Escuela Politécnica Superior

Departamento de Tecnología Electrónica y de las Comunicaciones

QUALITY-DRIVEN VIDEO ANALYSIS FOR THE IMPROVEMENT OF FOREGROUND SEGMENTATION

PhD Thesis written by
Diego Ortego Hernández
under the supervision of
Dr. José María Martínez Sánchez
and
Dr. Juan Carlos San Miguel Avedillo

Madrid, May 2018

Department: Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid, Spain

PhD Thesis: Quality-driven video analysis for the improvement of foreground segmentation

Author: **Diego Ortego Hernández**
Ingeniero de Telecomunicación
(Universidad Autónoma de Madrid)

Supervisor: **Jose María Martínez Sánchez**
Doctor Ingeniero de Telecomunicación
(Universidad Politécnica de Madrid)
Universidad Autónoma de Madrid, Spain

Supervisor: **Juan Carlos San Miguel Avedillo**
Doctor Ingeniero de Telecomunicación
Universidad Autónoma de Madrid, Spain

Year: 2018

Comittee: **Jesús Bescós Cano**
Universidad Autónoma de Madrid, Spain

Andrea Cavallaro
Queen Mary University of London, United Kingdom

Noel E. O'Connor
Dublin City University, Ireland

The work described in this Thesis was carried out within the Video Processing and Understanding Lab at the Department of Tecnología Electrónica y de las Comunicaciones, Escuela Politécnica Superior, Universidad Autónoma de Madrid (from 2014 to 2018). It was partially supported by the Spanish Government (TEC2014-53176-R, HAVideo) and by the Department of Tecnología Electrónica y de las Comunicaciones (Universidad Autónoma de Madrid).

To Karen and my family.

It is easier to go down a hill than up, but the view is from the top.
- Arnold Bennett (1867-1931)

Acknowledgments

This thesis is the result of four years of work. There has been ups and downs but I've enjoyed every moment. It is a pleasure to thank the many people who made this thesis possible.

First, I would like to express my gratitude to Dr. José María Martínez Sánchez (Chema) and Dr. Jesús Bescós Cano for giving me the opportunity to do this thesis, as I have felt their support and transparency from the beginning. Second, but not less important, I would like to thank Dr. Juan Carlos San Miguel Avedillo (JC), for engaging my interest for this amazing world.

Concerning the thesis, I would like to thank my supervisors Chema and JC for their advice and guidance throughout all this time. On the one hand, they have progressively taught me how to work methodologically which, for sure, has been one key ingredient to successfully end this thesis. On the other hand, I have had (and still having) many helpful research discussions that have oriented me in the proper direction. I expect to be able to continue collaborating with them in the future years. Beyond my supervisors, I would like to thank all the VPULab members (Álvaro, Marcos, Pencho and Rafa) for contributing, in different forms, to this thesis. Specially, I would like to thank Dr. Marcos Escudero Viñolo for all the fruitful discussions that we have had during this process, I believe that they have helped me a lot to evolve as a researcher.

One part of the doctoral training that has not been included in this document is the work I did during my research stay at Dublin City University in the beginning of the fourth year. First, I would like to thank Noel O'Connor for giving me the opportunity to spend three months at his fantastic lab. I found many friendly people that helped me during my stay there. Second, I would like to thank Kevin McGuinness for his advice and guidance there, sharing with me many ideas and having fruitful discussions that have helped me to start in a new manner of approaching computer vision tasks. Furthermore, I would like to make special mention to Camille, Eric and Eva as they made me enjoy my time there.

Going back to the degree that led me to the thesis, I would like to recognize and thank the key role of my cousin, Carlos, for his valuable help that shaped me to address a thesis in the best conditions. Carlos is not only a cousin, but a really good friend.

I do not want to forget my family (Emilio, Encarnita and Elena). Since I was a child, they created a fantastic environment at home that has led me to achieve all the targets that I have

pursued during my life. I could never pay all this in a different way that trying to follow their values and principles.

Finally, I am very grateful for the love, patience and support that Karen has provided me during this thesis. She is the most important ingredient in it and she has much responsibility in the person that I am now (a better one I guess). Karen, you deserve my love and gratitude now and in the future years.

Diego Ortego Hernández
May 2018

Abstract

Nowadays, the huge amount of available video content demands the creation of automatic systems for its understanding. In this context, the research community continuously improves the performance of these systems developing new algorithms that are methodologically evaluated in benchmarks via annotated ground-truth data. However, little interest is directed towards understanding the performance of the results when ground-truth is not available (stand-alone evaluation or quality estimation), which enables both an evaluation without costly annotation processes and an online understanding of errors that might be useful to improve results during run-time. In particular, the segmentation of objects of interest in videos or foreground segmentation is a relevant research area motivated by its variety of applications in topics such as video-surveillance or video edition. This thesis addresses tasks related to foreground segmentation that can improve its results while being independent of its internal details, background estimation from video frames and stand-alone quality estimation of foreground segmentation masks. Furthermore, it proposes a foreground segmentation improvement framework based on quality information.

In the first part of this thesis, two algorithms are proposed for both overcoming background estimation and applying it to stationary object detection. Therefore, this part starts by developing a block-level background estimation algorithm robust to stationary objects due to the combination of a temporal analysis to obtain a set of background candidates and spatial analysis to enforce smoothness constraints selecting the right background candidate in each image location. Then, a practical use of background estimation for stationary object detection is explored by continuously estimating background images at different sampling instants and comparing them to determine stationarity. This approach is based on an online clustering that enables fast adaptation to scene variations while analyzing spatio-temporal changes to detect the stationary objects. Experiments on a variety of datasets demonstrate the efficiency of the two proposed background estimation related approaches proposed.

In the second part, this thesis estimates the quality of foreground segmentation algorithms from a stand-alone perspective and proposes a post-processing framework that exploits quality information to improve algorithm results. Firstly, this part addresses the stand-alone evaluation of foreground masks by extracting properties over their connected components (blobs). In

particular, an extensive comparison in terms of correlations with ground-truth based evaluation metrics and capabilities for quality-levels discrimination for 21 measures, revealing that fitness between blobs and segmented image regions (fitness-to-regions) is a good quality estimator. Afterwards, this thesis proposes a post-processing framework to improve foreground segmentation performance exploiting fitness-to-regions. To do so, a motion-aware hierarchical image segmentation of each frame is built to allow quality estimation at different degrees of detail (without merging foreground and background image regions). This hierarchical framework enables the estimation of a combined quality. Finally, this foreground quality is transformed and exploited together with spatial color relations to improve the foreground mask via an optimal labeling process. The experiments conducted over large and heterogeneous datasets with varied challenges validate the utility of this approach.

Resumen

Actualmente, la gran cantidad de contenido visual disponible demanda la creación de herramientas automáticas de análisis. En este sentido, la comunidad investigadora mejora continuamente estos sistemas mediante el desarrollo de nuevos algoritmos que se evalúan de manera metodológica en conjuntos de datos de referencia que disponen de anotaciones humanas del resultado esperado. No obstante, existen pocos trabajos centrados en entender cómo evaluar los algoritmos cuando no hay datos anotados disponibles, situación que permitiría tanto evitar los costosos procesos de anotación humana, como entender los errores de los algoritmos en tiempo de ejecución para poder mejorarlos. En particular, la segmentación de objetos de primer plano o frente en secuencias de vídeo es un área de investigación de gran relevancia debido a sus múltiples aplicaciones en tareas tales como la vídeo-vigilancia o la edición de vídeo. En este sentido, esta tesis aborda tareas relacionadas con la segmentación de frente que tienen capacidad para mejorar sus resultados a la vez que son independientes de sus particularidades, la estimación de fondo de escena y la estimación de calidad sin utilizar datos anotados. Además, esta tesis propone un marco para la mejora de la segmentación de objetos de frente utilizando información de calidad.

En la primera parte de esta tesis, se proponen un algoritmo de estimación de fondo y otro para la detección de objetos estáticos. Esta parte comienza con el desarrollo de un algoritmo de estimación de fondo a nivel de bloque que es robusto a los problemas derivados de los objetos estáticos gracias a la combinación de una etapa temporal que obtiene un conjunto de candidatos de fondo y una etapa espacial que selecciona el candidato adecuado siguiendo criterios de continuidad espacial. A continuación, se explora la detección de objetos estáticos como uso práctico de la estimación de imágenes de fondo mediante la utilización de dichas imágenes en instantes temporales sucesivos para compararlas y generar las detecciones. Este algoritmo se basa en un agrupamiento temporal de bloques que permite una actualización rápida a la vez que se analizan las variaciones espacio-temporales para detectar los objetos estáticos. Los experimentos realizados en múltiples conjuntos de datos demuestran la utilidad de los dos algoritmos desarrollados.

En la segunda parte, esta tesis estima la calidad de algoritmos de segmentación de objetos de frente sin utilizar datos anotados y propone un esquema que utiliza esta calidad para mejorar los resultados de los algoritmos. Esta parte comienza abordando la evaluación sin datos anotados

mediante el cálculo de propiedades de las componentes conexas de la máscara de objetos de frente. En particular, se hace un extenso estudio comparativo de 21 propiedades mediante el análisis de la correlación con las medidas calculadas empleando datos anotados y de la separación de calidad que se obtiene entre distintos niveles de calidad. Este estudio revela que el ajuste entre regiones de la imagen y las componentes conexas de la máscara de segmentación (ajuste a regiones) es un buen estimador de calidad. A continuación, esta tesis propone un algoritmo de pos-procesado que emplea el ajuste a regiones para mejorar la calidad de los resultados. Para lograrlo, se emplea una segmentación jerárquica de la imagen que considera información de movimiento para prevenir la fusión de regiones de frente y fondo. Esta jerarquía permite la estimación de múltiples calidades a distintos niveles de detalle para combinarlas en una única calidad. Finalmente, esta calidad de la máscara se utiliza junto con las relaciones de color de la imagen para mejorar la máscara de segmentación de frente mediante un proceso de etiquetado óptimo. Los experimentos realizados sobre numerosos conjuntos de datos y algoritmos demuestran la utilidad del algoritmo propuesto.

Contents

I Introduction **1**

1 Introduction **3**

1.1 Motivation . . . 3
1.2 Objectives . . . 8
1.3 Major contributions . . . 9
1.4 Structure of the document . . . 10

II Background estimation **13**

2 Background estimation in videos with stationary objects **15**

2.1 Introduction . . . 15
2.2 Related work . . . 16
2.3 Proposed approach: overview . . . 18
2.4 Temporal Analysis . . . 19
2.4.1 Motion filtering . . . 19
2.4.2 Dimensionality Reduction . . . 20
2.4.3 Clustering . . . 20
2.5 Spatial Analysis . . . 23
2.5.1 Seed Selection . . . 24
2.5.2 Sequential Multipath Reconstruction . . . 25
2.5.3 Rejection based Multipath Reconstruction . . . 28
2.6 Experimental work . . . 32
2.6.1 Evaluation framework . . . 34
2.6.2 Temporal analysis evaluation . . . 36
2.6.3 Seed selection technique evaluation . . . 37
2.6.4 Spatial analysis evaluation . . . 39
2.6.5 Comparison against related approaches . . . 39
2.6.6 Evaluation in SBMnet2016 dataset . . . 45

2.6.7 Evaluation framework . . . 45
2.6.8 Parametrization . . . 45
2.6.9 Results in SBMnet dataset . . . 46
2.7 Conclusions . . . 52

3 Background updating for stationary object detection 55
3.1 Introduction . . . 55
3.2 Overview . . . 56
3.3 Online Block Clustering . . . 57
3.3.1 Matching metric . . . 58
3.4 Stationary Block Detection . . . 60
3.5 Experimental Results . . . 61
3.5.1 Setup . . . 61
3.5.2 Comparative evaluation . . . 62
3.6 Conclusions . . . 63

III Foreground segmentation 65

4 Foreground segmentation quality 67
4.1 Introduction . . . 67
4.2 Related work . . . 69
4.3 Stand-alone generic quality measures . . . 72
4.3.1 Contrast-based measures . . . 72
4.3.2 Uniformity-based measures . . . 75
4.3.3 Shape-based measures . . . 76
4.3.4 Fitness-based measures . . . 77
4.3.5 Density-based measures . . . 78
4.4 Experimental methodology . . . 79
4.4.1 Dataset and algorithms . . . 79
4.4.2 Blob-level performance measures . . . 81
4.4.3 Similarity of measures . . . 82
4.5 Experimental results . . . 83
4.5.1 Measures relationships . . . 83
4.5.2 Quality levels separation . . . 87
4.5.3 Ranking . . . 88
4.5.4 Discussion . . . 89
4.6 Conclusions . . . 91

5 Foreground segmentation improvement **93**
5.1 Introduction . . . 93
5.2 Related Work . . . 95
5.3 Foreground mask improvement . . . 96
5.3.1 Overview . . . 96
5.3.2 Description . . . 97
5.4 Experimental work . . . 107
5.4.1 Experimental methodology . . . 107
5.4.2 Effect of parameters in performance improvement . . . 109
5.4.3 Improvement over the original algorithms in CDNET2014, LASIESTA, SABS and BMC datasets . . . 111
5.4.4 Comparison against the state-of-the-art . . . 114
5.4.5 Applying foreground quality to algorithm combination . . . 115
5.4.6 Discussion . . . 115
5.5 Conclusions . . . 117

IV Conclusions **119**

6 Achievements, conclusions and future work **121**
6.1 Summary of achievements and main conclusions . . . 121
6.2 Future Work . . . 123

V Appendixes **127**

A Publications **129**

B Logros, conclusiones y trabajo futuro **131**
B.1 Resumen de logros y principales conclusiones . . . 131
B.2 Trabajo futuro . . . 133

Glossary **137**

Bibliography **139**

List of Figures

1.1 Segmentation tasks with increasing semantic knowledge. 4
1.2 Background subtraction challenges: illumination changes, camera jitter and ghosts. 5
1.3 Background subtraction challenges: stationary objects, dynamic background, shadows and camouflage. 6
1.4 Schemes to improve a background subtraction result. 7
1.5 Dependence among the chapters of this thesis. 11

2.1 Overview of the proposed multipath approach for temporal-spatial block-level background estimation. 19
2.2 Example of a dendrogram to detect the optimal clustering partition for a 8-block set. 22
2.3 Example of normalized scores for clustering validation. 23
2.4 Seed Selection example. 25
2.5 Multipath reconstruction scheme. 26
2.6 Example of reconstruction scheme. 27
2.7 RMR diagram of operations. 29
2.8 Scheme used to compute the inter-block color dissimilarity measure. 30
2.9 Example of the benefits that introduces the use of intra-block heterogeneity and inter-block dissimilarity in RMR. 31
2.10 Visual examples of the selected sequences for evaluation. 34
2.11 Clustering evaluation. 37
2.12 Comparison of SMR against RMR. 39
2.13 Examples of failures of SMR solved by RMR. 40
2.14 Evaluation of RMR against state-of-the-art methods for the task of BE in terms of average AE for the proposed dataset. 40
2.15 Sequence by sequence AUC of RMR (blue) against DCT (black) and SGMM-SOD (red) for the task of BE. 43
2.16 Qualitative results showing the estimated background of top selected approaches for the BE task. 44

2.17 Example of estimated backgrounds for Basic category of SBMnet2016 dataset. . 46
2.18 Example of estimated backgrounds for Intermittent Motion category of SBMnet2016 dataset. 47
2.19 Example of estimated backgrounds for Clutter category of SBMnet2016 dataset. 48
2.20 Example of estimated backgrounds for Jitter category of SBMnet2016 dataset. . 49
2.21 Example of estimated backgrounds for Jitter category of SBMnet2016 dataset. . 50
2.22 Example of estimated backgrounds for Background Motion category of SBMnet2016 dataset. 51
2.23 Example of estimated backgrounds for Very Long category of SBMnet2016 dataset. 52
2.24 Example of estimated backgrounds for Very Short category of SBMnet2016 dataset. 53

3.1 Block diagram of the proposed approach. 57
3.2 Example of the temporal analysis for a block location where the stability is modified changing from the empty scene to a suitcase. 58
3.3 Examples of ratio of two blocks. 59
3.4 Sequence of operations for Stationary Block Detection in a sampling instant. . . 61
3.5 Examples of detections in each dataset. 63

4.1 Taxonomy for evaluation measures of foreground segmentation masks. 69
4.2 Examples of contrast-based measures using inner and outer regions in the contour of a blob mask. 73
4.3 Examples of stand-alone generic measures motion difference (MD) and spatial uniformity (SU). 75
4.4 Examples of stand-alone generic measures boundary complexity (BX) and superpixel straddling (SS). 77
4.5 Examples of external density measure (SE). 79
4.6 Example images from selected categories of CDNET2014 dataset. 80
4.7 Example of ground-truth (GT) based quality measures. 82
4.8 Cross-correlation among quality measures and clustering obtained via agglomerative hierarchical clustering. 84
4.9 Self-Organizing Map of stand-alone generic quality measures. 85
4.10 Correlation among blob-level performance F and the subset of interesting stand-alone measures. 86
4.11 Example of SS failures due to erroneously segmented image regions that merge objects with the environment. 87
4.12 Probability density functions (pdf) of the scores for top-3 quality measures to discriminate different ground-truth performance levels. 88

4.13 Example of replication of ground-truth based ranking (frame 2530 from Fall sequence). 88
4.14 Example of superpixel straddling (SS) capabilities to estimate low quality in False Positive Blobs. 90
4.15 Example of Superpixel Straddling (SS) capabilities to estimate quality. 90

5.1 Foreground improvement framework overview. 97
5.2 Example for the ultrametric contour map (UCM). 98
5.3 Example of motion-aware image segmentation. 99
5.4 Hierarchical quality estimation. 102
5.5 Weighting function to combine all hierarchy levels. 103
5.6 Example for the effect of the proposed weighted average. 104
5.7 Example of the foreground segmentation process. 106
5.8 Example of foreground segmentation improvement when using or not the pairwise potential. 107
5.9 Examples of the effect of parameter in the performance for CDNET2014 and LASIESTA datasets. 111
5.10 Example of foreground improvements in LASIESTA dataset. 113
5.11 Example of foreground improvements in SABS and BMC datasets. 114
5.12 Example of foreground improvements in CDNET2014 dataset. 116
5.13 Relative computational complexity for the proposed approach. 117

List of Tables

2.1 Key symbols and notations 21
2.2 Dataset description. 35
2.3 Seed selection technique evaluation. 38
2.4 Comparison against state-of-the-art methods in terms of AUC and SBMI2015 error measures for the proposed dataset. 41
2.5 Comparison against state-of-the-art methods in terms of AUC and SBMI2015 error measures for the SBMI dataset. 42
2.6 Results for Basic category of SBMnet2016 dataset. 46
2.7 Results for Intermittent Motion category of SBMnet2016 dataset. 47
2.8 Results for Clutter category of SBMnet2016 dataset. 48
2.9 Results for Jitter category. 49
2.10 Results for Illumination Changes category of SBMnet2016 dataset. 50
2.11 Results for Background Motion category. 51
2.12 Results for Very Long category of SBMnet2016 dataset. 52
2.13 Results for Very Short category. 53

3.1 Comparative evaluation. 62

4.1 Description of selected quality measures. 72
4.2 Background subtraction algorithms selected to analyze blob properties. 81
4.3 Selected measures to estimate quality. 86
4.4 Ranking obtained by stand-alone generic measures compared to blob-level performance F for TPBs and FPBs. 89
4.5 Ranking obtained by stand-alone generic measures compared to blob-level performance F for TPBs. 89

5.1 Selected background subtraction algorithms. 108
5.2 Example of the effect of and in the F-score. 109
5.3 Overall average performance for LASIESTA, SABS and BMC datasets. 111
5.4 Per-category average foreground segmentation performance in CDNET2014 dataset. 112

5.5 Performance comparison of the proposed framework against related work. 114
5.6 Algorithm combination comparisons against related work. 115

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

Computer vision is the research field that aims to provide machines or computers with the capability to visually sense the world around them as good or better than humans do. In this sense, computer vision includes methods for acquiring, processing, analyzing and understanding image data to act accordingly. This data can take many forms, such as video sequences from one or multiple cameras and multi-dimensional data from a medical scanner.

As a wide field, computer vision has numerous applications; in agriculture, augmented reality, autonomous vehicles, biometrics, character recognition, forensics, industrial quality inspection, face recognition, gesture analysis, geoscience, image restoration, medical image analysis, pollution monitoring, process control, remote sensing, robotics, security and surveillance transport; where there is a common need: results with reliable performance. Therefore, obtaining a meaningful performance implicitly involves the development of algorithms that seek improvements in those cases where previous algorithms do not succeed.

One fundamental task in computer vision is the partition or segmentation of an image into meaningful regions (see Figure 1.1) that can serve as base information to a wide variety of applications, such as action recognition [Ghodrati et al., 2014], event detection [Fan et al., 2013] or autonomous driving [Siam et al., 2017]. Firstly, the simplest form of segmentation attending to semantic knowledge is superpixel segmentation, which groups sets of connected pixels sharing spatial [Felzenszwalb and Huttenlocher, 2004][Achanta et al., 2012] or spatio-temporal [Brendel and Todorovic, 2009][Galasso et al., 2012] properties without knowledge of each superpixel semantic category or label. Secondly, foreground segmentation reveals interesting objects in a scene or foreground by segregating them from the rest of the scene or background [Bouwmans, 2014], thus providing some semantic information. Thirdly, as an evolution of object detection, instance segmentation [He et al., 2017][Hu et al., 2017] individually segments each object in the scene with a semantic label and without labeling all the "stuff" classes, i.e. sky, tree, etc.

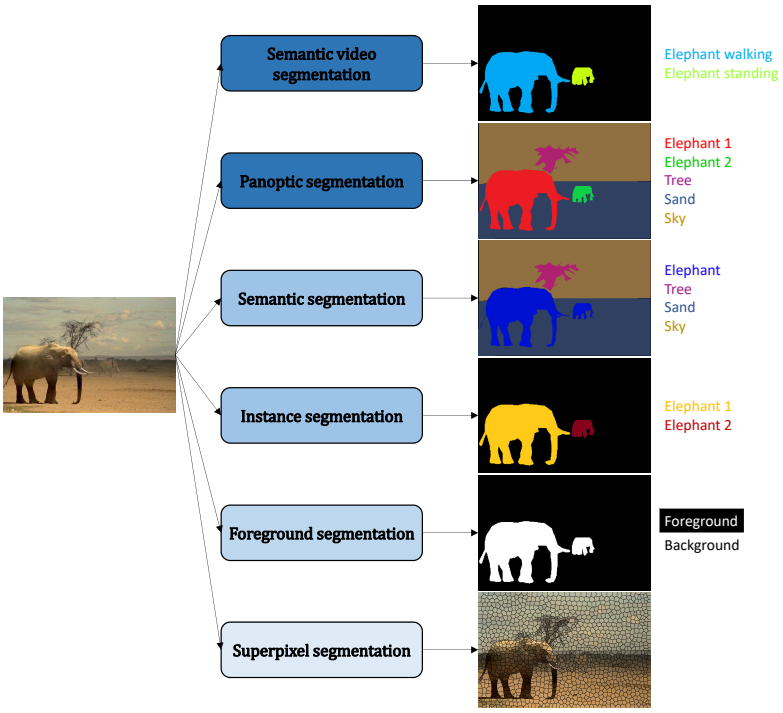


Figure 1.1: Segmentation tasks with increasing semantic knowledge.

Fourthly, semantic segmentation aims to label each pixel of the scene with a semantic category [Zhao et al., 2017a], thus requiring a learning processes to previously discover the appearance of each category. Fifthly, as a natural extension of semantic segmentation and instance segmentation, panoptic segmentation [Kirillov et al., 2018] has been named as the task of evolving semantic segmentation to a labeling process where also each instance of a non-stuff semantic category receives a different label. Finally, segmentation can go a step further and associate semantic labels to those objects involved in an action [Xu and Corso, 2016][Qiu et al., 2017], i.e. a label could be "Elephant walking".

Among these segmentation tasks, foreground segmentation in videos plays and important role in many applications such as video-surveillance [Bouwmans, 2014] and video edition [Hu et al., 2017] and has greatly evolved during the last two decades [Stauffer and Grimson, 1999][St-Charles et al., 2015][Babaee et al., 2018]. Foreground segmentation aims to detect the objects of interest or foreground in images or videos [Bouwmans, 2014][Borji et al., 2015][Perazzi et al., 2016][Minaee and Wang, 2017] where such "interest" depends on the application domain. For example, foreground in images can be defined as salient or co-salient objects [Borji et al., 2015][Tsai et al., 2016][Zhang et al., 2017a] or as generic objects [Alexe et al., 2012][Jain et al., 2017]. In videos, foreground may correspond to all moving objects [Bouwmans, 2014] or specific objects relying on saliency [Wang et al., 2015b] or co-saliency [Yao et al., 2017], spatio-temporal patterns

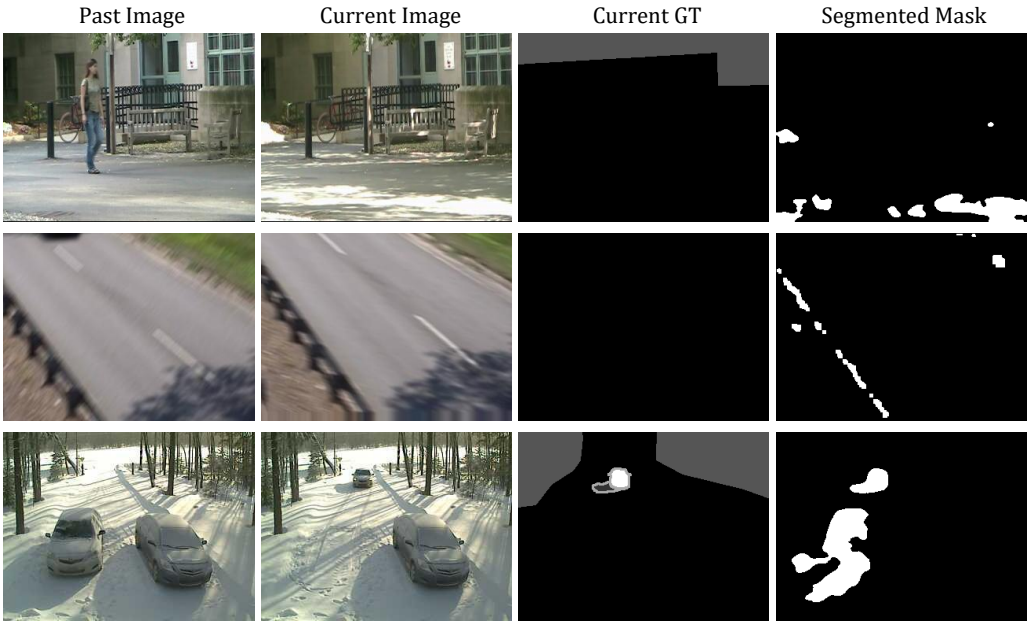


Figure 1.2: Background subtraction challenges: illumination changes, camera jitter and ghosts. Each row shows, from left to right: a previous image in the video, current image, current ground-truth (labels are presented for objects, their shadows and contours and regions to discard) and segmented foreground mask. The first row presents how scene illumination changes over time can lead to false positives. The second row shows that camera jitter induces false positives. The third row presents a ghost detection of a car that was part of the background but has moved.

[Lee et al., 2011] or weak labels [Zhang et al., 2017b]. Moreover, unconstrained video object segmentation addresses challenges related to camera motion, shape deformations of objects or motion blur [Faktor and Irani, 2014]. Existing approaches are unsupervised (e.g. detect spatio-temporal relevant objects [Papazoglou and Ferrari, 2013][Wang et al., 2015a]), semi-supervised (e.g. propagate initially segmented objects [Jain and Grauman, 2014]) or supervised (e.g. frame-by-frame human intervention [Maninis et al., 2017]). Furthermore, there are scenarios with a relative control of camera motion where video object segmentation is tackled through background subtraction [Bouwmans, 2014][Yang et al., 2015], which compares each frame with a background model of the sequence.

Focusing on controlled camera motion scenarios, background subtraction algorithms usually have four stages [Bouwmans, 2014]: *Modeling*, to statistically represent the background of the scene; *Initialization*, to acquire the first model; *Maintenance*, to adapt the model to scene variations over time; and *Detection*, to segment foreground objects by comparing each frame and the model. This foreground segmentation process poses several challenges [Bouwmans, 2014] that have a direct impact in the foreground segmentation performance. Figure 1.2 presents false positives caused by illumination changes (new illumination not included in the model), camera

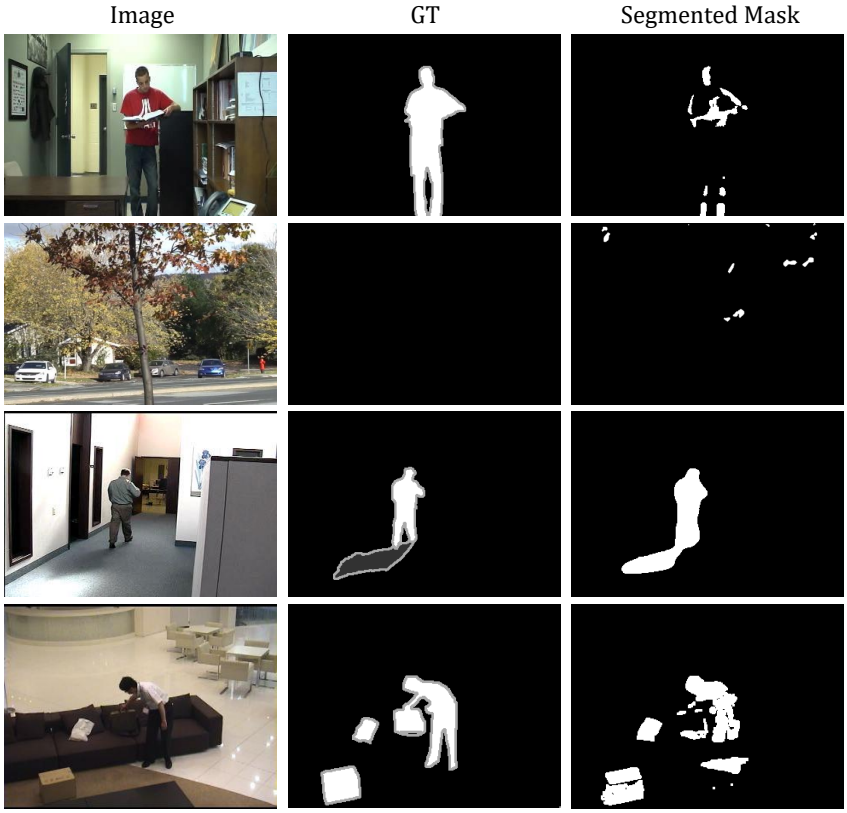


Figure 1.3: Background subtraction challenges: stationary objects, dynamic background, shadows and camouflage. Each row shows, from left to right: image under analysis, current ground-truth (labels are presented for objects, their shadows and contours and regions to discard) and segmented foreground mask. The first row presents a stationary person with undetected parts as it is being absorbed by the background model. The second row presents a scene with a dynamic background, i.e. a waving tree, that induces false positives. The third row presents a classical detection of a cast shadow. Finally, the fourth row presents how the brown binder carried by a person loses some parts due to similarities with the brown color of the sofa.

jitter (camera motion shifts spatially image pixels from their corresponding background ones) and ghosts (objects included in the background move, revealing a background that is not part of the model); whereas Figure 1.3 presents issues with stationary objects (motionless or stationary foreground may be erroneously incorporated into the model), dynamic backgrounds (motion associated with the background that the model is not able to handle), cast shadows (shadows from objects are sometimes detected) and camouflages (strong similarity between objects and background).

Therefore, connecting with the need of reliable results, background subtraction can be improved through several strategies: model changes, features changes and output changes (see Figure 1.4). Firstly, changing the background model to select an appropriate one is closely re-

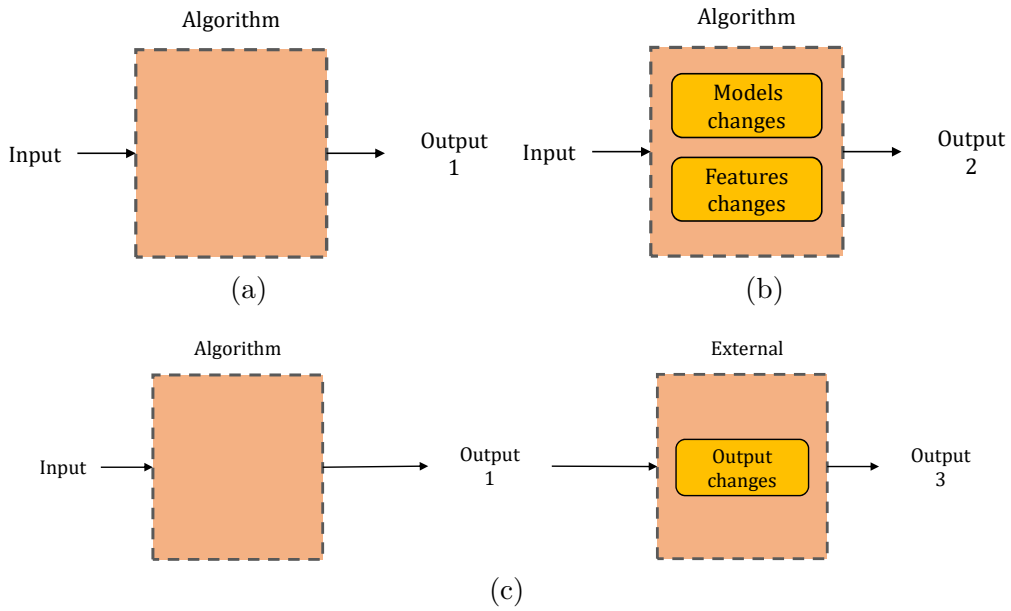


Figure 1.4: Schemes to improve a background subtraction result. The original algorithm (a) computes an Output 1 that would like to be improved. Traditional approaches seek algorithm-dependent changes in the model and the features (b) to obtain an improved result Output 2. Alternatively, changes in the output (c) can be done to obtain an improved result Output 3, which is external and do not need to modify internal algorithm details, but to estimate some output properties.

lated with the ability to deal with several challenges at the same time [Bouwmans, 2014] while properly adapting the background model to scene variations. Secondly, changing the features (e.g. color, gradient, texture, motion) is widely done by the background subtraction algorithms [López-Rubio and López-Rubio, 2015a][Dey and Kundu, 2016][Bouwmans et al., 2016] to deal with the wide variety of challenges posed by background subtraction. Moreover, deep learning models [Braham and Droogenbroeck, 2016][W. et al., 2017] have recently emerged as promising frameworks to unify modeling and feature selection. Finally, foreground segmentation masks can be also improved by adopting output changes or post-processing techniques from extracted output (foreground) properties to either remove false positives or recover false negatives [Parks and Fels, 2008]. These techniques stand out as a very interesting alternative for the improvement of foreground segmentation masks as they can be performed independently of the algorithm, thus avoiding the complex task of modifying features or models that are inherent to each algorithm.

In the literature, performance improvement through post-processing has been mainly addressed using morphological operations [Dougherty, 1992][St-Charles et al., 2015] to fill holes or remove small blobs and inspecting generic foreground mask properties [Schick et al., 2012][Giordano et al., 2015][Braham et al., 2017] to filter erroneous foreground and expand to undetected areas. Furthermore, there are algorithms dealing with false positives, such as illumination

changes [Chen and Ellis, 2014][López-Rubio and López-Rubio, 2015b], shadows [Sanin et al., 2012][Huerta et al., 2015] or dynamic backgrounds [St-Charles et al., 2015][Pham et al., 2015], but they develop robust features that are dependent on the algorithm, as they use the background model, rather than using image and foreground information that are algorithm independent.

Therefore, among the post-processing approaches, the use of generic foreground mask properties provides independence of specific phenomena (e.g. illumination or shadows) and, unlike morphological operations, introduces complementary information to the foreground mask. Furthermore, these foreground mask properties provide insights of foreground segmentation performance [Correia and Pereira, 2002][Erdem et al., 2004][SanMiguel and Martinez, 2010] and can be used to improve the foreground masks [Giordano et al., 2015].

Regarding the performance evaluation of background subtraction algorithms, there are many proposals in the literature ranging from qualitative visualization tools [Ramadan, 2006][Song et al., 2014][Sánchez Rodríguez et al., 2014] to quantitative reference-based evaluations [Nascimento and Marques, 2006][Brutzer et al., 2011][Wang et al., 2014b] that require human annotation [Cuevas et al., 2015]. Despite the plethora of existing algorithms and reference-based evaluation measures, little attention has been directed towards the reference-free or stand-alone evaluation of foreground masks. Such evaluation is a complex task that requires the estimation of the performance of segmented foreground masks without using any ground-truth data nor human intervention. However, a stand-alone evaluation enables performance estimation during run-time, thus entailing the possibility of improving the foreground mask. Furthermore, computing performance without ground-truth means no need of hours of manual annotation, thus making possible to evaluate algorithms with non-annotated data.

1.2 Objectives

The main objective of this thesis is to explore ways of improving background subtraction using information independent of a particular algorithm. We propose to approach the performance improvement goal by analyzing properties from the elements that are common to all algorithms, i.e. the input and the output. For achieving this objective, we propose to study the following areas:

- Background estimation: A background image can be a useful tool to improve the segmentation performance by correcting the errors of the background maintenance stage through a re-initialization of the background model. Therefore, we analyze the capabilities of background estimation algorithms to compute a good background image.

- Stand-alone performance evaluation for background subtraction: Knowing the performance of a segmentation output during run-time provides a valuable information to be able

to improve it online either by adapting an algorithm configuration or by post-processing the segmentation output. Therefore, we investigate how this performance can be estimated through properties extracted from the foreground masks computed by background subtraction algorithms.

- Improvement schemes for background subtraction: We study how to improve background subtraction by introducing information independent of particular algorithms in order to be able to generalize over all algorithms.

- Practical application: We investigate further utilities of background estimation and stand-alone evaluation to perform additional tasks, such as stationary object detection and algorithm combination.

1.3 Major contributions

The main contributions of this thesis are summarized below:

1. We propose a block-level approach to estimate the background image of video sequences with moving and stationary objects using temporal and spatial analysis to identify non-moving background candidates and select those ones that best fit in terms of spatial continuity using a Rejection based Multipath Reconstruction scheme.

2. We introduce a taxonomy of performance evaluation of foreground masks that extends the empirical-analytical taxonomy [SanMiguel and Martinez, 2010][Vojodi et al., 2013][Shi et al., 2015] of the state-of-the-art by considering stand-alone evaluation measures. Furthermore, we survey these stand-alone measures by considering properties used in different research areas where segmentation is performed.

3. We analyze 21 evaluation measures to understand which are the good properties to achieve stand-alone evaluation in terms of correlation to ground-truth based evaluation. In particular, we demonstrate that the fitness between the foreground mask and image regions or superpixels (fitness-to-regions property) is a good indicator.

4. We propose a framework for the improvement of foreground segmentation masks using a fitness-to-regions based information. In particular, we use a hierarchical approach that combines the fitness between the foreground mask and image segmentation partitions obtained at different degrees of detail that prevent foreground-background merging due to motion constraints.

5. We investigate the utility of a background image for the task of stationary object detection by exploiting spatio-temporal changes in background images over time.

6. We investigate the utility of fitness-to-regions as base information for algorithm combination.

The first contribution corresponds to the journal paper [Ortego et al., 2016a] and the conference paper [Ortego et al., 2016b], while the second and third contributions are included in a journal paper [Ortego et al., 2017]. Furthermore, the fourth and sixth contributions are compiled in a journal paper under review. Moreover, the journal paper [Ortego et al., 2015] comprises the fifth contribution.

