{L}_{k+1}\mathbf{W}_k - \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}}\right)}{\eta_1}\right]. \tag{A.21}$$

Now, to obtain the updates for $\mathbf{S}_{k+1}$ we will take the derivatives of Eq. A.5 with respect to $\mathbf{S}$. Since the derivatives of all terms that do not depend on $\mathbf{S}$ are zero,

we only need to consider part of the terms of Eq. A.5 to find the update of $\mathbf{S}$.

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right], \tag{A.22}$$

To find the updates of $\mathbf{S}$ matrix we will consider the case where the $\mathbf{\Pi}$ is a all-ones matrix, therefore leaving us with the following expanded cost function

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{S}\|_1 + \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \right]. \tag{A.23}$$

Linearizing the function around the $\mathbf{S}_k$ operation point

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{S}\|_1 + \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{1(k)}\eta_2}{2} \|\mathbf{S} - \mathbf{S}_k\|_2^2 \right]. \tag{A.24}$$

We can now use the quadratic approximation the function of $\mathbf{S}$ as $F(\mathbf{S}) \approx F(\mathbf{S}_k) + \langle \mathbf{S} - \mathbf{S}_k, \bar{\nabla} F(\mathbf{S}_k) \rangle$ according to [105], with

$$F(\mathbf{S}) = \langle (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2. \tag{A.25}$$

In this formulation we have

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{S}\|_1 + \frac{\mu_{1(k)}\eta_2}{2} \left\| \mathbf{S} - \left(\mathbf{S}_k - \frac{1}{\mu_{1(k)}\eta_2} \bar{\nabla} F(\mathbf{S}_k) \right) \right\|_2^2 \right]. \tag{A.26}$$

We can, therefore, write it as [107]

$$\mathbf{S}_{k+1} = \tau_{\frac{\lambda_1}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k - \frac{\bar{\nabla} F(\mathbf{S}_k)}{\mu_{1(k)}\eta_2} \right]. \tag{A.27}$$

Which means that, to find the final update for $\mathbf{S}_{k+1}$ we only need to find $\bar{\nabla} F(\mathbf{S}_k)$. To do so we will re-write $F(\mathbf{S})$, so that

$$F(\mathbf{S}) = \text{Tr} \left[\mathbf{Y}_{1(\text{k})}^{\text{T}} (-\mathbf{\Theta} \odot \mathbf{S})\hat{\mathbf{W}}_{\text{k+1}} \right] + \frac{\mu_{1(\text{k})}}{2} \|\mathbf{L}_{\text{k+1}}\mathbf{W} - \mathbf{L}_{\text{k}}\|_2^2 \tag{A.28}$$

$$= \text{Tr} \left[-\hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} (-\mathbf{\Theta} \odot \mathbf{S})^{\text{T}} \mathbf{Y}_{1(\text{k})} \right] + \frac{\mu_{1(k)}}{2} \times 2 \langle \mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}, (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}_{k+1})\hat{\mathbf{W}}_{k+1} \rangle \tag{A.29}$$

$$= \text{Tr} \left[-(-\mathbf{\Theta} \odot \mathbf{S})^{\text{T}} \mathbf{Y}_{1(\text{k})} \hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} \right] + \mu_{1(k)} \text{Tr} \left[\hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}_{\text{k+1}})^{\text{T}} \mathbf{L}_{\text{k+1}} \hat{\mathbf{W}}_{\text{k+1}} \right]. \tag{A.30}$$

Since we are going to take the derivative with respect to $\mathbf{S}$, we can simplify all the terms of $F(\mathbf{S})$ that do not depend on $\mathbf{S}$, so that

$$F(\mathbf{S}) = \mathrm{Tr}\left[-\mathbf{S}^{\mathrm{T}}(-\boldsymbol{\Theta} \odot \mathbf{Y}_{1(\mathrm{k})})\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\right] + \mu_{1(\mathrm{k})}\mathrm{Tr}\left[-\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}(\boldsymbol{\Theta} \odot \mathbf{S})^{\mathrm{T}}\mathbf{L}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}}\right] \tag{A.31}$$

$$= \langle -(\boldsymbol{\Theta} \odot \mathbf{Y}_{1(k)})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{S}\rangle + \mu_{1(k)}\mathrm{Tr}\left[-\mathbf{S}^{\mathrm{T}}(\boldsymbol{\Theta} \odot \mathbf{L}_{\mathrm{k+1}})\hat{\mathbf{W}}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\right] \tag{A.32}$$

$$= \langle -(\boldsymbol{\Theta} \odot \mathbf{Y}_{1(k)})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{S}\rangle + \mu_{1(k)}\langle -(\boldsymbol{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{S}\rangle. \tag{A.33}$$