1.4 Structure of the document

This document is structured in five parts, which are organized as follows:

- Part I: Introduction
  - *Chapter 1: Introduction.* This chapter presents the motivation, the objectives, the main contributions and the structure of this thesis.
- Part II: Background estimation
  - *Chapter 2: Background estimation in videos with stationary objects.* It describes the temporal-spatial strategy proposed to reconstruct an object-free background in presence of moving and stationary objects.
  - *Chapter 3: Background updating for stationary object detection.* It proposes a stationary object detector for long-term video analysis based on spatio-temporal changes in the most stable scene representations or background.
- Part III: Foreground segmentation
  - *Chapter 4: Foreground segmentation quality.* It studies several measures computed over connected components of foreground masks to identify the properties of high-performance foreground segmentation masks.
  - *Chapter 5: Foreground segmentation improvement.* It proposes a fitness-to-regions hierarchical post-processing framework to improve foreground segmentation masks.
- Part IV: Conclusions
  - *Chapter 6: Achievements, conclusions and future work.* It concludes this document summarizing the main results and future work for its extension.
- Part V: Appendixes

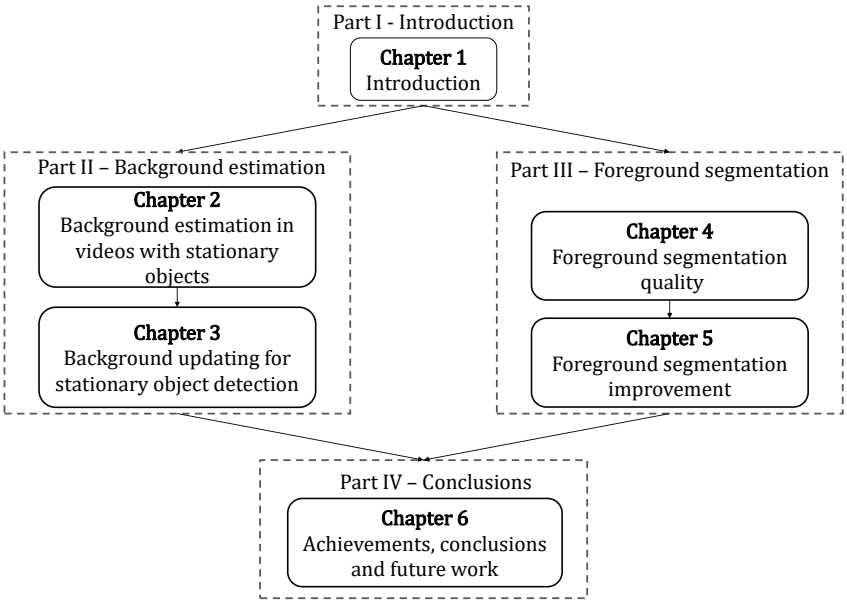


Figure 1.5: Dependence among the chapters of this thesis.

– *Appendix A: Publications.*

The relationships between chapters and parts of the thesis are depicted in Figure 1.5.

Part II

Background estimation

Chapter 2

Background estimation in videos with stationary objects

2.1 Introduction[1]

Segregating relevant moving objects is widely used in several applications of image processing and computer vision. This task often requires to estimate a foreground-free image (or background) under several visual challenges such as in background subtraction algorithms [Bouwmans, 2014][Sobral and Vacavant, 2014]. Background estimation (BE) finds applications not only in moving object segregation from video sequences [Park and Byun, 2013] but also to represent redundancy in video compression [Paul, 2012], to repair deteriorated images for inpainting [Chen et al., 2010], to implement video-based privacy protection [Nakashima et al., 2011] and to obtain object-free images for computational photography [Granados et al., 2008].

Several state-of-the-art BE approaches easily capture the background by assuming the availability of a set of frames without foreground objects (*training frames*) [Bouwmans, 2014]. This assumption may not be correct in many video-surveillance scenarios (e.g. shopping malls, airports or train stations) where many foreground objects may exist due to crowds and stationary objects, making very challenging the capture of the background. In general, BE faces two problems related with spatio-temporal scene variations: background visibility and photometric factors. The former occurs when pixels or regions of the background are seen for short periods of time in the training frames (e.g. due to stationary objects or to high-density of moving foreground), thus the predominant temporal data is not the background. The latter affects BE performance by modifying the background (illumination changes) or by affecting to the employed features (shadows and camouflages). The presence of stationary objects is a major limitation in current approaches as background visibility is highly decreased in the training frames.

[1]This chapter is an adapted version of the publications [Ortego et al., 2016a][Ortego et al., 2016b]

To overcome the above-mentioned limitations, we propose a block-level BE approach based on a temporal-spatial strategy that reconstructs an object-free background in presence of moving and stationary objects. For each spatial location, a temporal analysis module obtains a number of background candidates (blocks) via motion filtering, dimensionality reduction and threshold-free hierarchical clustering. Then, the spatial analysis module selects the most suitable candidate for each spatial location according to available candidates in neighboring locations. Firstly, the spatial strategy partially approximates the background by setting a number of initial locations (seeds) based on the motion activity along the training frames. Secondly, an iterative process estimates the remaining background based on inter-block and intra-block smoothness constraints. The experimental work validates the utility of the proposed approach, outperforming selected approaches in various datasets especially when dealing with stationary objects.

The contribution of the proposed approach is fourfold. Firstly, we propose a threshold-free clustering technique to determine background candidates without requiring parameter tuning to achieve optimal performance [Reddy et al., 2011][Hsiao and Leou, 2013]. Secondly, we obtain an initial background estimation (seeds selection) containing more data than state-of-the-art approaches [Reddy et al., 2009][Baltieri et al., 2010][Reddy et al., 2011] without introducing additional errors. Thus, fewer spatial locations need to be reconstructed, making the proposed approach less prone to estimation errors as compared to related approaches. Thirdly, the iterative reconstruction estimates different hypotheses of the neighboring background at each location and selects one of them, unlike approaches based on single-hypothesis estimations which may have low-accuracy [Reddy et al., 2009][Baltieri et al., 2010][Colombari and Fusiello, 2010][Reddy et al., 2011]. Fourthly, a new performance measure is proposed to avoid the use of a unique threshold [Reddy et al., 2009][Baltieri et al., 2010][Reddy et al., 2011].

The chapter is organized as follows: Section 2.2 discusses the related work and Section 2.3 overviews the proposed approach. Sections 2.4 and 2.5 describe the temporal and spatial analysis, respectively. Section 2.6 shows the experimental work. Finally, Section 2.7 presents some conclusions.

2.2 Related work

Different terms are used for BE [Reddy et al., 2011][Balcilar and Sonmez, 2015]: bootstrapping [Maddalena and Petrosino, 2012][Hsiao and Leou, 2013], background initialization [Colombari and Fusiello, 2010][Park and Byun, 2013], background generation [Colque and Camara-Chavez, 2011][Zhang et al., 2012] or background reconstruction [Crivelli et al., 2011]. Moreover, BE literature can be categorized as [Maddalena and Petrosino, 2014b]: temporal statistics, sub-intervals of stable intensity, iterative model completion and optimal labeling. In this section, we instead review related approaches focusing on the applied strategy: temporal and spatial. These

strategies may use data in a batch or an online fashion, operating at pixel or region (block) level.

Approaches using *temporal* strategies are common in background subtraction [Maddalena and Petrosino, 2014b], where the first frame is taken as the background image, which is updated by the successive frames [Maddalena and Petrosino, 2012][Chen and Ellis, 2014][St-Charles et al., 2015]. Beyond these techniques, Robust Principal Component Analysis (RPCA) [Bouwmans and Zahzah, 2014] models the background image of a video sequence by low-rank subspace analysis while the foreground is represented by the correlated sparse outliers. However, RPCA methods lose the temporal and spatial structure when representing each frame as a column vector, thus limiting the initialization capabilities. EigenBackground (EB) methods compute a basis of eigenvectors from the training frames to model the background at image [Oliver et al., 2000] or block [Hu et al., 2009] level. EB methods require a temporal consistency of the background for successful performance where short-term background occlusions are assumed [Tian et al., 2013]. RPCA and EB methods do not consider multiple basis to account for the range of appearances exhibited in the training frames and the relations between the basis of adjacent spatial locations, thus decreasing their performance in presence of slow-motion or stationary foreground. The temporal median at pixel level is widely used [Eng et al., 2003][Maddalena and Petrosino, 2014a], but stationary objects for more than 50% of the training frames are included in the background. Motion information can be used to remove foreground objects from the background model such as optical flow [Gutchess et al., 2001][Chia-Chih and Aggarwal, 2008][Lin et al., 2009] or inter-frame differences [Lin et al., 2009][Zhang et al., 2012][Hsiao and Leou, 2013]. Temporal continuous stability of pixel intensity is also employed to obtain hypotheses for the background model in each spatial location [Gutchess et al., 2001][Wang and Suter, 2006][Chia-Chih and Aggarwal, 2008] where non-continuous intervals are wrongly assumed as different background representations. Therefore, clustering of non-continuous intervals is preferred to address such assumption [Reddy et al., 2009][Baltieri et al., 2010][Colombari and Fusiello, 2010][Reddy et al., 2011][Benalia and Ait-Aoudia, 2012]. Furthermore, temporal variability of pixel values is used to keep occluded background values and to avoid wrong model updates with foreground data [Park and Byun, 2013].

Although some approaches only use *temporal* analysis [Wang and Suter, 2006][Maddalena and Petrosino, 2014a], a *spatial* analysis is needed in presence of moving and stationary objects since background may no longer be the dominant temporal information in the training frames. Smoothness constraints may be imposed in the background to decide whether new pixels or blocks belong to the background employing features such as color [Zhang et al., 2012]. In [Reddy et al., 2009] and [Reddy et al., 2011], the Discrete Cosine Transform (DCT) is embedded in a Markov Random Field (MRF) framework to introduce smoothness in neighbors while iterative background estimations correct possible errors [Reddy et al., 2011]. Alternatively, DCT can be replaced by the Hadamard transform to decrease computational complexity,

which is combined with iterative corrections based on gradient features between candidates and their neighbors [Baltieri et al., 2010]. Smoothness can also be cast as finding the best partially-overlapping block between candidates and the already set background locations [Colombari and Fusiello, 2010]. Moreover, block-level color and gradient constraints with the neighborhood can be applied to estimate the background [Shrotre and Karam, 2013]. Furthermore, other approaches encode spatial smoothness and temporal information in energy minimization frameworks such as Loopy Belief Propagation [Xun and Huang, 2008][Guo et al., 2012], Graph Cuts [Chen et al., 2010], Conditional Mixed-State MRFs [Crivelli et al., 2011] or dynamic MRFs [Park and Byun, 2013]. Recently, [Chacon-Murguia et al., 2014] introduces spatial constraints through image segmentation. Additionally, spatial information also considers optical flow in the neighborhood [Gutchess et al., 2001], correcting its density by handling objects moving at different depths [Chia-Chih and Aggarwal, 2008].

In summary, several BE strategies have been proposed where recent approaches use temporal information and apply smoothness constraints over the estimated background. The main limitation of current approaches involves situations of low background visibility where existing smoothness schemes do not successfully deal with stationary objects.

2.3 Proposed approach: overview

The proposed approach performs a temporal-spatial analysis at block level (see Figure 2.1) over a set of T training frames $\mathcal{I}_t$, $\mathbb{F} = \{\mathcal{I}_1 ... \mathcal{I}_T\}$, to extract the reconstructed background image $\mathcal{B}$ free of moving and stationary objects. Firstly, the *Splitting* module divides each $\mathcal{I}_t$ into non-overlapping blocks $R_t^{\mathbf{s}}$ of size $W \times W$, where $\mathbf{s}$ is the bi-dimensional index for the spatial location of each block. Secondly, the *Temporal Analysis* module creates a number of background candidates $C_l^{\mathbf{s}}$ for each spatial location $\mathbf{s}$, where $l \in \{1 \dots N^{\mathbf{s}}\}$ and $N^{\mathbf{s}} \leq T$ is the number of candidates. The *Temporal Analysis* consists of the *Motion Filtering* stage to discard $R_t^{\mathbf{s}}$ blocks where moving objects exist and the *Dimensionality Reduction* stage to decrease the amount of data analyzed by the *Clustering* stage which obtains a set of background candidates. Finally, the *Spatial Analysis* module reconstructs the background of each spatial location $\mathbf{s}$, partially estimated in the *Seed Selection* stage, by the *Multipath Reconstruction* stage to iteratively fill each spatial location $\mathbf{s}$ with the optimal candidate $C_*^{\mathbf{s}}$ using inter-block and intra-block smoothness constraints. The temporal and spatial analysis modules are described in Section 2.4 and Section 2.5, respectively. The key symbols we use in this chapter are given in Table 2.1.

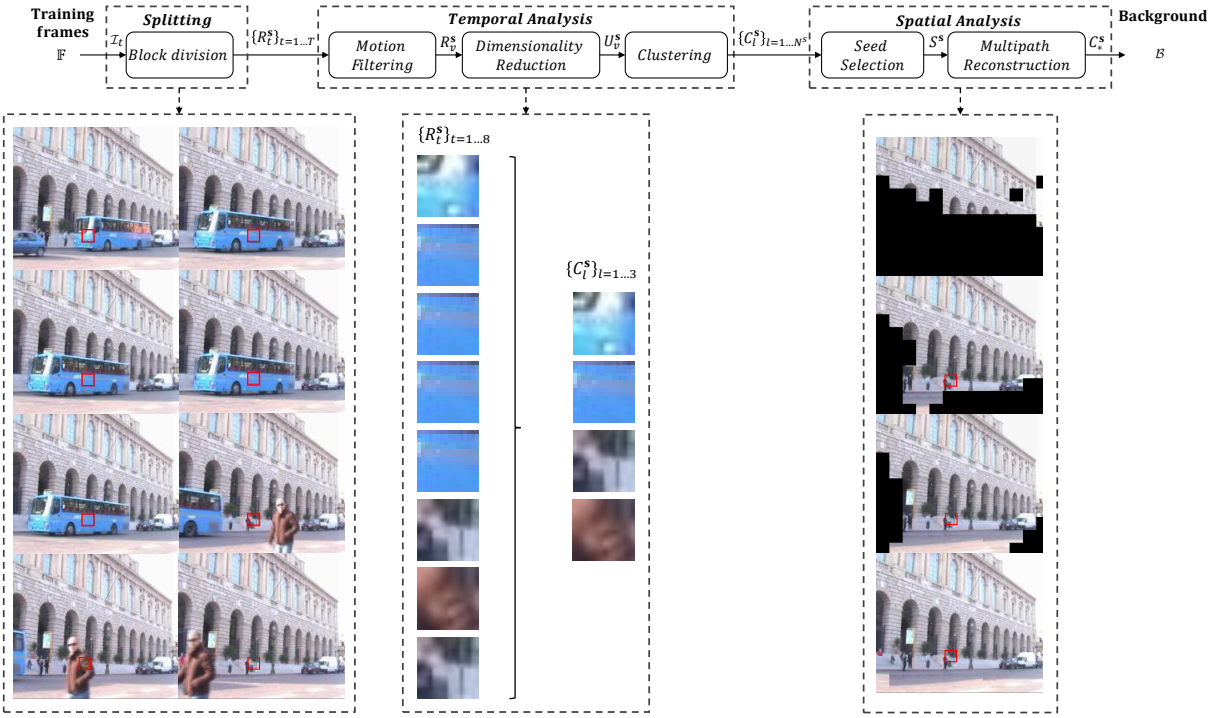


Figure 2.1: Overview of the proposed multipath approach for temporal-spatial block-level background estimation. Below each module, visual examples are provided for a selected spatial location $\mathbf{s}$ (marked in red). Firstly, the *Splitting* module divides into blocks the training frames $\mathbb{F}$ (the selected block is shown for the training frames: 28, 109, 190, 191, 192, 354, 371 and 386 of the sequence *guardia*). Secondly, the *Temporal Analysis* groups all blocks $R_t^{\mathbf{s}}$ extracted from $\mathbb{F}$, thus obtaining background candidates $C_l^{\mathbf{s}}$ via clustering as seen in the visual example (left: $R_t^{\mathbf{s}}$ blocks from previous example, right: $C_l^{\mathbf{s}}$ clusters computed at $\mathbf{s}$). Finally, the *Spatial Analysis* reconstructs the background, starting from some selected seeds $S^{\mathbf{s}}$ and iteratively filling all spatial location $\mathbf{s}$ until the whole background is obtained as illustrated in the visual example (from top to bottom: initial selected seeds, two iterations of the multipath reconstruction and the final reconstructed background, where the red rectangle corresponds to the selected candidate $C_*^{\mathbf{s}}$).

2.4 Temporal Analysis

The *Temporal Analysis* module generates the background candidates of each spatial location $\mathbf{s}$. It contains three stages (Figure 2.1): *Motion Filtering*, *Dimensionality Reduction* and *Clustering*.

2.4.1 Motion filtering

The *Motion filtering* stage discards $R_t^{\mathbf{s}}$ blocks corresponding to moving objects that cannot be candidates for the reconstructed background $\mathcal{B}$. For all training frames, we compute the motion activity at block level $\lambda_t^{\mathbf{s}}$:

$$\lambda_t^{\mathbf{s}} = \begin{cases} 1 & if \quad \exists \mathbf{p} \in \mathbf{s} : \left| \mathcal{I}_t^{\mathbf{p}} - \mathcal{I}_{t-k}^{\mathbf{p}} \right| > \eta \\ 0 & otherwise \end{cases} , \tag{2.1}$$

where $\mathbf{p}$ is the bi-dimensional index for pixel locations in $\mathbf{s}$ and the threshold η is computed automatically [Kapur et al., 1985] to detect intensity changes between $k-$separated frame differences due to moving objects (k should be small). $\lambda_t^{\mathbf{s}}$ takes the value 1(0) when motion (no motion) is detected, thus rejecting (keeping) the associated block R_t^s. Note that Eq. 2.1 implies the visualization of the background for k consecutive frames, as often assumed in existing literature [Chia-Chih and Aggarwal, 2008][Reddy et al., 2011]. Finally, the selected data to compose the background at each location $\mathbf{s}$ is represented by $\mathbb{Y}^{\mathbf{s}} = \{R_v^{\mathbf{s}}\}_{v=1\ldots M^{\mathbf{s}}}$, where $M^{\mathbf{s}}$ is the number of blocks without motion and $M^{\mathbf{s}} \leq T$, $\forall \mathbf{s}$.

2.4.2 Dimensionality Reduction

To further reduce the data to process, we apply Principal Component Analysis (PCA) [Jolliffe, 2005] to $\mathbb{Y}^{\mathbf{s}}$ as the useful data to generate background candidates is driven by the block variance. Pixel locations with variations over time are relevant to group blocks whereas pixel locations without variability are redundant. PCA determines a transformation basis to project data where pixels with low variance over time are removed. PCA is applied to all blocks in $\mathbb{Y}^{\mathbf{s}}$, where each block is previously rasterized into a column vector of size $3W^2$ by concatenating its RGB channels. Finally, we obtain a matrix $\mathbb{Z}^{\mathbf{s}} = \{U_v^{\mathbf{s}}\}_{v=1\ldots M^{\mathbf{s}}}$, where $|U_v^{\mathbf{s}}| \leq |R_v^{\mathbf{s}}|$ and $|\cdot|$ denotes the cardinality, i.e. the number of elements, representing the data in the PCA domain which is used exclusively for the clustering stage (Subsection 2.4.3). Note that the *Spatial Analysis* module (Section 2.5) uses the $W \times W$ blocks $R_t^{\mathbf{s}}$ to estimate the background image $\mathcal{B}$ instead of the PCA-reduced data $U_v^{\mathbf{s}}$.

2.4.3 Clustering

This stage generates a number of candidates $C_l^{\mathbf{s}}$ to be the background $B^{\mathbf{s}}$ for each location $\mathbf{s}$. Instead of using the raw data, we group the PCA-reduced data $\mathbb{Z}^{\mathbf{s}}$ into clusters $K_l^{\mathbf{s}}$ which are structured as partitions $\mathbb{P}_{N^{\mathbf{s}}}^{\mathbf{s}} = \{K_1^{\mathbf{s}} \ldots K_{N^{\mathbf{s}}}^{\mathbf{s}}\}$ where $N^{\mathbf{s}}$ is the total number of clusters. As the optimum $N^{\mathbf{s}}$ is not known for each $\mathbf{s}$, hypotheses for the partitions are created for different values of $N^{\mathbf{s}}$. The optimal partition is found by validation indexes that maximize inter-cluster differences and intra-cluster similarities. The proposed approach provides a threshold-free clustering that leads to sub-optimal solutions containing the desired candidates. The candidates $C_l^{\mathbf{s}}$ represent each cluster $K_l^{\mathbf{s}}$ where the best candidate $C_*^{\mathbf{s}}$ is selected in the *Spatial Analysis* module (Section 2.5).

For generating the clusters, we employ agglomerative hierarchical clustering (AHC) [Jain

Symbol	Notation
t	Temporal index.
$\mathbf{p}$	Bi-dimensional index for pixel locations.
$\mathbf{s}$	Bi-dimensional index for block locations.
$\mathbb{F}$	Set of T training frames to reconstruct the background image.
$\mathcal{I}_t$	Training frame at time t.
$\mathcal{B}$	Reconstructed background image using $\mathbb{F}$.
$R_t^{\mathbf{s}}$	$W \times W$ block of $\mathcal{I}_t$ at time t and location $\mathbf{s}$.
$\lambda_t^{\mathbf{s}}$	Score for block-level activity at location $\mathbf{s}$.
$\mathbb{Y}^{\mathbf{s}}$	Set containing $M^{\mathbf{s}}$ motion-filtered blocks $R_t^{\mathbf{s}}$.
$U_v^{\mathbf{s}}$	PCA-reduced block v at location $\mathbf{s}$, where $v \in [1, M^{\mathbf{s}}]$.
$\mathbb{Z}^{\mathbf{s}}$	Set containing $M^{\mathbf{s}}$ PCA-reduced blocks $U_v^{\mathbf{s}}$.
$N^{\mathbf{s}}$	Number of clusters at location $\mathbf{s}$.
l	Index to denote a cluster at location $\mathbf{s}$, where $l \in [1, N^{\mathbf{s}}]$.
$K_l^{\mathbf{s}}$	Cluster l at location $\mathbf{s}$ that groups $U_t^{\mathbf{s}}$ (i.e. $R_t^{\mathbf{s}}$).
$\mathbb{P}_b^{\mathbf{s}}$	Cluster partition at location $\mathbf{s}$ with b clusters.
$\theta_{SI}(\mathbb{P}_b^{\mathbf{s}})$	Score for cluster partition $\mathbb{P}_b^{\mathbf{s}}$ (Silhouette).
$\theta_{DB}(\mathbb{P}_b^{\mathbf{s}})$	Score for cluster partition $\mathbb{P}_b^{\mathbf{s}}$ (Davies-Bouldin).
$\mathbb{P}_*^{\mathbf{s}}$	Optimal partition at location $\mathbf{s}$. It contains $N^{\mathbf{s}}$ clusters.
$C_l^{\mathbf{s}}$	Candidate to be background (i.e. represents the cluster K_l^s).
$S^{\mathbf{s}}$	Seed block at location $\mathbf{s}$ for the seed image $\mathcal{S}$.
$\xi^{\mathbf{s}}$	Activity score to compute seeds at location $\mathbf{s}$.
$\tilde{\mathcal{B}}$	Iteratively reconstructed background image. $\tilde{\mathcal{B}}$ is initialized with $\mathcal{S}$ and contains blocks $\tilde{B}^{\mathbf{s}}$.
$\mathbb{V}_8^{\mathbf{s}}$	8-connected block neighborhood at location $\mathbf{s}$.
$\mathbb{V}_4^{\mathbf{s}}$	4-connected block neighborhood at location $\mathbf{s}$.
$\Phi\left(C_l^{\mathbf{s}'}\right)$	Inter-block color discontinuity for candidate $C_l^{\mathbf{s}'}$.
$\Psi\left(C_l^{\mathbf{s}'}\right)$	Intra-block heterogeneity for candidate $C_l^{\mathbf{s}'}$.
$\Omega\left(C_l^{\mathbf{s}'}\right)$	Inter-block color dissimilarity for candidate $C_l^{\mathbf{s}'}$.
$\tilde{C}_{\Phi}^{\mathbf{s}',m}$	Temporary candidate selected using Φ, at location $\mathbf{s}'$ for path m.
$\tilde{C}_{\Psi}^{\mathbf{s}',m}$	Temporary candidate selected using Ψ, at location $\mathbf{s}'$ for path m.
$\tilde{C}_{\Omega}^{\mathbf{s}',m}$	Temporary candidate selected using Ω, at location $\mathbf{s}'$ for path m
$\tilde{C}^{\mathbf{s}',m}$	Temporary candidate selected at location $\mathbf{s}'$ for path m. Is selected among $\tilde{C}_{\Phi}^{\mathbf{s}',m}$, $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$.
$C_*^{\mathbf{s}'}$	Selected candidate at location $\mathbf{s}'$.
$\mathcal{G}$	Ground-truth background image that contains blocks $G^{\mathbf{s}}$.
$B_{best}^{\mathbf{s}}$	Best background block selecting in the location $\mathbf{s}$ the block $\tilde{B}^{\mathbf{s}}$ with lowest distance to the ground-truth block $G^{\mathbf{s}}$.

Table 2.1: Key symbols and notations

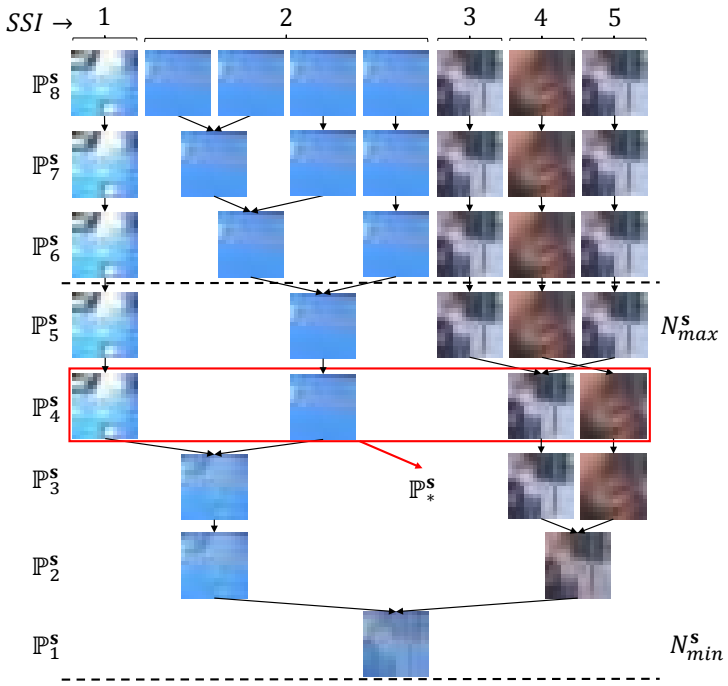


Figure 2.2: Example of a dendrogram to detect the optimal clustering partition $\mathbb{P}_*^{\mathbf{s}}$ for a 8-block set. Only partitions between $N_{min}^{\mathbf{s}}$ and $N_{max}^{\mathbf{s}}$ are considered (dashed lines). $\mathbb{P}_4^{\mathbf{s}}$ is selected as optimal partition as it has the highest $\theta_{SI}(\mathbb{P}_b^{\mathbf{s}}) + \theta_{DB}(\mathbb{P}_b^{\mathbf{s}})$, thus $N^{\mathbf{s}} = 4$. Albeit clustering uses PCA-reduced blocks $U_v^{\mathbf{s}}$, we show the associated blocks $R_t^{\mathbf{s}}$ for visualization purposes.

et al., 1999] over matrices $\mathbb{Z}^{\mathbf{s}}$ where the distance between two clusters is defined as the highest Euclidean distance among members $U_v^{\mathbf{s}}$ of both clusters. The AHC cluster structure can be represented as dendrograms, i.e. tree-like diagrams depicting partition hypotheses at different cluster distances. Thus, we limit the number of clustering hypotheses between a minimum and maximum value ($N_{min}^{\mathbf{s}}$ and $N_{max}^{\mathbf{s}}$, respectively). $N_{min}^{\mathbf{s}}$ is set to 1 (i.e. one cluster) which corresponds to an always-visible background. For each location $\mathbf{s}$, $N_{max}^{\mathbf{s}}$ is set to the number of identified Sub-intervals of Stable Intensity (SSI) [Gutchess et al., 2001][Chia-Chih and Aggarwal, 2008], as SSIs may be caused by objects or background. SSIs are continuous temporal intervals without intensity variations, computed at block level using motion information from Eq. 2.1. Finally, partition hypotheses $\{\mathbb{P}_b^{\mathbf{s}}\}_{b=N_{min}^{\mathbf{s}},\dots,N_{max}^{\mathbf{s}}}$ are generated where b is the number of clusters in the partition. Figure 2.2 shows a dendrogram for clustering eight blocks and an example of SSIs on top of Figure 2.2, where $N_{max}^{\mathbf{s}} = 5$.

Subsequently, clustering validation determines the best partition $\mathbb{P}_*^{\mathbf{s}}$ containing the optimal number of clusters $N^{\mathbf{s}}$. This validation employs the Silhouette θ_{SI} and Davies-Bouldin θ_{DB} indexes [Wang et al., 2009]. θ_{SI} measures the compactness and separation among clusters; a higher average value of this measure implies a better quality of the cluster. θ_{DB} measures the

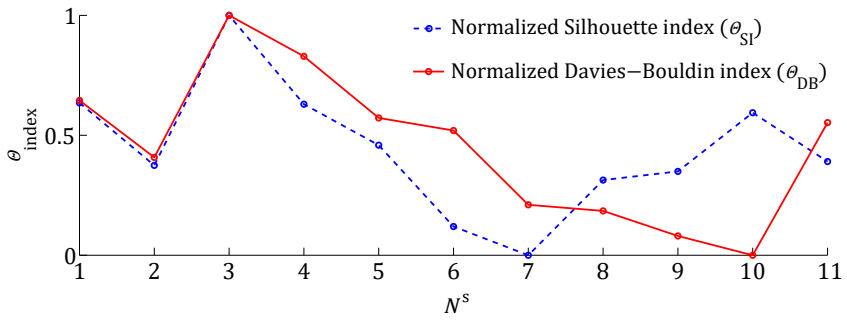


Figure 2.3: Example of normalized scores for clustering validation. Scores $\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b)$ and $\theta_{DB}(\mathbb{P}^{\mathbf{s}}_b)$ are shown for each partition ranging from $N^{\mathbf{s}}_{min} = 1$ to $N^{\mathbf{s}}_{max} = 11$ clusters. The optimum is $N^{\mathbf{s}} = 3$ as partition $\mathbb{P}^{\mathbf{s}}_3$ obtains the highest $\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b) + \theta_{DB}(\mathbb{P}^{\mathbf{s}}_b)$ score.

similarity between each cluster and its highest similar one; small values in this index correspond to compact clusters whose centroid is far from the others. After computing both indexes for each hypothesized partition $\mathbb{P}^{\mathbf{s}}_b$, we normalize them by considering that maximum θ_{SI} and minimum θ_{DB} are preferred:

$$\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b) = \frac{SI(\mathbb{P}^{\mathbf{s}}_b) - min(\mathbb{L})}{max(\mathbb{L}) - min(\mathbb{L})}, \tag{2.2}$$

$$\theta_{DB}(\mathbb{P}^{\mathbf{s}}_b) = \frac{DB(\mathbb{P}^{\mathbf{s}}_b) - max(\mathbb{M})}{max(\mathbb{M}) - min(\mathbb{M})}, \tag{2.3}$$

where the sets $\mathbb{L} = \{\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b)\}_{b=N^{\mathbf{s}}_{min},\dots,N^{\mathbf{s}}_{max}}$ and $\mathbb{M} = \{\theta_{DB}(\mathbb{P}^{\mathbf{s}}_b)\}_{b=N^{\mathbf{s}}_{min},\dots,N^{\mathbf{s}}_{max}}$ are all θ_{SI} and θ_{DB} scores, respectively. Then, both scores are combined for each partition $\mathbb{P}^{\mathbf{s}}_b$ to determine the optimal $\mathbb{P}^{\mathbf{s}}_*$:

$$\mathbb{P}^{\mathbf{s}}_* = \underset{b=N^{\mathbf{s}}_{min},\dots,N^{\mathbf{s}}_{max}}{argmax} \left(\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b) + \theta_{DB}(\mathbb{P}^{\mathbf{s}}_b)\right). \tag{2.4}$$

Figure 2.3 presents an example of clustering validation with 11 partitions, where the optimal one contains 3 clusters with the highest $\theta_{SI}(\mathbb{P}^{\mathbf{s}}_b) + \theta_{DB}(\mathbb{P}^{\mathbf{s}}_b)$ value.

Finally, we compute each background candidate $C^{\mathbf{s}}_l$ as the average of members in the cluster $K^{\mathbf{s}}_l$, using the $W \times W$ blocks $R^{\mathbf{s}}_t$ instead of the PCA-reduced data $U^{\mathbf{s}}_v$, similarly to the widely used K-means clustering [Hartigan, 1975], which also reduces noise in the final candidate.

2.5 Spatial Analysis

This module obtains each background block $B^{\mathbf{s}}$ by selecting the best candidate $C^{\mathbf{s}}_*$ among the set of background candidates $C^{\mathbf{s}}_l$. For each location, a multipath reconstruction of the background is proposed to enforce background smoothness among selected candidates in neighboring locations. The reconstruction process is divided in two stages (see Figure 2.1): *Seed Selection* and

Multipath Reconstruction. For the latter, the explanation is divided into *Sequential Multipath Reconstruction* (Subsection 2.5.2) and *Rejection based Multipath Reconstruction* (Subsection 2.5.3) for readability.

2.5.1 Seed Selection

An initial partial background estimation is provided for selected locations by seed blocks $S^{\mathbf{s}}$ defined as highly-reliable background candidates. Existing approaches often establish this candidate-seed correspondence for the $\mathbf{s}$ locations with one cluster and, therefore, a unique candidate $C_l^{\mathbf{s}}$ for $B^{\mathbf{s}}$ is selected in such locations [Reddy et al., 2009][Baltieri et al., 2010][Reddy et al., 2011]. When these single-candidate clusters do not exist, a *major cluster* $\hat{C}_{l'}^{\mathbf{s}}$ at each spatial location $\mathbf{s}$ can be identified as the cluster with maximum size:

$$\hat{C}_{l'}^{\mathbf{s}} = C_{l'}^{\mathbf{s}} : |K_{l'}^{\mathbf{s}}| > |K_l^{\mathbf{s}}|, \forall l = 1, \ldots, N^{\mathbf{s}}, \tag{2.5}$$

where major clusters are selected as seeds when their cardinality is equal to the maximum one for all locations $\max_{\mathbf{s}} \{|K_{l'}^{\mathbf{s}}|\}$. However, Eq. (2.5) initializes few blocks where stationary objects may be temporally dominant and be wrongly selected as seeds. Errors in this initial background estimation are critical since they are propagated in the subsequent stages.

We address such limitation by proposing a unified analysis of stationarity and motion activity along training frames. We detect locations $\mathbf{s}$ with low motion or without stationary objects over time as suitable locations to initialize with seeds. For such detection, we assume that stationary objects occluding the background in $\mathcal{I}_1$ are not going to remain in the same location in $\mathcal{I}_T$. This assumption is reasonable, as objects not moving for all training frames can be considered as background. Hence, an activity score at block level $\xi^{\mathbf{s}}$ is computed as:

$$\xi^{\mathbf{s}} = max\left\{ f\left(\mathcal{I}_1^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_1^{\mathbf{p}}\}\right) + f\left(\mathcal{I}_T^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_T^{\mathbf{p}}\}\right) \right\}_{\forall\, \mathbf{p} \in \mathbf{s}}, \tag{2.6}$$

where $\mathbf{p}$ is a pixel location; $\mathbb{F}^{\mathbf{p}}$, $\mathcal{I}_1^{\mathbf{p}}$ and $\mathcal{I}_T^{\mathbf{p}}$ are the gray-level pixel values at location $\mathbf{p}$ of the training sequence, initial and final frame, respectively; $\mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_1^{\mathbf{p}}\}$ and $\mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_T^{\mathbf{p}}\}$ are the set of training frames except the initial and final ones, respectively. The function $f(\cdot, \cdot)$ computes the average value for the absolute pixel-level difference:

$$f\left(\mathcal{I}_t^{\mathbf{p}}, \mathbb{I}^{\mathbf{p}}\right) = \frac{1}{|\mathbb{I}^{\mathbf{p}}|} \sum_{q=1}^{|\mathbb{I}^{\mathbf{p}}|} \begin{cases} 1\, if & \left|\mathcal{I}_t^{\mathbf{p}} - \mathcal{I}_q^{\mathbf{p}}\right| > \tau \\ 0 & otherwise \end{cases}, \tag{2.7}$$

where $\mathbb{I}^{\mathbf{p}} = \left\{\mathcal{I}_q^{\mathbf{p}}\right\}_{q=1:|\mathbb{I}^{\mathbf{p}}|}$ is a generic set of pixels at location $\mathbf{p}$ and τ is a detection threshold computed automatically [Kapur et al., 1985]. The forward activity score $f\left(\mathcal{I}_1^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_1^{\mathbf{p}}\}\right)$ compares the pixels of the first frame against the other frames. Similarly, the backward activity score $f\left(\mathcal{I}_T^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_T^{\mathbf{p}}\}\right)$ compares the pixels of the last frame against the other frames. Finally,

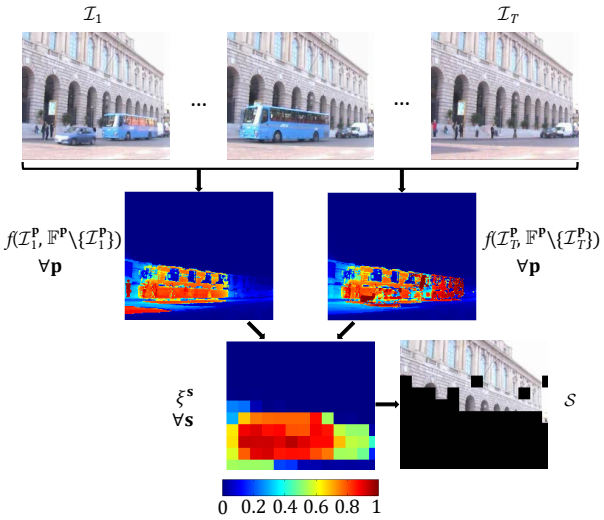


Figure 2.4: Seed Selection example. From top (set of frames) to bottom ($S^{\mathbf{s}}$) the *Seed Selection* process is presented. Key. $f\left(\mathcal{I}_1^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_1^{\mathbf{p}}\}\right)$: forward activity score (pixel level). $f\left(\mathcal{I}_T^{\mathbf{p}}, \mathbb{F}^{\mathbf{p}} \setminus \{\mathcal{I}_T^{\mathbf{p}}\}\right)$: backward activity score (pixel level). $\xi^{\mathbf{s}}$: activity score (block level). $\mathcal{S}$: seeds image.

the initial background estimation with seeds $S^{\mathbf{s}}$ is obtained only in locations with minimum $\xi^{\mathbf{s}}$:

$$S^{\mathbf{s}} = \begin{cases} \hat{C}_l^{\mathbf{s}} & if\ \xi^{\mathbf{s}} = min\{\xi^{\mathbf{s}'}\}_{\forall \mathbf{s}' \in I} \\ \emptyset & otherwise \end{cases}, \tag{2.8}$$

where $\hat{C}_l^{\mathbf{s}}$ is the major cluster and the empty locations $\mathbf{s}$ will be filled by the *Multipath Reconstruction*. Figure 2.4 presents an example of the activity scores where locations with minimum $\xi^{\mathbf{s}}$ conform the seeds $S^{\mathbf{s}}$. The initial partial background $\tilde{\mathcal{B}}$ to be reconstructed is obtained using the seed image $\mathcal{S}$, i.e. $\tilde{B}^{\mathbf{s}} = S^{\mathbf{s}}$, where $\mathcal{S} = \{S^{\mathbf{s}}\}_{\forall s}$.

2.5.2 Sequential Multipath Reconstruction

This subsection describes the framework for Sequential Multipath Reconstruction (SMR) to iteratively reconstruct the background from the initial estimation (Eq. 2.8).

If we consider the location index $\mathbf{s}$ as a bi-dimensional vector (i.e. $B^{\mathbf{s}} \equiv B^{(i,j)}$), the 4-connected neighborhood $\mathbb{V}_4^{\mathbf{s}}$ is defined as:

$$\mathbb{V}_4^{\mathbf{s}} = \left\{B^{(i-1,j)}, B^{(i,j+1)}, B^{(i+1,j)}, B^{(i,j-1)}\right\}, \tag{2.9}$$

whereas the 8-connected neighborhood $\mathbb{V}_8^{\mathbf{s}}$ is defined as:

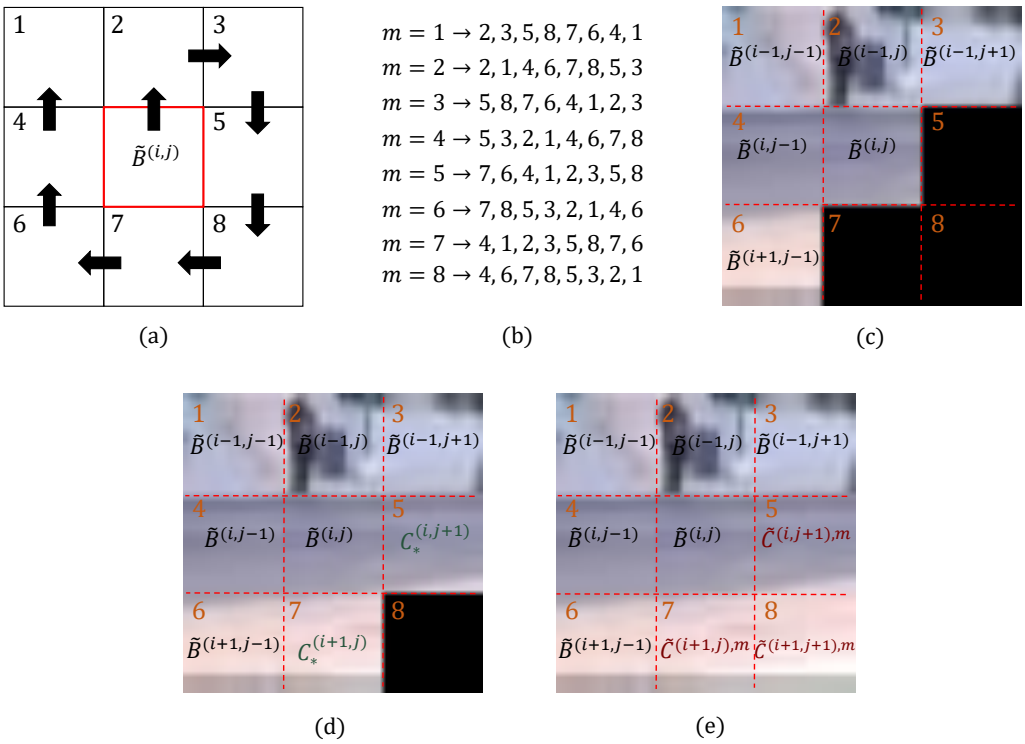
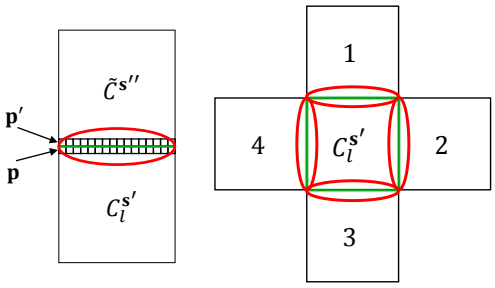


Figure 2.5: Multipath reconstruction scheme for each iteration of $\tilde{B}^{\mathbf{s}} \equiv \tilde{B}^{(i,j)}$. (a) First path ($m = 1$) to reconstruct $\mathbb{V}_8^{(i,j)}$. Black arrows describe the path direction. (b) Locations explored for the $m = 1 \dots 8$ paths, which assign a temporary block $\tilde{C}^{\mathbf{s}',m}$ for each empty location $\mathbf{s}' \in \mathbb{V}_8^{(i,j)}$. (c) Example of a seed $\tilde{B}^{(i,j)}$ and its $\mathbb{V}_8^{(i,j)}$. (d) Result of the reconstruction of $\mathbb{V}_8^{(i,j)}$ in (c) using the $m = 1$ path, thus temporary blocks $\tilde{C}^{\mathbf{s}',m}$ are selected (dark red in locations 5, 7 and 8). (e) Final reconstruction of $\mathbb{V}_4^{(i,j)}$ for $\tilde{B}^{(i,j)}$ where the reconstructed blocks $C_*^{\mathbf{s}'}$ (dark green in locations 5 and 7) are selected.

$$\mathbb{V}_8^{\mathbf{s}} = \{B^{(i-1,j)}, B^{(i-1,j+1)}, B^{(i,j+1)}, B^{(i+1,j+1)}, B^{(i+1,j)}, B^{(i+1,j-1)}, B^{(i,j-1)}, B^{(i-1,j-1)}\}. \tag{2.10}$$

SMR starts each iteration of the background reconstruction from a partial background $\tilde{\mathcal{B}}$ with empty locations. Then, SMR chooses a background block $\tilde{B}^{\mathbf{s}}$ with maximum number of non-empty neighbors in $\mathbb{V}_8^{\mathbf{s}}$, where empty locations are reconstructed by m paths or hypotheses. Each path starts from one side of $\tilde{B}^{\mathbf{s}}$ (top, bottom, left or right), employs a direction (clockwise or counterclockwise) and sequentially fills all empty locations $\mathbf{s}' \in \mathbb{V}_8^{\mathbf{s}}$ with candidates $C_l^{\mathbf{s}'}$. Multi-path reconstruction improves robustness against wrong candidate selections due to objects or other artifacts. Figure 2.5(a) shows the selected block $\tilde{B}^{\mathbf{s}}$ whose neighborhood $\mathbb{V}_8^{\mathbf{s}}$ is explored using 8 paths traversed as presented in Figure 2.5(b). Some blocks already exist in each path and therefore, they are not reconstructed. Figure 2.5(c) presents an example where some $\tilde{B}^{\mathbf{s}}$



(a) (b)

Figure 2.6: Example of reconstruction scheme using fitness function Φ. (a) Border between $C_l^{\mathbf{s}'}$ and one neighboring block where discontinuities are analyzed. (b) Color discontinuity scheme of $\Phi(C_l^{\mathbf{s}'})$, where $\mathbb{V}_4^{\mathbf{s}'}$, i.e. 1, 2, 3 and 4, are used to analyze discontinuities with $C_l^{\mathbf{s}'}$. Borders between blocks are marked in green and adjacent pixels of the border are circled in red.

neighbors exist.

For each m-path, we select suitable candidates to fill empty locations by employing a fitness function Φ based on the inter-block color discontinuity in the neighborhood $\mathbb{V}_4^{\mathbf{s}'}$ of the location to be filled:

$$\Phi\left(C_l^{\mathbf{s}'}\right) = \frac{1}{\left|\mathbb{V}_4^{\mathbf{s}'}\right|} \sum_{\mathbf{s}'' \in \mathbb{V}_4^{\mathbf{s}'}} \left(\frac{1}{W} \sum_{p,p' \in \mathbb{E}} \left| C_l^{\mathbf{s},'p'} - \tilde{C}^{\mathbf{s},''p'} \right| \right), \tag{2.11}$$

where the $\tilde{C}^{\mathbf{s}''}$ are the already set neighbors in $\mathbf{s}'' \in \mathbb{V}_4^{\mathbf{s}'}$ (temporary blocks selected during the path reconstruction or previously estimated $\tilde{B}^{\mathbf{s}'}$). $\mathbb{E}$ denotes the set of pixel locations pairs $\mathbf{p}$ and $\mathbf{p}'$ in the border between blocks $C_l^{\mathbf{s}'}$ and $\tilde{C}^{\mathbf{s}''}$, respectively. Therefore, Φ employs 1 to 4 borders depending on the non-empty locations in $\mathbb{V}_4^{\mathbf{s}'}$. Figure 2.6 shows the $\mathbb{V}_4^{\mathbf{s}'}$ reconstruction scheme using Φ for the location $\mathbf{s}' \in \mathbb{V}_4^{\mathbf{s}}$. Figure 2.6(a) presents the pixel locations considered to compare two adjacent blocks and Figure 2.6(b) illustrates the $\mathbb{V}_4^{\mathbf{s}'}$ neighborhood employed.

For each m-path, the candidate $\tilde{C}^{\mathbf{s}',m}$ is selected by minimizing Φ:

$$\tilde{C}^{\mathbf{s}',m} = \underset{\forall l \in \{1,..,N^{\mathbf{s}'}\}}{argmin} \; \Phi(C_l^{\mathbf{s}'}), \tag{2.12}$$

where $m \in \{1 \dots 8\}$ and $C_l^{\mathbf{s}'}\, \forall l$ are the available candidates. Figure 2.5(e) shows the reconstruction of $\mathbb{V}_4^{\mathbf{s}}$ starting from the initial estimation in Figure 2.5(c) where Figure 2.5(d) presents a temporary $\mathbb{V}_8^{\mathbf{s}}$ single-path reconstruction.

Finally, we obtain the best estimation for the $\mathbb{V}_4^{\mathbf{s}}$ neighborhood using the $m = 1 \dots 8$ paths. We select the best candidate $C_*^{\mathbf{s}'}$among the temporary blocks $\tilde{C}^{\mathbf{s}',m}$:

$$C_*^{\mathbf{s}'} = \underset{\forall m \in \{1,\dots,8\}}{argmin} \; \Phi(\tilde{C}^{\mathbf{s}',m}), \tag{2.13}$$

Algorithm 2.1 Sequential Multipath Reconstruction (SMR).

Input: $S^{\mathbf{s}}$ seeds and $C_l^{\mathbf{s}}$ candidates
Output: $\mathcal{B} = \{B^{\mathbf{s}}\}_{\forall s} : B^{\mathbf{s}} \neq \emptyset, \forall \mathbf{s}$.

while $(\exists\, \tilde{B}^{\mathbf{s}} = \emptyset)$
  Selection of $\mathbf{s} : \tilde{B}^{\mathbf{s}} \neq \emptyset$
  for $m = 1$ to 8 **do**
    for $\mathbf{s}' \in \mathbb{V}_8^{\mathbf{s}}$
      if $\tilde{B}^{\mathbf{s}'} = \emptyset$ **then**
        Select $\tilde{C}^{\mathbf{s}',m}$ with Eq. 2.12
      end
    end
  end
  for $\mathbf{s}' \in \mathbb{V}_4^{\mathbf{s}}$
    Select $C_*^{\mathbf{s}'}$ with Eq. 2.13
    $\tilde{B}^{\mathbf{s}'} = C_*^{\mathbf{s}'}$
  end
end
$\mathcal{B} = \tilde{\mathcal{B}}$

where $\Phi(\tilde{C}^{\mathbf{s}',m})$ is the Φ value obtained by the candidate during the m-path reconstruction (Eq. 2.11). As temporary blocks in $\mathbb{V}_4^{\mathbf{s}}$ (top-center, bottom-center, middle-left and middle-right locations) employ three borders and temporary blocks in $\mathbb{V}_8^{\mathbf{s}}$ (top-left, top-right, bottom-left and bottom-right locations) employ only two borders, we only select $C_*^{\mathbf{s}'}$ for $\mathbb{V}_4^{\mathbf{s}}$ due to its higher reliability. Then, the process of selecting $\tilde{B}^{\mathbf{s}}$ and reconstructing its $\mathbb{V}_4^{\mathbf{s}}$ is repeated until the complete background is generated. A summary of SMR is given in Algorithm 2.1.

2.5.3 Rejection based Multipath Reconstruction

SMR focuses on smoothness between adjacent blocks (external continuity, Φ similarity in Eq. 2.11) and, therefore, objects far from block boundaries may be unnoticed (e.g. stationary objects). These objects may have the minimum Φ value and be wrongly selected as the best candidate (Eq. 2.13). Moreover, another source of error exists as all external borders are not analyzed in $\mathbb{V}_8^{\mathbf{s}}$.

Extending SMR, we propose a *Rejection based Multipath Reconstruction* (RMR) scheme to overcome these limitations by rejecting reconstructions with high uncertainty, i.e. where some candidates $C_l^{\mathbf{s}'}$ have similar Φ value to the selected $C_*^{\mathbf{s}'}$ in Eq. 2.13. We disambiguate such selection by analyzing internal variations via intra-block heterogeneity Ψ and similarities to adjacent neighbors via inter-block color dissimilarity Ω. Figure 2.7 presents the diagram of operations performed by RMR.

RMR starts from an initial background estimation $\tilde{\mathcal{B}}$ containing seeds $S^{\mathbf{s}}$ and empty locations (*Estimate initial background* stage in Figure 2.7). Then, RMR iteratively chooses a location $\mathbf{s}$ to reconstruct its empty neighbors via multiple paths $m \in \{1 \dots 8\}$ similarly to SMR (*Find location*

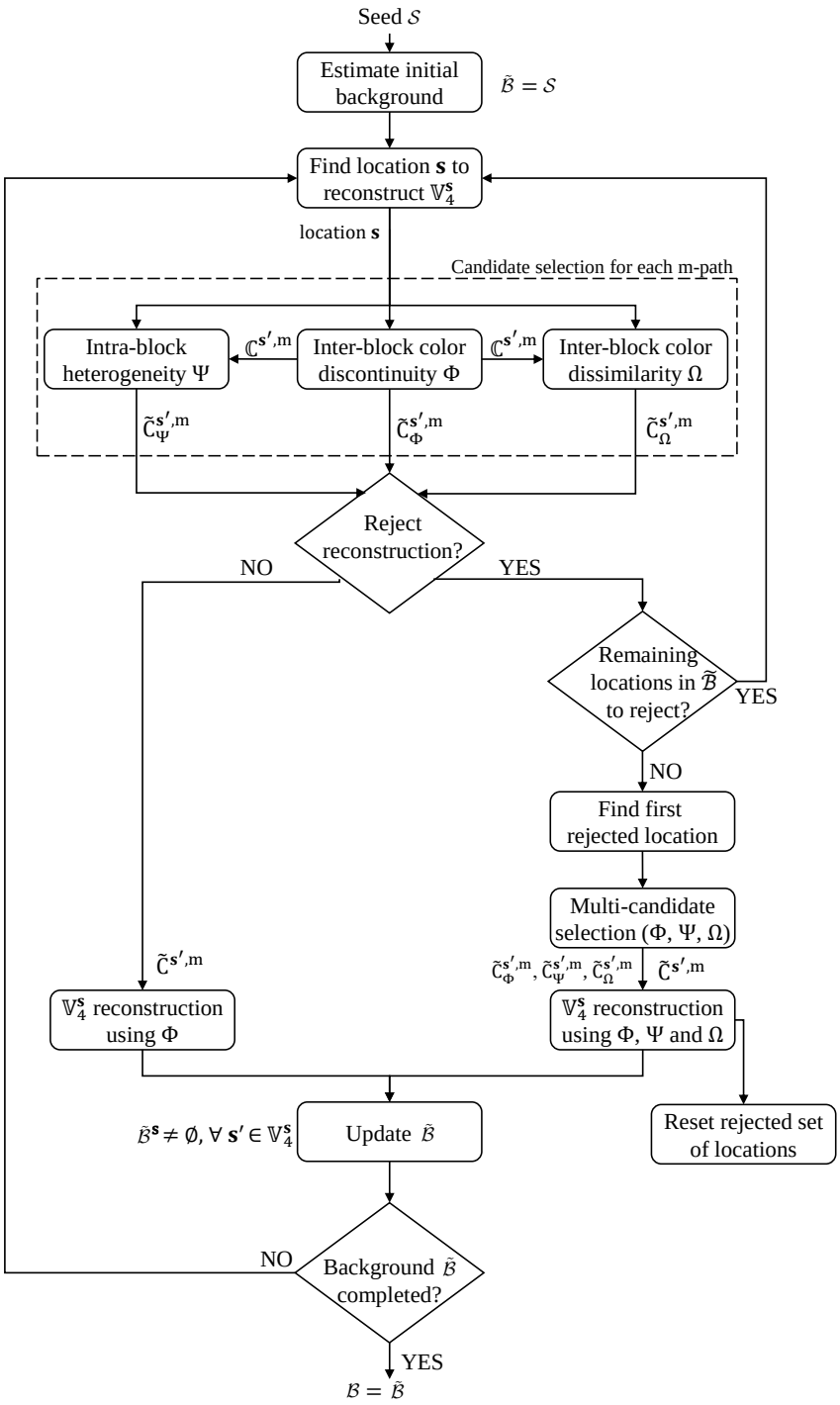


Figure 2.7: RMR diagram of operations. The diagram starts in the the top and ends in the bottom.

s stage in Figure 2.7).

For each m-path, we then obtain the best candidate $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ using Φ as in Eq. 2.12. To infer

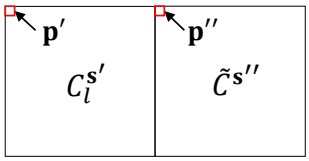


Figure 2.8: Scheme used to compute the inter-block color dissimilarity measure Ω. Pixel distances between $\mathbf{p}'$ and $\mathbf{p}''$ from blocks $C_l^{\mathbf{s}'}$ and $\tilde{C}^{\mathbf{s}''}$ are computed.

high uncertain selections in the location s', a subset of candidates is obtained from the available ones $\left\{C_l^{\mathbf{s}'}\right\}_{l=1\ldots N^{\mathbf{s}'}}$:

$$\mathbb{C}^{\mathbf{s}',m} = \left\{C_l^{\mathbf{s}'} \,\forall l \,:\, \left|\Phi(C_l^{\mathbf{s}'}) - \Phi(\tilde{C}_{\Phi}^{\mathbf{s}',m})\right| < \rho\right\}, \tag{2.14}$$

where ρ is a similarity threshold with a small value to obtain highly similar candidates to the best selection $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ that satisfy the smoothness constraints of the neighborhood.

To resolve such uncertainty in the selection using Φ, we employ intra-block heterogeneity Ψ and inter-block color dissimilarity Ω to the subset of candidates $C_l^{\mathbf{s}'} \in \mathbb{C}_l^{\mathbf{s}',m}$:

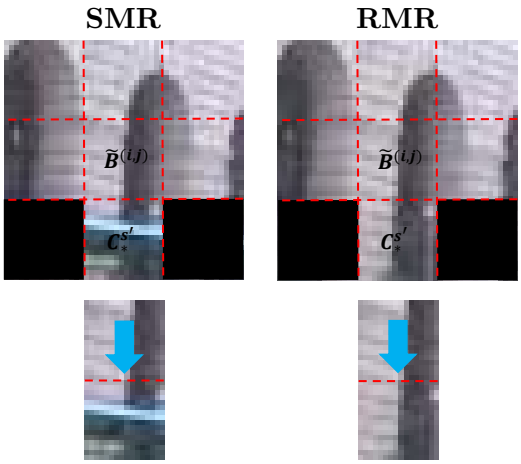
$$\Psi\left(C_l^{\mathbf{s}'}\right) = \sum_{q=1}^{64} \left|A_q(C_l^{\mathbf{s}'})\right|^2, \tag{2.15}$$

$$\Omega\left(C_l^{\mathbf{s}'}\right) = \frac{1}{\left|\mathbb{V}_4^{\mathbf{s}'}\right|} \sum_{\mathbf{s}''\in\mathbb{V}_4^{\mathbf{s}'}} \sum_{\substack{\mathbf{p}'\in\mathbf{s}' \\ \mathbf{p}''\in\mathbf{s}''}} 1 - g\left(C_l^{\mathbf{s}'}\left(\mathbf{p}'\right), \tilde{C}^{\mathbf{s}''}\left(\mathbf{p}''\right)\right), \tag{2.16}$$

where A_q are the coefficients of the Discrete Cosine Transform (A_1 is set to 0 to remove zero-frequency data) [Ahmed et al., 1974] and $g(\cdot,\cdot)$ is the cosine similarity [Dony and Wesolkowski, 1999] between two pixels $\mathbf{p}'$ and $\mathbf{p}''$ from blocks $C_l^{\mathbf{s}'}$ and $\tilde{C}^{\mathbf{s}''}$. Figure 2.8 illustrates the scheme to compute Ω between blocks $C_l^{\mathbf{s}'}$ and $\tilde{C}^{\mathbf{s}''}$. $\Psi(C_l^{\mathbf{s}'})$ measures the variability of RGB values for the block considered whereas $\Omega(C_l^{\mathbf{s}'})$ measures the average pixel-level difference between RGB values of pixels in $C_l^{\mathbf{s}'}$ and $\tilde{C}^{\mathbf{s}''}$. Figure 2.9 presents a comparative example of the $\mathbb{V}_4^{\mathbf{s}}$ reconstruction. SMR selects a wrong candidate when an artifact appears in Figure 2.9(a) (e.g. block $C_*^{\mathbf{s}'}$ with part of a blue bus occluding the background). As the measures Ψ and Ω have high values for this artifact, RMR correctly reconstructs the background as depicted in Figure 2.9(b). Note that the use of inter-block measures (Φ and Ω) minimizes discontinuities between blocks, thus reducing the block effect.

For each m-path, we apply $\Psi(C_l^{\mathbf{s}'})$ and $\Omega(C_l^{\mathbf{s}'})$ to the subset of candidates $C_l^{\mathbf{s}'} \in \mathbb{C}_l^{\mathbf{s}',m}$ in order to obtain two additional best candidates $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ as:

$$\tilde{C}_{\Psi}^{\mathbf{s}',m} = \underset{\forall C_l^{\mathbf{s}'}\in\mathbb{C}_l^{\mathbf{s}'}, l\in\{1\ldots N^{\mathbf{s}'}\}}{argmin} \Psi(C_l^{\mathbf{s}'}), \tag{2.17}$$



(a) (b)

Figure 2.9: Example of the benefits that introduces the use of intra-block heterogeneity (Ψ) and inter-block dissimilarity (Ω) in RMR. (a) and (b) show the reconstruction of $\mathbb{V}_4^{(i,j)}$ of $\tilde{B}^{(i,j)}$ using SMR and RMR respectively, where the reconstructed block $C_*^{s'}$ was the only one unset from $\mathbb{V}_4^{(i,j)}$. Note that RMR is able to select the correct background through Ψ and Ω as they enforce the background smoothness, thus preventing the selection of artifacts done by SMR.

$$\tilde{C}_{\Omega}^{\mathbf{s}',m} = \underset{\forall C_l^{\mathbf{s}'} \in \mathbb{C}_l^{\mathbf{s}'},\, l \in \{1 \ldots N^{\mathbf{s}'}\}}{argmin} \Omega(C_l^{\mathbf{s}'}). \tag{2.18}$$

Thus, we infer highly-uncertain candidates when the three best selections $\tilde{C}_{\Phi}^{\mathbf{s}',m}$, $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ disagree (*Reject reconstruction?* stage in Figure 2.7). Therefore, we reject the assignment of a candidate to the background when:

$$Rejection = \begin{cases} 1 \; if & \neg(\tilde{C}_{\Phi}^{\mathbf{s}',m} = \tilde{C}_{\Psi}^{\mathbf{s}',m} = \tilde{C}_{\Omega}^{\mathbf{s}',m}) \\ 0 & otherwise \end{cases}. \tag{2.19}$$

This rejection identifies when the candidate $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ is more homogeneous (low Ψ) or the candidate $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ is more similar to its neighborhood (low Ω) as compared to $\tilde{C}_{\Phi}^{\mathbf{s}',m}$. Hence, no assignment is done since a more suitable candidate may be employed ($\tilde{C}_{\Psi}^{\mathbf{s}',m}$ or $\tilde{C}_{\Omega}^{\mathbf{s}',m}$). The $\mathbb{V}_4^{\mathbf{s}}$ reconstruction of $\tilde{B}^{\mathbf{s}}$ is not performed when any m-path is rejected. Conversely, the $\mathbb{V}_4^{\mathbf{s}}$ reconstruction is performed as for SMR when none of the m-paths is rejected.

After rejecting all remaining locations (*Remaining locations in $\tilde{\mathcal{B}}$ to reject?* stage in Figure 2.7), we analyze these rejected locations to complete the background reconstruction. Another iterative process begins to determine the next location $\tilde{B}^{\mathbf{s}}$ (*Find first rejected location* stage in Figure 2.7) and to select the candidate $\tilde{C}^{\mathbf{s}',m}$ for each m-path (*Multi-candidate selection* stage

in Figure 2.7) using a set of rules:

$$\tilde{C}^{\mathbf{s}',m} = \begin{cases} \tilde{C}_{\Phi}^{\mathbf{s}',m} & if \quad \tilde{C}_{\Phi}^{\mathbf{s}',m} = \tilde{C}_{\Psi}^{\mathbf{s}',m} = \tilde{C}_{\Omega}^{\mathbf{s}',m} \\ \tilde{C}_{\Psi}^{\mathbf{s}',m} & if \quad \tilde{C}_{\Phi}^{\mathbf{s}',m} \neq \tilde{C}_{\Psi}^{\mathbf{s}',m} \\ \tilde{C}_{\Omega}^{\mathbf{s}',m} & if \quad \tilde{C}_{\Phi}^{\mathbf{s}',m} = \tilde{C}_{\Psi}^{\mathbf{s}',m} \wedge \tilde{C}_{\Omega}^{\mathbf{s}',m} \neq \tilde{C}_{\Phi}^{\mathbf{s}',m} \end{cases}, \tag{2.20}$$

where $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ is selected when all blocks are the same, $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ is selected when it has better homogeneity than $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ as this may denote the presence of an artifact and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ is selected when the second condition does not occur and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ has better color similarity than $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ with its neighbors, i.e. there is a block with better Ω denoting that $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ may contain an artifact.

After selecting the m-candidates $\tilde{C}_{\Phi}^{\mathbf{s}',m}$, $\tilde{C}_{\Psi}^{\mathbf{s}',m}$ and $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ for all m-paths in Eq. (2.20), we combine them to obtain the best candidate $C_{*}^{\mathbf{s}'}$ for the location $\mathbf{s}'$:

$$C_{*}^{\mathbf{s}'} = \underset{m \in \{1,\ldots,8\}}{argmin} \ \Gamma\left(\tilde{C}_{\Phi}^{\mathbf{s}',m}, \tilde{C}_{\Psi}^{\mathbf{s}',m}, \tilde{C}_{\Omega}^{\mathbf{s}',m}\right), \tag{2.21}$$

where Γ combines the Φ, Ψ and Ω measures for the candidates for each m-path as:

$$\Gamma = \begin{cases} \overline{\Phi}(\tilde{C}^{\mathbf{s}',m}) & if \quad \tilde{C}^{\mathbf{s}',m} = \tilde{C}_{\Phi}^{\mathbf{s}',m} \\ \overline{\Phi}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) + \overline{\Psi}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) + \overline{\Omega}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) & if \quad (\tilde{C}^{\mathbf{s}',m} = \tilde{C}_{\Psi}^{\mathbf{s}',m}) \wedge \\ & \qquad (\overline{\Omega}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) \leq \overline{\Omega}(\tilde{C}_{\Phi}^{\mathbf{s}',m})) \\ \overline{\Phi}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) + \overline{\Psi}(\tilde{C}_{\Psi}^{\mathbf{s}',m}) & if \quad \tilde{C}^{\mathbf{s}',m} = \tilde{C}_{\Psi}^{\mathbf{s}',m} \\ \overline{\Phi}(\tilde{C}_{\Omega}^{\mathbf{s}',m}) + \overline{\Omega}(\tilde{C}_{\Omega}^{\mathbf{s}',m}) & if \quad \tilde{C}^{\mathbf{s}',m} = \tilde{C}_{\Omega}^{\mathbf{s}',m} \end{cases}, \tag{2.22}$$

where the location $\mathbf{s}' \in \mathbb{V}_4^{\mathbf{s}}$; $\overline{\Phi}$, $\overline{\Psi}$ and $\overline{\Omega}$ are the normalized measures to the range [0,1] by their maximum value for all m-paths. Each case represents a different rejection, where the first one is applied when no rejection is detected in $\mathbf{s}'$, while the second, third and fourth cases apply to rejections due to Ψ and Ω, only Ψ and only Ω, respectively. This reconstruction of $\mathbb{V}_4^{\mathbf{s}}$ updates $\tilde{B}$ and it is iteratively performed until the entire background $\tilde{\mathcal{B}}$ is reconstructed (*Background $\tilde{\mathcal{B}}$ completed?* stage in Figure 2.7). The final estimated background $\mathcal{B}$ corresponds to the last iterative update of $\tilde{\mathcal{B}}$. A summary of RMR is presented in algorithm 2.2.

2.6 Experimental work

We evaluate the temporal and spatial analysis of the proposed approach, *Rejection based Multipath Reconstruction* (RMR), and provide comparisons against representative state-of-the-art approaches.

Algorithm 2.2 Rejection based Multipath Reconstruction (RMR).

Input: $S^{\mathbf{s}}$ seeds and $C_l^{\mathbf{s}}$ candidates
Output: $\mathcal{B} = \{B^{\mathbf{s}}\}_{\forall s} : B^{\mathbf{s}} \neq \emptyset$, $\forall \mathbf{s}$.

while $(\exists\, \tilde{B}^{\mathbf{s}} = \emptyset)$
  $\mathbb{K} = \emptyset$ (set of currently rejected locations)
  Selection of $\mathbf{s} : \tilde{B}^{\mathbf{s}} \neq \emptyset \wedge \mathbf{s} \notin \mathbb{K}$
  $Assigned = 0$
  $allR = 0$
  while $(Assigned = 0)$
    $Rejection = 0$
    for $m = 1$ to 8 **do**
      for $\mathbf{s}' \in \mathbb{V}_8^{\mathbf{s}}$
        if $\tilde{B}^{\mathbf{s}'} = \emptyset$ **then**
          Select $\tilde{C}_{\Phi}^{\mathbf{s}',m}$, $\tilde{C}_{\Psi}^{\mathbf{s}',m}$, $\tilde{C}_{\Omega}^{\mathbf{s}',m}$ with Eqs. 2.12, 2.17, 2.18
          if $\tilde{C}_{\Psi}^{\mathbf{s}',m} \neq \tilde{C}_{\Phi}^{\mathbf{s}',m} \vee \tilde{C}_{\Omega}^{\mathbf{s}',m} \neq \tilde{C}_{\Phi}^{\mathbf{s}',m} \wedge allR = 0$ **then**
            add $\mathbf{s}$ to $\mathbb{K}$
            $Rejection = 1$
            break
          else
            Select $\tilde{C}^{\mathbf{s}',m}$ with Eq. 2.20
          end
        end
      end
      if $Rejection = 1$ **then**
        break
      end
    end
    if $Rejection = 1$ **then**
      if all $\mathbf{s}$ are rejected **then**
        $\mathbb{K} = \emptyset$, $Rejection = 0$
        $allR = 1$
      else
        break
    else
      $Assigned = 1$
      for $\mathbf{s}' \in \mathbb{V}_4^{\mathbf{s}}$
        Select $C_*^{\mathbf{s}'}$ using Eq. 2.21
        $\tilde{B}^{\mathbf{s}'} = C_*^{\mathbf{s}'}$
      end
    end
  end
end
$\mathcal{B} = \tilde{\mathcal{B}}$

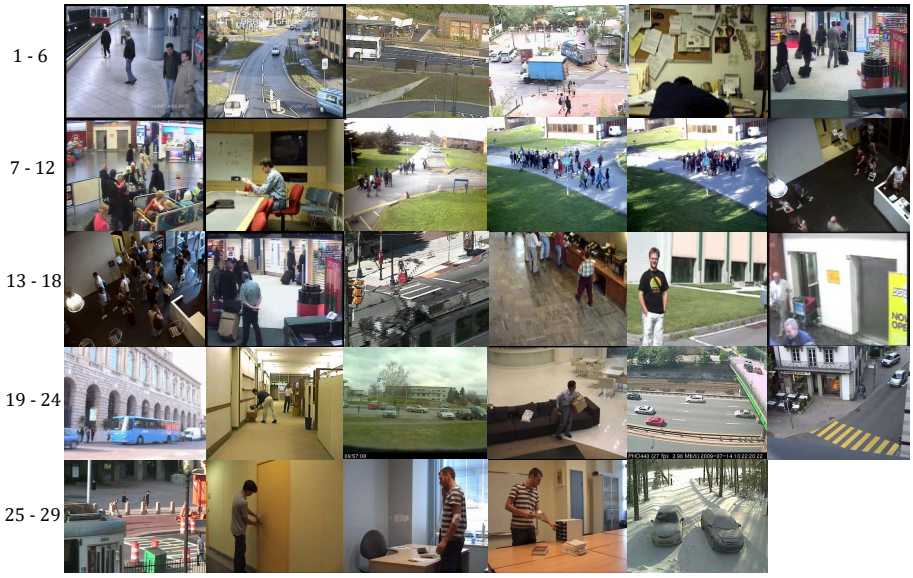


Figure 2.10: Visual examples of the selected sequences for evaluation. The IDs on the left correspond to the ones in Table 2.2. Ground-truth backgrounds are available at http://www-vpu.eps.uam.es/publications/BE_RMR.

2.6.1 Evaluation framework

2.6.1.1 Dataset

For evaluation we use 29 real sequences selected from public datasets (Wallflower[2] [Toyama et al., 1999], COST211[3], AVSS 2007[4], LIMU[5], TRECVID[6], PBI[7] [Colombari and Fusiello, 2010], IDIAP[8] [Varadarajan and Odobez, 2009], PETS 2009[9] [Ellis et al., 2009], SAIVT-Campus [Xu et al., 2012], CUHK[10] [Wang et al., 2012], LIRIS 2012[11] [Wolf et al., 2014], CDNET[12] [Wang et al., 2014b]), covering different scenarios and complexities (see Figure 2.10), mainly stationary objects and crowds. Ground-truth data[13] has been manually composed from instants where the scene (or part of it) is foreground-free. Table 2.2 describes the properties of video sequences in terms of foreground *Stationarity*, according to size and duration; *Visibility*, according to duration

[2]http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm
[3]http://www.csd.uoc.gr/~tziritas/cost.html
[4]http://www.eecs.qmul.ac.uk/~andrea/avss2007.html
[5]http://limu.ait.kyushu-u.ac.jp/dataset/en/
[6]http://trecvid.nist.gov/trecvid.data.html
[7]http://www.diegm.uniud.it/fusiello/demo/bkg/
[8]http://www.idiap.ch/~odobez/RESSOURCES/DataRelease-TrafficJunction.php
[9]http://www.cvg.reading.ac.uk/PETS2009/
[10]http://www.ee.cuhk.edu.hk/~xgwang/CUHK_square.html
[11]http://liris.cnrs.fr/voir/activities-dataset/videoframes.html
[12]http://changedetection.net/
[13]Software and ground-truth data available at http://www-vpu.eps.uam.es/publications/BE_RMR

ID	Video	Dataset	#f	T	S	V	SI
1	*AB_H*	AVSS 2007	400	I	H	M	M
2	*PV_E*	AVSS 2007	500	I	H	L	M
3	*BSM*	LIMU	400	O	H	L	L
4	*SQ*	CUHK	500	O	H	L	L
5	*FGA*	Wallflower	400	I	H	L	L
6	*TREC1*	TRECVID	498	I	H	H	M
7	*TREC2*	TRECVID	699	I	L	H	M
8	MO	Wallflower	300	I	H	L	L
9	*PETS1*	PETS 2009	221	O	L	H	H
10	*PETS2*	PETS 2009	240	O	M	H	H
11	*PETS3*	PETS 2009	378	O	H	H	M
12	*Test*	SAIVT Campus	500	I	L	M	M
13	*Train*	SAIVT Campus	500	I	L	H	H
14	*TREC3*	TRECVID	400	I	M	M	M
15	*AB_Box*	CDNET	500	O	H	M	L
16	*bootstrap*	Wallflower	294	I	L	L	H
17	*ca_vignal*	PBI	258	O	M	L	L
18	*cam4*	TRECVID	300	I	M	L	L
19	*guardia*	PBI	400	O	H	M	L
20	*hall_m*	COST	300	I	M	M	L
21	*parking*	CDNET	400	O	H	L	L
22	*sofa*	CDNET	400	I	H	L	L
23	*st_light*	CDNET	400	O	H	H	L
24	*traffic*	IDIAP	500	O	H	L	L
25	*tramp*	CDNET	400	O	H	H	L
26	*vid16*	LIRIS 2012	380	I	H	L	L
27	*vid22*	LIRIS 2012	345	I	M	M	L
28	*vid36*	LIRIS 2012	128	I	M	M	L
29	*winter*	CDNET	500	O	H	L	M

Table 2.2: Dataset description. Key. #f: Number of frames. T: Type. I: Indoor. O: Outdoor. S: Stationary region complexity. V: Visibility of empty scene complexity. SI: Shadows and Illumination changes complexity. L, M and H mean low, medium and high levels, respectively.

and size of the background visualized along time; *Shadows and Illumination changes*, according to the amount of these photometric factors. The ID of the video sequences displayed in Table 2.2 is used to report results. Additionally, comparisons are provided for the SBMI2015 dataset[14] [Maddalena and Petrosino, 2015] that contains 7 video sequences with their ground-truth images for the task of BE.

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

2.6.1.2 Evaluation measures

We compute performance via six different error measures adopted from SBMI2015 [Maddalena and Petrosino, 2015]. Three SBMI2015 measures employ the absolute gray-level difference Δ, which is defined for each pixel as:

$$\Delta^{\mathbf{p}} = |\mathcal{B}^{\mathbf{p}} - \mathcal{G}^{\mathbf{p}}|_Y \ , \tag{2.23}$$

where $\mathcal{B}$ and $\mathcal{G}$ denote the estimated and the ground-truth backgrounds, respectively. $|\cdot|_Y$ is the pixel-level absolute difference using the luminance information Y. The first measure, Average Gray-level Error (AGE), is the mean Δ value over the image. The second measure, Average of Error pixels (AE), determines pixel errors by thresholding Δ with $\alpha = 20$ and computes the percentage of error pixels in the image. The third measure, Average of Clustered Error pixels (ACE), considers the average number of error pixels where their 4-connected neighbors are error pixels. The lower the value, the better performance for AGE, AE and ACE. The remaining three measures are Peak-Signal-to-Noise-Ratio (PSNR), Multi-Scale Structural Similarity index (MS-SSIM) and Color image Quality Measure (CQM). The higher the value, the better performance for these three measures.

Additionally, we propose a threshold-free error measure to avoid the threshold dependency exhibited by AE. A number of thresholds α_i are employed to generate a curve with the corresponding AE values where the Area Under the Curve (AUC) is reported for performance evaluation.