Which leads to

$$\bar{\nabla}F(\mathbf{S}) = -(\boldsymbol{\Theta} \odot \mathbf{Y}_{1(k)})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} - \mu_{1(k)}(\boldsymbol{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \tag{A.34}$$

and

$$\mathbf{S}_{k+1} \;=\; \tau_{\frac{\lambda_1}{\mu_{1(k)}\eta_2}}\left[\mathbf{S}_k + \frac{\left((\boldsymbol{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1} + \dfrac{(\boldsymbol{\Theta} \odot \mathbf{Y}_{1(k)})}{\mu_{1(k)}}\right)\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_2}\right]. \tag{A.35}$$

In the case where $\boldsymbol{\Pi}$ is not the identity matrix the update becomes

$$\mathbf{S}_{k+1} \;=\; \tau_{\frac{\lambda_1^*}{\mu_{1(k)}\eta_2}}\left[\mathbf{S}_k + \frac{\left((\boldsymbol{\Theta} \odot \mathbf{L}_{k+1})\hat{\mathbf{W}}_{k+1} + \dfrac{(\boldsymbol{\Theta} \odot \mathbf{Y}_{1(k)})}{\mu_{1(k)}}\right)\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_2}\right], \tag{A.36}$$

where the value of λ_1^* depends on the value of $\boldsymbol{\Pi}$ for the current point.

Finally, to derive the update for $\mathbf{E}_{k+1}$ we need to write the augmented cost function with the terms that depend on $\mathbf{E}_k$

$$\mathbf{E}_{k+1} = \mathrm{argmin}_{\mathbf{E}}\Big[\lambda_2\|\mathbf{E}\|_1 + \langle -\mathbf{E}\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)}\rangle + \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)}\rangle + \frac{\mu_1}{2}\|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2\Big]. \tag{A.37}$$

Linearizing the function around the $\mathbf{E}_k$ operation point

$$\mathbf{E}_{k+1} = \mathrm{argmin}_{\mathbf{E}}\Big[\lambda_2\|\mathbf{E}\|_1 + \langle -\mathbf{E}\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)}\rangle + \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)}\rangle + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_F^2 + \frac{\mu_{2(k)}}{2}\|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 + \frac{\mu_{2(k)}\eta_3}{2}\|\mathbf{E} - \mathbf{E}_k\|_2^2\Big]. \tag{A.38}$$

We can now use the quadratic approximation of the function of $\mathbf{E}$ as $F(\mathbf{E}) \approx$

$F(\mathbf{E}_k) + \langle \mathbf{E} - \mathbf{E}_k, \bar{\nabla} F(\mathbf{E}_k) \rangle$ according to [105], with

$$F(\mathbf{E}) = \langle -\mathbf{E}\hat{\mathbf{W}}_{k+1}, \mathbf{Y}_{1(k)} \rangle + \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)} \rangle + \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W}_{k+1} - \mathbf{L}_{k+1}\|_F^2 + \frac{\mu_{2(k)}}{2} \|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \quad \text{(A.39)}$$

In this formulation we have

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \left[\lambda_2 \|\mathbf{E}\|_1 + \frac{\mu_{2(k)}\eta_3}{2} \|\mathbf{E} - \frac{1}{\mu_{2(k)}\eta_3} \bar{\nabla} F(\mathbf{E}_k)\|_2^2 \right]. \quad \text{(A.40)}$$

We can, therefore, write it as [107]

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}} \left[\mathbf{E}_k - \frac{\bar{\nabla} F(\mathbf{E}_k)}{\mu_{2(k)}\eta_3} \right]. \quad \text{(A.41)}$$

Which means that, to find the final update for $\mathbf{E}_{k+1}$, we only need to find $\bar{\nabla} F(\mathbf{E}_k)$. To do so we will re-write $F(\mathbf{E}) = F(\mathbf{E}^{(1)}) + F(\mathbf{E}^{(2)})$, with

$$F(\mathbf{E}^{(1)}) = \text{Tr}\left[-\mathbf{Y}_{1(\text{k})}^{\text{T}} \mathbf{E}\hat{\mathbf{W}}_{\text{k+1}} \right] + \frac{\mu_{1(\text{k})}}{2} \|\mathbf{L}_{\text{k+1}}\mathbf{W}_{\text{k+1}} - \mathbf{L}_{\text{k}}\|_2^2, \quad \text{(A.42)}$$

and

$$F(\mathbf{E}^{(2)}) = \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)} \rangle + \frac{\mu_{2(k)}}{2} \|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2. \quad \text{(A.43)}$$

Now, we re-write those terms as

$$F(\mathbf{E}^{(1)}) = \text{Tr}\left[-\mathbf{Y}_{1(\text{k})}^{\text{T}} \mathbf{E}\hat{\mathbf{W}}_{\text{k+1}} \right] + \frac{\mu_{1(\text{k})}}{2} \|\mathbf{L}_{\text{k+1}}\mathbf{W}_{\text{k+1}} - \mathbf{L}_{\text{k}}\|_2^2 \quad \text{(A.44)}$$

$$= \text{Tr}\left[-\hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} \mathbf{E}^{\text{T}} \mathbf{Y}_{1(\text{k})} \right] + \frac{\mu_{1(\text{k})}}{2} \|(\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{\text{k+1}} - \mathbf{E})\hat{\mathbf{W}}_{\text{k+1}}\|_2^2 \quad \text{(A.45)}$$

$$= \text{Tr}\left[-\mathbf{E}^{\text{T}} \mathbf{Y}_{1(\text{k})} \hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} \right] + \frac{\mu_{1(k)}}{2} \langle (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E})\hat{\mathbf{W}}_{k+1}, (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E})\hat{\mathbf{W}}_{k+1} \rangle \quad \text{(A.46)}$$

$$= \langle -\mathbf{Y}_{1(k)} \hat{\mathbf{W}}_{k+1}^{\text{T}}, \mathbf{E} \rangle + \frac{\mu_{1(k)}}{2} \times 2 \langle (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E}_k)\hat{\mathbf{W}}_{k+1}, (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E})\hat{\mathbf{W}}_{k+1} \rangle \quad \text{(A.47)}$$

$$= \langle -\mathbf{Y}_{1(k)} \hat{\mathbf{W}}_{k+1}^{\text{T}}, \mathbf{E} \rangle + \mu_{1(k)} \langle \mathbf{L}_k \hat{\mathbf{W}}_{k+1}, (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{k+1} - \mathbf{E})\hat{\mathbf{W}}_{k+1} \rangle \quad \text{(A.48)}$$

$$= \langle -\mathbf{Y}_{1(k)} \hat{\mathbf{W}}_{k+1}^{\text{T}}, \mathbf{E} \rangle + \mu_{1(k)} \text{Tr}\left[((\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S}_{\text{k+1}} - \mathbf{E})\hat{\mathbf{W}}_{\text{k+1}})^{\text{T}} \mathbf{L}_{\text{k}} \hat{\mathbf{W}}_{\text{k+1}} \right]. \quad \text{(A.49)}$$

Since we are going to take the derivative with respect to $\mathbf{E}$, we can simplify all the

terms of $F(\mathbf{E})$ that do not depend on $\mathbf{E}$, so that

$$F(\mathbf{E}^{(1)}) = \langle -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle + \mu_{1(k)}\mathrm{Tr}\left[-(\mathbf{E}\hat{\mathbf{W}}_{\mathrm{k+1}})^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\right] \tag{A.50}$$

$$= \langle -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle + \mu_{1(k)}\mathrm{Tr}\left[-\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\right] \tag{A.51}$$

$$= \langle -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle + \mu_{1(k)}\mathrm{Tr}\left[-\mathbf{E}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\right] \tag{A.52}$$

$$= \langle -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle + \mu_{1(k)}\langle -\mathbf{L}_{k}\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle. \tag{A.53}$$

Which leads to

$$\bar{\nabla}F(\mathbf{E}^{(1)}) = -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} - \mu_{1(k)}\mathbf{L}_{k}\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}. \tag{A.54}$$

Now, for $F(\mathbf{E}^{(2)})$ we have

$$F(\mathbf{E}^{(2)}) = \langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{Y}_{2(k)}\rangle + \frac{\mu_{2(k)}}{2}\|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \tag{A.55}$$

$$= \mathrm{Tr}\left[\mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}(\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon)\right] + \frac{\mu_{2(\mathrm{k})}}{2}\langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\rangle, \tag{A.56}$$

again, ignoring what does not depend on $\mathbf{E}$ and fixing twice one of the $\mathbf{E}$ to $\mathbf{E}_k$ in the second part

$$\begin{aligned} F(\mathbf{E}^{(2)}) = {} & \mathrm{Tr}\left[\mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}\mathbf{E}(\mathbf{I} - \mathbf{D}) - \mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}\mathbf{1}\epsilon\right] + \\ & \frac{\mu_{2(k)}}{2} \times 2 \times \mathrm{Tr}\left[(\mathbf{E} - \mathbf{E}\mathbf{D})^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D} - \mathbf{1})\epsilon\right] \end{aligned} \tag{A.57}$$