2.6.1.3 Parametrization

For the proposed approach, we use $W = 16$ as the block size similarly to [Reddy et al., 2009][Baltieri et al., 2010][Reddy et al., 2011]. We heuristically set $k = 3$ for inter-frame differences in Eq. 2.1 to increase the motion detected as compared to consecutive frame differences. Finally, $\rho = 5$ is heuristically set to select candidates with color discontinuity similar to the minimum value in Eq. 2.14, as they may be part of the background. Note that we use less heuristic parameters than related state-of-the-art approaches [Reddy et al., 2009][Baltieri et al., 2010][Reddy et al., 2011][Hsiao and Leou, 2013].

2.6.2 Temporal analysis evaluation

We compare the proposed clustering to generate background candidates (Subsection 2.4.3) against the sequential clustering of algorithm DCT [Reddy et al., 2011], which is chosen as a top-ranked state-of-the-art result (as shown in Subsection 2.6.5). DCT clustering requires two thresholds to associate blocks into clusters; while the proposed clustering is automatic. We measure performance by inspecting whether any of the candidates $C_l^{\mathbf{s}}$ contains G^s so the spatial

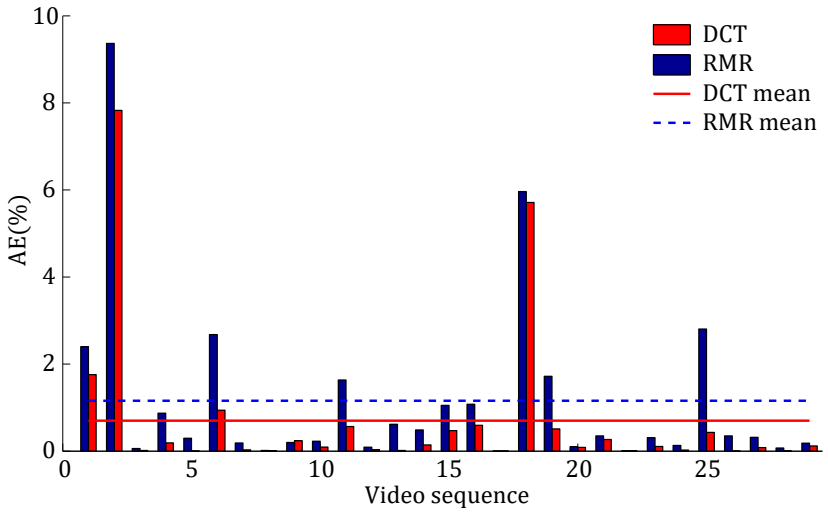


Figure 2.11: Clustering evaluation. The figure shows the AE error measure. The x-axis is the video sequence ID referenced in Table 2.2. The lower AE the better.

analysis may be able to reconstruct the background. Firstly, we determine the best matching between candidates and ground-truth $B^{\mathbf{s}}_{best}$ as follows:

$$B^{\mathbf{s}}_{best} = \underset{C^{\mathbf{s}}_l}{argmin} \left(\max_{\mathbf{p}} \left(\Delta \left(C^{s,\mathbf{p}}_l, G^{s,\mathbf{p}} \right) \right) \right). \tag{2.24}$$

Secondly, we compute the AE measure ($\alpha = 20$) between $B^{\mathbf{s}}_{best}$ and the ground-truth data. Figure 2.11 compares mean AE performance for the proposed and selected approaches where both present similar scores, 1.157% for RMR and 0.699% for DCT. Attending to each sequence performance, both algorithms achieve low errors for all sequences except for 2 and 18, where some selected blocks of $B^{\mathbf{s}}_{best}$ differ from the ground-truth data due to variations in the illumination and reflections, respectively. Although RMR clustering slightly reduces performance compared to DCT [Reddy et al., 2011], it has the advantage of being automatic (threshold-free), thus avoiding the adjustment needed in DCT clustering for different environments.

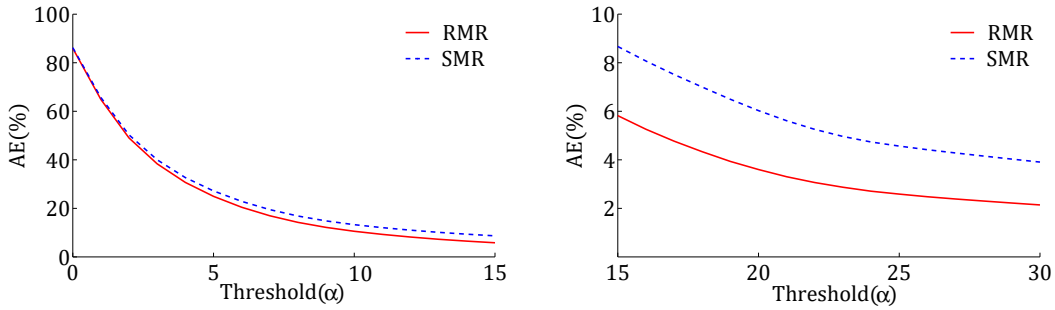
2.6.3 Seed selection technique evaluation

We compare the performance of the RMR *Seed Selection* with the one proposed in DCT [Reddy et al., 2011] where seed locations are selected when only a single candidate exists. As shown in Table 2.3, RMR initializes a higher percentage of the reconstructed background $\tilde{\mathcal{B}}$ (19.02%) than DCT (10.01%), measured with Reconstruction Percentage RP, i.e. amount of reconstructed blocks in the initialization, while keeping the correct selection of initial $\tilde{B}^{\mathbf{s}}$ blocks, i.e. S^s, measured with AE ($\alpha = 20$). Low RP occurs when many block locations contain variations along the training frames, which is induced by low background visibility (7, 25, 28), background variations

	RP		AE	
ID	**DCT [Reddy et al., 2011]**	**RMR**	**DCT [Reddy et al., 2011]**	**RMR**
1	4.11	12.80	0.14	1.92
2	3.62	4.11	3.70	3.40
3	0.33	13.33	0.00	0.00
4	0.48	34.30	0.00	0.2
5	1.25	1.25	0.39	0.017
6	5.31	24.88	0.00	0.00
7	0.72	2.17	0.13	0.00
8	13.75	1.25	0.00	0.00
9	14.12	56.71	0.00	0.00
10	0.23	26.62	0.00	0.16
11	3.70	18.29	0.00	0.00
12	13.89	18.18	0.05	0.00
13	1.52	10.10	0.00	0.00
14	6.28	15.22	0.00	0.89
15	0.41	12.35	0.00	0.00
16	1.25	3.75	0.00	0.00
17	22.22	11.11	0.00	3.13
18	51.25	5.00	5.68	0.00
19	39.16	46.15	0.00	0.00
20	2.120	31.82	0.00	0.10
21	62.67	59.00	0.25	0.00
22	15.33	45.00	0.00	0.00
23	12.33	14.00	0.00	0.00
24	0.97	17.87	0.00	0.00
25	0.21	0.21	0.00	0.00
26	0.48	1.69	0.00	0.00
27	0.24	40.58	0.00	0.00
28	0.24	0.24	0.00	0.00
29	12.00	23.67	0.00	0.00
Mean	10.01	19.02	0.004	0.004

Table 2.3: Seed selection technique evaluation. Comparison between the selection described in DCT algorithm [Reddy et al., 2011] and the proposed approach in RMR. As measures, we report the reconstruction percentage (RP) of the initial $\tilde{\mathcal{B}}$ and AE. ID denotes the number of the video sequence referenced in Table 2.2. The higher RP the better. The lower AE the better. Green, black and red denotes better, equal and worse result than [Reddy et al., 2011], respectively.

due to shadows (2, 16) or changing backgrounds (18) and large stationary objects (5, 8 and 26). Starting with a higher amount of initialized background blocks $\tilde{B}^{\mathbf{s}}$ provides more information for the iterative reconstructions which leads to improvements in the background estimation performance. Note that AE is computed over a partially reconstructed background whose average



(a) (b)

Figure 2.12: Comparison of SMR against RMR. Lower means better performance. (a) Comparison with $\alpha \in [0, 15]$. (b) Comparison with $\alpha \in [15, 30]$.

percentage RP is almost the double in RMR than in DCT, i.e. the initial estimation of the background contains more pixels and it may lead to more error pixels.

2.6.4 Spatial analysis evaluation

We compare RMR with SMR to show the benefits of iterative rejection. We avoid the threshold dependency of AE by computing multiple results using $\alpha \in [0, 30]$. The overall results for all sequences are shown in Figure 2.12 in terms of average AE. Figure 2.12(a) shows that SMR and RMR present similar results for low α values whereas Figure 2.12(b) indicates that RMR outperforms SMR due to its rejection capability. The Area Under the Curve (AUC) for SMR and RMR (lower area means better performance) is 392.86 and 359.13 for the evaluation interval $\alpha \in [0, 15]$ and 83.40 and 49.79 for $\alpha \in [15, 30]$.

The RMR improvement over SMR is illustrated by the examples in Figure 2.13, where reconstructions of $\mathbb{V}_4^{\mathbf{s}}$ for SMR and RMR are presented. For reconstructing the blue locations, SMR selects erroneous blocks, corresponding to artifacts (stationary objects), while RMR selects proper blocks. This occurs as SMR does not cope with the lack of not analyzing external edges of $\mathbb{V}_4^{\mathbf{s}}$ (black arrows), thus allowing discontinuities in that areas and due to failures of the fact that $\tilde{C}_{\Phi}^{\mathbf{s}',m}$ is the best candidate (as can be shown in the three examples of the figure). RMR solves these problems by analyzing Ψ and Ω of similar blocks belonging to $\mathbb{C}^{\mathbf{s}',m}$ (Eq. 2.14) and performing the rejection scheme.

2.6.5 Comparison against related approaches

We compare the proposed approach RMR against BE-specific approaches and top-ranked background subtraction algorithms. For BE, we select RSM [Wang and Suter, 2006], DCT [Reddy et al., 2011], the Median (MED) [Maddalena and Petrosino, 2014a] and IMBS-1 [Bloisi et al., 2014]. For background subtraction, we use Fuzzy [El Baf et al., 2008], SC-SOBS [Maddalena

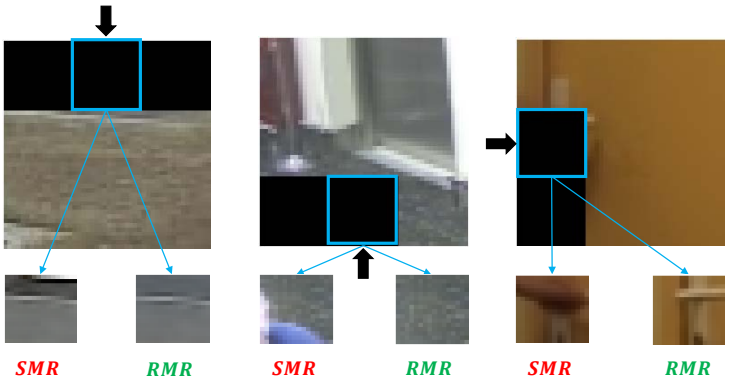


Figure 2.13: Examples of failures of SMR solved by RMR for the sequences *BSM* (left), *cam4* (middle) and *vid16* (right).

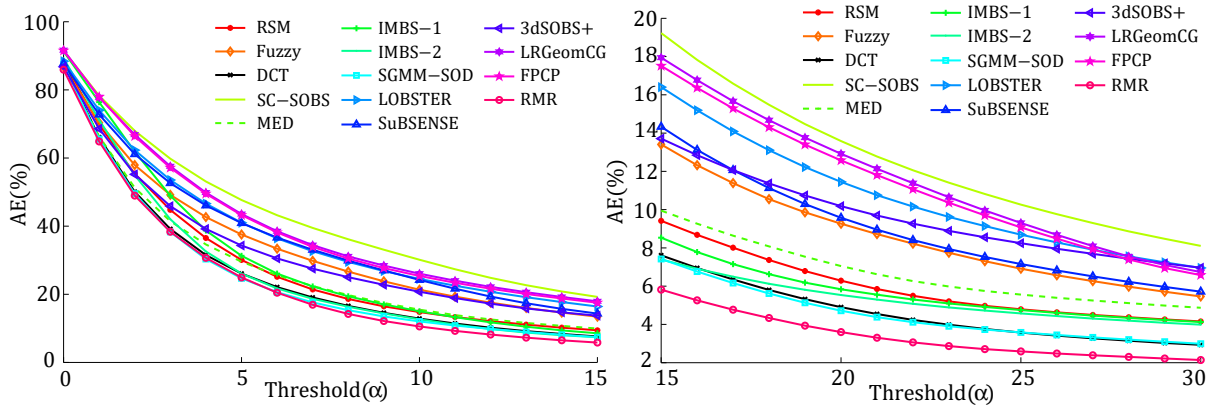


Figure 2.14: Evaluation of RMR against state-of-the-art methods for the task of BE and using the average of AE for all sequences. (a) Comparison with $\alpha \in [0, 15]$. (b) Comparison with $\alpha \in [15, 30]$.

and Petrosino, 2012], 3dSOBS+ [Maddalena and Petrosino, 2014a], IMBS-2 [Bloisi et al., 2014], LOBSTER [St-Charles and Bilodeau, 2014], SGMM-SOD [Evangelio et al., 2014], SuBSENSE [St-Charles et al., 2015] and two algorithms based on low-rank and sparse decomposition, LRGeomCG [Vandereycken, 2013] and FPCP [Rodriguez and Wohlberg, 2013]. For these non-specific BE algorithms, we use the estimated background after processing all training frames. Note that their BE results may not reflect their performance for foreground detection. We use the BGSLibrary [Sobral and Vacavant, 2014] (Fuzzy, LOBSTER and SuBSENSE) and the LRSLibrary [Sobral et al., 2015] (LRGeomGC and FPCP). IMBS-1 uses IMBS initialization over the training frames, while IMBS-2 uses the default algorithm. We use default parameters for all approaches.

Figure 2.14 compares AE performance for the threshold $\alpha \in [0, 30]$ where results are split in

Approach	AUC		AGE	AE	ACE	MS-SSIM	PSNR	CQM
	$\alpha \in [0, 15]$	$\alpha \in [15, 30]$						
RMR	**359.13** +3.6%	**49.79** +25.0%	**5.37** +10.3%	**3.60** +23.7%	**1.67** +19.7%	**0.955** +1.6%	**30.17** +6.3%	**40.77** +2.4%
SGMM-SOD	372.56	66.38	5.99	4.72	2.08	0.940	28.37	39.83
DCT	384.79	67.55	6.12	4.90	2.35	0.939	27.66	38.94
IMBS-2	396.47	78.14	7.08	5.54	2.60	0.908	25.23	37.07
IMBS-1	451.83	83.15	7.83	5.84	2.74	0.907	24.44	36.00
RSM	428.39	88.00	7.65	6.29	2.96	0.899	24.40	36.73
MED	420.94	98.88	7.86	7.05	4.35	0.900	24.52	37.85
Fuzzy	510.20	126.24	8.29	9.26	5.79	0.911	25.67	39.06
SuBSENSE	549.23	131.69	9.36	9.57	4.36	0.899	24.31	35.83
3dSOBS+	486.64	142.45	9.18	10.18	6.10	0.881	24.32	36.72
LOBSTER	559.79	157.10	10.07	11.45	4.75	0.875	23.94	34.95
FPCP	588.08	166.71	9.79	12.57	8.56	0.898	24.52	-
LRGeomCG	594.87	170.99	9.95	12.92	8.57	0.896	24.41	-
SC-SOBS	639.78	185.33	11.68	13.58	6.46	0.839	22.24	35.23

Table 2.4: Comparison against state-of-the-art methods in terms of AUC and SBMI2015 error measures for the proposed dataset of 29 video sequences. The lower AUC, AGE, AE and ACE the better performance, while the higher MS-SSIM, PSNR and CQM the better the performance. Methods are presented in descending ranking order according to AUC for $\alpha \in [15, 30]$. Note that CQM measure is not computed for FPCP and LRGemoCG as background is obtained in gray-scale. The percentage of improvement compared to best state-of-the-art approach is shown under RMR performance.

two intervals to improve visibility. RMR has the best performance for both threshold intervals, followed by SGMM-SOD and DCT. The first sweep of the AE threshold ($\alpha \in [0, 15]$) presents high variation as low α values do not allow small variability with the ground-truth which should be handled as training frames may contain additive noise. Therefore, the sweep $\alpha \in [15, 30]$ is preferable to compute the performance. Table 2.4 includes further details in terms of AUC and SBMI2015 error measures. For all measures RMR outperforms state-of-the-art results, coping with stationary objects much better. Due to the variability of AE for $\alpha \in [0, 15]$, AUC from $\alpha \in [15, 30]$ better reflects the performance, being the best state-of-the-art approaches SGMM-SOD and DCT as both use smoothness constraints. Improvements can be analyzed regarding two sets of measures; the first includes AUC (significant AUC interval $\alpha \in [15, 30]$), AGE, AE and ACE; and the second one includes MS-SSIM, PSNR and CQM. For the first set of measures, we reduce the error in a range of 10.3 % (AGE) to 25.0% (AUC) compared to SGMM-SOD. For the second set of measures, the improvement compared to SGMM-SOD ranges from 1.6% (MS-SSIM) to 6.3% (PSNR). Additionally, experiments in the SBMI2015 dataset have been carried out (see Table 2.5) where again the proposed approach RMR outperforms the related

Approach	AUC		AGE	AE	ACE	MS-SSIM	PSNR	CQM
	$\alpha \in [0, 15]$	$\alpha \in [15, 30]$						
RMR	**692.06**	**79.49**	**9.75**	**5.21**	**3.61**	**0.964**	**28.52**	39.54
	+6.1%	+50.0%	+23.9%	+50.2%	+49.1%	+6.5%	+8.6%	-1.7%
DCT	743.88	158.97	12.81	10.47	7.09	0.905	26.25	37.50
SGMM-SOD	755.26	209.84	16.19	13.34	9.83	0.884	25.73	35.52
RSM	737.00	236.63	17.00	15.96	10.55	0.816	23.30	35.13
IMBS-1	852.01	247.03	19.40	16.57	8.85	0.831	22.78	33.67
IMBS-2	834.12	279.84	20.72	19.25	10.32	0.795	22.37	33.60
LOBSTER	800.89	347.98	19.06	24.52	14.86	0.812	20.99	31.66
3dSOBS+	794.30	381.02	22.17	25.95	20.78	0.772	21.92	35.94
MED	771.76	393.81	21.31	27.19	22.39	0.806	23.41	37.27
Fuzzy	809.71	449.53	18.87	32.28	26.44	0.882	24.46	**40.23**
SuBSENSE	819.26	453.56	20.89	31.79	23.46	0.845	22.63	37.09
SC-SOBS	912.81	497.13	22.91	35.26	24.91	0.810	21.00	36.77
FPCP	1003.50	646.32	22.53	46.34	40.84	0.891	21.59	-
LRGeomCG	1012.30	656.29	22.90	47.37	40.26	0.885	21.41	-

Table 2.5: Comparison against state-of-the-art methods in terms of AUC and SBMI2015 error measures for the SBMI dataset. The lower AUC, AGE, AE and ACE the better performance, while the higher MS-SSIM, PSNR and CQM the better the performance. Methods are presented in descending ranking order according to AUC for $\alpha \in [15, 30]$. Note that CQM measure is not computed for FPCP and LRGemoCG as background is obtained in gray-scale.The percentage of improvement compared to best state-of-the-art approach is shown under RMR performance.

work and where best compared approaches are again SGMM-SOD and DCT.

In Figure 2.15, sequence results are shown in terms of AUC against the DCT and SGMM-SOD approach (best related works), for $\alpha \in [15, 30]$. As shown in Figure 2.15, the proposed approach is better than DCT in 23 sequences and worse in 6, while compared to SGMM-SOD the proposed approach is better in 19 and worse in 10. The reasons of performance decrease can be compiled into failure of background smoothness assumption (sequences 4, 20 and 23), block effect (sequences 13 and 26), differences between reconstructed background and ground-truth caused by illumination changes or dynamic objects (sequences 2, 5, 12, 18, 28) and erroneous initialization in all algorithms where high error propagation occurs (sequences 1 and 25). Therefore, regarding the stationarity challenge, improvement is obtained in almost all sequences by RMR.

Figure 2.16 shows eight examples of the qualitative results in presence of stationarity, low visibility and camouflages issues. In these examples, unlike most of the state-of-the-art approaches, the proposed approach removes long-term stationary objects and crowds from the reconstructed background $\mathcal{B}$ (see 3, 4, 6, 13, 19, 24 and 29). However, video sequence 4 (*CUHK*) introduces erroneous white blocks due to a higher continuity of a block $C_l^{\mathbf{s}'}$ with a white car that is later

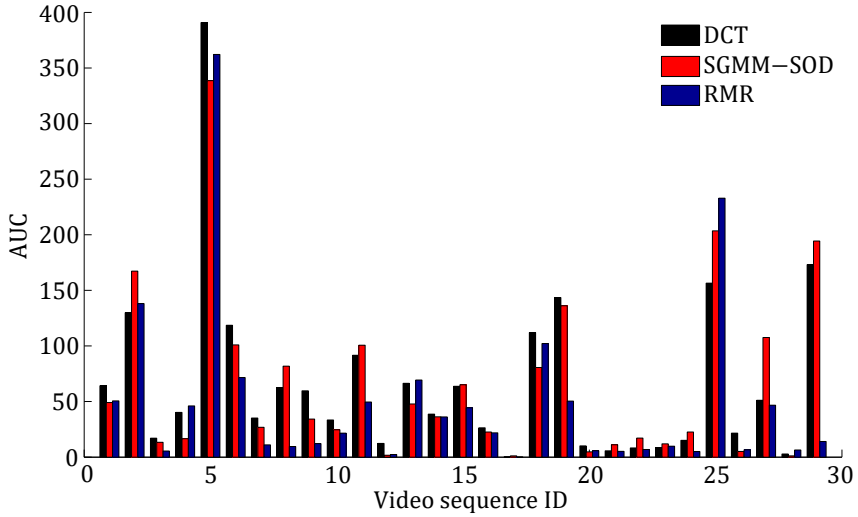


Figure 2.15: Sequence by sequence AUC ($\alpha \in [15, 30]$) of RMR (blue) against DCT (black) and SGMM-SOD (red) for the task of BE. The x-axis is the video sequence ID referenced in Table 2.2. The lower AUC the better performance.

propagated. Also, video sequence 25 (*tramp*) induces errors (also in all the compared state-of-the-art) due to the combination of several problems: inter-block color discontinuity measure Φ fails in one iteration, correct $C_l^{\mathbf{s}'}$ does not belong to $\mathbb{C}^{\mathbf{s}',m}$ so the failure of Φ is not handled and the blocks with foreground motion are not correctly removed due to moving regions bigger than the block size.

The comparative evaluation shows low performance of recent background subtraction algorithms (IMBS-2, LOBSTER, SuBSENSE, SC-SOBS, 3dSOBS+ LRGeomCG and FPCP) when applied to capture the background in situations with crowds or stationary objects. Some of these algorithms (IMBS-2, LOBSTER, SuBSENSE, 3dSOBS+ and SC-SOBS) are much faster than DCT and RMR at the cost of significant performance decreases because of the background assumptions, i.e. foreground is not representative in the training frames, which does not apply to stationary objects or crowds. Therefore, the spatial constrains introduced by RMR or DCT are needed to improve performance for background estimation in complex situations. One exception is SGMM-SOD that removes foreground ghosts based on spatial constraints, allowing a faster background update when stationary objects leave the scene. However, such update depends on the temporal duration of the stationary objects and training frames, obtaining errors when background has low visibility (see sequences 19, 24 and 29 in Figure 2.16) whereas RMR does not have such duration constraints.

The computational cost of the proposed approach is mainly due to the Clustering and Multipath Reconstruction stages, that require approximately 28% and 70% of processing time. Our un-optimized MATLAB implementation of the proposed approach has an average running time of 5.3 μs/pixel (e.g. 200 color 350x240 frames in around 4.5 minutes). Regarding the state-

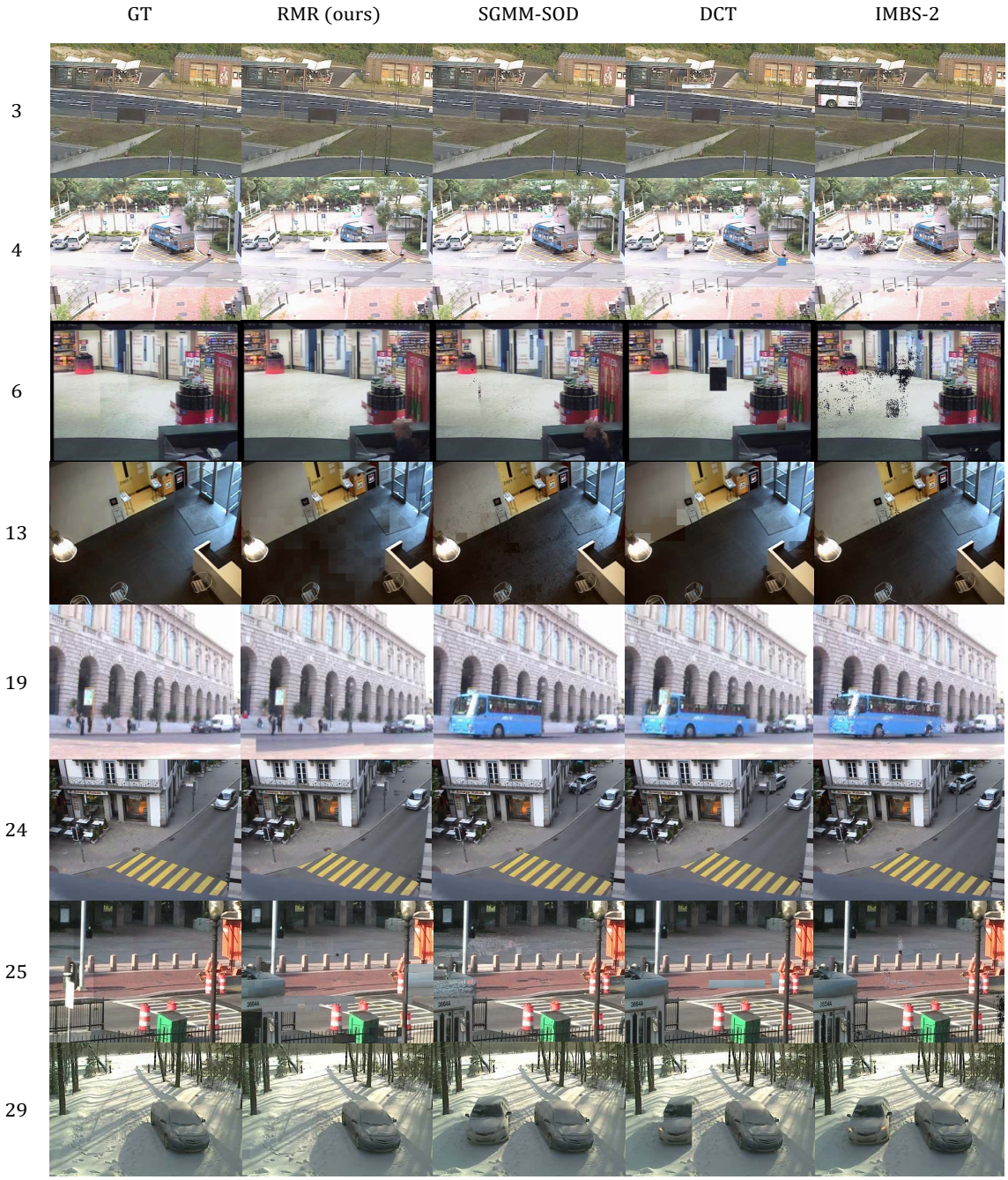


Figure 2.16: Qualitative results showing the estimated background $\mathcal{B}$ of top selected approaches for the BE task. From top to bottom rows: 3 (*BSM*), 4 (*CUHK*), 6 (*TREC1*), 19 (*guardia*), 24 (*traffic*) and 29 (*winter*) are examples with high complexity of stationarity solved successfully, while many approaches of the literature fail; 6 (*TREC1*) and 13 (*Train*) are examples where the background is successfully estimated under low visibility conditions; 25 (*tramp*) is an example of erroneous reconstruction due to non compliance of the rejection conditions. Each column corresponds to the results of a selected approach (first column is the manually extracted GT).

of-the-art, our proposal performs faster than other approaches. For example, RPCA methods use MATLAB implementations to run in the range 9.82-476 μs/pixel [Bouwmans and Zahzah, 2014]. More complex background estimation approaches report a running time ranging from 65 to 312 μs/pixel [Xun and Huang, 2008][Colombari and Fusiello, 2010], all using MATLAB. The implementation of the proposed approach is currently restricted to offline operation, however significant speedups can be achieved by using other programming languages or by parallel processing.

2.6.6 Evaluation in SBMnet2016 dataset

2.6.7 Evaluation framework

The evaluation framework used is the one defined in SBMnet[15], where videos and metrics are proposed to evaluate Background Estimation algorithms. These videos cover a wide range of BE challenges and are representative of typical indoor and outdoor scenarios. In particular, 79 videos organized in 8 categories (Basic, Intermittent Motion, Clutter, Jitter, Illumination Changes, Background Motion, Very Long and Very Short) are presented. The videos contain moving background, camera jitter, crowds, illumination changes and shadows, still or stationary objects and pedestrians with a wide range of duration. Furthermore, there are specific categories to test Background Estimation capabilities in very long and very short videos. Moreover, to evaluate the algorithm performance by comparisons between estimated and ground-truth backgrounds, the six metrics presented in Subsection 2.6.1.2 (i.e. AGE, AE, ACE, MS-SSIM, PSNR and CQM) are provided.

2.6.8 Parametrization

The proposed approach has been tested in all videos from SBMnet. We use $W = 16(32)$ as the block size for sequences lower (higher) than 400 pixels in any of the resolution dimensions. We heuristically set $k = 3$ for inter-frame differences (Motion filtering stage) and $\rho = 5$ to determine if uncertain candidates are selected (Subsection 2.5). Moreover, to operate in this dataset we have adapted our algorithm designed for relatively short sequences by estimating background of certain videos in intervals of 600 frames and then selecting the smoothest background using the energy potential (i.e. DCT coefficients with DC coefficient set to zero) as the final one. Additionally, to be able to evaluate our approach in camera jitter or background motion scenarios, we decided to avoid the application of the motion filter in order to assure the existence of candidates in areas with more than 80% of the sequence blocks (i.e. camera jitter and dynamic background).

[15] http://pione.dinf.usherbrooke.ca/

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
511	5.3709	0.0674	0.0036	0.9457	26.3268	28.3708
Blurred	2.9910	0.0169	0.0072	0.9699	30.4749	31.0951
CamouflageFgObjects	7.8394	0.0947	0.0561	0.9281	23.3538	24.0034
ComplexBackground	9.0284	0.0763	0.0185	0.9355	22.6907	23.6944
fluidHighway	9.6826	0.0479	0.0300	0.9359	25.7876	26.1176
highway	8.5028	0.0394	0.0017	0.9427	26.4460	27.6270
Hybrid	15.0338	0.2440	0.0940	0.7375	20.8193	21.5506
I_SI_01	2.2996	0.0013	0.0001	0.9879	37.8160	38.2460
IntelligentRoom	3.0674	0.0060	0.0004	0.9907	37.816	34.7119
Intersection	2.7770	0.0021	0.0000	0.9874	35.9573	36.5337
IPPR2	4.6470	0.0077	0.0008	0.9689	32.1837	32.4051
MPEG4_40	4.2090	0.0316	0.0092	0.9508	28.3177	29.6966
PETS2006	2.4113	0.0008	0.0002	0.9893	37.7516	37.9057
skating	9.8116	0.1238	0.0408	0.8365	22.4678	23.9420
streetCornerAtNight	3.1218	0.0040	0.0027	0.9761	34.8402	35.9937
wetSnow	3.3928	0.0046	0.0017	0.9575	34.1038	34.5768

Table 2.6: Results for Basic category of SBMnet2016 dataset.

Figure 2.17: Example of estimated backgrounds for Basic category of SBMnet2016 dataset. From left to right: *IPPR2*, *CamouflageFgObjects* and *Hybrid* video sequences.

2.6.9 Results in SBMnet dataset

2.6.9.1 Basic

The results from the video sequences of this category are presented in Table 2.6. The proposed algorithm, RMR, is able to correctly estimate the background in almost all sequences (see *IPPR2* estimated background in Figure 2.17), achieving high performance in most of them. However, there are some errors that decrease performance, e.g. in *CamounflageFgObjects*, *ComplexBackground*, *Hybrid* and *skating* video sequences (see artifacts in *CamouflageFgObjects* and *Hybrid* estimated backgrounds in Figure 2.17). In these sequences foreground artifacts are included into the estimated background due to better spatial continuation with their surroundings than the correct background with the used smoothness measures. Note that once there is a fail-

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
AVSS2007	9.2767	0.0663	0.0513	0.9094	20.3096	21.3404
busStation	3.1366	0.0134	0.0053	0.9631	30.3210	31.4297
Candela_m1.10	2.5884	0.0032	0.0000	0.9949	36.1408	36.1531
CaVignal	1.2354	0.0002	0.0000	0.9962	40.7606	41.2068
copyMachine	6.1841	0.0259	0.0136	0.9667	29.4652	30.4191
I_CA_01	3.2538	0.0076	0.0015	0.9655	34.1952	34.4539
I_CA_02	5.1246	0.0388	0.0212	0.9639	26.4139	27.1210
I_MB_01	3.0618	0.0033	0.0011	0.9859	34.2275	35.0617
I_MB_02	3.4959	0.0050	0.0027	0.9840	34.1510	34.7315
office	10.4576	0.0685	0.0155	0.9716	25.8505	26.6531
sofa	2.2413	0.0034	0.0016	0.9922	36.8434	37.1986
streetCorner	5.2663	0.0156	0.0061	0.9818	29.3515	30.2434
Teknomo	5.7299	0.0363	0.0063	0.9716	26.8961	27.9404
trampstop	4.0753	0.0184	0.0009	0.9884	31.2114	31.8158
UCF-traffic	1.9553	0.01153	0.0058	0.9654	32.6634	34.997
Uturn	2.6871	0.0234	0.0131	0.9680	29.3923	30.0893

Table 2.7: Results for Intermittent Motion category of SBMnet2016 dataset.

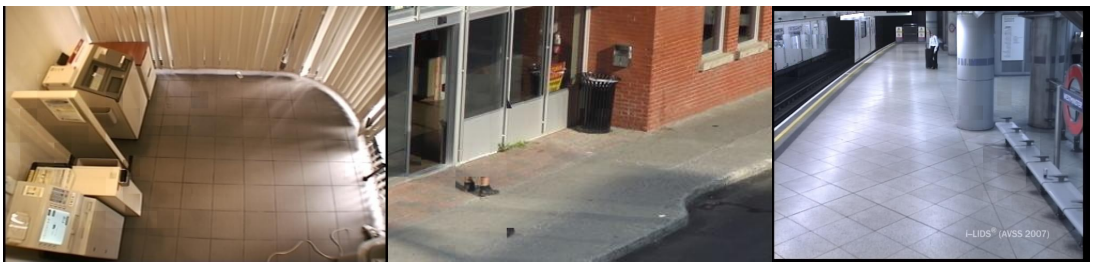


Figure 2.18: Example of estimated backgrounds for Intermittent Motion category of SBMnet2016 dataset. From left to right: *copyMachine*, *busStation* and *AVSS2007* video sequences.

ure, it could be propagated by selecting erroneous neighboring candidates with good spatial continuation with such failure.

2.6.9.2 Intermittent Motion

In Table 2.7 results from this category are presented. Intermittent Motion category contains sequences with stationary objects. The criterion assumed is that the background is not represented by objects that arrive or leave during the analyzed sequence. This assumption is handled by considering that a background representation is spatially smoother with its surroundings than an object. In general, backgrounds for all sequences are correctly generated (see example of *copyMachine* estimated background in Figure 2.18), in spite of some small artifacts, e.g. *busStation*, *I_CA_02* and *office* (see *busStation* estimated background in Figure 2.18), due to

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
Board	7.0139	0.0401	0.0079	0.8337	28.3130	29.3061
boulevardJam	4.8947	0.0388	0.0128	0.9282	29.2511	30.5310
Crowded	7.9463	0.0574	0.0341	0.9423	27.5218	28.7301
Foliage	49.6680	0.7049	0.5006	-0.0870	12.1684	12.9637
groupCampus	22.6134	0.2826	0.1673	0.4652	16.0017	16.8798
HumanBody2	13.6012	0.1138	0.0731	0.7685	16.9066	18.0912
ICRA3	10.1774	0.1915	0.1633	0.9019	21.7150	22.5227
IndianTraffic3	3.3920	0.0361	0.0240	0.9417	30.6872	32.1454
People&Foliage	33.3270	0.33132	0.2906	0.5342	12.2052	13.384
tramway	14.7508	0.2043	0.0482	0.8268	19.4502	20.9041
UCF-fishes	1.0457	0.0005	0.0002	0.9806	42.7483	44.4171

Table 2.8: Results for Clutter category of SBMnet2016 dataset.

Figure 2.19: Example of estimated backgrounds for Clutter category of SBMnet2016 dataset. From left to right: *IndianTraffic3*, *Foliage* and *Board* video sequences.

the smoothness constraints applied. However, despite obtaining good background estimations in *AVSS2007* sequence (see *AVSS2007* estimated background in Figure 2.18), performance metrics experiment a decrease. This may occur due to a ground-truth image that includes a background representation that is never visualized during the video sequence. This situation could only be handled by inpainting algorithms.

2.6.9.3 Clutter

Results from Clutter category are presented in Table 2.8. RMR has weak performance in many sequences of this category due to failures in the motion analysis (see Figure 2.19). The original algorithm [Ortego et al., 2016a] discarded all motion blocks as it considered that motion is part of the foreground. However, as presented in Subsection 2.4, if high motion activity is detected blocks from all frames are analyzed in order to handle dynamic background and camera jitter situations. This fact leads to the generation of many background candidates, thus including more distractors than in the original configuration used in [Ortego et al., 2016a], where they were automatically discarded, leading to estimated backgrounds with artifacts (see

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
badminton	8.4681	0.1227	0.0811	0.7365	23.8541	24.7652
boulevard	13.4511	0.1842	0.0566	0.8198	19.5784	21.0043
CMU	6.9662	0.0781	0.0051	0.9742	25.5719	26.4675
I_MC_02	15.4106	0.1895	0.0785	0.7366	18.6677	19.9169
I_SM_04	5.1498	0.0426	0.01121	0.9670	25.0802	26.200
O_MC_02	12.0355	0.1476	0.0470	0.8010	20.9252	21.8259
O_SM_04	8.3818	0.0879	0.0073	0.9254	25.0287	26.2648
sidewalk	25.3898	0.3482	0.2089	0.3395	15.3927	17.0788
traffic	9.1391	0.1202	0.0656	0.7254	25.1776	26.1749

Table 2.9: Results for Jitter category.

Figure 2.20: Example of estimated backgrounds for Jitter category of SBMnet2016 dataset. From left to right: *badminton, traffic* and *sidewalk* video sequences.

IndianTraffic3 estimated background in Figure 2.19). Additionally, as previously mentioned in Basic category, once that foreground objects are erroneously selected as background the error may be propagated, as it occurs in *Foliage*, *groupCampus*, *HumanBody2* and *People&Foliage* (see *Foliage* estimated background in Figure 2.19). Moreover, it is important to highlight that despite correctly estimating the background for *Board* sequence (see estimated background in Figure 2.19), metrics experiments an unusual performance decrease that may be induced by an erroneous ground-truth or metrics weaknesses.