$$\begin{aligned} = {} & \mathrm{Tr}\left[(\mathbf{I} - \mathbf{D})^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{Y}_{2(\mathrm{k})}\right] - \mathrm{Tr}\left[\mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}\mathbf{1}\epsilon\right] + \\ & \mu_{2(k)}\mathrm{Tr}\left[\mathbf{E}^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D}_{\mathrm{k}} - \mathbf{1}\epsilon) - \mathbf{D}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D} - \mathbf{1}\epsilon)\right] \end{aligned} \tag{A.58}$$

$$\begin{aligned} = {} & \mathrm{Tr}\left[\mathbf{E}^{\mathrm{T}}\mathbf{Y}_{2(\mathrm{k})}(\mathbf{I} - \mathbf{D})^{\mathrm{T}}\right] - \mathrm{Tr}\left[\mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}\mathbf{1}\epsilon\right] + \\ & \mu_{2(k)}\langle \mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon, \mathbf{E}\rangle + \mu_{2(k)}\langle -(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)\mathbf{D}^{\mathrm{T}}, \mathbf{E}\rangle \end{aligned} \tag{A.59}$$

$$\begin{aligned} = {} & \langle \mathbf{Y}_{2(k)}(\mathbf{I} - \mathbf{D}^{\mathrm{T}}), \mathbf{E}^{\mathrm{T}}\rangle - \mathrm{Tr}\left[\mathbf{Y}_{2(\mathrm{k})}^{\mathrm{T}}\mathbf{1}\epsilon\right] + \\ & \mu_{2(k)}\langle (\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}), \mathbf{E}\rangle, \end{aligned} \tag{A.60}$$

which gives us

$$\bar{\nabla}F(\mathbf{E}^{(2)}) = \mathbf{Y}_{2(k)}(\mathbf{I} - \mathbf{D}^{\mathrm{T}}) + \mu_{2(k)}(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}). \tag{A.61}$$

Putting both parts together we have

$$\begin{aligned} \bar{\nabla}F(\mathbf{E}) = {} & -\mathbf{Y}_{1(k)}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} - \mu_{1(k)}\mathbf{L}_{k}\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} + \mathbf{Y}_{2(k)}(\mathbf{I} - \mathbf{D}^{\mathrm{T}}) + \\ & \mu_{2(k)}(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}), \end{aligned} \tag{A.62}$$

which leads us to the final update function

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}}\left[\mathbf{E}_k + \frac{\frac{\mu_{1(k)}}{\mu_{2(k)}}\left(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1} + \frac{\mathbf{Y}_{1(k)}}{\mu_{1(k)}}\right)\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_3} - \frac{\left(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon + \frac{\mathbf{Y}_{2(k)}}{\mu_{2(k)}}\right)(\mathbf{I} - \mathbf{D}^{\mathrm{T}})}{\eta_3}\right]. \quad \text{(A.63)}$$

A.2 Sequential Algorithm Derivation

For the sequential algorithm the goal is to find the updates that allow one to minimize the following function

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{E},\mathbf{S}} \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1, \\ &\text{s.t.} \begin{cases} \mathbf{A} = \mathbf{L} + \mathbf{\Theta} \odot \mathbf{S} + \mathbf{E} \\ \mathbf{LW} = \mathbf{L} \\ \mathbf{W}_{ii} = 0, \ \forall i \\ (\mathbf{E} - \mathbf{ED})\mathbf{1} \succeq \epsilon, \text{ where } \mathbf{D} = \begin{bmatrix} \mathbf{0^T} & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \end{cases}, \end{aligned} \quad \text{(A.64)}$$

where $\mathbf{W}$ is the weight matrix bearing the relations between the columns of the low-rank representation $\mathbf{L}$ of the data matrix $\mathbf{A}$, $\mathbf{S}$ is the sparse matrix where foreground objects shall lie, and $\mathbf{E}$ is the sparse residue matrix composed of the parts of $\mathbf{A}$ not represented by neither $\mathbf{L}$ nor $\mathbf{S}$. The matrix $\mathbf{1}$ of dimensions $m \times n$ has all entries equal to 1. And $\mathbf{I}$ is the identity matrix.

To find a solution for this optimization problem, we will employ a similar solution to that used in the previous section, with the exception we are not using the duals during the convexification step. The expanded function will feature only the quadratic penalty term from the previously used ALM expansion. The new expanded cost function will be obtained through the use of incremental subgradient-proximal methods described in details in [103] due to its strong convergence guarantees and stability, as explained in the referred article, since in our sequential implementation we will use few iterations to minimize the cost function. This solution is detailed in the sequel. From Eq. (A.64), one has that

$$\mathbf{LW} - \mathbf{L} = \mathbf{0}, \quad \text{(A.65)}$$

and

$$\mathbf{L} = \mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}. \quad \text{(A.66)}$$