2.6.9.4 Jitter

The results obtained for sequences with camera jitter are presented in Table 2.9. Backgrounds estimated by RMR almost eliminate all foreground objects, including few artifacts. However, due to camera motion, backgrounds are generated with neighboring candidates from different temporal instants (see *badminton* and *traffic* estimated backgrounds in Figure 2.20), thus leading to performance decreases when comparing with ground-truth. Figure 2.20 shows the estimated background of *sidewalk*, a scenario where selecting displaced background blocks leads to weak performance due to high differences with the background in areas of high background contrast.

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
CameraParameter	1.2104	0.0004	0.0000	0.9910	41.7392	42.5813
cubicle	7.4869	0.0547	0.0210	0.9685	26.0071	26.8601
Dataset3Camera1	7.7915	0.0405	0.0112	0.9162	28.0960	29.1434
Dataset3Camera2	7.1402	0.0365	0.0034	0.9310	28.0124	29.0944
I_IL_01	9.7467	0.1207	0.1000	0.9437	25.4816	26.6659
I_IL_02	9.7454	0.1407	0.1064	0.9408	23.4406	24.4845

Table 2.10: Results for Illumination Changes category of SBMnet2016 dataset.

Figure 2.21: Example of estimated backgrounds for Jitter category of SBMnet2016 dataset. From left to right: *CameraParameter*, *Dataset3Camera1* and *Dataset3Camera2* video sequences.

2.6.9.5 Illumination Changes

Results obtained for Illumination Changes category are presented in Table 2.10. Backgrounds are correctly estimated as the illumination does not impact the background smoothness criterion. However, in cases of global illumination changes, the representation chosen depends on the seeds selected, i.e. in the temporary dominant illumination (see *CameraParameter* estimated background in Figure 2.21, that is reconstructed with switched off lights and not with switched on). Moreover, for local illumination changes, there is block effect in the estimated backgrounds as sometimes different illumination representations are selected in neighboring locations, e.g. in *Dataset3Camera1* and *Dataset3Camera2* (see both estimated backgrounds in Figure 2.21).

2.6.9.6 Background Motion

Sequences containing dynamic background are compiled in Background Motion category and their results for RMR are presented in Table 2.11. Dynamic background areas are correctly reconstructed as reconstruction errors are not related with background motion (see *fountain01* estimated background in Figure 2.22), but with foreground objects that experiment better continuation with neighboring locations than background representation, e.g. in *overpass* sequence there is good continuation of a green foreground object with green trees (see *overpass* estimated background in Figure 2.22). Moreover, it is important to highlight that performance decrease in *fall* sequence may be related with errors in the ground-truth image, as it is correctly estimated

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
advertisementBoard	1.5575	0.0015	0.0000	0.9964	39.4799	39.7909
canoe	19.5947	0.3265	0.0856	0.6064	17.9059	18.4874
fall	23.0484	0.3225	0.0920	0.7611	16.2215	17.2962
fountain01	7.1950	0.0821	0.0179	0.9022	23.9172	25.2021
fountain02	7.3068	0.0654	0.0082	0.9278	26.7854	27.5155
overpass	15.0570	0.2111	0.0746	0.6966	19.1595	19.9321

Table 2.11: Results for Background Motion category.

Figure 2.22: Example of estimated backgrounds for Background Motion category of SBMnet2016 dataset.

(see estimated background in Figure 2.22).

2.6.9.7 Very Long

Results obtained by RMR for very long video sequences are presented in Table 2.12. RMR estimates correctly most of the backgrounds, except for Terrace sequence where foreground is erroneously selected as background due to similar color with its surroundings (see *BusStopMorning* and *Terrace* estimated backgrounds in Figure 2.23). Despite introducing only a small artifact for *PedAndStorrowDrive* (see its estimated background in Figure 2.23), there is a considerable performance decrease that may occur due to differences in the location of shadows inherent to the scene, as their location vary along time.

2.6.9.8 Very Short

Results obtained by RMR for very short video sequences are presented in Table 2.13. RMR estimates correctly most of the backgrounds (see *peopleInShade* estimated background in Figure 2.24), however there are some performance decreases in *Toscana*, *DynamicBackground* and *TwoLeaveShop1cor* sequences due to artifacts (see both estimated backgrounds in Figure 2.24). This artifacts are included due to the failure of background smoothness in some locations.

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
BusStopMorning	5.9804	0.0372	0.0010	0.9793	28.4728	29.1260
Dataset4Camera1	4.9501	0.0097	0.0005	0.9920	31.4085	32.0248
PedAndStorrowDrive	26.0220	0.5813	0.3349	0.7422	18.2712	19.7232
PedAndStorrowDrive3	4.2348	0.0398	0.0040	0.9779	28.7352	29.7659
Terrace	25.0421	0.6056	0.4450	0.8662	18.7667	19.7293

Table 2.12: Results for Very Long category of SBMnet2016 dataset.

Figure 2.23: Example of estimated backgrounds for Very Long category of SBMnet2016 dataset.

2.7 Conclusions

We presented a block-level BE approach to estimate the background of video sequences with moving and stationary objects. A clustering approach without the need of thresholds is performed over motion-filtered and dimension reduced data, which determines the candidates blocks to be background. Subsequently, a *Rejection based Multipath Reconstruction* based on background smoothness constraints selects the most suitable candidate. This multipath scheme includes a *Seed Selection* stage to initially estimate the background which is locally reconstructed using different paths (hypotheses), thus increasing the robustness against errors. An evaluation metric based on a sweep of threshold values is proposed to avoid the threshold dependency of existing metric AE. The experiments validate the performance of the clustering analysis and the *Seed Selection* technique and provide comparisons against related work, demonstrating the advantages of the proposed approach. The results show that our spatial strategy is effective to operate in presence of stationary objects, but may not be the best choice under challenges such as illumination changes, dynamic backgrounds or very long sequences, where temporal strategies may be preferred. Regarding the evaluation in the SBMnet2016 dataset, we had to adapt our algorithm to assure operation under different challenges. However, this change substantially degraded the performance in clutter and background motion categories.

	AGE	AE	ACE	MS-SSIM	PSNR	CQM
Toscana	6.9606	0.0554	0.0359	0.8981	22.9475	23.7080
CUHK_square	6.7243	0.0662	0.0074	0.9359	25.3190	26.4075
DynamicBackground	14.5281	0.2038	0.0400	0.8664	19.8128	20.6954
MIT	6.5953	0.0680	0.0092	0.9417	25.6235	26.9524
NoisyNight	6.4003	0.0382	0.0095	0.8929	27.5206	28.7724
pedestrians	1.6400	0.0004	0.0000	0.9951	39.7712	40.1219
peopleInShade	9.2772	0.0668	0.0166	0.9568	26.8045	27.7174
snowFall	2.5053	0.0008	0.0001	0.9538	36.9386	37.4674
TownCentre	4.5600	0.0156	0.0028	0.9616	30.4174	31.0549
TwoLeaveShop1cor	4.8674	0.0256	0.0155	0.9269	25.6549	26.3624

Table 2.13: Results for Very Short category.

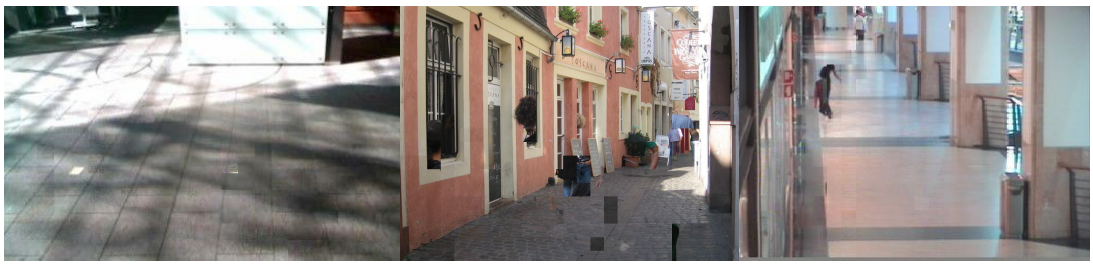


Figure 2.24: Example of estimated backgrounds for Very Short category of SBMnet2016 dataset.

Chapter 3

Background updating for stationary object detection

3.1 Introduction[1]

Stationary object detection (SOD) has recently experienced extensive research [Fan et al., 2013] due to its contribution to prevent terrorist attacks by detecting abandoned objects [Lin et al., 2015] and illegal parked vehicles [Albiol et al., 2011]. SOD aims to detect the objects in the scene that remain stationary after previous motion. Typically, a background subtraction (BS) algorithm extracts the objects and SOD decides whether they are stationary or not [Bayona et al., 2009]. However, current BS algorithms present many shortcomings to label foreground and background regions in real situations [Bouwmans, 2014], thus highly determining the SOD accuracy.

Recent SOD approaches employ different strategies based on BS. Whilst temporal accumulation of foreground masks [Guler et al., 2007] is widely used, post-processing [Pan et al., 2011][Kim and Kim, 2014] and combination [Ortego and SanMiguel, 2014] of additional features are required to address BS limitations in presence of crowds and illumination challenges. Temporal sampling of foreground and motion masks enables operation in complex scenes [Bayona et al., 2010], being the sample selection crucial for the detection accuracy. Dual BS approaches rely on fast and slow updated BS algorithms to identify the stationary objects [Porikli et al., 2008][Lin et al., 2015]. However, BS failures require additional post-processing, such as edge features or fast-slow model interaction [Evangelio and Sikora, 2011], which avoids the detection of background objects that are removed from the scene. Other approaches take advantage of multilayer BS algorithms to model moving objects, stationary objects and background [YingLi et al., 2011]. Nevertheless, the validation of the detected candidates via patch features [Fan

[1]This chapter is an adapted version of the publication [Ortego et al., 2015]

et al., 2013] and edge features [Szwoch, 2014] is again needed to handle BS errors. Moreover, [Albiol et al., 2011] detects parked vehicles over time using stable keypoints instead of BS. Many SOD challenges addressed in previous research are related to BS difficulties with illumination changes, crowds, intermittent object motion and required temporal adaptation. These aspects are pivotal to transfer SOD research to real situations, where long-term operation may be required. The increasing interest in long-term operation is reflected in recent works for abandoned object detection [Fan et al., 2013] and vehicle tracking [Fan et al., 2014], where BS limitations are addressed to reduce the high number of false alarms.

This chapter proposes a SOD approach for long-term operation that has three main contributions. Firstly, it does not use BS to perform SOD, thus not being constrained to BS limitations. Secondly, the scene is modeled by an Online Block Clustering approach that describes the stationarity of the scene, i.e. the background. The proposed approach is robust to illumination changes and quickly adapts to scene variations while identifying the stationary objects. Finally, few parameters are needed to operate with the proposed approach, unlike most of the state-of-the-art where handling stationary objects with BS introduces many parameters and thresholds. We validate the proposed approach for short-term and long-term scenarios, outperforming the state-of-the-art results.

This reminder of this chapter is organized as follows: Section 3.2 overviews the proposed approach whereas Section 3.3 and 3.4 describe the clustering and the stationary detection. Section 3.5 presents the experiments. Finally, Section 3.6 concludes this chapter.

3.2 Overview

The proposed approach detects stationary objects without using BS (see Figure 3.1). A block-level online clustering of the scene detects spatio-temporal stability changes (i.e. variations over time of the most stable scene representation or background in each block location) at regular sampling instants. Those changes are exploited to identify stationary objects. Firstly, a *Block Division* stage decomposes each frame $\mathcal{I}_t$ into non-overlapping $W \times W$ blocks $R_t^{\mathbf{s}}$ at each instant t, where $\mathbf{s}$ denotes the block location. Secondly, an *Online Block Clustering* stage (see Section 3.3) models each location $\mathbf{s}$ over time, updating a cluster partition $\mathbb{P}^{\mathbf{s}}$. This stage handles the temporal adaptation to scene changes, by assigning each incoming block $R_t^{\mathbf{s}}$ to one cluster of the partition $\mathbb{P}^{\mathbf{s}}$ or creating a new one. Only stationary blocks $R_t^{\mathbf{s}}$ (i.e., without motion with respect to $R_{t-1}^{\mathbf{b}}$) are analyzed at this stage. This clustering provides robustness against illumination changes by considering pixel ratios at block level which groups blocks even if their illumination has changed. Finally, a *Stationary Block Detection* stage (see Section 3.4) outputs a result image $\mathcal{D}_b$ with stationary objects, where b defines the sampling instant each k frames. Data associated to the last stable cluster $S^{\mathbf{s}}$, old stable clusters $\mathbb{O}^{\mathbf{s}}$ and the alarm

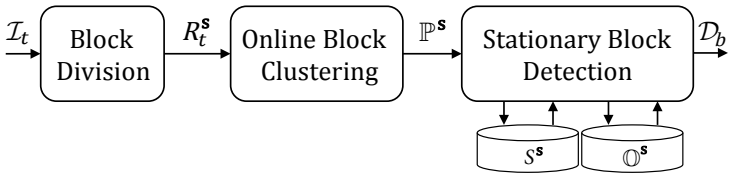


Figure 3.1: Block diagram of the proposed approach.

time T is used to respectively detect the spatio-temporal stability changes, discard those changes caused by previously visualized clusters (i.e. the empty scene or a previous detection) and detect stationarity for changes longer than the alarm time. This last stage improves the state-of-the-art by reducing false alarms due to intermittent object motion and allowing to detect stationarity for objects not fully visible during T. Figure 3.2 presents an example of the scene analysis.

3.3 Online Block Clustering

Once each $\mathcal{I}_t$ is divided into non-overlapping blocks $R_t^{\mathbf{s}}$, the Online Block Clustering models the temporal scene evolution by grouping similar blocks over time into clusters. Thus, cluster partitions $\mathbb{P}^{\mathbf{s}} = \left\{C_q^{\mathbf{s}}\right\}_{q=1:|\mathbb{P}^{\mathbf{s}}|}$ are created for each block location $\mathbf{s}$, where $C_q^{\mathbf{s}}$ is the block representing each cluster and $|\cdot|$ denotes the cardinal. For a more readable notation, we omit the index $\mathbf{s}$ since the clustering operations apply to the same location.

As the target is to identify stationarity, blocks containing moving objects are not necessary. Therefore, matching between blocks from consecutive frames is first performed to discard each incoming R_t not matching R_{t-1}. For each R_t without motion, this stage determines the matches with existent clusters from $\mathbb{P}$. Each cluster models a spatio-temporal scene pattern and it is described by the first instant of visualization f_q, last instant of visualization l_q, repeatability w_q and the cluster representative C_q. To update $\mathbb{P}$, if no matching is found a new cluster with representative $C_{q\prime} = R_t$ is created, where $q\prime = |\mathbb{P}| + 1$. On the contrary, a match exists and an existent cluster is updated applying a cumulative moving average:

$$C_q = \frac{C_q^{match} \cdot w_q + R_t}{w_q + 1}, \tag{3.1}$$

where C_q^{match} is the cluster matching R_t and w_q is the block repeatibility computed as $w_q = f\left(C_q, R_{t-T:t}\right)$. This function $f\left(\cdot\right)$ is computed online without keeping every R_t and counts each matching between C_q and R_t, also decreasing the w_q value in each sampling instant b to reduce the contribution of old visualizations. For example, given a sampling t, such decrease removes from the repeatability old visualizations summed before the instant $t-T$, i.e. older than the alarm time T from the current instant t, thus facilitating a fast adaptation for long-term operation. Hence, $R_{t-T:t}$ depicts that visualizations from $t-T$ to current instant t guide the

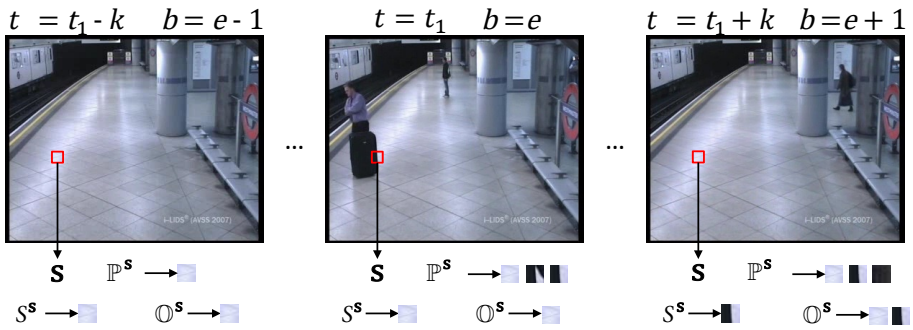


Figure 3.2: Example of the temporal analysis for a block location $\mathbf{b}$ where the stability is modified changing from the empty scene to a suitcase. $\mathbb{P}^{\mathbf{s}}$ keeps clusters from $\mathbf{s}$, while $S^{\mathbf{s}}$ and $\mathbb{O}^{\mathbf{s}}$ keep, respectively, the information from the last stable cluster and old stable clusters. Additionally, the relation between the temporal index t and the sampling instant b is shown.

computation of the w_q associated to C_q, operation that is performed online and without buffer. Occasionally, R_t can match different members of $\mathbb{P}$ and C_q^{match} is selected as:

$$C_q^{match} = \underset{\forall C_q \in \mathbb{C}}{argmax}\ f\left(C_q, R_{t-T:t}\right), \tag{3.2}$$

where $\mathbb{C} = \{C_q : C_q\ matches\ R_t\ \forall q\}$. Thus, the selected match with C_q, i.e. C_q^{match} to use in Eq. 3.1, has the highest repeatability w_q. Furthermore, in each sampling instant $\mathbb{P}$ is pruned to keep the z most visualized clusters in order to speedup the matching search.

3.3.1 Matching metric

We determine the distance between two blocks R and R' based on pixel ratios, which are known to be robust against illumination for motion detection [Pilet et al., 2008][Wu et al., 2005]. Inspired by these works, we define the ratio between two RGB pixels of R and R' as the maximum of their three-channel pixel ratio:

$$r_{max}\left(R^{\mathbf{p}}, R'^{\mathbf{p}}\right) = max\left\{r_i\left(R_i^{\mathbf{p}}, R_i'^{\mathbf{p}}\right)\right\}_{i=R,G,B}, \tag{3.3}$$

where $\mathbf{p}$ denotes a pixel location and r_i is the pixel ratio of each image channel i. The ratio of each channel is:

$$r_i\left(R_i^{\mathbf{p}}, R_i'^{\mathbf{p}}\right) = 1 - \frac{min\left\{R_i^{\mathbf{p}}, R_i'^{\mathbf{p}}\right\} + m}{max\left\{R_i^{\mathbf{p}}, R_i'^{\mathbf{p}}\right\} + m}, \tag{3.4}$$

being m a correction constant [Pilet et al., 2008] to manage the ratio instability in low intensity values. Unlike [Pilet et al., 2008], when comparing two pixel values we divide the minimum

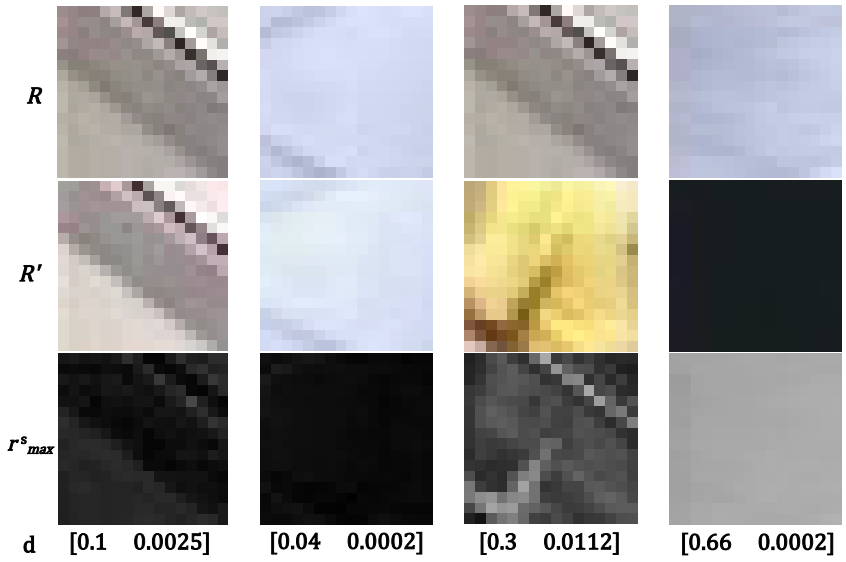


Figure 3.3: Ratio of blocks R and R', where $r^{\mathbf{s}}_{max}$ is the pixel ratio for every pixel $\mathbf{p} \in R$. The higher the difference, the brighter the pixel ratio. First and second columns are examples of equal blocks where $\mu(R, R')$ and $\sigma^2(R, R')$ have low values. Third and fourth columns are different blocks where $\mu(R, R')$ has a high value, while $\sigma^2(R, R')$ is high (low) in the third (fourth) column due to a heterogeneous (homogeneous) change.

between the maximum value in order to obtain $r^{\mathbf{p}} \in [0, 1]$. Thus, $r^{\mathbf{p}} = 0\,(1)$ means maximum (minimum) pixel similarity.

To model the block as a whole, we use a feature vector $\mathbf{d}$ composed of mean μ and variance σ^2 of the pixel ratio:

$$\mu(R, R') = \frac{1}{|R|} \sum_{\mathbf{p} \in R} r_{max}(R^{\mathbf{p}}, R'^{\mathbf{p}}), \tag{3.5}$$

$$\sigma^2(R, R') = \frac{1}{|R| - 1} \sum_{\mathbf{p} \in R} \left(r_{max}(R^{\mathbf{p}}, R'^{\mathbf{p}}) - \mu(R, R')\right)^2. \tag{3.6}$$

Using mean and variance allows measuring, respectively, the intensity and heterogeneity of the variations between the blocks R and R'. The higher (lower) the intensity change is, the higher (lower) the $\mu(R, R')$ will be, while the higher (lower) the heterogeneity in the change is, the higher (lower) $\sigma^2(R, R')$ will be. This behavior is depicted in Figure 3.3. The matching between R and R' is modeled by a pre-trained SVM based on the two-dimensional feature vector $\mathbf{d}$. Thus, for the Online Block Clustering, $\mu(R_t, R_{t-1})$ and $\sigma^2(R_t, R_{t-1})$ are computed to match R_t and R_{t-1} when discarding motion blocks, while $\mu(R_t, C_q)$ and $\sigma^2(R_t, C_q)$ are obtained to associate a non-moving R_t to any of the C_q members of $\mathbb{P}$.

3.4 Stationary Block Detection

This stage analyzes stability in regular sampling instants (i.e. each k frames) to identify stationarity. To that end, data associated to old stable clusters $\mathbb{O}^{\mathbf{s}}$ and the last stable cluster $S^{\mathbf{s}}$ in each block location is kept. The former contains the clusters generating stationary detections, i.e. $\mathbb{O}^{\mathbf{s}} = \{O_h^{\mathbf{s}}\}_{h=1:|\mathbb{O}^{\mathbf{s}}|}$, while the latter has the last stable cluster that either induced stationarity or was an old visualization.

In each sampling instant b, a sequence of operations is performed (summarized in Figure 3.4) to determine stationarity. First, the most stable cluster from $\mathbb{P}^{\mathbf{s}}$, $C_b^{\mathbf{s}}$, is obtained:

$$C_b^{\mathbf{s}} = \underset{C_q^{\mathbf{s}} \in \mathbb{P}^{\mathbf{s}}\, \forall q}{argmax}\ w_q^{\mathbf{s}}. \tag{3.7}$$

Therefore, selecting a $C_b^{\mathbf{s}}$ for each spatial location $\mathbf{s}$ reveals the stability of the scene or background. Subsequently, the occurrence of the spatio-temporal stability change is verified by comparing first visualization instant from $C_b^{\mathbf{s}}$ and $S^{\mathbf{s}}$, i.e. $f_b^{\mathbf{s}}$ and $f_*^{\mathbf{s}}$:

$$\mathcal{T}_b^{\mathbf{s}} = \begin{cases} 1 & if \quad f_b^{\mathbf{s}} \neq f_*^{\mathbf{s}} \\ 0 & otherwise \end{cases}, \tag{3.8}$$

where $\mathcal{T}_b^{\mathbf{s}}$ denotes whether a stability change is occurring (1) or not (0). Note that the first instant of visualization is sufficient to verify equality between clusters, as it is an exclusive cluster footprint at each block location $\mathbf{s}$. Then, the buffer $\mathbb{O}^{\mathbf{s}}$ is consulted to determine if $C_b^{\mathbf{s}}$ was previously seen:

$$\mathcal{N}_b^{\mathbf{s}} = \begin{cases} 1 & if \quad C_b^{\mathbf{s}} \notin \mathbb{O}^{\mathbf{s}} \\ 0 & otherwise \end{cases}, \tag{3.9}$$

where $\mathcal{N}_b^{\mathbf{s}}$ determines whether $C_b^{\mathbf{s}}$ is a new stable cluster (1) or not (0). The matching measure from Section 3.3 is used for this task, i.e. $\mu\left(C_b^{\mathbf{s}}, O_h^{\mathbf{s}}\right)$ and $\sigma^2\left(C_b^{\mathbf{s}}, O_h^{\mathbf{s}}\right)$ are computed for every h to classify $C_b^{\mathbf{s}}$ as equal or different to any $O_h^{\mathbf{s}}$. Furthermore, in case of been and old cluster ($\mathcal{N}_b^{\mathbf{s}} = 0$), $S^{\mathbf{s}}$ is updated with $C_b^{\mathbf{s}}$ and $f_b^{\mathbf{s}}$ to be the last stable cluster from $\mathbf{s}$. When $\mathcal{N}_b^{\mathbf{s}} = 1$, the stationary detection is performed as:

$$\mathcal{L}_b^{\mathbf{s}} = \begin{cases} 1 & if \quad (\mathcal{T}_b^{\mathbf{s}} = 1) \wedge (\mathcal{N}_b^{\mathbf{s}} = 1) \wedge \\ & (lifetime \geq T) \\ 0 & otherwise \end{cases}, \tag{3.10}$$

where $lifetime = l_b^{\mathbf{s}} - f_b^{\mathbf{s}}$ is the amount of frames from the first to the last visualization of $C_b^{\mathbf{s}}$ and $\mathcal{L}_b^{\mathbf{s}} = 1$ denotes stationarity ($\mathbf{s}$ contains a new stable cluster). In case of stationarity, $C_b^{\mathbf{s}}$,

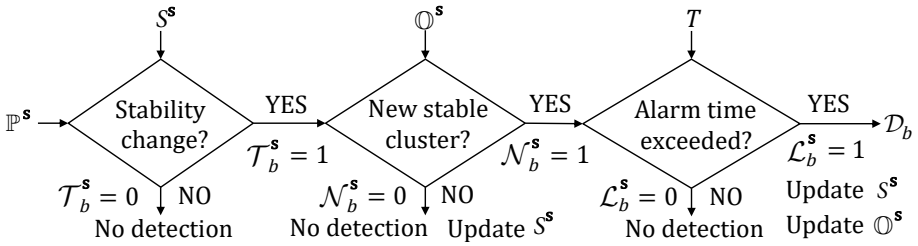


Figure 3.4: Stationary Block Detection. Sequence of operations to determine stationarity in a b sampling instant.

$f_b^{\mathbf{s}}$, $l_b^{\mathbf{s}}$ and $w_b^{\mathbf{s}}$ become the last stable representation $S^{\mathbf{s}}$ and they are included in $\mathbb{O}^{\mathbf{s}}$ as an old visualized cluster.

As stationary objects may spread across several blocks, detections of neighboring locations are associated via connected component analysis. They are visualized as bounding boxes in the result image $\mathcal{D}_b$ during a user-defined time.

3.5 Experimental Results

3.5.1 Setup

We use short sequences for evaluation from AVSS07[2] and PETS06[3] datasets as in other works [Porikli et al., 2008][Kim and Kim, 2014]. The *PV_M* sequence from AVSS07 is not included as it contains camera jitter and the Online Block Clustering filters every motion block. Moreover, long sequences from CUHK[4], VIRAT[5], IDIAP Traffic Junction[6] and AVSS2007 datasets have been also used for long-term test conditions. CUHK and IDIAP datasets contain one sequence each, while we have merged all the short clips from VIRAT which are continuous in time and useful for SOD, conforming 4 sequences. Overall, 364951 frames (∼4.05 h) have been tested and we have visually identified 51 abandoned objects and stopped vehicles as ground-truth[7]. We assume that people are not considered neither false or correct detections as in [Pan et al., 2011][YingLi et al., 2011] and previous visualization of the empty scene due to the long-term focus of the proposed approach.

To evaluate the results, we use standard Precision $P = TP/(TP + FP)$, Recall $R = TP/(TP + FN)$ and F-score $F = 2 \cdot P \cdot R/(P + R)$ measures, where TP, FP and FN denote, respectively, correct, false and missed detections.

For the proposed approach, we use W=16 (16×16 blocks) and k=50 frames (sampling rate).

[2] http://www.avss2007.org/
[3] http://www.cvg.reading.ac.uk/PETS2006/data.html
[4] http://www.ee.cuhk.edu.hk/∼xgwang/CUHK_square.html
[5] http://www.viratdata.org/
[6] http://www.idiap.ch/∼odobez/datasets.php
[7] http://www-vpu.eps.uam.es/publications/SOD_STSC/

Algorithm		Short-term									Long-term									
		AVSS07					PETS06	Mean			AVSS07	IDIAP	VIRAT				CUHK	Mean		
		AB_E	*AB_M*	*AB_H*	*PV_E*	*PV_H*	*Cam3*	P	R	F	*AB_EV*	*I_1*	*V_1*	*V_2*	*V_3*	*V_4*	*C_1*	P	R	F
sACC	GT/TP/FP	1/1/0	1/1/6	1/1/3	1/1/5	1/1/10	1/1/0	.20	1	.33	5/5/21	7/7/10	3/3/8	3/3/10	4/4/10	15/15/6	8/8/20	.35	1	.51
mACC	GT/TP/FP	1/1/0	1/1/0	1/1/0	1/1/0	1/0/2	1/1/0	.71	.83	.77	5/5/5	7/7/2	3/3/6	3/3/6	4/4/9	15/15/0	8/8/10	.54	**1**	.70
TS	GT/TP/FP	1/1/0	1/1/2	1/1/1	1/1/4	1/1/10	1/1/0	.26	1	.41	5/5/12	7/7/10	3/3/8	3/3/10	4/4/10	15/15/6	8/8/20	.37	1	.54
DB	GT/TP/FP	1/1/1	1/1/4	1/1/3	1/1/1	1/1/10	1/1/0	.24	1	.39	5/4/25	7/7/6	3/3/0	3/3/0	4/4/0	15/13/3	8/6/7	.49	.89	.63
Proposed	GT/TP/FP	1/1/0	1/1/0	1/1/0	1/1/0	1/1/0	1/1/0	**1**	**1**	**1**	5/5/3	7/6/4	3/3/3	3/3/6	4/4/4	15/14/0	8/8/2	**.66**	.96	**.78**

Table 3.1: Comparative evaluation. GT, TP and FP denote, respectively, ground-truth, correct and false detections. The proposed approach achieves best results (bold) in short-term and long-term sequences.

The lower (higher) the sampling rate is the lower (higher) the delay in the detections is, thus its value does not have a significant impact on performance. For the online clustering (Section 3.3), we set m=64 as in [Pilet et al., 2008] and z=3 to keep at most 3 clusters for each location $\mathbf{s}$ after each sampling instant. The SVM to match blocks is trained with a balanced set of about 2000 positive and negative samples, i.e. 2000 labeled block comparisons randomly collected from different sequences. Finally, the alarm time T is set to 30 seconds for PETS06 (ground-truth value), 1 minute for AVSS07 (minimum value of the ground-truth ones) and 1 minute for the rest of the datasets without annotations.

3.5.2 Comparative evaluation

The proposed approach is compared with four state-of-the-art approaches: temporal accumulation for single [Guler et al., 2007] (sACC) and multiple [Ortego and SanMiguel, 2014] (mACC) features, temporal sampling [Bayona et al., 2010] (TS) and dual background [Porikli et al., 2008] (DB). These approaches are tested both in short-term and long-term sequences. It is fair to mention that other state-of-the-art approaches [Pan et al., 2011][Kim and Kim, 2014] report similar results in the short-term sequences, however as there is no available software to analyze the long-term ones, they have not been included.

The left part of Table 3.1 shows the results for short-term sequences. The proposed approach detects all objects without false positives, thus demonstrating the robustness in crowded situations such as *AB_M*, *AB_H* and *PV_H*, where illumination changes and cast shadows take place. This robustness against crowds is due to the Online Block Clustering that models the stationarity while discarding moving appearances, thus preventing from alarms triggered by moving crowds. However, the state-of-the-art approaches have serious difficulties to cope with moving crowds (except [Ortego and SanMiguel, 2014]) as continuously detecting foreground in the same spatial location produces false positives.

The right part of Table 3.1 shows the results for long-term sequences which pose additional challenges besides crowds or illumination changes, such as temporal adaptation and correctly handling intermittent object motion. The results show that the proposed approach outper-

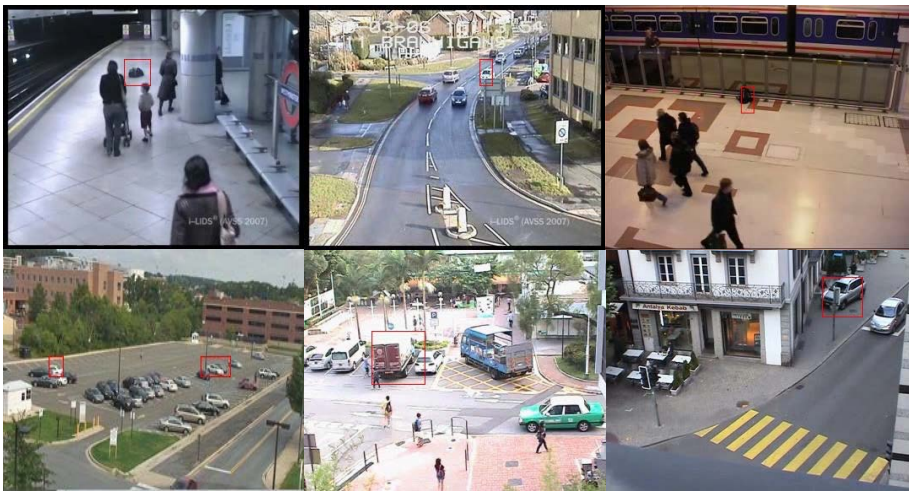


Figure 3.5: Examples of detections in each dataset. First row, from left to right: *AB_H*, *PV_H* and *C_1*. Second row, from left to right: *I_1*, *Cam3* and *V_4*.

forms the state-of-the-art, specially in sequences which contain intermittent object motion and crowded situations, such as *C_1* and *AB_EV*. This is due to the ability of the Online Block Clustering to perform a fast and illumination-robust adaptation and due to the capability of the Stationary Block Detection stage to identify stationary objects while avoiding the intermittent object motion issue by the buffer $\mathbb{O}^{\mathbf{s}}$. However, the proposed approach fails in few situations, due to uncovered regions of the empty scene where its appearance changes from the one in $\mathbb{O}^{\mathbf{s}}$ and due to camouflage effects. An example of the proposed approach detections is shown in Figure 3.5.

The computational cost of the proposed approach is mainly due to the clustering stage since the stationary detection performs simple operations each k frames. A non-optimized MATLAB implementation runs at 5 *fps* on a standard PC (P-IV 2.8 GHz and 2 GB RAM). Regarding the state-of-the-art [Guler et al., 2007][Porikli et al., 2008][Bayona et al., 2010][Ortego and SanMiguel, 2014], the overall cost depends on BS where recent MATLAB implementations are in the range 8-10 *fps* [Chen and Ellis, 2014][Seo and Kim, 2014].

3.6 Conclusions

This chapter proposes a SOD approach suitable for long-term operation due to its robustness to crowds, illumination changes and intermittent object motion. The proposed approach presents a new strategy to identify stationarity by detecting spatio-temporal stability changes in the scene. The proposed approach employs Online Block Clustering robust to illumination changes, being able to distinguish between equal and different spatial representations of the scene over time. Future work will mainly explore dynamic block size and additional features for the clustering.

Part III

Foreground segmentation

Chapter 4

Foreground segmentation quality

4.1 Introduction[1]

The systematic evaluation of algorithmic performance in challenging situations has recently gained interest in computer vision [Goyette et al., 2014][Borji et al., 2015][Pont-Tuset and Marques, 2015][Cehovin et al., 2016][Lopez-Molina et al., 2016]. The algorithm results are often compared to a reference or ground-truth data, which need a time-consuming annotation process prone to human errors. To overcome these limitations, reference-free evaluation measures have been proposed for video quality estimation [Park et al., 2013], color-to-gray image conversion [Ma et al., 2015], image quality assessment [Zhang et al., 2016], image segmentation quality [Zhang et al., 2008], video object segmentation [Erdem et al., 2004], salient object detection [Mai and Liu, 2014] or video object tracking [SanMiguel and Cavallaro, 2015], to name a few. Moreover, these reference-free measures can have further uses beyond off-line evaluations such as improving run-time performance for video segmentation [Min et al., 2014], video quality [Seshadrinathan and Caviedes, 2012] and video tracking [Yuan et al., 2014].

In this context, video object segmentation is a popular low-level task in computer vision which aims to segment the objects of interest in a video sequence. In unconstrained situations [Perazzi et al., 2016], video object segmentation is cast as detecting spatio-temporal salient objects, propagating initially segmented objects or using frame-by-frame human intervention, respectively, for unsupervised, semi-supervised and supervised measures. These situations present challenges related to camera motion, shape deformations of objects or motion blur [Faktor and Irani, 2014]. However, this chapter focuses on the segmentation of foreground objects based on background subtraction [Bouwmans, 2014]. Assuming a relative control of camera motion, such objects are extracted by comparing each frame with a model of the background in the video sequence. This comparison results in binary masks describing the foreground objects for each frame.

[1]This chapter is an adapted version of the publication [Ortego et al., 2017]

The evaluation of background subtraction algorithms has been widely explored, ranging from qualitative visualization tools [Ramadan, 2006][Song et al., 2014][Sánchez Rodríguez et al., 2014] to quantitative reference-based evaluations [Nascimento and Marques, 2006][Brutzer et al., 2011][Wang et al., 2014b] that require human annotation [Cuevas et al., 2015]. Despite the plethora of existing algorithms and reference-based evaluation measures, little attention has been directed towards the reference-free or stand-alone evaluation of foreground masks. Such evaluation is a complex task that requires the estimation of the performance or quality of segmented foreground masks without using any ground-truth data nor human intervention.

Existing measures for stand-alone background subtraction evaluation provide a rough estimation of ground-truth performance (i.e. quality) rather than a fine-grained one. These measures can be categorized into: *assisted*, when other algorithms validate background subtraction performance; *specific*, when a measure is designed for a particular phenomenon that may degrade performance such as shadows or dynamic backgrounds; and *generic*, when object mask properties are exploited to estimate performance. Generic measures for stand-alone background subtraction evaluation are preferred to be independent of external algorithms or particular phenomena, unlike other stand-alone measures. However, a thorough quantitative comparison of current measures for stand-alone background subtraction evaluation has yet to be reported.