Replacing Eq. (A.65) into (A.3), we get

$$(\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E})\mathbf{W} - (\mathbf{A} - \mathbf{\Theta} \odot \mathbf{S} - \mathbf{E}) = \mathbf{0}, \tag{A.67}$$

which yields the following expanded cost function

$$\Gamma(\mathbf{W}, \mathbf{E}, \mathbf{S}, \mu_1, \mu_2) = \|\mathbf{W}\|_1 + \lambda_1\|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \lambda_2\|\mathbf{E}\|_1 + \frac{\mu_1}{2}\|\mathbf{LW} - \mathbf{L}\|_F^2 + \frac{\mu_2}{2}\|\mathbf{E} - \mathbf{ED} - \mathbf{1}\epsilon\|_F^2. \tag{A.68}$$

To obtain the update function for $\mathbf{W}_{k+1}$ we need to write the augmented cost function with the terms that depend on $\mathbf{W}_k$

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2\right], \tag{A.69}$$

Linearizing the function around the $\mathbf{W}_k$ operation point

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{1(k)}\eta_1}{2}\|\mathbf{W} - \mathbf{W}_k\|_2^2\right]. \tag{A.70}$$

We can now use the quadratic approximation the function of $\mathbf{W}$ as $F(\mathbf{W}) \approx F(\mathbf{W}_k) + \langle \mathbf{W} - \mathbf{W}_k, \bar{\nabla}F(\mathbf{W}_k)\rangle$ according to [105], with

$$F(\mathbf{W}) = \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2 \tag{A.71}$$

In this formulation we have

$$\mathbf{W}_{k+1} = \text{argmin}_{\mathbf{W}} \left[\|\mathbf{W}\|_1 + \frac{\mu_{1(k)}\eta_1}{2}\left\|\mathbf{W} - \left(\mathbf{W}_k - \frac{1}{\mu_{1(k)}\eta_1}\bar{\nabla}F(\mathbf{W}_k)\right)\right\|_2^2\right]. \tag{A.72}$$

We can, therefore, write it as [107]

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}}\left[\mathbf{W}_k - \frac{\bar{\nabla}F(\mathbf{W}_k)}{\mu_{1(k)}\eta_2}\right]. \tag{A.73}$$

Which means that, to find the final update for $\mathbf{W}_{k+1}$ we only need to find $\bar{\nabla}F(\mathbf{W}_k)$. To do so we will re-write $F(\mathbf{W})$, so that

$$F(\mathbf{W}) = \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 \tag{A.74}$$

$$= \frac{\mu_{1(k)}}{2}\langle \mathbf{L}_{k+1}\hat{\mathbf{W}}, \mathbf{L}_{k+1}\hat{\mathbf{W}}\rangle, \tag{A.75}$$

here we can expand the righthand side of the equation fixing one of the $\mathbf{W}$ at $\mathbf{W}_k$ and letting the other as it stands. By doing so with both the original $\mathbf{W}$, and since

$\hat{\mathbf{W}} = \mathbf{LW} - \mathbf{L}$ we get

$$F(\mathbf{W}) = \frac{\mu_{1(k)}}{2} \times 2\langle \mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}, \mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\rangle \tag{A.76}$$

$$= \mu_{1(k)}\mathrm{Tr}\left[(\mathbf{L}_{\mathrm{k+1}}\mathbf{W} - \mathbf{L}_{\mathrm{k+1}})^{\mathrm{T}}(\mathbf{L}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}})\right] \tag{A.77}$$

$$= \mu_{1(k)}\mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k+1}}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}}) - \mathbf{L}_{\mathrm{k+1}}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}})\right]. \tag{A.78}$$

Since we are going to take the derivative with respect to $\mathbf{W}$, we can simplify all the terms of $F(\mathbf{W})$ that do not depend on $\mathbf{S}$, so that

$$F(\mathbf{W}) = \mu_{1(k)}\mathrm{Tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{L}_{\mathrm{k+1}}^{\mathrm{T}}(\mathbf{L}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}})\right] \tag{A.79}$$

$$= \mu_{1(k)}\langle \mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}), \mathbf{W}\rangle \tag{A.80}$$

Which leads to

$$\bar{\nabla}F(\mathbf{W}) = \mu_{1(k)}\mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}), \tag{A.81}$$

and

$$\mathbf{W}_{k+1} = \tau_{\frac{1}{\mu_{1(k)}\eta_1}}\left[\mathbf{W}_k + \frac{\mathbf{L}_{k+1}^{\mathrm{T}}(\mathbf{L}_{k+1} - \mathbf{L}_{k+1}\mathbf{W}_k)}{\eta_1}\right], \tag{A.82}$$

Now, to obtain the updates for $\mathbf{S}_{k+1}$ we will take the derivatives of Eq. A.68 with respect to $\mathbf{S}$. Since the derivatives of all terms that do not depend on $\mathbf{S}$ are zero, we only need to consider part of the terms of Eq. A.68 to find the update of $\mathbf{S}$.

$$\mathbf{S}_{k+1} = \mathrm{argmin}_{\mathbf{S}}\left[\lambda_1\|\mathbf{\Pi} \odot \mathbf{S}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2\right]. \tag{A.83}$$

To find the updates of $\mathbf{S}$ matrix we will consider the case where the $\mathbf{\Pi}$ is the identity matrix, therefore leaving us with the following expanded cost function

$$\mathbf{S}_{k+1} = \mathrm{argmin}_{\mathbf{S}}\left[\lambda_1\|\mathbf{S}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2\right]. \tag{A.84}$$

Linearizing the function around the $\mathbf{S}_k$ operation point

$$\mathbf{S}_{k+1} = \mathrm{argmin}_{\mathbf{S}}\left[\lambda_1\|\mathbf{S}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2 + \frac{\mu_{1(k)}\eta_2}{2}\|\mathbf{S} - \mathbf{S}_k\|_2^2\right]. \tag{A.85}$$

We can now use the quadratic approximation of the function of $\mathbf{S}$ as $F(\mathbf{S}) \approx F(\mathbf{S}_k) + \langle \mathbf{S} - \mathbf{S}_k, \bar{\nabla}F(\mathbf{S}_k)\rangle$ according to [105], with

$$F(\mathbf{S}) = \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_{k+1}\|_2^2. \tag{A.86}$$

In this formulation we have

$$\mathbf{S}_{k+1} = \text{argmin}_{\mathbf{S}} \left[\lambda_1 \|\mathbf{S}\|_1 + \frac{\mu_{1(k)}\eta_2}{2} \left\| \mathbf{S} - \left(\mathbf{S}_k - \frac{1}{\mu_{1(k)}\eta_2} \bar{\nabla} F(\mathbf{S}_k) \right) \right\|_2^2 \right]. \tag{A.87}$$

We can, therefore, write it as [107]

$$\mathbf{S}_{k+1} \; = \; \tau_{\frac{\lambda_1}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k - \frac{\bar{\nabla} F(\mathbf{S}_k)}{\mu_{1(k)}\eta_2} \right]. \tag{A.88}$$

Which means that, to find the final update for $\mathbf{S}_{k+1}$ we only need to find $\bar{\nabla} F(\mathbf{S}_k)$. To do so we will re-write $F(\mathbf{S})$, so that

$$\begin{aligned} F(\mathbf{S}) &= \frac{\mu_{1(k)}}{2} \|\mathbf{L}_{k+1}\mathbf{W} - \mathbf{L}_k\|_2^2 && \text{(A.89)} \\ &= \frac{\mu_{1(k)}}{2} \times 2\langle \mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1}, (\mathbf{A} - \boldsymbol{\Theta} \odot \mathbf{S} - \mathbf{E}_{k+1})\hat{\mathbf{W}}_{k+1} \rangle && \text{(A.90)} \\ &= \mu_{1(k)} \text{Tr} \left[\hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} (\mathbf{A} - \boldsymbol{\Theta} \odot \mathbf{S} - \mathbf{E}_{\text{k+1}})^{\text{T}} \mathbf{L}_{\text{k+1}} \hat{\mathbf{W}}_{\text{k+1}} \right]. && \text{(A.91)} \end{aligned}$$

Since we are going to take the derivative with respect to $\mathbf{S}$, we can simplify all the terms of $F(\mathbf{S})$ that do not depend on $\mathbf{S}$, so that

$$\begin{aligned} F(\mathbf{S}) &= \mu_{1(k)} \text{Tr} \left[-\hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} (\boldsymbol{\Theta} \odot \mathbf{S})^{\text{T}} \mathbf{L}_{\text{k+1}} \hat{\mathbf{W}}_{\text{k+1}} \right] && \text{(A.92)} \\ &= \mu_{1(k)} \text{Tr} \left[-\mathbf{S}^{\text{T}} (\boldsymbol{\Theta} \odot \mathbf{L}_{\text{k+1}}) \hat{\mathbf{W}}_{\text{k+1}} \hat{\mathbf{W}}_{\text{k+1}}^{\text{T}} \right] && \text{(A.93)} \\ &= \mu_{1(k)} \langle -(\boldsymbol{\Theta} \odot \mathbf{L}_{k+1}) \hat{\mathbf{W}}_{k+1} \hat{\mathbf{W}}_{k+1}^{\text{T}}, \mathbf{S} \rangle. && \text{(A.94)} \end{aligned}$$