While many recent proposals for background subtraction evaluation [Wang et al., 2014b] [Cuevas et al., 2016] use ground-truth to determine frame-level performance, this chapter addresses the evaluation from a stand-alone perspective using measures computed over connected components (i.e. blobs) for the estimation of foreground segmentation masks quality. We discuss the available measures in the literature to identify the properties of high-performance foreground segmentation masks. To compare these measures, the results of eight state-of-the-art background subtraction algorithms are analyzed using the CDNET2014 dataset [Wang et al., 2014b]. We first cluster these measures according to their linear and non-linear relations using the Pearson's correlation coefficient [Pearson, 1896] and Self-Organizing Maps [Kohonen, 1982]. Then, we select the most useful measures of each cluster to analyze their capabilities for discriminating low, medium and high performance (i.e. quality levels). Finally, we explore the application of these measures to rank algorithms as compared to rankings based on ground-truth performance. To the best of our knowledge, this is the first attempt to provide a comprehensive study of stand-alone performance estimation for connected components in foreground segmentation masks (i.e. blobs) as previous works [Correia and Pereira, 2002][Erdem et al., 2004][SanMiguel and Martinez, 2010] are mainly focused on frame-level measures in simple scenarios. Such frame-level evaluations combine all blob qualities per frame, thus restricting a detailed analysis of relevant blob mask properties.

The contribution is threefold. Firstly, we survey and categorize a large set of quality estimation measures for foreground segmentation masks extracted from background subtraction,

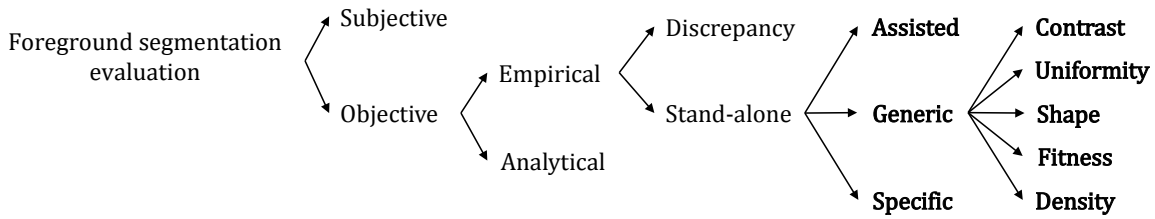


Figure 4.1: Taxonomy for evaluation measures of foreground segmentation masks. Bold denotes the categories proposed in this chapter.

video object segmentation, image segmentation, image co-segmentation and object recognition. Secondly, we provide an extensive comparison of 21 measures to analyze their strengths and weaknesses. The third contribution focuses on key properties of well segmented blob masks and analyzes their discrimination capabilities for common background subtraction challenges.

The reminder of this chapter is organized as follows: Section 4.2 overviews the related work in performance evaluation of segmentation masks, whereas Section 4.3 describes selected stand-alone quality measures. Section 4.4 presents the experimental methodology before discussing the experimental results in Section 4.5. Finally, Section 4.6 summarizes the main conclusions.

4.2 Related work

Performance evaluation of foreground masks is categorized in the state-of-the-art using an empirical-analytical taxonomy [SanMiguel and Martinez, 2010][Vojodi et al., 2013][Shi et al., 2015] that we extend in this chapter to consider stand-alone measures (see Figure 4.1).

In the literature, existing measures are frequently classified into *subjective* and *objective*, denoting whether human perception is (or not) used to quantify the performance [Villegas and Marichal, 2004]. Subjective measures [Shi et al., 2015] are sometimes approximated by objective measures [Gelasca and Ebrahimi, 2009] in order to reduce human intervention. Furthermore, the objective evaluation is divided into *analytical* and *empirical*, where the former evaluates an algorithm considering its theoretical description, requirements and complexity, while the latter uses the video properties and the algorithm results. Although there are some analytical measures [Gao et al., 2000], the evaluation in background subtraction has been mainly studied empirically in the state-of-the-art, either by using ground-truth annotated objects (discrepancy evaluation) or without any ground-truth data (stand-alone evaluation).

Regarding the use of ground-truth data, *discrepancy* evaluations assess algorithm performance through comparisons between expected and segmented foreground masks. To homogenize evaluation, benchmarks have been proposed for salient object detection [Borji et al., 2015], text detection [Veit et al., 2016], video object segmentation [Perazzi et al., 2016] and background sub-

traction [Wang et al., 2014b]. Evaluation measures used in these benchmarks can be categorized into region similarity, contour similarity and temporal stability [Perazzi et al., 2016]. Firstly, region similarity measures employ true positive, false positive, true negative and false negative pixel decisions to compare detected and ground-truth pixels. This category covers measures such as Precision, Recall, F-score [Goyette et al., 2014], Jaccard index [Everingham et al., 2010], Precision-Recall (PR) curves, Receiver Operating Characteristic (ROC) curves, Area Under the Curve (AUC) and Mean Absolute Error (MAE) [Borji et al., 2015]. Some of these measures have a strong similarity, such as F-score and Jaccard index, which are monotonically related and could be both generalized by the Tversky index [Tversky, 1977]; or PR and ROC curves whose relation is studied in [Davis and Goadrich, 2006]. Additionally, there are some efforts to analyze and overcome limitations inherent to these traditional measures by considering the inaccuracy of ground-truth masks, the location and distribution of errors and the error type itself [Liu and Sang, 2011][Lallier et al., 2011][Margolin et al., 2014]. Despite commonly reporting measures at frame or sequence level, there are evaluations based on pixel-level [Wang et al., 2014b][Perazzi et al., 2016] and blob-level [Lazarevic-McManus et al., 2008][Calarasanu et al., 2016] information. Note that introducing blob-level information requires establishing the correspondence between the segmented blobs and the ground-truth blobs [Lazarevic-McManus et al., 2008][Calarasanu et al., 2016], as many segmented blobs may belong to the same ground-truth blob. Secondly, contour similarity measures compute the accuracy of blob contours by comparing segmented and ground-truth foreground masks through a bipartite graph matching that provides robustness against small contour inaccuracies when computing F-score [Perazzi et al., 2016]. Finally, temporal stability measures penalize inter-frame variations of segmented foreground masks. Such stability is computed in [Liu and Sang, 2011] as the variation of the region similarity measures between frames, while [Perazzi et al., 2016] finds transformations of blob masks from one frame to the following and penalizes non-smooth and imprecise transformations.

Dependence on ground-truth data limits the usability of the aforementioned evaluation protocols to labeled datasets and, even more important, it relegates the performance evaluation to an off-line procedure as ground-truth is not available in real applications during running time. Therefore, in the absence of such ground-truth data (or when it is difficult to obtain it), discrepancy measures cannot be applied and stand-alone evaluation is required. This stand-alone evaluation is based on a rough estimation of ground-truth performance (i.e. quality) following different strategies. In particular, these strategies can be categorized as *assisted*, *specific* and *generic*, and they differ from traditional discrepancy measures in the sense that they can adopt different forms.

Stand-alone assisted evaluations measure segmented foreground masks quality by employing the results of external algorithms to assist the video object segmentation with confidence maps of reliable areas to segment, such as tracking algorithms assisting segmentation. Specifically,

some supervised approaches start from a manual object initialization to jointly model segmentation and tracking in the same framework using multi-part tracking information [Wen et al., 2015] or a Bayesian tracker [Salti and Stefano, 2015] as priors of high quality areas to segment. Additionally, [Ling et al., 2014] proposes a background modeling approach that improves the segmented foreground mask using feedback of tracked moving objects as quality to guide the background adaptation to scene changes depending of the tracked object type (moving, stationary or background) and the foreground segmentation by merging blobs belonging to the same tracked object. Also, it is possible to use two independent data sources, each representing a quality estimation for the other one, and maximize their agreement to improve the foreground segmentation [Conaire et al., 2006]. Beyond the use of other algorithms data, assistance can also be understood as exploiting relationships between algorithm stages as done in [García and Bescós, 2008] to improve segmentation through quality estimations provided by low-level scene ontologies.

Moreover, *stand-alone specific* evaluation targets challenging situations with an expected decrease in performance such as illumination changes, shadows or dynamic background. Illumination changes have been analyzed by exploiting the image entropy over time [Cheng et al., 2011][Ramirez-Quintana and Chacon-Murguia, 2015] and the relations between an image and the background model, directly using foreground information [López-Rubio and López-Rubio, 2015b][Cheng et al., 2015] and raw data [Withagen et al., 2009][Chen and Ellis, 2014]. Moreover, detecting cast shadows has been tackled by analyzing chromatic, physical, geometric or texture properties of foreground masks [Conaire et al., 2007][Sanin et al., 2012][Al-Najdawi et al., 2012][Huerta et al., 2015]. However, some approaches automatically detect and remove shadows in single images without using foreground masks information, but analyzing image properties [Guo et al., 2013][Khan et al., 2016]. For instance, shadow removal is cast as a matting problem to build data-driven approaches using region and boundary properties [Khan et al., 2016] and pair-wise region relationships [Guo et al., 2013]. Additionally, detecting dynamic background motion is desirable due to their difficult modeling [Pham et al., 2015]. For example, dynamic background can be locally detected by, respectively, analyzing blinking pixels over time and motion features to determine characteristic background motions [Pham et al., 2015][St-Charles et al., 2015]. Additionally, such dynamism can be also globally detected by accumulating adjacent frame differences in temporal windows [Ramirez-Quintana and Chacon-Murguia, 2015].

Finally, *stand-alone generic* evaluation focuses on estimating quality by inspecting properties of the foreground masks. These generic measures have been weakly explored for the evaluation of segmented foreground masks in videos obtained from background subtraction algorithms. However, closely related areas, such as image segmentation [Zhang et al., 2008], image co-segmentation [Li et al., 2014], video object segmentation [Erdem et al., 2004] or object recognition [Zitnick and Dollár, 2014], have studied stand-alone generic measures. We select

Category	Measure	Acronym	Type	Features				
				Color channels	Motion	Edges	Segmented image regions	Foreground mask
Contrast	Spatial Color Contrast [Erdem et al., 2004]	SC	Accuracy	✓				
	Motion Difference [Erdem et al., 2004]	MD	Accuracy		✓			
	Spatial Clique Potential [Min et al., 2014]	SP	Accuracy	✓				
	Temporal Clique Potential [Min et al., 2014]	TC	Accuracy		✓			
	Local Contrast to Neighbors [Correia and Pereira, 2002]	LN	Accuracy	✓				
	Local Contrast [Li et al., 2014]	LC	Accuracy	✓				
	Color Contrast [Giordano et al., 2015]	CC	Accuracy	✓				
	Motion Contrast [Giordano et al., 2015]	MC	Accuracy		✓			
Uniformity	Spatial Uniformity [Correia and Pereira, 2002]	SU	Error	✓				
	Motion Uniformity [Correia and Pereira, 2002]	MU	Error		✓			
	Color Homogeneity [Giordano et al., 2015]	CH	Accuracy	✓				
	Motion Homogeneity [Giordano et al., 2015]	MH	Accuracy		✓			
Shape	Shape Regularity [Correia and Pereira, 2002]	SH	Accuracy					✓
	Boundary Turning Points [Li et al., 2014]	BT	Error					✓
	Boundary Curvature [Li et al., 2014]	BC	Error					✓
	Boundary Complexity [Giordano et al., 2015]	BX	Error					✓
Fitness	Edge fitness [Min et al., 2014]	E2	Error			✓		
	Edge fitness [Li et al., 2014]	E1	Accuracy			✓		
	Superpixel Straddling [Giordano et al., 2015]	SS	Accuracy				✓	✓
Density	Separability [Li et al., 2014]	SE	Error					✓
	Edge Density [Giordano et al., 2015]	ED	Accuracy			✓		

Table 4.1: Description of selected quality measures.

and detail representative measures from these areas in Section 4.3 and we extensively compare their performance for the estimation of background subtraction quality in the experimental work (Section 4.5).

4.3 Stand-alone generic quality measures

Table 4.1 summarizes the measures selected for the quality assessment of background subtraction algorithms based on blob mask properties. These measures aim to estimate accuracy (error) when they focus, respectively, on the correctness (failures) of the results. Furthermore, existing measures can be classified into five groups: *contrast*, *uniformity*, *shape*, *fitness* and *density*. In this context, the relation between objects and blobs is established by defining objects as entities composed by one or several connected components (i.e. blobs). Therefore, stand-alone generic quality measures can be defined operating over a set of O individual blobs $\{\mathcal{S}_i\}_{i=1}^{O}$ extracted from the foreground segmentation mask $\mathcal{M}$ corresponding to the current image $\mathcal{I}^t$, where t is a temporal index. Each blob $\mathcal{S}_i$ is a set of connected foreground pixels in $\mathcal{M}$. For clarity, the temporal index is omitted in this section for all measures except for those where different temporal instants are used.

4.3.1 Contrast-based measures

These measures compute a spatial or temporal contrast between regions around blob masks contours, where higher contrast indicates higher quality. In consequence, contrast measures can be defined as a function $g^X(\mathcal{S}_i)$, where X denotes an arbitrary measure and the function

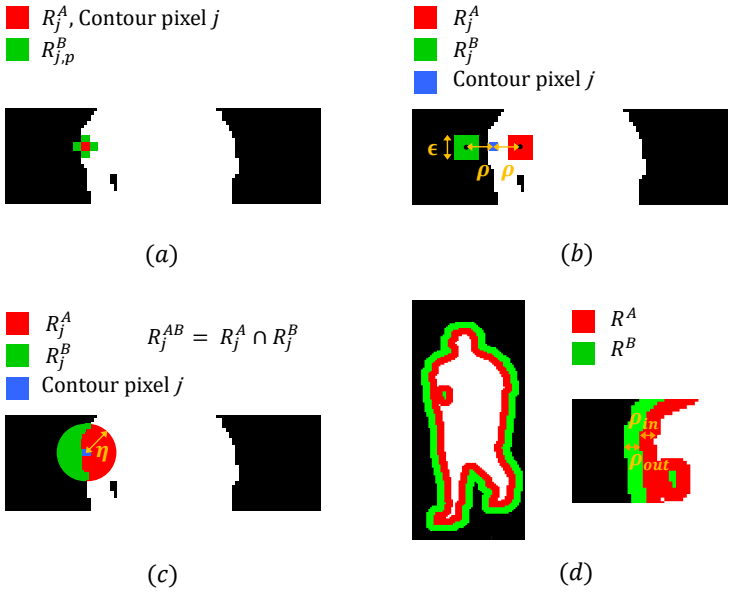


Figure 4.2: Examples of contrast-based measures using the regions R_j^A and R_j^B. From a) to d) regions for [Correia and Pereira, 2002], [Erdem et al., 2004], [Min et al., 2014] and [Li et al., 2014][Giordano et al., 2015]. In b) ϵ defines the side length of a squared region and ρ the distance from the contour pixel j to the centers of the inner and outer squared regions. In c) η is the radius of the circular region R_j^{AB} around contour pixel j, whose inner and outer regions are defined by the blob mask. In d) the inner and outer regions are defined, respectively, with size ρ_{in} and ρ_{out} around the blob mask.

depends on $\mathcal{S}_i$ by employing its contours and its regions (see Figure 4.2) in the image $\mathcal{I}$ or the optical flow fields $\mathcal{O}$. In [Correia and Pereira, 2002], g is particularized for the Local Contrast to Neighbors (LN) measure as:

$$g^{LN} = \frac{1}{N}\sum_{j=1}^{N}\sum_{c=1}^{3} w_c \cdot \max_p \left\| \mathcal{I}_c\left(R_j^A\right) - \mathcal{I}_c\left(R_{j,p}^B\right) \right\|, \tag{4.1}$$

where j is the contour pixel index; c is the color channel index; N is the number of contour pixels; w_c weights each color channel $\mathcal{I}_c$; and $\mathcal{I}_c\left(R_j^A\right)$ and $\mathcal{I}_c\left(R_{j,p}^B\right)$ are, respectively, the region R_j^A for each contour pixel and each p adjacent region $R_{j,p}^B$ in $\mathcal{I}_c$ (see Figure 4.2(a) for the definitions of R_j^A and $R_{j,p}^B$). This measure operates in the YUV color space computing differences between each contour pixel and its adjacent regions, resulting in a high quality score whenever large contrast is found between them. Also, contrast is defined in [Erdem et al., 2004] for spatial color contrast (SC) and motion difference (MD) as:

$$g^{SC} = \frac{1}{N}\sum_{j=1}^{N} \frac{\left\| \mathrm{E}\left[\mathcal{I}\left(R_j^A\right)\right] - \mathrm{E}\left[\mathcal{I}\left(R_j^B\right)\right] \right\|}{\sqrt{3 \cdot 255^2}}, \tag{4.2}$$

$$g^{MD} = \frac{1}{N}\sum_{j=1}^{N}\frac{w_j\left(1-\exp\left(-\left\|\mathrm{E}\left[\mathcal{O}(R_j^A)\right]-\mathrm{E}\left[\mathcal{O}(R_j^B)\right]\right\|\right)\right)}{\sum_{j=1}^{N} w_j}, \tag{4.3}$$

where $\|\cdot\|$ denotes the euclidean norm; $\mathcal{O}$ is the two-dimensional optical flow field associated with image $\mathcal{I}$; $\mathrm{E}[\cdot]$ is the average computed over the ϵ-squared regions R_j^A and R_j^B from Figure 4.2(b) in the image pixels (Eq. 4.2) and optical flow vectors (Eq. 4.3); and w_j is the pixel-level optical flow reliability that weights the optical flow at contour pixel j as defined in [Erdem et al., 2004]. Figure 4.2(b) shows the definition of the adjacent regions R_j^A and R_j^B whose contrast is computed in the YCbCr color space (g^{SC}) or using the optical flow fields (g^{MD}), thus expecting higher differences for well segmented blobs. Furthermore, [Min et al., 2014] implements g for spatial and temporal contrast by, respectively, spatial clique potential (SP) and temporal clique potential (TC) by:

$$g^{SP} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{\left|R_j^B\right|}\sum_{k=1}^{|R_j^B|} w_k \cdot \left\|\mathcal{I}\left(R_{j,k}^B\right) - \frac{1}{\left|R_j^A\right|}\sum_{l=1}^{|R_j^A|} w_l \cdot \mathcal{I}_c\left(R_{j,l}^A\right)\right\|^2, \tag{4.4}$$

$$g^{TC} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{\left|R_j^{AB}\right|}\sum_{k=1}^{|R_j^{AB}|} w_k \cdot \left\|\mathcal{I}^t\left(R_{j,k}^{AB}\right) - \mathcal{I}^{t-1}\left(R_{j,k}^{AB}\right)\right\|^2, \tag{4.5}$$

where R_j^{AB} is the circular patch around a pixel contour j with, respectively, inner and outer regions R_j^A and R_j^B (see Figure 4.2(c)); $|\cdot|$ denotes cardinality (i.e. the number of pixels in a region); w_k is a pixel-level Gaussian decay to penalize the contribution of distant pixels to the center of R_j^{AB} as defined in [Min et al., 2014]; and $\mathcal{I}^t$ and $\mathcal{I}^{t-1}$ are the frames at temporal instants t and $t-1$. For these measures, highly contrasted regions in terms of gray-level values correspond to high quality scores. Moreover, the function g is particularized for the local contrast (LC) [Li et al., 2014] as g^{LC} and the color contrast (CC) [Giordano et al., 2015] as g^{CC} by $\chi\left(\mathrm{H}\left(\mathcal{I}\left(R^A\right)\right),\mathrm{H}\left(\mathcal{I}\left(R^B\right)\right)\right)$, where $\chi(\cdot,\cdot)$ denotes Chi-squared distance and $\mathrm{H}\left(\mathcal{I}\left(R^A\right)\right)$ and $\mathrm{H}\left(\mathcal{I}\left(R^B\right)\right)$ are the histograms of the image data in R^A and R^B (see Figure 4.2(d)). Both measures assume that the Chi-squared distance between R^A and R^B is high when a mask is properly segmented, whilst they differ in the color space used (g^{LC} uses RGB and g^{CC} hue from HSV) and in the definition of R^A and R^B lengths. Finally, g^{MC} is defined for motion Contrast (MC) similarly to g^{LC} and g^{CC} but comparing inner and outer optical flow fields as $\chi\left(\mathrm{H}\left(\mathcal{O}\left(R^A\right)\right),\mathrm{H}\left(\mathcal{O}\left(R^B\right)\right)\right)$. An example of contrast measure is presented in Figure 4.3, where low values of the motion difference (MD) measure indicates low quality for an erroneously segmented blob.

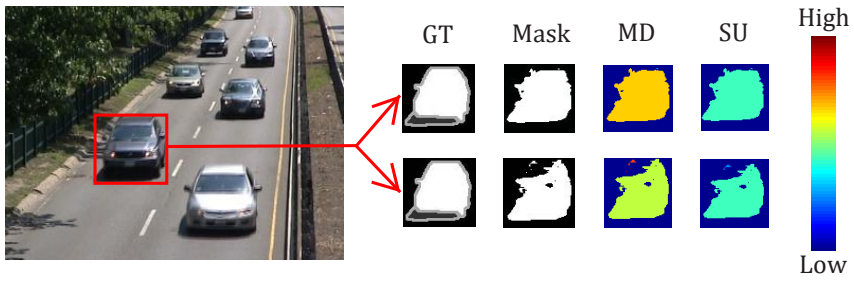


Figure 4.3: Examples of stand-alone generic measures motion difference (MD) and spatial uniformity (SU). On the left, image under analysis marking with a red rectangle a car to segment. On the right, ground-truth binary mask (GT), segmented masks from algorithms (FTSG [Wang et al., 2014a] on top and GMM [Stauffer and Grimson, 1999] on bottom) and segmented masks filled with the estimated score of MD and SU. MD, based on optical flow contrast, is able to capture the quality reduction from FTSG to GMM (yellow to green color), while SU does not capture such reduction in this example (same color for both masks with different quality).

4.3.2 Uniformity-based measures

These measures analyze the internal homogeneity of the blob mask region in terms of color or motion. A high homogeneity is assumed as a high quality indicator which can be defined as a function $u^X(\mathcal{S}_i)$ for each measure X and blob $\mathcal{S}_i$. The function u is implemented in [Correia and Pereira, 2002] for spatial and motion uniformity in the blob mask as:

$$u^{SU} = \sum_{c=1}^{3} w_c \cdot \text{var}\left[\mathcal{I}_c\left(R^{full}\right)\right], \tag{4.6}$$

$$u^{MU} = \sum_{c=1}^{2} \text{var}\left[\mathcal{O}_c\left(R^{full}\right)\right], \tag{4.7}$$

where w_c weights the uniformity score computed by each YUV color channel giving higher importance to Y; $\text{var}[\cdot]$ denotes variance; $\mathcal{I}_c$ and $\mathcal{O}_c$ are, respectively, an image $\mathcal{I}$ channel and an optical flow field $\mathcal{O}$ dimension; and R^{full} are all the pixels in the blob $\mathcal{S}_i$. Moreover, in [Giordano et al., 2015] color and motion homogeneity are extracted, respectively, by the average hue values $\text{E}\left[\mathcal{I}_c\left(R^{full}\right)\right]$ in the HSV color space and the average of the optical flow module $\text{E}\left[\left\|\mathcal{O}\left(\text{R}^{\text{full}}\right)\right\|\right]$ in the blob pixels. Both measures assume that high homogeneity is preferred for high quality masks; however an object that loses part of it in the segmented blob can keep a good homogeneity as presented in Figure 4.3, where spatial uniformity (SU) does not measure effectively a performance decrease. Note that, despite aiming to estimate homogeneity, measures proposed in [Giordano et al., 2015] are based on averaging color and motion, operation that does not define an homogeneity criterion as it does not consider variability.

4.3.3 Shape-based measures

Regarding shape properties, some measures estimate blob quality by a function $f^X(\mathcal{S}_i)$ for each shape complexity measure X, which associates complex shapes with poor segmentation. In [Correia and Pereira, 2002], shape regularity (SH) is cast as a combination of circularity and elongation of the blob as:

$$f^{SH} = \max\left(\frac{4 \cdot \pi \cdot \left|R^{full}\right|}{N^2}, \max\left(1, \frac{\left|R^{full}\right|}{20 \cdot T}\right)\right), \tag{4.8}$$

where left and right terms of $\max(\cdot,\cdot)$ denote, respectively, circularity and elongation; R^{full} are all the pixels in the blob $\mathcal{S}_i$; N the number of contour pixels; and T the blob thickness, i.e. number of morphological erosions of $\mathcal{S}_i$ until disappearance. Therefore, this measure computes shape regularity through geometric properties, matching high f^{SH} with well segmented blobs. Furthermore, in case of [Li et al., 2014], shape regularity is explored through detecting boundary turning points (BT), by defining:

$$f^{BT} = \frac{1}{N}\sum_{j=1}^{N} d\left(\tilde{\mathcal{S}}_{i,j} < \tau_1\right), \tag{4.9}$$

where N is the number of contour pixels; and $d(\cdot)$ has the value 0(1) when the smoothed blob mask $\tilde{\mathcal{S}}_i$ in each contour pixel j exceeds (or not) a threshold τ_1, thus defining the percentage of boundary turning points in the blob mask contour. These turning points of a blob contour are associated with irregular shapes, thus the higher f^{BT} the lower the blob quality. Furthermore, [Li et al., 2014] defines shape complexity by estimating the boundary curvature (BC) as:

$$f^{BC} = \sum_{j=1}^{N} \frac{\det\left(\text{Hess}\,\mathcal{S}_{i,j}\right)}{\left(1 + \|\nabla\mathcal{S}_{i,j}\|^2\right)}, \tag{4.10}$$

where $\det(\cdot)$ is the determinant of a matrix; Hess is the Hessian matrix; and $\|\nabla\mathcal{S}_{i,j}\|$ is the norm of the gradient of $\mathcal{S}_{i,j}$. This measure associates good quality with smooth contours, which are expected to present a small curvature, i.e. low f^{BC}. Also, [Giordano et al., 2015] defines the boundary curvature, but jointly with boundary concavity, thus defining boundary complexity (BX) by:

$$f^{BX} = \frac{1}{N}\sum_{j=1}^{N} \phi_{curv}\left(\mathcal{M}\left(R_j^C\right)\right) \cdot \phi_{conc}\left(\mathcal{M}\left(R_j^C\right)\right), \tag{4.11}$$

where N is the number of contour pixels; R_j^C is a squared region centered at each contour pixel j; and $\phi_{curv}\left(\mathcal{M}\left(R_j^C\right)\right)$ and $\phi_{conc}\left(\mathcal{M}\left(R_j^C\right)\right)$ compute, respectively, boundary curvature and

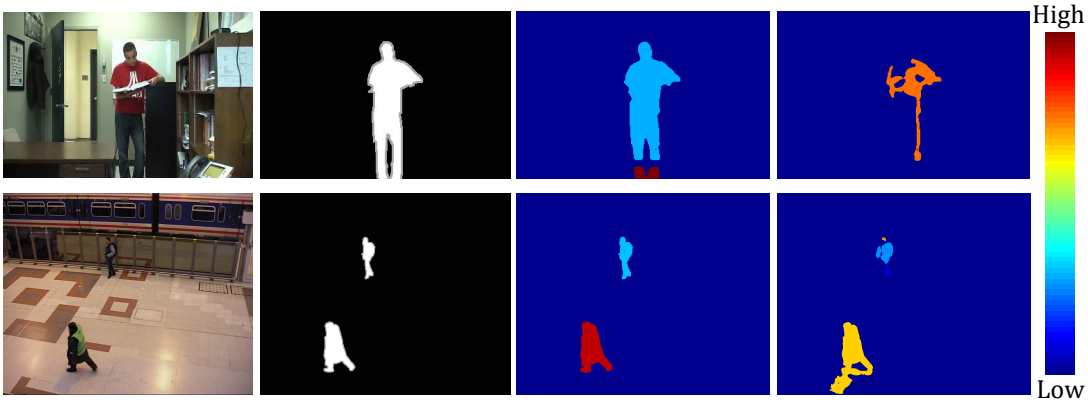


Figure 4.4: Examples of stand-alone generic measures boundary complexity (BX) and superpixel straddling (SS). Each row shows, from left to right, the original image, ground-truth binary mask and two algorithm results (MBS [Sajid and Samson Cheung, 2015] and GMM [Stauffer and Grimson, 1999]) with their estimated quality. First row shows that BX associates higher error with higher variation of the blob contour (right). Second row shows SS results where low quality is exhibited for shadows attached to the bottom blob (right).

boundary concavity as detailed in [Giordano et al., 2015]. The computation of $\phi_{curv}\left(\mathcal{M}\left(R_j^C\right)\right)$ consists in scanning each region R_j^C by comparing the distance between the contour segment's end-points and the segment's actual length inside the region, while $\phi_{conc}\left(\mathcal{M}\left(R_j^C\right)\right)$ looks for non-convex contour pixels. Therefore, convex shapes will lead to low values of f^{BX}, whereas the more irregular the shape the higher the measure value will be. First row in Figure 4.4 presents an example of quality estimation through shape complexity, showing that sometimes complex blob masks shapes are associated with low quality.

4.3.4 Fitness-based measures

These measures provide an estimation of fitness between blob mask and edges [Min et al., 2014][Li et al., 2014] or color-segmented image regions [Giordano et al., 2015] by a function $z^X\left(\mathcal{S}_i\right)$, where measure X defines the fitness function z. A segmented image region is a set of pixels grouped according to a similarity criterion, e.g. color [Felzenszwalb and Huttenlocher, 2004]. For adjustment to edges, [Li et al., 2014] defines z as edge fitness (E1) by:

$$z^{E1} = \frac{1}{N}\sum_{j=1}^{N} d\left(\mathcal{U}_j \geq \tau_2\right), \tag{4.12}$$

where N is the number of contour pixels; and $d\left(\cdot\right)$ has value 1 or 0 when contour pixel j in the ultrametric contour map $\mathcal{U}$ [Arbelaez et al., 2009] exceeds or not a threshold τ_2, thus defining the percentage of contour pixels fitted to edges. The higher the value of z^{E1}, the higher the

quality as more blob contour pixels are fitted to $\mathcal{U}$. Furthermore, [Min et al., 2014] implements edge fitness (E2) by defining:

$$z^{E2} = \frac{1}{N}\sum_{j=1}^{N}\mathcal{E}_j \tag{4.13}$$

where N is the number of contour pixels; and $\mathcal{E}$ is a binary edge map, being $\mathcal{E}_j$ its corresponding value in the contour pixel j. Similarly to z^{E1}, z^{E2} is expected to have higher values for well segmented blobs. For adjustment to image color regions, superpixel straddling (SS) [Giordano et al., 2015] (originally defined in [Alexe et al., 2012] for bounding boxes) defines z as:

$$z^{SS} = 1 - \sum_{q=1}^{Q}\frac{\min\left(\left|R_q \setminus R^{full}\right|, \left|R_q \cap R^{full}\right|\right)}{|R^{full}|}, \tag{4.14}$$

where Q is the number of overlapping color-segmented image regions R_q; R^{full} is the whole region of blob mask $\mathcal{S}_i$; $\left|R_q \setminus R^{full}\right|$ denotes the number of foreground pixels from the *q-th* segmented image region R_q outside $\mathcal{S}_i$; $\left|R_q \cap R^{full}\right|$ is the number of pixels from R_q included in $\mathcal{S}_i$; and $\left|R^{full}\right|$ is the number of pixels of blob $\mathcal{S}_i$. Superpixel Straddling z^{SS} establishes that high fitness values are obtained for high quality blobs, while wrongly segmented blobs present low fitness values. The second row in Figure 4.4 presents an example of quality estimation through fitness to color-segmented image regions, leading to a quality loss when a shadow is included in a blob as such shadow area is part of a segmented image region that does not fit the blob mask.

4.3.5 Density-based measures

Finally, density measures associate external and internal density properties of a blob with, respectively, low and high quality by defining a measure X that estimates density by a function $h^X(\mathcal{S}_i)$. Regarding external density, [Li et al., 2014] computes a separability measure (SE) as:

$$h^{SE} = \frac{\left|R^B\right|}{|R^A|} + \lambda N_l, \tag{4.15}$$

where R^A and R^B are the regions of the largest and second largest blobs in the neighborhood of $\mathcal{S}_i$ (including $\mathcal{S}_i$); λ is a scaling factor; and N_l is the number of neighboring blobs. Therefore, a correctly segmented blob is expected to be a dominant blob in its neighborhood with a small number of blobs in such neighborhood, thus leading to low h^{SE} values. Note that [Li et al., 2014] defines the neighborhood as the entire image as it is applied for images containing a unique object to segment, which is not the case in background subtraction. Therefore, we define the neighborhood as an extended bounding box of 10 pixels by all sides. Moreover, [Giordano et al., 2015] uses in blob masks a measure defined in [Alexe et al., 2012] for bounding boxes that

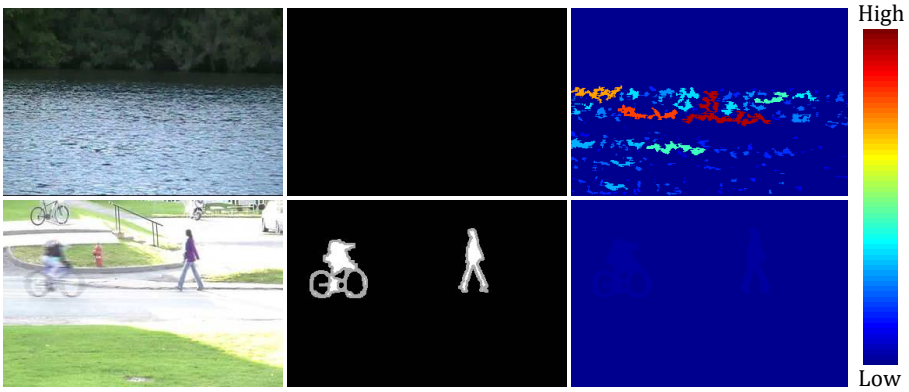


Figure 4.5: Examples of external density measure (SE). Each row shows, from left to right, original image, ground-truth mask and detection using GMM [Stauffer and Grimson, 1999] with its estimated quality. This measure detects high error when there are neighboring blobs of different size (first row), while measures low error for correct blobs (second row).

captures edge density in the inner ring region R^{in} of $\mathcal{S}_i$ through the function:

$$h^{ED} = \frac{1}{|R^{in}|} \sum_{k=1}^{|R^{in}|} \mathcal{E}_k, \tag{4.16}$$

where $\mathcal{E}$ is a binary edge map, being $\mathcal{E}_k$ its corresponding value for each pixel $k \in R^{in}$. A higher density of edges is expected in the inner ring for blobs that are properly segmented. Figure 4.5 presents an example of external density that leads to higher error when a blob has surrounding blobs of different size.

4.4 Experimental methodology

To analyze the discussed quality measures applied to blobs extracted from background subtraction algorithms, we have selected datasets and algorithms and defined blob-level ground-truth measures to perform our experiments.

4.4.1 Dataset and algorithms

We use the CDNET2014 dataset [Wang et al., 2014b] that consists of 53 video sequences containing common background subtraction challenges with their corresponding ground-truth data. We select eight of the eleven categories (*PTZ*, *Thermal* and *Turbulence* are excluded) as many of the selected measures are defined for color images and most of the background subtraction algorithms are designed for static cameras. These eight categories include 40 video sequences (113848 frames in total); visual examples are illustrated in Figure 4.6. To extract blobs from

Figure 4.6: Example images from selected categories of CDNET2014 dataset. From top-left to bottom-right: *Baseline*, *Dynamic Background*, *Camera Jitter*, *Intermittent Object Motion*, *Shadows*, *Bad Weather*, *Low Framerate* and *Night Videos* categories.

the video sequences, we employ eight relevant background subtraction algorithms (see Table 4.2 for a brief summary) according to their CDNET2014 results (ordered in increasing order of ground-truth performance[2]): GMM [Stauffer and Grimson, 1999], KDE [Elgammal and Davis, 2000], SC-SOBS [Maddalena and Petrosino, 2012], AMBER [Wang and Dudek, 2014], CwisarDH [Gregorio and Giordano, 2014], FTSG [Wang et al., 2014a], MBS [Sajid and Samson Cheung, 2015] and SuBSENSE [St-Charles et al., 2015]. KDE (Kernel Density Estimation) builds a non-parametric kernel using pixel values observed over time for each pixel location and detects foreground by comparing new frames to these kernels in each pixel location; GMM (Gaussian Mixture Model) models each pixel as an adaptive mixture of Gaussians and detects foreground pixels when the Gaussian distribution associated with the pixel is not part of the dominant data; SC-SOBS (Spatially Coherent Self-Organizing Background Subtraction) is based on a self-organizing neural network that learns pixel variations along a spatially restricted local neighborhood of the most active neurons and detects foreground pixels by comparing each pixel location in the image to that location and its neighbors in the background model; AMBER (Adapting Multi-resolution Background ExtractoR) implements a temporarily multi-resolution background model that contains pixel level values and their associated confidence based on their occurrence and detects foreground pixels with individual decision thresholds based on neighborhood information; CwisarDH (Change detection WiSARD system with History support) models the background through a weightless neural network that learns and updates pixel-level foreground discriminators and uses a buffer of repetitive foreground colors to determine when re-training is needed to handle stationary objects and sudden color variations; FTSG (Flux Tensor with Split Gaussian models) exploits the computation of motion information using a flux tensor formulation and the foreground and background modeling with a split Gaussian method to compare

[2] `http://changedetection.net/`

Algorithm	Model type	Model features	Average frame-level F-score
SuBSENSE	Non-parametric sample-based	Color, Texture	.7408
MBS	Single Gaussians of multiple features	Color	.7288
FTSG	Flux tensor and mixture of Gaussians	Color, Motion	.7283
CwisarDH	Weightless neural network	Color	.6812
AMBER	Multi-resolution temporal templates	Color	.6577
SC-SOBS	Self-organized neural network	Color	.5961
GMM	Mixture of Gaussians	Color	.5707
KDE	Non-parametric kernel	Color	.5688

Table 4.2: Background subtraction algorithms selected to analyze blob properties.

them and perform a robust foreground detection; MBS (Multi-mode Background Subtraction) selects in a training procedure the appropriate color space for each scene using single Gaussian models for each color channel, thus leading to accurate foreground detection when comparing image pixels with the model in the test phase; and SuBSENSE (Self-Balanced SENsitivity SEgmenter) provides a robust foreground detection using a non-parametric background model based on samples of spatio-temporal binary similarity descriptors and color values, which is dynamically tuned following a feedback scheme. We have selected these algorithms based on their overall ground-truth performance in CDNET2014, including algorithms of low, medium and high performance in order to build a set of data with variable performance and enable the inspection of stand-alone measures utility. We use the results provided in CDNET2014 and apply the quality measures on every 30th frame, obtaining approximately 240000 blobs from the selected background subtraction algorithms.

4.4.2 Blob-level performance measures

To assess the selected stand-alone generic measures, we define new blob-level ground-truth based measures. This is necessary as objects may be fragmented into several blobs and no prior knowledge is assumed for the correspondence between blobs from the segmented foreground mask and available blobs from the ground-truth (ground-truth blobs usually correspond to objects). Firstly, we define a set of M ground-truth blobs $\mathcal{G} = \{GT_j\}_{j=1}^{M}$ extracted from their ground-truth masks. Secondly, Precision $P(\mathcal{S}_i)$ and Recall $R(\mathcal{S}_i)$ of the extracted blob S_i are computed and combined to define a unique blob-level performance measure by F-score $F(S_i)$:

$$P(\mathcal{S}_i) = TP(\mathcal{S}_i) / (TP(\mathcal{S}_i) + FP(\mathcal{S}_i)), \tag{4.17}$$

$$R(\mathcal{S}_i) = TP(\mathcal{S}_i) / (TP(\mathcal{S}_i) + FN(\mathcal{S}_i)), \tag{4.18}$$

$$F(\mathcal{S}_i) = 2 \cdot P(\mathcal{S}_i) \cdot R(\mathcal{S}_i) / (P(\mathcal{S}_i) + R(\mathcal{S}_i)), \tag{4.19}$$

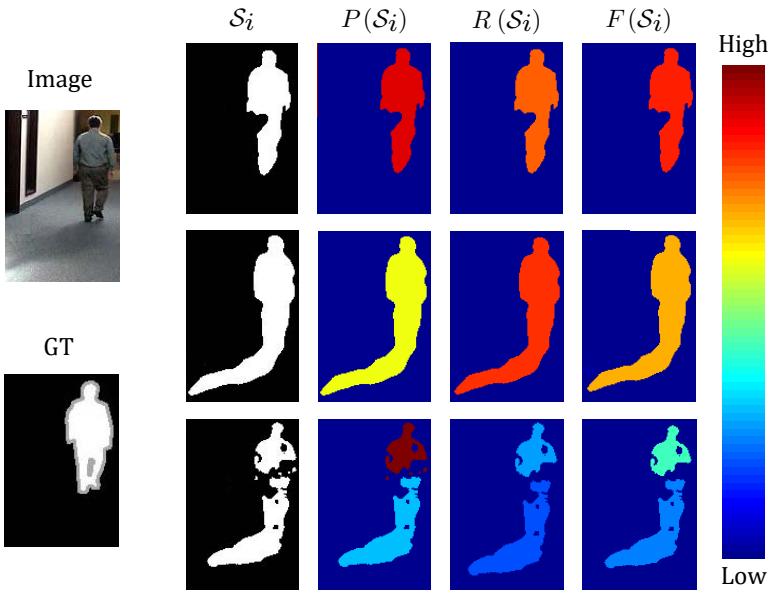


Figure 4.7: Example of ground-truth (GT) based quality measures. On the left, the original image and its associated ground-truth are presented. On the right, each row shows an example of blob-level ground-truth performance with the blob-level Precision $P(\mathcal{S}_i)$, Recall $R(\mathcal{S}_i)$ and F-score $F(\mathcal{S}_i)$. First row corresponds to a well segmented foreground with high performance. Second and third rows show performance decreases due to, respectively, including wrongly segmented pixels (shadows) and blob fragmentation of the object mask. The former situation impacts $P(\mathcal{S}_i)$ value (yellow), while the latter reduces $R(\mathcal{S}_i)$ (dark blue), thus decreasing $F(\mathcal{S}_i)$ in both cases.

where $TP(\mathcal{S}_i) = |\mathcal{G}(\mathcal{S}_i) \cap \mathcal{S}_i|$ denotes the number of correctly detected pixels in the blob mask $\mathcal{S}_i$ as compared to the associated ground-truth object $\mathcal{G}(\mathcal{S}_i)$; $FP(\mathcal{S}_i) = |\mathcal{S}_i \setminus \mathcal{G}(\mathcal{S}_i)|$ is the number of wrongly detected pixels in the blob mask $\mathcal{S}_i$; and $FN(\mathcal{S}_i) = |\mathcal{G}(\mathcal{S}_i) \setminus TP(\mathcal{S}_i)|$ is the number of missed pixels from the associated ground-truth objects. The ground-truth objects associated with each blob $\mathcal{S}_i$ are defined as $\mathcal{G}(\mathcal{S}_i) = \{GT_j : GT_j \cap \mathcal{S}_i > 0 \ \& \ GT_j \in \mathcal{G}\}$, i.e. the set of ground-truth objects overlapping the blob. Figure 4.7 depicts examples for foreground segmentation masks with different quality. These ground-truth based measures can be computed for those blobs with $TP(\mathcal{S}_i) > 0$ (True Positive Blobs or TPBs), whereas we set all measures to zero for blobs with $TP(\mathcal{S}_i) = 0$ (False Positive Blobs or FPBs).

4.4.3 Similarity of measures

To understand the similarity between stand-alone measures and ground-truth data, we use the Pearson's correlation coefficient [Pearson, 1896]:

$$\rho_{G,X} = \frac{\mathrm{E}\left[(G - \mu_G)\left((X - \mu_X)\right)\right]}{\sigma_G \sigma_X}, \tag{4.20}$$

where G and X are, respectively, the ground-truth performance and a stand-alone quality estimation for a given set of blobs. It is expected for quality measures based on accuracy (error) to obtain positive (negative) correlations when compared to the ground-truth measure $F(\mathcal{S}_i)$. In order to get all correlation results in the positive range $[0, 1]$ and make them comparable, we transform stand-alone error measures by applying a unity-based normalization to all measures, i.e. using the maximum and minimum value to bound the measures to $[0, 1]$. Then, the measures values are subtracted from the value 1 for the measures whose results estimate the error of the segmented blob masks (on average over the whole dataset).

We explore both linear and non-linear similarities among generic quality measures by exploiting, respectively, Pearson's correlation [Pearson, 1896] and Self-Organizing Map (SOM) [Kohonen, 1982]. On the one hand, we compute the correlation-based similarity among stand-alone measures (Eq. 4.20) and use agglomerative hierarchical clustering (AHC) with a complete linkage criterion in order to group the generic quality measures as strongly correlated measures result in similar quality scores. AHC provides a tree-like diagram where a cutoff correlation 0.1 is employed to get the grouping. Note that such a parameter setting is common practice in other semi-automatic clustering techniques such as affinity propagation [Frey and Dueck, 2007] applied to ground-truth based evaluation of visual tracking [Cehovin et al., 2016]. On the other hand, we have performed a two-stage clustering [Vesanto and Alhoniemi, 2000] by first using SOM, a neural network that presents in a bi-dimensional grid the relations among multidimensional data, and then grouping the SOM neurons through k-means clustering. Note that we have used the SOM Toolbox [Vatanen et al., 2015] and a balanced set of TPBs and FPBs across the eight algorithms used (approximately 46000 blobs extracted from the original 240000 blobs) to analyze the relationships among stand-alone measures.

4.5 Experimental results

To analyze the stand-alone generic measures listed in Table 4.1, we have identified their relationships in Subsection 4.5.1, studied their discrimination capabilities for different performance levels in Subsection 4.5.2 and analyzed their algorithm ranking capabilities in Subsection 4.5.3. For simplicity, we use F to refer to the blob-level ground-truth performance $F(\mathcal{S}_i)$ in Eq. 4.19.

4.5.1 Measures relationships

Using Pearson's correlation we have discovered eight clusters for the generic quality measures as depicted in Figure 4.8. Despite the moderate cross-correlation obtained, we can observe that the first cluster groups measures based on spatial information. Moreover, the second and third clusters group, respectively, motion and shape measures, the latter including a contrast measure. Finally, the fifth cluster groups measures of different nature, while the remaining clusters contain

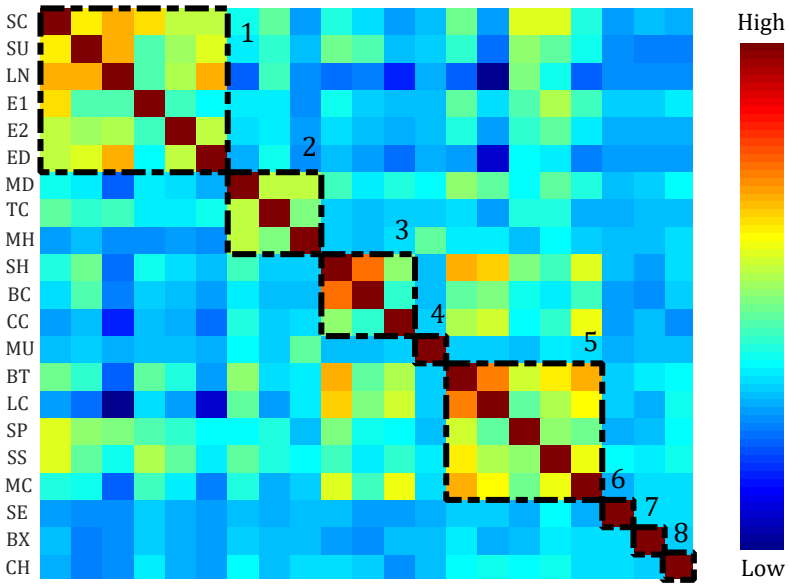


Figure 4.8: Cross-correlation among quality measures and clustering obtained via agglomerative hierarchical clustering.

single measures with low cross-correlation.

Furthermore, we explore more complex relations among quality measures by training a SOM with a lattice of hexagonal neurons or cells as done in [Vatanen et al., 2015]. In Figure 4.9(a) we present the neurons' lattice where quality measures have been assigned to the closest neuron. These assignments directly establish close relations among edge based measures E2 and ED; uniformity measures SU, SH and MH with motion difference MD; spatial contrast measures CC and LC with fitness to regions SS and external density SE; two motion-based measures MU and MC; and between shape measure BT and contrast measure TC. Additionally, in Figure 4.9(b) we present the Unified distance matrix (U-matrix) which shows the distance between neighboring neurons by extending the lattice and including hexagonal cells between real neurons to represent distances; thus spatially close neurons may have a large distance while distant neurons may be closer in the feature space. Analyzing the distances of the U-matrix, one can observe that two shape measures are closely related (BC and BX with a low distance cell between them) and that the neuron assigned to contrast MC is close to the neuron with contrast measures LC and CC. Also homogeneity CH and fitness to edges E1 are close and contrasts SP and LN have a medium distance between them. To determine clusters from the SOM lattice, we have applied k-means as done in [Vesanto and Alhoniemi, 2000], setting the number of clusters to six based on Davies Bouldin validation index [Vesanto and Alhoniemi, 2000], thus clustering the SOM as presented in Figure 4.9(c).

To determine a interesting subset of measures to estimate quality, the highest correlated

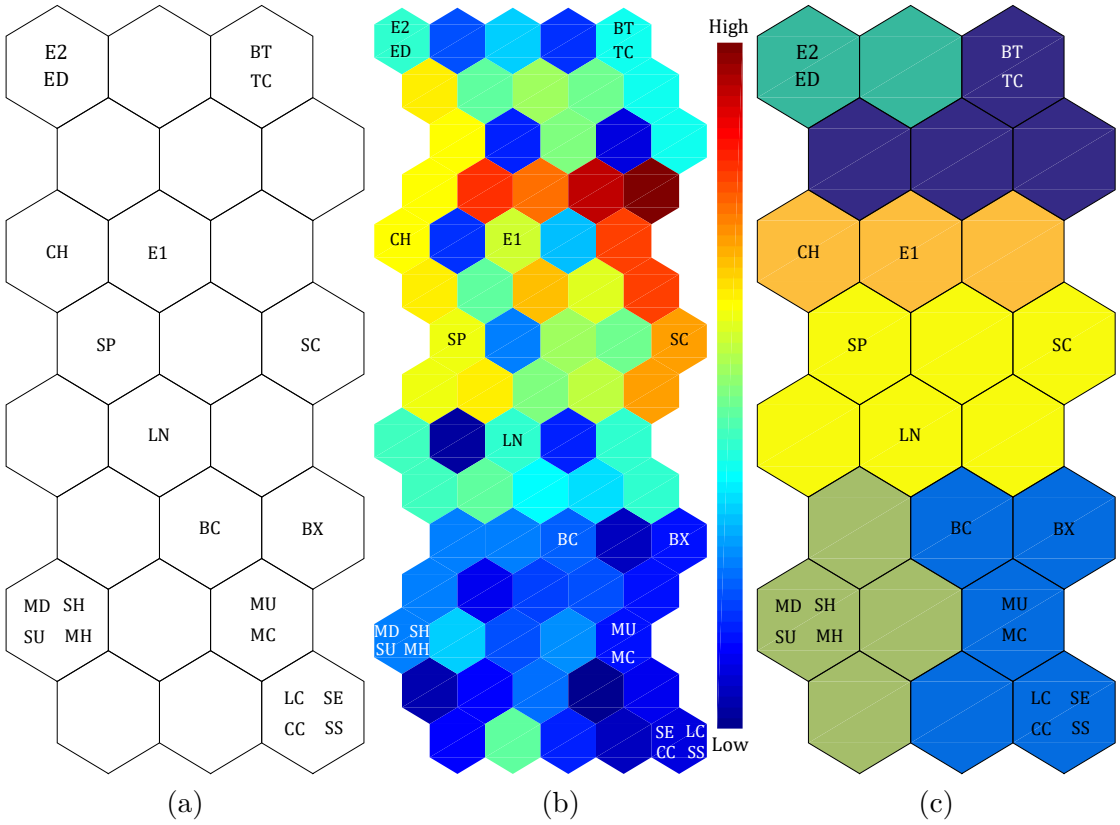


Figure 4.9: Self-Organizing Map of stand-alone generic quality measures. (a) Labeled lattice, (b) U-matrix, where empty cells denoting distances between neurons are introduced (red means high distance) and (c) Clustered lattice using k-means (the colors represent the same cluster).

measure with blob-level performance F of each cluster is selected (see Table 4.3), being the most promising superpixel straddling (SS) that is based on fitness between the blob mask and the segmented image regions. Note that top-correlated measures with F from the clustered SOM are SS, E1, SC, MD, BT and E2, among which SC, E2 and BT are not top-correlated measures of the Pearson's correlation based clusters. Therefore, we have included in Table 4.3 top-correlation for both linear and non-linear relations.

Moreover, we have studied the performance variability of the selected measures in Table 4.3 for each of the eight categories used from CDNET2014 as shown in Figure 4.10. Only the top-six are shown in this analysis for visualization reasons. Clearly, SS performs better than other measures across all categories, but it suffers diminishing correlation in some of them. For example, in *Bad Weather*, *Night Videos* and *Low Framerate*, the performance of SS decreases due to image segmentation errors that merge objects with their surroundings, thus diminishing their fitness value (see Figure 4.11). Therefore, the performance of SS is highly affected by the image segmentation algorithm used [Felzenszwalb and Huttenlocher, 2004]. Analyzing BT results, a

Measure	**SS**	**BT**	**E1**	**SC**	**MD**	**CC**	**E2**	**SE**	**BX**	**CH**	**MU**
Type	Fitness	Shape	Fitness	Contrast	Contrast	Contrast	Fitness	Density	Shape	Uniformity	Uniformity
Cluster	5	5	1	1	2	3	1	6	7	8	4
$\rho_{G,F}$	.6944	.4275	.3040	.3029	.2622	.1867	.1842	.1011	.0799	.0760	.0714

Table 4.3: Selected measures to estimate quality. The highest correlated measure of each cluster determined by agglomerative hierarchical clustering are selected. $\rho_{G,F}$ denotes Pearson's correlation between each stand-alone generic quality measure and ground-truth based blob-level performance F .

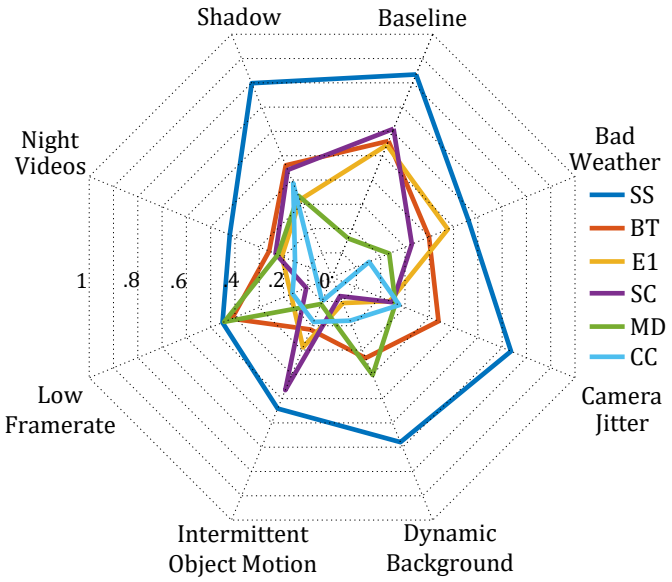


Figure 4.10: Correlation among blob-level performance F and the subset of interesting stand-alone measures. Data used involves the balanced set of blobs from algorithms in Table 4.2 split into categories.

high correlation is obtained due to the ability to distinguish between FPB and TPB, thus leading to diminishing correlations when there are less FPB (low quality blobs) and the challenge lies in the estimation of different medium and high performance values, as occurs in *Bad Weather* and *Intermittent Object Motion* categories. For *Night Videos* category, BT also experiences low correlation caused by the smooth shapes of FPB. Moreover, E1 is affected not only in the aforementioned challenging categories for SS, but in *Camera Jitter* and *Low Framerate* due to erroneous contour probability maps. The motion contrast measure MD has diminishing correlation when measuring quality of non-moving objects as evidenced in the *Intermittent Object Motion* category that contains stationary objects. Also, optical flow inaccuracies in the *Camera Jitter* category lead to low correlation for MD. The remaining measures, despite useful in some cases, are not good quality estimators from a global perspective.

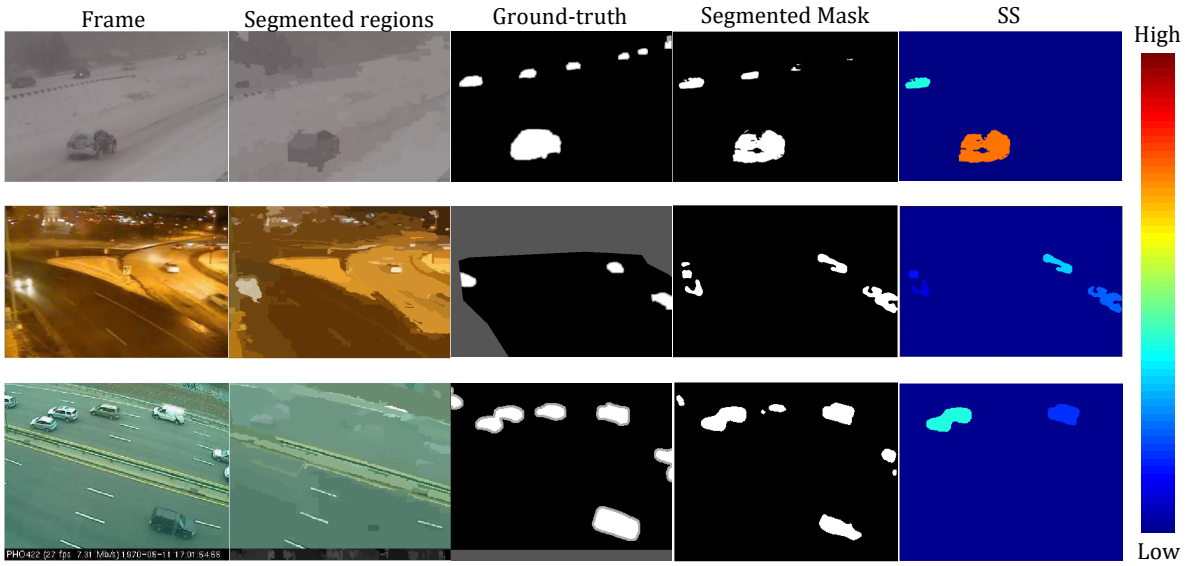
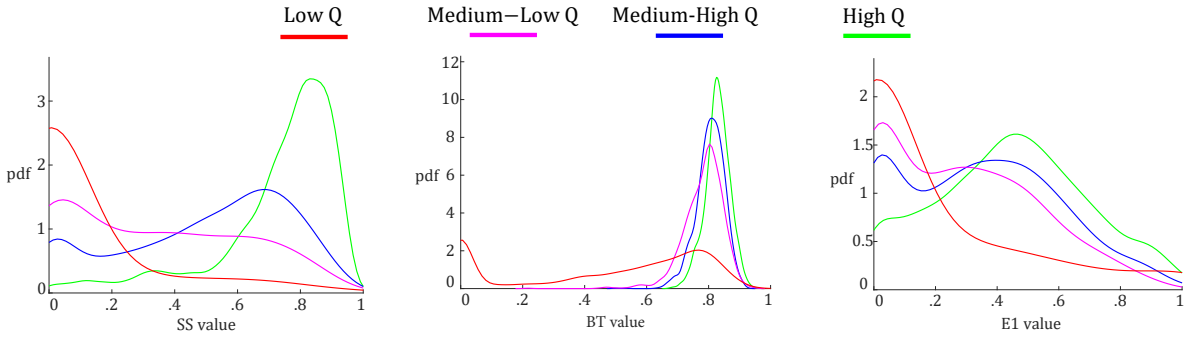


Figure 4.11: Example of SS failures due to erroneously segmented image regions that merge objects with the environment. Each row represents an example, from top to bottom: *Bad Weather* (frame 1218 of blizzard sequence and segmented foreground of GMM), *Night Videos* (frame 901 of winterStreet sequence and segmented foreground of MBS) and *Low Framerate* (frame 881 of turnpike_0_5fps sequence and segmented foreground of SuBSENSE). The erroneously segmented image regions overlapping some blobs, lead to low quality in such blobs. Note that this low quality in the SS scores is erroneously obtained for some properly segmented TPB.

4.5.2 Quality levels separation

We now further study the utility of stand-alone generic measures to qualitatively discriminate different ground-truth performance levels. We have defined four levels (low, medium-low, medium-high and high) by dividing the range for ground-truth performance $[0, 1]$ into four quartiles. Figure 4.12 shows the estimated probability density functions of the scores for the top-3 quality measures. The expected behavior revealing a good quality levels separation would be to have probability density functions with zero value out of each quality level range, i.e. zero for higher than 0.25 in low quality (red), lower than 0.25 and higher than 0.5 in medium-low quality (pink), lower than 0.5 and higher than 0.75 in medium-high quality (blue) and lower than 0.75 in high quality (green). Globally, none of the quality measures is able to clearly distinguish among the four levels. However, SS achieves a moderate separation which explains the highest value obtained for ground-truth correlation in Subsection 4.5.1. However, attending to the task of separating high and low quality, boundary turning points (BT) and edge fitness (E1) also show some potential as depicted, respectively, in Figure 4.12(b) and (c). Note that the remaining measures are not shown due to their weak separation capability among the performance levels.



(a) Superpixel straddling (SS) (b) Boundary turning points (BT) (c) Edge fitness constraint (E1)

Figure 4.12: Probability density functions (pdf) of the scores for top-3 quality measures to discriminate different ground-truth performance levels: High ($1 \geq F \geq 0.75$), Medium-High ($0.75 > F \geq 0.5$), Medium-Low ($0.5 > F \geq 0.25$) and Low ($0.25 > F \geq 0$).

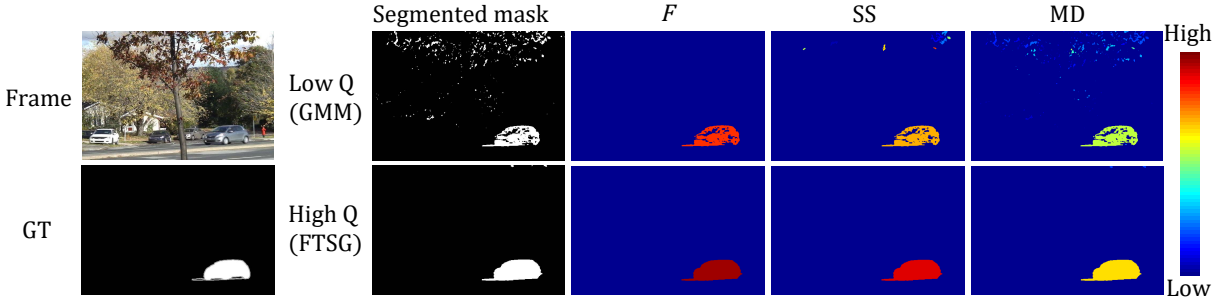


Figure 4.13: Example of replication of ground-truth based ranking (frame 2530 from Fall sequence). On the left, frame under analysis and its associated ground-truth (GT) are shown, while on the right, segmented mask, ground-truth measure F and stand-alone quality measures SS and MD are presented.

4.5.3 Ranking

From a practical point of view and given a set of algorithms with different performance, stand-alone measures are expected to replicate the ranking of algorithms given by the ground-truth evaluation. Table 4.4 shows the capabilities of the selected cluster measures from Subsection 4.5.1 to rank the selected algorithms (GMM, SC-SOBS, KDE, AMBER, CwisarDH, MBS, FTSG and SuBSENSE). Each measure ranking is computed by averaging the scores of all blobs of each algorithm, without introducing balancing processes. As posed by previous experiments, SS almost replicates the algorithm ranking, switching AMBER and CwisarDH, while providing a reasonable separation among algorithms. For BT, only algorithms with consecutive order in the ranking are switched, thus exhibiting ranking capabilities explained by the TPB and FPB separation presented in Subsection 4.5.2. The remaining measures do not achieve correct ranking, as expected by the correlation results in Table 4.3. Figure 4.13 depicts an example

Algorithm	F	SS	BT	E1	SC	MD	CC	E2	SE	BX	MU	CH
SuBSENSE	.203 (1)	**.276 (1)**	.646 (1)	.237 (1)	.177 (2)	.314 (1)	.657 (3)	.019 (2)	.993 (1)	.966 (1)	$.239\times10^{-3}$ (1)	.365 (3)
FTSG	.149 (2)	**.219 (2)**	.595 (3)	.200 (3)	.167 (5)	.305 (2)	.668 (2)	.019 (1)	.992 (2)	.933 (4)	$.167\times10^{-3}$ (2)	.382 (2)
MBS	.088 (3)	**.150 (3)**	.598 (2)	.208 (2)	.172 (4)	.260 (3)	.684 (1)	.015 (6)	.989 (3)	.956 (2)	$.103\times10^{-3}$ (3)	.385 (1)
CwisarDH	.052 (4)	**.080 (5)**	.490 (5)	.167 (6)	.154 (7)	.257 (4)	.654 (4)	.019 (3)	.978 (8)	.941 (3)	$.085\times10^{-3}$ (4)	.344 (4)
AMBER	.042 (5)	**.122 (4)**	.495 (4)	.176 (4)	.158 (6)	.170 (7)	.573 (6)	.010 (8)	.988 (4)	.887 (5)	$.045\times10^{-3}$ (6)	.335 (5)
KDE	.023 (6)	**.066 (6)**	.328 (6)	.161 (7)	.174 (3)	.193 (5)	.572 (7)	.018 (4)	.981 (5)	.875 (6)	$.038\times10^{-3}$ (7)	.276 (6)
SC-SOBS	.018 (7)	**.053 (7)**	.319 (7)	.123 (8)	.153 (8)	.164 (8)	.578 (5)	.013 (7)	.979 (6)	.872 (7)	$.050\times10^{-3}$ (5)	.272 (8)
GMM	.015 (8)	**.033 (8)**	.254 (8)	.172 (5)	.179 (1)	.180 (6)	.542 (8)	.017 (5)	.978 (7)	.862 (8)	$.026\times10^{-3}$ (8)	.285 (7)

Table 4.4: Ranking obtained by stand-alone generic measures compared to blob-level performance F for TPBs and FPBs. Results are reported as the *score (ranking)* obtained for each stand-alone measure.

Algorithm	F	SS	BT	E1	SC	MD	CC	E2	SE	BX	MU	CH
SuBSENSE	.442 (1)	**.446 (1)**	.146 (2)	.335 (1)	.228 (1)	.369 (1)	.683 (1)	.231 (1)	**.993 (1)**	.965 (1)	$.423\times10^{-3}$ (1)	.405 (2)
FTSG	.394 (2)	**.404 (2)**	.147 (1)	.295 (3)	.208 (3)	.358 (3)	.676 (4)	.213 (2)	**.992 (2)**	.937 (3)	$.326\times10^{-3}$ (2)	.399 (3)
MBS	.345 (3)	**.335 (4)**	.131 (5)	.294 (4)	.215 (2)	.368 (2)	.678 (3)	.197 (4)	**.989 (4)**	.950 (2)	$.305\times10^{-3}$ (3)	.407 (1)
AMBER	.298 (4)	**.342 (3)**	.131 (4)	.332 (2)	.190 (4)	.316 (6)	.666 (6)	.198 (3)	**.990 (3)**	.930 (4)	$.228\times10^{-3}$ (7)	.390 (5)
KDE	.244 (5)	**.265 (5)**	.132 (3)	.226 (5)	.179 (5)	.347 (4)	.672 (5)	.193 (5)	**.985 (5)**	.906 (6)	$.303\times10^{-3}$ (4)	.393 (4)
SC-SOBS	.201 (6)	**.218 (6)**	.120 (8)	.220 (7)	.161 (7)	.295 (8)	.639 (8)	.149 (8)	**.984 (6)**	.881 (8)	$.235\times10^{-3}$ (6)	.359 (8)
GMM	.187 (7)	**.188 (7)**	.125 (7)	.225 (6)	.171 (6)	.319 (5)	.658 (7)	.174 (7)	**.983 (7)**	.894 (7)	$.243\times10^{-3}$ (5)	.367 (7)
CwisarDH	.167 (8)	**.176 (8)**	.127 (6)	.176 (8)	.158 (8)	.312 (7)	.680 (2)	.178 (6)	**.979 (8)**	.920 (5)	$.218\times10^{-3}$ (8)	.374 (6)

Table 4.5: Ranking obtained by stand-alone generic measures compared to blob-level performance F for TPBs. Results are reported as the *score (ranking)* obtained for each stand-alone measure.

where the rank of two low-high quality algorithms is kept by SS and MD (the top correlated fitness and contrast measures).

Moreover, we consider the capabilities to rank results when only True Positive Blobs (TPBs) appear, thus excluding the False Positive Blobs (FPBs) in this experiment. Table 4.5 shows the results where similar rankings are maintained as compared to the experiment with all data (TPBs and FPBs) in Table 4.4. However, SE keeps now the correct rankings unlike the previous experiment, thus denoting that FPBs quality is poorly estimated by this measure and BT deteriorates its ranking, thus denoting that it is not useful for the estimation of TPBs quality.

4.5.4 Discussion

The presented quality measures aim to estimate quality of blob masks when no ground-truth data is available. SS is the best measure being capable of measuring quality for both TPBs and FPBs as shown, respectively, in Figures 4.13 and 4.14. SS advantages can be explained from the Gestalt principles of grouping [Wertheimer, 1938] such as similarity (SS segmentation groups similar regions), proximity (SS only considers neighboring regions) and closure (SS penalizes regions not filled by the blob mask). Figure 4.15, presents an example of SS capability to decrease quality when false pixels or incomplete objects are detected. Other measures cover only one single principle, such as contrast/uniformity (similarity), shape (closure, by looking for simple shapes) or SE (proximity). However, SS fails theoretically when large FPBs cover

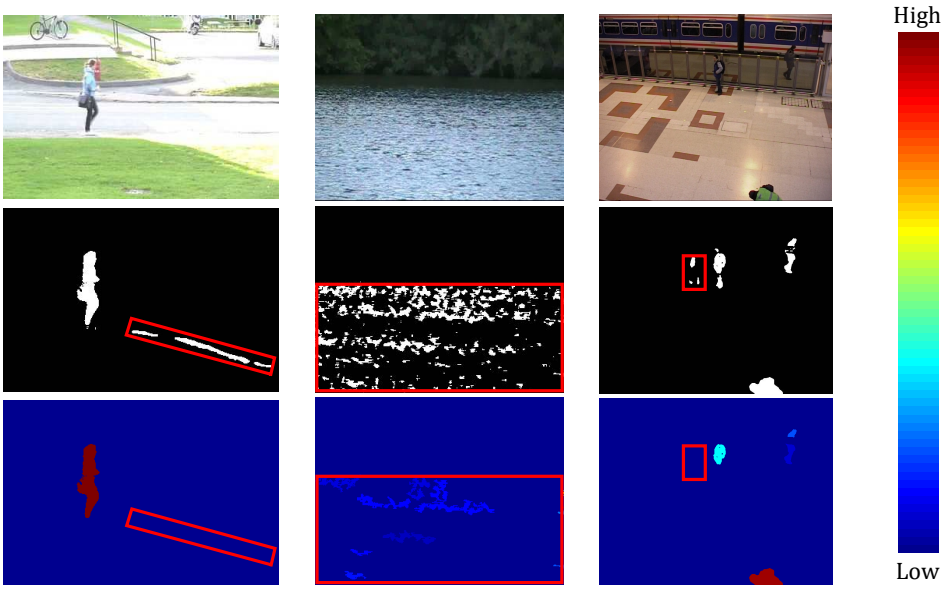


Figure 4.14: Example of superpixel straddling (SS) capabilities to estimate low quality in False Positive Blobs. Each column presents, from top to bottom, image under analysis, associated segmented foreground mask with False Positive Blobs marked in red and superpixel straddling value. First, second and third column show False Positive Blobs due to, respectively, shadows (frame 351 from pedestrians sequence), dynamic background (frame 316 from canoe sequence) and ghost detection (frame 656 from PETS2006 sequence), being the value of SS low in all cases.

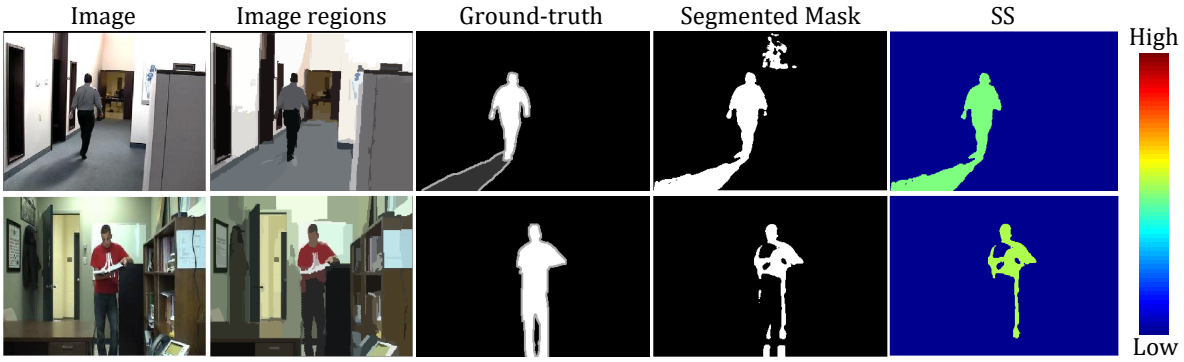


Figure 4.15: Example of Superpixel Straddling (SS) capabilities to estimate quality. First row (frame 1201 from cubicle sequence of Shadow category): the segmented mask has a weak fitness to its overlapping region in the shadow area (room floor), thus decreasing the SS value; second row (frame 1111 from office sequence of Baseline category): the segmented mask has not detected part of the person legs, thus SS value is reduced via the weak fitness between legs regions and the segmented mask.

several image regions and experimentally when image segmentation has errors merging different regions, thus leading to weak fitness. Moreover, BT has the capability of distinguishing between TPBs and FPBs, thus denoting that both can be distinguished from their shapes. Furthermore,

although MD is capable of estimating quality decreases for moving objects (see Figure 4.13), it is not useful for stationary objects as it employs motion contrast. Despite its poor potential to estimate quality, the separability SE may be interesting to detect situations of close blobs of different size, thus detecting fragmentation of objects and typical concentration of FPBs that dynamic backgrounds tend to generate. In addition, spatial uniformity measures are not useful to estimate quality in background subtraction as objects are either homogeneous and heterogeneous, but the motion homogeneity concept finds its utility in discriminating among FPBs (tend to be homogeneous) and moving objects (tend to be heterogeneous).

Besides the aforementioned discussion concerning measures utility in background subtraction, it is key to consider that identifying properties of high-performance foreground segmentation masks is application-dependent. We have presented the low performance of spatial homogeneity to estimate quality as objects could be either homogeneous or heterogeneous in background subtraction. However, if we consider spatial uniformity in a text segmentation application it will probably capture desired properties of segmented letters or words as they tend to have a unique color. Also, spatial contrast measures fail in background subtraction as the contrast between objects and their surroundings ranges from low to high values, thus high contrasted objects with many errors may have similar quality than low-contrasted objects with few errors. Nevertheless, in other related areas such as salient object segmentation, the spatial contrast measures could find their utility as salient objects tend to be contrasted with their surroundings. As opposed to background subtraction, motion contrast will be less useful in unconstrained scenarios where objects and camera may share the motion, leading to weak motion contrast. Moreover, considering proximity to neighboring blobs, like SE does, may find higher utility when a unique object is segmented in an image as usually occurs in salient object detection or co-segmentation. This dependency on the application exhibits the complexity of stand-alone evaluation and indicates that measures should be defined keeping in mind the target application, thus allowing to capture the nature of performance decreases.

Finally, a major drawback of stand-alone generic quality measures are the missed blobs. As these measures only analyze the segmented blobs, a completely undetected or missed object is not inspected and, therefore, leads to non-accurate estimations of overall quality of algorithms.

4.6 Conclusions

In this chapter, we provide a comprehensive study on stand-alone measures for quality estimation of foreground segmentation masks in background subtraction. We select from related literature a diverse set of measures that are thoroughly analyzed in terms of correlation with ground-truth, quality levels separation and algorithm ranking capabilities. Experiments with eight algorithms over a large background subtraction dataset shows that edge and region fitness and

motion contrast properties can be used to approximate ground-truth performance. Future work will explore developing new stand-alone quality measures and their application for performance improvement.

Chapter 5

Foreground segmentation improvement

5.1 Introduction[1]

Benchmarking computer vision algorithms has recently garnered remarkable attention as a methodological performance assessment [Wang et al., 2014b][Menze and Geiger, 2015][Borji et al., 2015][Kristan et al., 2016][Perazzi et al., 2016] driving the development of better algorithms. Alternatively, one may focus on improving the results of algorithms by post-processing techniques. This scheme may be of interest when the details of algorithms are not available and, therefore, making further changes or adjusting parameters is not possible.

In this context, foreground segmentation is a popular low-level task in computer vision to detect the objects of interest or foreground in images or videos [Bouwmans, 2014][Borji et al., 2015][Perazzi et al., 2016][Minaee and Wang, 2017] where such "interest" depends on the application domain. In this chapter, we focus on video sequences with a relative control of camera motion, where video object segmentation is tackled through background subtraction (BS) [Bouwmans, 2014][Yang et al., 2015] which compares each frame with a background model of the sequence.

Boosting BS performance has been mainly addressed by making use of three strategies. Firstly, selecting appropriate background models is akin to the ability of simultaneously dealing with several challenges [Bouwmans, 2014] while accurately adapting the background model to sequence variations. For example, Gaussian and support vector models [Tavakkoli et al., 2008][Lin et al., 2009] deal effectively with dynamic background; subspace learning models [Tsai and Lai, 2009][Tian et al., 2013] handle better illumination changes; neural networks [Maddalena and Pet-

[1]This chapter is an adapted version of the manuscript under review *"Diego Ortego, Juan C. SanMiguel and José M. Martínez, "Hierarchical improvement of foreground segmentation masks in background subtraction", in minor revision in IEEE Transaction on Circuits and Systems for Video Technology.*

rosino, 2014a][Gregorio and Giordano, 2014] offer a good computation-accuracy trade-off; and RPCA (Robust Principal Component Analysis) and sparse models [Sobral et al., 2014][Bouwmans et al., 2017][Erfanian Ebadi and Izquierdo, 2016] provide suitable frameworks to integrate constraints for foreground segmentation under different challenges. Secondly, properly choosing BS features [López-Rubio and López-Rubio, 2015a][Dey and Kundu, 2016][Bouwmans et al., 2016] is key as each feature type (e.g. color, gradient, texture, motion) exhibits robustness against different BS challenges, thus combining them may overcome single-feature shortcomings. Moreover, deep learning models [Braham and Droogenbroeck, 2016][W. et al., 2017] have recently emerged as promising frameworks to unify modeling and feature selection. However, current models [Braham and Droogenbroeck, 2016][W. et al., 2017][Ang Lim and Yalim Keles, 2018] are limited to employ train and test data from the same video sequence. Thirdly, post-processing techniques may improve foreground segmentation masks by either removing false positives or recovering false negatives [Parks and Fels, 2008]. For instance, there are techniques independent of the BS algorithm such as morphological operations [St-Charles et al., 2015][Dougherty, 1992] to fill holes or remove small regions; and inspection foreground mask properties [Schick et al., 2012][Giordano et al., 2015] to filter false positives and expand to undetected areas. Moreover, specific post-processing may tackle errors due to illumination changes [Chen and Ellis, 2014][López-Rubio and López-Rubio, 2015b], shadows [Sanin et al., 2012][Huerta et al., 2015] or dynamic backgrounds [St-Charles et al., 2015][Pham et al., 2015]; but the designed features depend on the employed background model, thus limiting their applicability.