Which leads to

$$\bar{\nabla} F(\mathbf{S}) = -\mu_{1(k)} (\boldsymbol{\Theta} \odot \mathbf{L}_{k+1}) \hat{\mathbf{W}}_{k+1} \hat{\mathbf{W}}_{k+1}^{\text{T}}, \tag{A.95}$$

and

$$\mathbf{S}_{k+1} \; = \; \tau_{\frac{\lambda_1}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k + \frac{(\boldsymbol{\Theta} \odot \mathbf{L}_{k+1}) \hat{\mathbf{W}}_{k+1} \hat{\mathbf{W}}_{k+1}^{\text{T}}}{\eta_2} \right] \tag{A.96}$$

In the case where $\boldsymbol{\Pi}$ is not the identity matrix the update becomes

$$\mathbf{S}_{k+1} \; = \; \tau_{\frac{\lambda_1^*}{\mu_{1(k)}\eta_2}} \left[\mathbf{S}_k + \frac{(\boldsymbol{\Theta} \odot \mathbf{L}_{k+1}) \hat{\mathbf{W}}_{k+1} \hat{\mathbf{W}}_{k+1}^{\text{T}}}{\eta_2} \right], \tag{A.97}$$

where the value of λ_1^* depends on the value of $\boldsymbol{\Pi}$ for the current point.

Finally, to derive the update for $\mathbf{E}_{k+1}$ we need to write the augmented cost function with the terms that depend on $\mathbf{E}_k$

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \left[\lambda_2 \|\mathbf{E}\|_1 + \frac{\mu_1}{2} \|\mathbf{L}\mathbf{W} - \mathbf{L}\|_F^2 + \frac{\mu_2}{2} \|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \right]. \tag{A.98}$$

Linearizing the function around the $\mathbf{E}_k$ operation point

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \Big[\lambda_2\|\mathbf{E}\|_1 + \frac{\mu_{1(k)}}{2}\|\mathbf{LW}-\mathbf{L}\|_F^2 + \frac{\mu_{2(k)}}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_2^2 + \frac{\mu_{2(k)}\eta_3}{2}\|\mathbf{E}-\mathbf{E}_k\|_2^2\Big]. \quad \text{(A.99)}$$

We can now use the quadratic approximation of the function of $\mathbf{E}$ as $F(\mathbf{E}) \approx F(\mathbf{E}_k) + \langle \mathbf{E}-\mathbf{E}_k, \bar{\nabla}F(\mathbf{E}_k)\rangle$ according to [105], with

$$F(\mathbf{E}) = \frac{\mu_{1(k)}}{2}\|\mathbf{LW}-\mathbf{L}\|_F^2 + \frac{\mu_{2(k)}}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_2^2 + \frac{\mu_{2(k)}\eta_3}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_2^2 \quad \text{(A.100)}$$

In this formulation we have

$$\mathbf{E}_{k+1} = \text{argmin}_{\mathbf{E}} \left[\lambda_2\|\mathbf{E}\|_1 + \frac{\mu_{2(k)}\eta_3}{2}\|\mathbf{E}-\frac{1}{\mu_{2(k)}\eta_3}\bar{\nabla}F(\mathbf{E}_k)\|_2^2\right]. \quad \text{(A.101)}$$

We can, therefore, write it as [107]

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}}\left[\mathbf{E}_k - \frac{\bar{\nabla}F(\mathbf{E}_k)}{\mu_{2(k)}\eta_3}\right]. \quad \text{(A.102)}$$

Which means that, to find the final update for $\mathbf{E}_{k+1}$ we only need to find $\bar{\nabla}F(\mathbf{E}_k)$. To do so we will re-write $F(\mathbf{E}) = F(\mathbf{E}^{(1)}) + F(\mathbf{E}^{(2)})$, with

$$F(\mathbf{E}^{(1)}) = \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W}-\mathbf{L}_k\|_2^2, \quad \text{(A.103)}$$

and

$$F(\mathbf{E}^{(2)}) = \frac{\mu_{2(k)}}{2}\|\mathbf{E}-\mathbf{ED}-\mathbf{1}\epsilon\|_2^2. \quad \text{(A.104)}$$

Now, we re-write those terms as

$$\begin{aligned}
F(\mathbf{E}^{(1)}) &= \frac{\mu_{1(k)}}{2}\|\mathbf{L}_{k+1}\mathbf{W}_{k+1}-\mathbf{L}_k\|_2^2 && \text{(A.105)}\\
&= \frac{\mu_{1(k)}}{2}\|(\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}\|_2^2 && \text{(A.106)}\\
&= \frac{\mu_{1(k)}}{2}\langle(\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}, (\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}\rangle && \text{(A.107)}\\
&= \frac{\mu_{1(k)}}{2}\times 2\langle(\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E}_k)\hat{\mathbf{W}}_{k+1}, (\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}\rangle && \text{(A.108)}\\
&= \mu_{1(k)}\langle\mathbf{L}_k\hat{\mathbf{W}}_{k+1}, (\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}\rangle && \text{(A.109)}\\
&= \mu_{1(k)}\text{Tr}\left[\Big((\mathbf{A}-\mathbf{\Theta}\odot\mathbf{S}_{k+1}-\mathbf{E})\hat{\mathbf{W}}_{k+1}\Big)^{\mathrm{T}}\mathbf{L}_k\hat{\mathbf{W}}_{k+1}\right]. && \text{(A.110)}
\end{aligned}$$

Since we are going to take the derivative with respect to $\mathbf{E}$, we can simplify all the

terms of $F(\mathbf{E})$ that do not depend on $\mathbf{E}$, so that

$$F(\mathbf{E}^{(1)}) = \mu_{1(k)}\mathrm{Tr}\left[-(\mathbf{E}\hat{\mathbf{W}}_{\mathrm{k+1}})^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\right] \tag{A.111}$$

$$= \mu_{1(k)}\mathrm{Tr}\left[-\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\right] \tag{A.112}$$

$$= \mu_{1(k)}\mathrm{Tr}\left[-\mathbf{E}^{\mathrm{T}}\mathbf{L}_{\mathrm{k}}\hat{\mathbf{W}}_{\mathrm{k+1}}\hat{\mathbf{W}}_{\mathrm{k+1}}^{\mathrm{T}}\right] \tag{A.113}$$

$$= \mu_{1(k)}\langle -\mathbf{L}_k\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}, \mathbf{E}\rangle. \tag{A.114}$$

Which leads to

$$\bar{\nabla}F(\mathbf{E}^{(1)}) = -\mu_{1(k)}\mathbf{L}_k\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}. \tag{A.115}$$

Now, for $F(\mathbf{E}^{(2)})$ we have

$$F(\mathbf{E}^{(2)}) = \frac{\mu_{2(k)}}{2}\|\mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\|_2^2 \tag{A.116}$$

$$= \frac{\mu_{2(k)}}{2}\langle \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon, \mathbf{E} - \mathbf{E}\mathbf{D} - \mathbf{1}\epsilon\rangle, \tag{A.117}$$

again, ignoring what does not depend on $\mathbf{E}$ and fixing twice one of the $\mathbf{E}$ to $\mathbf{E}_k$ in the second part

$$F(\mathbf{E}^{(2)}) = \frac{\mu_{2(k)}}{2} \times 2 \times \mathrm{Tr}\left[(\mathbf{E} - \mathbf{E}\mathbf{D})^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D} - \mathbf{1})\epsilon\right] \tag{A.118}$$

$$= \mu_{2(k)}\mathrm{Tr}\left[\mathbf{E}^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D}_{\mathrm{k}} - \mathbf{1}\epsilon) - \mathbf{D}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}(\mathbf{E}_{\mathrm{k}} - \mathbf{E}_{\mathrm{k}}\mathbf{D} - \mathbf{1}\epsilon)\right] \tag{A.119}$$

$$= \mu_{2(k)}\langle \mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon, \mathbf{E}\rangle + \mu_{2(k)}\langle -(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)\mathbf{D}^{\mathrm{T}}, \mathbf{E}\rangle \tag{A.120}$$

$$= \mu_{2(k)}\langle (\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}), \mathbf{E}\rangle, \tag{A.121}$$

which gives us

$$\bar{\nabla}F(\mathbf{E}^{(2)}) = \mu_{2(k)}(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}). \tag{A.122}$$

Putting both parts together we have

$$\bar{\nabla}F(\mathbf{E}) = -\mu_{1(k)}\mathbf{L}_k\hat{\mathbf{W}}_{k+1}\hat{\mathbf{W}}_{k+1}^{\mathrm{T}} + \mu_{2(k)}(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}}), \tag{A.123}$$

which leads us to the final update function

$$\mathbf{E}_{k+1} = \tau_{\frac{\lambda_2}{\mu_{2(k)}\eta_3}}\left[\mathbf{E}_k + \frac{\frac{\mu_{1(k)}}{\mu_{2(k)}}(\mathbf{L}_{k+1}\hat{\mathbf{W}}_{k+1})\hat{\mathbf{W}}_{k+1}^{\mathrm{T}}}{\eta_3} - \frac{(\mathbf{E}_k - \mathbf{E}_k\mathbf{D} - \mathbf{1}\epsilon)(\mathbf{I} - \mathbf{D}^{\mathrm{T}})}{\eta_3}\right]. \tag{A.124}$$