For BS post-processing, the use of generic properties from foreground masks is desired to provide independence of specific phenomena (e.g. illumination or shadows) and, unlike morphological operations, to exploit complementary features to the ones extracted from the mask only. A recent analysis of these properties to estimate performance without ground-truth data (i.e. quality) [Ortego et al., 2017] identified the best property as the fitness between connected components of the foreground mask (i.e. blobs) and the regions of the segmented image (fitness-to-regions). Therefore, in this chapter we propose to improve foreground segmentation masks in BS through the fitness to several segmented image regions partitions, which enables extending foreground masks to undetected areas while removing poorly fitted and isolated foreground regions.

The contribution of this chapter is five-fold. Firstly, we introduce motion constraints to build an image segmentation hierarchy without merging moving foreground and background regions. Secondly, unlike related state-of-the-art [Schick et al., 2012][Giordano et al., 2015], we apply the fitness-to-regions property to estimate the quality of the foreground mask using each image in the segmentation hierarchy. We obtain a hierarchy of foreground quality images leading to better improvement scores as compared to [Schick et al., 2012]. Thirdly, a motion-based combination of the foreground quality images hierarchy is proposed to prevent foreground-background merging

in absence of motion, while promoting the extension of foreground regions in presence of motion. Fourthly, we improve foreground mask by fusing the foreground quality images into a unique foreground quality that is later converted into a foreground probability map by applying a pixel-wise fully-connected Conditional Random Field (CRF). Fifthly, we demonstrate the utility of the proposed approach to improve BS results of both top and low performing algorithms as presented in the experimental comparisons conducted using fourteen algorithms over four heterogeneous datasets with varied challenges (CDNET2014 [Wang et al., 2014b], LASIESTA [Cuevas et al., 2016], SABS [Brutzer et al., 2011] and BMC [Vacavant et al., 2013]). Moreover, we also show the potential application of foreground quality images for algorithm combination.

The reminder of this chapter is organized as follows: Section 5.2 overviews existing post-processing techniques for BS. Section 5.3.2 details the proposed framework for BS post-processing. Subsequently, Section 5.4 presents the experimental methodology and the experimental results. Finally, Section 5.5 summarizes the main conclusions.

5.2 Related Work

Post-processing techniques for BS can be classified into model-dependent and model-independent. The former employs the background model, such as shadows detectors to compare image and background features in foreground areas [Sanin et al., 2012], whereas the latter only uses image and foreground properties [Schick et al., 2012][Giordano et al., 2015], thus being independent of a particular algorithm.

Model-dependent techniques target challenging situations that produce erroneous foreground such as illumination changes, shadows or dynamic background. Removing erroneously detected foreground due to illumination changes has been addressed through color relations between the image and the background model in foreground areas [Chen and Ellis, 2014][López-Rubio and López-Rubio, 2015b]. Furthermore, chromatic, physical, geometric or texture relations between images and its related background model can be exploited to detect cast shadows in foreground masks [Sanin et al., 2012][Al-Najdawi et al., 2012][Huerta et al., 2015]. Additionally, detecting dynamic background motion [Pham et al., 2015] has not directly been tackled to post-process the result but to guide parameter tuning [St-Charles et al., 2015][Ramirez-Quintana and Chacon-Murguia, 2015]. However, the joint analysis of blinking pixels and background to image differences performed in [St-Charles et al., 2015] could be directly applied to remove false positives rather than influence the background modeling. Similarly, one can find that contour based techniques for abandoned object detection [Campos et al., 2011][Kim et al., 2014], based on both image and background information in foreground areas, can be applied to remove foreground errors associated to ghosts.

Model-independent techniques are based on the analysis of foreground mask properties to

improve results. A common strategy is to post-process foreground masks through morphological operations [Dougherty, 1992][St-Charles et al., 2015]. This strategy only relies on the foreground mask, thus obviating useful information that can be extracted from a joint analysis of the foreground and the color image. In this sense, there are techniques that analyze generic foreground mask properties [Schick et al., 2012][Giordano et al., 2015] to filter erroneous foreground or to expand it to undetected foreground areas. In [Giordano et al., 2015], region or blob mask properties associated to the internal uniformity, contrast in contours, shape complexity and fitness-to-regions are used to remove false positives blobs. Furthermore, [Schick et al., 2012] employs fitness-to-regions embedded into a Markov Random Field framework where high (low) fitness is associated to good (poor) foreground probability. In [Raman et al., 2017], the coherence of optical flow directions in each individual frame and frame-by-frame coherence of optical flow are used to remove erroneous blobs, split blobs that contain different objects and merge blobs belonging to the same object, thus improving foreground segmentation performance in background subtraction. Moreover, in [Hassan et al., 2015] image boundaries are used to remove erroneously detected blobs caused by the effect of illumination. Also, ghosts can be post-processed using optical flow [Parks and Fels, 2008], as foreground objects often moves. However, absence of motion is not only characteristic in ghosts, but also in static foreground objects.

As a conclusion, *Model-independent* techniques stand out as very interesting alternatives due to their independence of BS algorithms. The fitness-to-regions property has demonstrated a great potential to both estimate foreground quality [Ortego et al., 2017] and improve results [Schick et al., 2012]. However, the use of over-segmented images (i.e. superpixels) in [Schick et al., 2012] highly limits the improvement capabilities, as superpixels normally do not extend over complete objects. In fact, such mapping between superpixels and objects remains an open issue in the object proposal literature [Uijlings et al., 2013][Arbeláez et al., 2014][Xiao et al., 2015], where superpixel merging to cover large or complete object regions is inspected.

5.3 Foreground mask improvement

5.3.1 Overview

We propose a framework to improve foreground masks $\mathcal{M}_t$ obtained by BS algorithms from an image $\mathcal{I}_t$ in the temporal instant t (see Figure 5.1). Firstly, we compute a motion-aware segmentation hierarchy $\mathbb{H}_t = \left\{\mathcal{R}_t^l\right\}_{l=1}^{L}$, where $\mathcal{R}_t^l = \left\{R_{t,i}^l\right\}_{i=1}^{k^l}$ is the image segmentation partition at hierarchy level l that is composed by k^l individual image regions $R_{t,i}^l$ and L is the number of hierarchy levels. This hierarchy contains several image segmentation partitions, each describing a degree of detail of the image $\mathcal{I}_t$ (from fine to coarse levels). The coarser the level the higher the merging of regions, thus covering larger object areas. We consider spatial similarities based on color and introduce motion constraints through the optical flow $\mathcal{O}_t$ in order to avoid merging

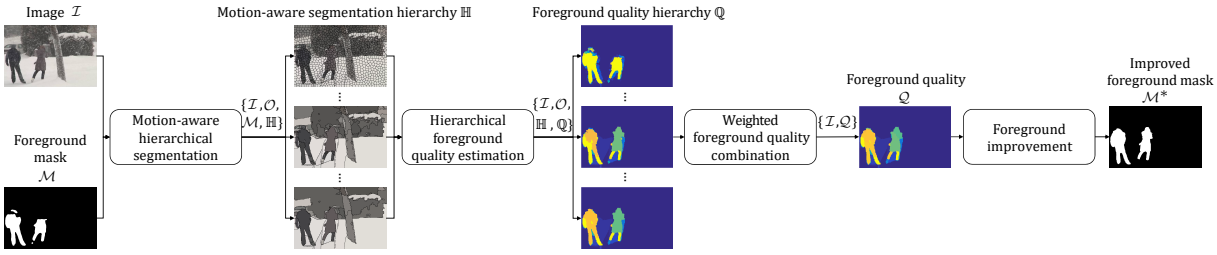


Figure 5.1: Foreground improvement framework overview. For clarity, we avoid the temporal index t (common to all notation). The motion-aware hierarchy $\mathbb{H}$ computed from the motion-aware color-based UCM (Eq. 5.1) is explained in Subsection 5.3.2.1, while the foreground quality hierarchy $\mathbb{Q}$ is computed using a fitness-to-regions property (Eq. 5.3) defined in Subsection 5.3.2.2. Then, a unique foreground quality $\mathcal{Q}$ is estimated using the weighted combination (Eq. 5.4) from Subsection 5.3.2.3. Finally, the improved foreground mask $\mathcal{M}^*$ is obtained via optimal labeling (Eq. 5.10) as presented in Subsection 5.3.2.4.

foreground and background regions in each partition of the hierarchy $\mathbb{H}_t$. Then, we estimate a foreground quality image for each level of the hierarchy $\mathcal{Q}_t^l$ using a fitness-to-region property, thus obtaining a foreground quality hierarchy $\mathbb{Q}_t = \left\{\mathcal{Q}_t^l\right\}_{l=1}^{L}$. The quality image Q_t^l of each level has the same size as $\mathcal{I}_t$ where each pixel is a score denoting its foreground quality. Subsequently, all levels of foreground qualities are combined to estimate a unique foreground quality image $\mathcal{Q}_t$ using a weighted average scheme based on the optical flow magnitude. This weighted average increases the importance of coarse levels in $\mathbb{H}_t$ for high optical flow magnitudes, as the presence of strong motion boundaries prevents an undesired foreground-background merging. Finally, we use a Conditional Random Field (CRF) to obtain an improved foreground mask $\mathcal{M}_t^*$ through an optimal labeling process that combines both foreground quality and spatial information. For simplifying notation, the temporal index is omitted in Figure 5.1 and in the following subsection.

5.3.2 Description

5.3.2.1 Motion-aware hierarchical segmentation

Merging superpixels to estimate semantically meaningful image regions containing objects is a common practice in the object proposal literature [Uijlings et al., 2013][Arbeláez et al., 2014][Xiao et al., 2015]. Building on such idea, we compute a motion-aware hierarchical image segmentation that extends over different degrees of details through each level partition into regions while preventing foreground-background merging.

A complete hierarchy of partitions can be defined as the set of all image segmentation results $\mathbb{H}' = \{\mathcal{R}^n\}_{n=1}^{N}$ where the level index n goes from the finest segmentation $\mathcal{R}^1$ (i.e. superpixels) to the coarsest segmentation $\mathcal{R}^N$ (i.e. complete image domain). The complete hierarchy can be understood as a dendrogram (tree) of regions where coarse levels are built merging regions from

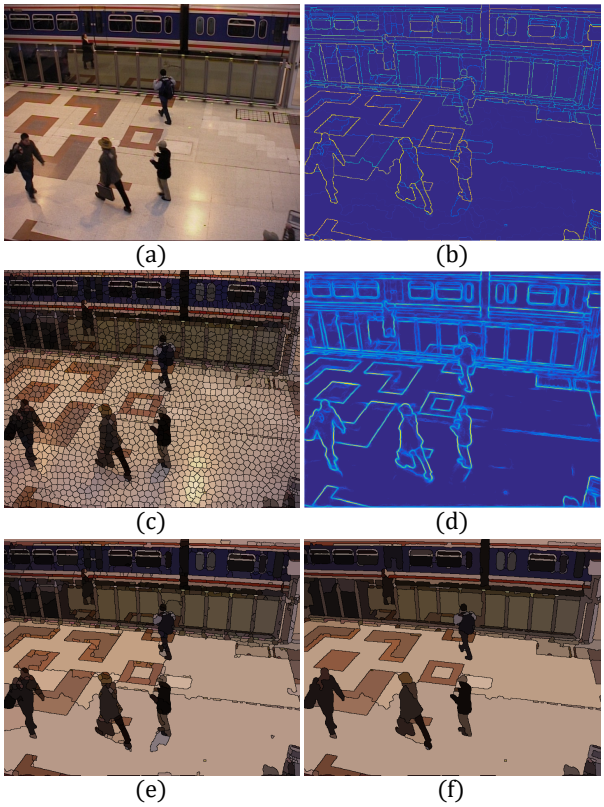


Figure 5.2: Example for the ultrametric contour map (UCM) Arbelaez et al. [2009]. The UCM (b) of an image (a) is obtained through superpixels (c) and their similarities (d), whereas different UCM thresholdings lead to different image segmentation partitions (e)-(f).

finer ones according to adjacent regions similarities. Such complete hierarchy can be computed through an ultrametric contour map (UCM) [Arbelaez et al., 2009], which is a boundary map that can be thresholded to obtain a set of closed boundaries containing segmented image regions. The lowest threshold leads to $\mathcal{R}^1$, while the highest threshold produces $\mathcal{R}^N$. Monotonically increasing the threshold merges the superpixels whose dissimilarity is under the threshold. Therefore, superpixels and their dissimilarities are required to compute the UCM by applying a greedy graph-based region merging algorithm [Arbelaez et al., 2009]. In particular, we have used the Piotr Dollar's proposal[2] which employs the mean boundary value [Dollár and Zitnick, 2013] as dissimilarity between SLIC based superpixels [Achanta et al., 2012]. Figure 5.2 presents an image (a), whose UCM [Arbelaez et al., 2009] (b) is extracted from superpixels (c) and dissimilarities defined by image boundaries [Dollár and Zitnick, 2013] (d). Therefore, thresholding the UCM with increasing values provides coarser partitions as presented in Figure 5.2 (e) and (f). We name this UCM based on color image properties as color-based UCM $\mathcal{U}^{col}$.

[2] `https://github.com/pdollar/edges`

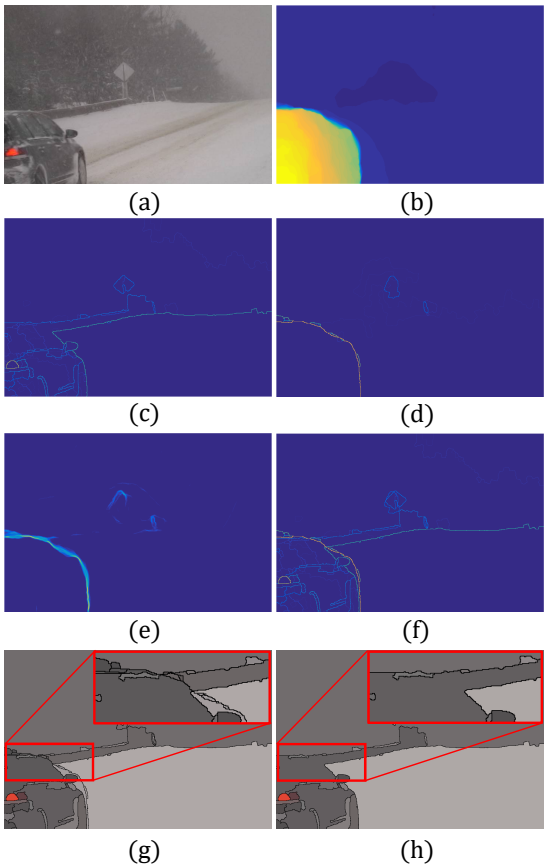


Figure 5.3: Example of motion-aware image segmentation. Given an image (a) and its associated optical flow magnitude (b), we compute, respectively, a color-based UCM $\mathcal{U}^{col}$ (c) and a motion-based UCM $\mathcal{U}^{mot}$ (d). This $\mathcal{U}^{mot}$ is obtained from motion boundaries (e) computed from the optical flow magnitude (b). Combining both UCMs we obtain a motion-aware color-based UCM $\mathcal{U}$ (f) that produces an image segmentation (g) with no foreground-background merging, unlike the direct use of $\mathcal{U}^{col}$ (h). The top-right rectangle of (g)(h) zooms an area to observe differences between merged regions.

While merging regions to fit foreground objects, merging between adjacent foreground regions is expected to occur before foreground-background merging. However, computing the hierarchy relying on appearance similarities as done by the color-based UCM $\mathcal{U}^{col}$ does not necessarily lead to the desired result (i.e. foreground and background not merged in the same regions). For example, in Figure 5.3 the color-based UCM $\mathcal{U}^{col}$ (c) of an image (a) lacks of boundaries in the top front part of a car due to color similarities with background regions. Therefore, we address such problem by including motion constraints to prevent foreground-background merging. We first create a motion-based UCM $\mathcal{U}^{mot}$ (see Figure 5.3(d)) based on per-pixel optical flow magnitude [Brox et al., 2004] (see Figure 5.3(b)) which defines moving object boundaries (see Figure 5.3(e)). To obtain $\mathcal{U}^{mot}$, we extract boundaries and superpixels over the optical

flow magnitude (replicated to 3 channels). Similarly to [Fragkiadaki et al., 2015], we do not re-train the boundary detector [Dollár and Zitnick, 2013] (trained for static image boundaries) as it effectively detects motion boundaries (see Figure 5.3(e)) and re-training may confuse the detector due to the misalignment of optical flow boundaries with the true image boundaries. Then, $\mathcal{U}^{mot}$ and $\mathcal{U}^{col}$ are combined into the motion-aware color-based UCM $\mathcal{U}$ (see Figure 5.3(f)):

$$\mathcal{U} = f_{ucm}\left(\mathcal{U}^{col}, \mathcal{U}^{mot}\right), \tag{5.1}$$

where $f_{ucm}(\cdot,\cdot)$ is the combination function applied to $\mathcal{U}^{mot}$ and $\mathcal{U}^{col}$. We propose a combination to keep only strong boundaries of the motion UCM $\mathcal{U}^{mot}$, thus obtaining the motion-aware color-based UCM $\mathcal{U}$ as:

$$\mathcal{U}^{\mathbf{p}} = \begin{cases} \max\left(\mathcal{U}^{\mathbf{p},col}, \mathcal{U}^{\mathbf{p},mot}\right) & if \quad \mathcal{U}^{\mathbf{p},mot} > \lambda^L \\ 0 & otherwise \end{cases}, \tag{5.2}$$

where $\mathbf{p}$ is the 2D pixel location in the UCM maps and λ^L is a threshold large enough to assure that only strong motion boundaries are added. This combination employs color merging while introducing only strong motion boundaries, thus preventing from over-segmentation due to weak motion boundaries that may appear. Therefore, the motion-aware color-based UCM $\mathcal{U}$ allows the computation of a complete hierarchy $\mathbb{H}'$ that prevents foreground-background merging.

Foreground segmentation requires foreground-background separation, thus we need each image region to contain foreground or background without merging both classes. This desired result does not occur for partitions close to $\mathcal{R}^N$ (i.e. partitions close to the complete image domain that tend to contain foreground and background merged), thus we sample the complete hierarchy to get a hierarchy $\mathbb{H} \subset \mathbb{H}'$ conformed by a subset of L levels (as introduced in Subsection 5.3.1) starting from the finest one. To that end, we threshold $\mathcal{U}$ to produce an image segmentation where foreground and background are not merged (see Figure 5.3(g)), whereas directly thresholding $\mathcal{U}^{col}$ merges both classes (see Figure 5.3(h)). We uniformly threshold $\mathcal{U}$ with L thresholds or levels ranging from the finest one (i.e. superpixels) to a maximum value. The result after applying the multiple thresholds is a motion-aware color segmentation hierarchy $\mathbb{H} = \left\{\mathcal{R}^l\right\}_{l=1}^{L}$ (see Figure 5.1), where each level l is composed by an image segmentation partition $\mathcal{R}^l$ obtained applying a threshold $\lambda^l = s(l-1)$ over $\mathcal{U}$ and s is the step between levels. We avoid using a single threshold λ generating a unique image segmentation that may have errors. Instead we consider selecting a number of levels L (i.e. $\left\{\lambda^l\right\}_{l=1}^{L}$) and defining the step between levels s to obtain each threshold λ^l (note that λ^L from Eq. 5.2 corresponds to the coarsest level threshold). Therefore, using a high (low) value of s means that there are less (more) λ^l possible values from the finest to the coarsest segmentation. Then, fixing the step between consecutive levels s and varying L reveals the effect of including more levels as analyzed in Subsection 5.4.2.1. This hierarchy $\mathbb{H}$ serves as the basis of the hierarchical foreground quality estimation, presented

in Subsection 5.3.2.2, to extend foreground for different image partitions.

5.3.2.2 Hierarchical foreground quality estimation

Based on the potential of image regions to estimate blob-level foreground quality [Ortego et al., 2017], we employ the property of fitness-to-regions to extend detected foreground blobs over foreground objects while removing erroneous foreground pixels. For each hierarchy level l, we compute a foreground quality for each region R_i^l as:

$$q_i^l = \frac{\sum\limits_{\mathbf{p} \in R_i^l} \mathcal{M}^{\mathbf{p}}}{|R_i^l|}, \tag{5.3}$$

where $|\cdot|$ denotes cardinality (i.e. $|R_i|$ is the number of pixels in region R_i) and $\mathcal{M}^{\mathbf{p}}$ is the pixel location $\mathbf{p}$ in the foreground segmentation mask $\mathcal{M}$ with values of 1(0) for foreground (background). This per-region quality q_i^l measures the fitness of the foreground mask to the region R_i through its percentage of foreground pixels. Therefore, the per-level foreground quality image is defined as an image $\mathcal{Q}^l = \left\{q_i^l\right\}_{i=1}^{k^l}$ with the same size of the image $\mathcal{I}$, where q_i^l is the quality per-region R_i^l and k^l is the number of regions in level l. Furthermore, the set of quality images per-level form a foreground quality hierarchy $\mathbb{Q} = \left\{\mathcal{Q}^l\right\}_{l=1}^{L}$ that is combined to obtain a unique foreground quality image as depicted in Subsection 5.3.2.3. Figure 5.4 shows examples of foreground qualities $\mathcal{Q}^l$ (g)-(i) extracted from fitness of the foreground segmentation mask $\mathcal{M}$ (b) of image $\mathcal{I}$ (a) to different segmentation partitions (d)-(f) of the hierarchy $\mathbb{H}$, having in fine (detailed) levels a weak spatial extension of the quality scores and high fitness to false positives of the foreground mask, while coarse levels enlarge regions covering foreground objects and diffusing foreground errors over background regions.

5.3.2.3 Weighted foreground quality combination

Given all the foreground quality images $\mathbb{Q} = \left\{\mathcal{Q}^l\right\}_{l=1}^{L}$, we obtain a unique foreground quality $\mathcal{Q}$ by combining all levels instead of selecting the best one, as such selection is not trivial. When no foreground-background merging is guaranteed, the coarsest level would be the best choice. However, stationary or slowly moving objects have, respectively, no motion boundaries or weak ones, thus easing foreground-background merging in coarse levels. Therefore, we perform a per-pixel weighted average to combine all levels by assigning different weights to each level based on the pixel optical flow magnitude.

In video sequences, we can distinguish between stationary objects or background and moving foreground objects through motion data. This premise has already been introduced in the hierarchy through strong motion boundaries provided by $\mathcal{U}^{\mathbf{p},mot}$ and it can also be used to

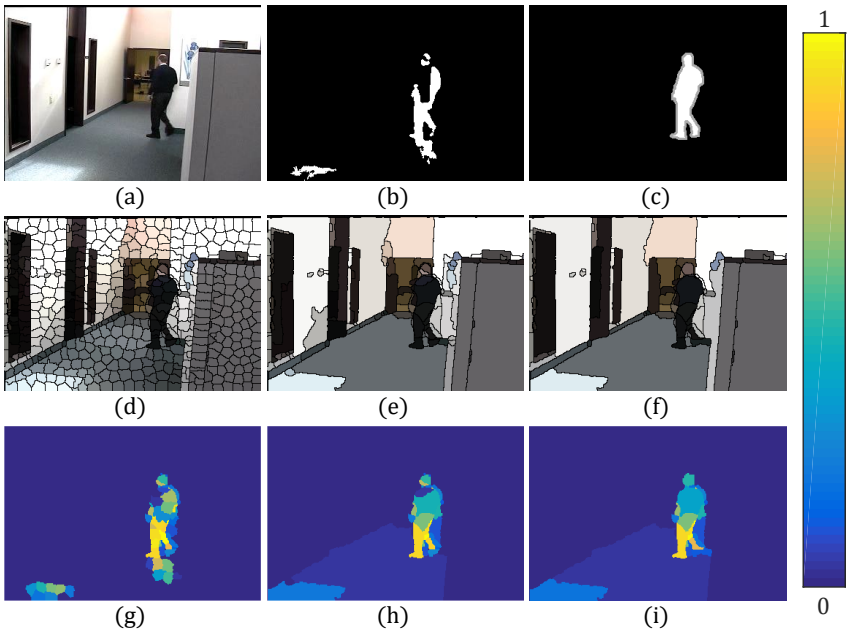


Figure 5.4: Hierarchical quality estimation. The image under analysis (a) has an associated foreground mask (b) that can be improved to accurately detect foreground as done by the ground-truth (c). The fitness between the foreground mask and the several image segmentation partitions (d)-(f) is the per-level quality $\mathcal{Q}^l$ shown in (g)-(i).

estimate $\mathcal{Q}$ through a weighted average as:

$$\mathcal{Q}^{\mathbf{p}} = \frac{\sum_l w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right)\mathcal{Q}^{l,\mathbf{p}}}{\sum_l w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right)}, \tag{5.4}$$

where $w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right)$ is the level l weighting function for pixel location $\mathbf{p}$ based on the optical flow magnitude $\|\mathcal{O}^{\mathbf{p}}\|$ associated to $\mathbf{p}$. We propose a weighting function linear with the level indexes and the motion values:

$$w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right) = \left[\frac{2d\,(l-1)-1}{m}\right]\|\mathcal{O}^{\mathbf{p}}\| + \left[1 - d\,(l-1)\right], \tag{5.5}$$

where $d = \frac{1}{L-1}$ and m is an upper bound for $\|\mathcal{O}^{\mathbf{p}}\|$ that assures maximum confidence in the coarsest level when there is enough motion (see Subsection 5.4.2.1 for an analysis of the effect of m in the performance). The higher the motion the higher the weight value for coarse levels (see the right subfigure in Figure 5.5(a)) where foreground is highly merged and the motion-aware UCM $\mathcal{U}$ has strong motion boundaries preventing foreground-background merging. However, for low $\|\mathcal{O}^{\mathbf{p}}\|$ values the combined UCM does not guarantee avoiding foreground-background merging, thus the coarser the level the lower the weight (see the left subfigure in Figure 5.5(a)) to reduce

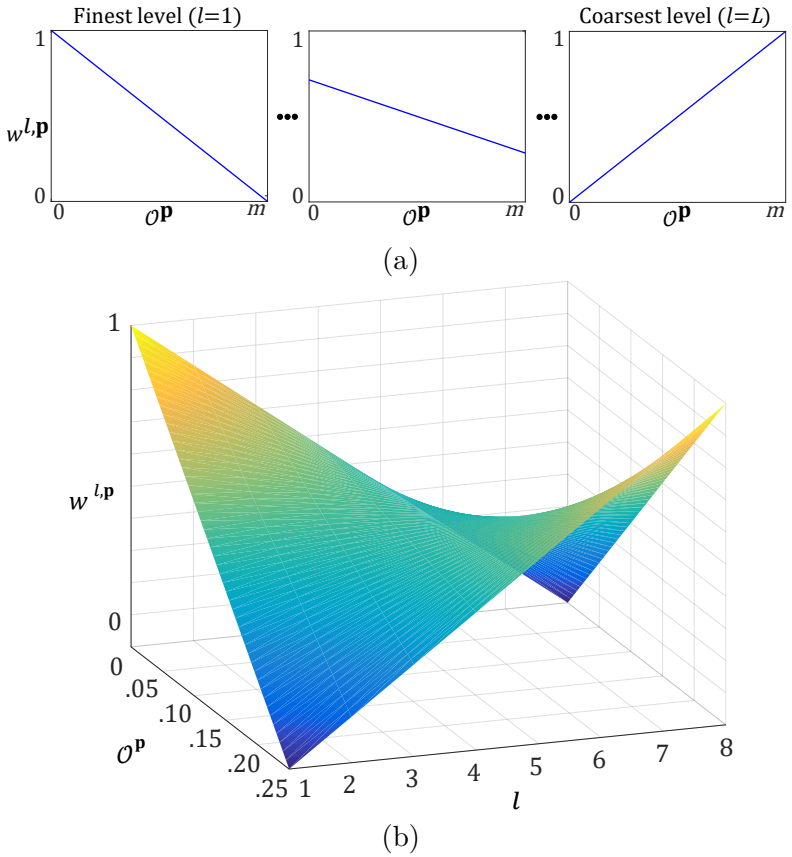


Figure 5.5: Weighting function to combine all hierarchy levels. Weights for each level are shown in (a), where the finest level (a)-left is weighted with maximum (minimum) weight in stationary (moving) pixels and the coarsest level (a)-right is weighted exactly in an opposite fashion. This assures that in cases of moving regions, where motion boundaries prevents from foreground-background merging, higher confidence is assigned to the coarsest level. The intermediate levels weights (a)-middle are defined to progressively move from the finest to the coarsest weight. In (b), the complete weighting function is presented with the parameters used, $L = 8$ and $m = 0.25$.

the contribution of coarse levels that may merge foreground and background. Therefore, the intermediate levels weights (see the middle subfigure in Figure 5.5(a)) range between the aforementioned finest and coarsest level weights. Additionally, the weighting function $w^{l,\mathbf{p}}(\|\mathcal{O}^{\mathbf{p}}\|)$ can be represented in 3D as depicted in Figure 5.5(b). In Figure 5.6 we present examples comparing the proposal and an equally weighted average (i.e. mean). In the first column, an image (a) and its foreground segmentation mask (b) contain a stationary person. The absence of motion induces a foreground-background merging that leads to the extension of scores out of the foreground area when equally weighting the per-level foreground qualities (c), whereas the proposed weighting palliates such merging by assigning a higher weight value to fine levels in absence of motion. Conversely, in the second column an image (e) contains moving people

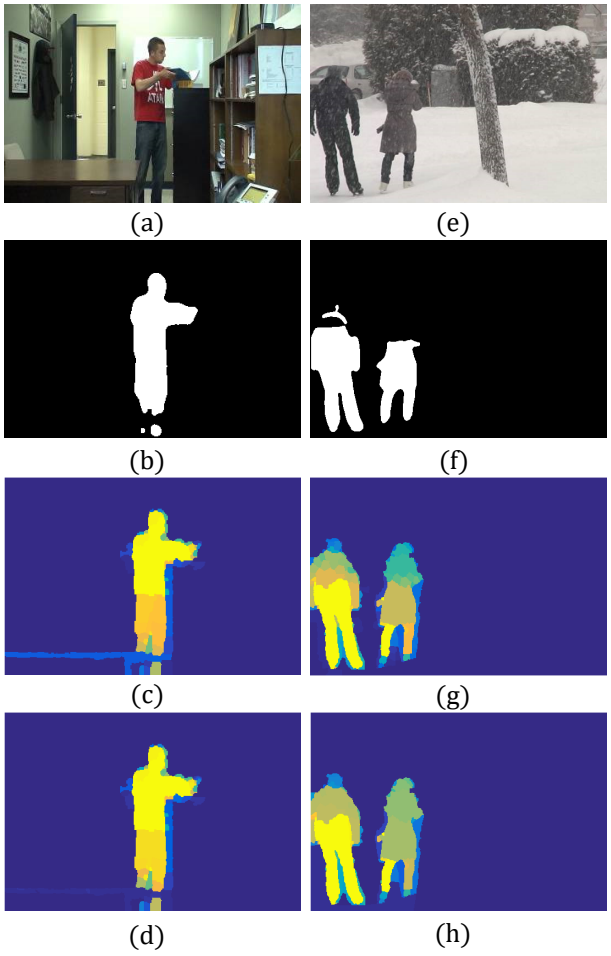


Figure 5.6: Example for the effect of the proposed weighted average. For each row, from top to bottom: images under analyses (a)(e), segmented foreground masks (b)(f) and foreground quality $\mathcal{Q}$ applying, respectively, an equally weighted average (c)(g) and the proposed weighting (d)(h).

that are not fully segmented in its foreground segmentation mask (f). The presence of motion allows improving the foreground quality obtained by applying equal weights (g) through the proposed weighting that assigns higher scores in the unsegmented top parts of the people due to the higher importance of coarse levels in presence of motion (h).

For scenarios with camera jitter, our assumption for the optical flow magnitude is not satisfied, leading to $\|\mathcal{O}^{\mathbf{P}}\|$ values exceeding m and therefore promoting coarse levels. In these cases we simply average all hierarchy qualities to compute $\mathcal{Q}$. The detection of frames affected by camera jitter is conducted using the average value of the temporal median of the optical flow magnitudes over large temporal windows.

5.3.2.4 Foreground improvement

Foreground mask improvement can be performed by thresholding the quality image $\mathcal{Q}$, as it expands over detected an undetected foreground regions. However, as motion boundaries used to restrict foreground-background merging are often not fitted to foreground object contours, a simple thresholding may add erroneous foreground pixels to the improved mask. Therefore, we introduce additional constraints to reduce such misclassifications near foreground object contours using a pixel-wise Conditional Random Field (CRF), which provides a robust framework to incorporate such constraints via spatial information potentials.

Using a CRF casts foreground segmentation into a binary pixel labeling problem, where a labeled image $\mathcal{C}$ has either foreground $\mathcal{C}^{\mathbf{p}} = 1$ or background $\mathcal{C}^{\mathbf{p}} = 0$ pixels. We use the fully-connected CRF model of [Krähenbühl and Koltun, 2011] to compute the optimal labeling $\mathcal{C}^*$ after an energy minimization process. The energy is defined over pixels and their labels as:

$$E\left(\mathcal{C}\right) = \sum_{\mathbf{p}\in\mathcal{I}} f_u\left(\mathcal{C}^{\mathbf{p}}\right) + \sum_{\mathbf{p}\in\mathcal{I}} \sum_{\mathbf{q}\in\mathbb{N}_{\mathbf{p}}} f_p\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right), \tag{5.6}$$

where f_u is a unary potential function to define the foreground probability, f_p is a pairwise potential function for labeling smoothness by penalizing neighboring pixels taking different labels and $\mathbb{N}_{\mathbf{p}}$ is the set of neighbors of pixel location $\mathbf{p}$.

For the pairwise potential f_p we use the model from [Krähenbühl and Koltun, 2011]:

$$f_p\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) = \mu\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) \left[w_1 \exp\left(-\frac{\left\|\mathbf{p}-\mathbf{q}\right\|^2}{2\sigma_\alpha^2} - \frac{\left\|\mathcal{I}^{\mathbf{p}}-\mathcal{I}^{\mathbf{q}}\right\|^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{\left\|\mathbf{p}-\mathbf{q}\right\|^2}{2\sigma_\gamma^2} \right) \right], \tag{5.7}$$

where each term is multiplied by $\mu\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) = 1$ if $\mathcal{C}^{\mathbf{p}} \neq \mathcal{C}^{\mathbf{q}}$ and zero otherwise to penalize locations with distinct labels; the first term is an appearance Gaussian kernel based on RGB and pixel location euclidean distances that aims to assign the same label to pixels with similar color and near positions; the second term is a Gaussian kernel dependent on pixel location euclidean distance to smooth the label assignment by removing isolated labels; the parameters σ_α, σ_β and σ_γ control the scale of the kernels; and w_1 and w_2 weight the contribution of each kernel to the pairwise potential. We heuristically set $\sigma_\alpha = 10$, $\sigma_\beta = 5$, $\sigma_\gamma = 3$, $w_1 = 1$ and $w_2 = 1$ that are all default parameters[3] in the implementation used, except σ_β that has been set to a smaller value in order to limit long range spatial connections that may decrease foreground segmentation performance due to similarities between foreground and background colors in the scene. The pairwise potential in [Krähenbühl and Koltun, 2011] was originally used for semantic segmentation in scenarios where foreground and background colors better define foreground and

[3] `https://github.com/johannesu/meanfield-matlab`

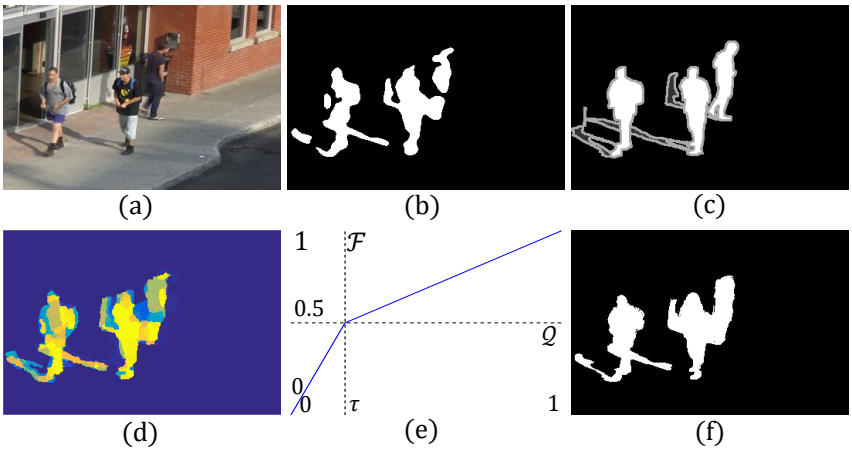


Figure 5.7: Example of the foreground segmentation process. An image $\mathcal{I}$ (a) with foreground segmentation mask $\mathcal{M}$ (b) has a ground-truth shown in (c). The foreground quality $\mathcal{Q}$ (d) is linearly transformed to resemble a probability (optimal foreground-background separation threshold τ for quality $\mathcal{Q}$ is mapped into 0.5 in $\mathcal{F}$) and compute an improved foreground mask $\mathcal{M}^*$(f) through a CRF.

background classes, thus higher σ_α and σ_β values lead to extremely accurate segmentation.

Moreover, we define the unary potential function f_u as:

$$f_u\left(\mathcal{C}^{\mathbf{p}}\right) = -\ln\left(\mathcal{F}^{\mathbf{p}}\right), \tag{5.8}$$

where $\mathcal{F}$ is a foreground probability estimated from the foreground quality image $\mathcal{Q}$. Such estimation is performed in order to transform $\mathcal{Q}$ into an information resembling a probability as needed by the CRF to correctly perform the foreground segmentation through maximum a posteriori inference. To that end, we perform a linear mapping between $\mathcal{Q}$ and $\mathcal{F}$ (see Figure 5.7(e)) as:

$$\mathcal{F}^{\mathbf{p}} = \begin{cases} \frac{0.5}{\tau}\,\mathcal{Q}^{\mathbf{p}}, & if \quad \mathcal{Q}^{\mathbf{p}} \leq \tau \\ \frac{0.5}{1-\tau}\,\mathcal{Q}^{\mathbf{p}} + \frac{0.5-\tau}{1-\tau}, & if \quad \mathcal{Q}^{\mathbf{p}} > \tau \end{cases}, \tag{5.9}$$

where τ is the foreground-background separation threshold associated to 0.5 foreground probability after the mapping (we analyze the effect of τ in the performance in Subsection 5.4.2.2). Note that linearly mapping fitness between superpixels and probability scores has been successfully performed in the literature [Schick et al., 2012].

Finally, we obtain the improved foreground mask $\mathcal{M}^*$ as the optimal labeling:

$$\mathcal{M}^* = \arg\min_{\mathcal{C}} E\left(\mathcal{C}\right). \tag{5.10}$$

Figure 5.7 depicts the foreground segmentation process of the image (a) given the foreground

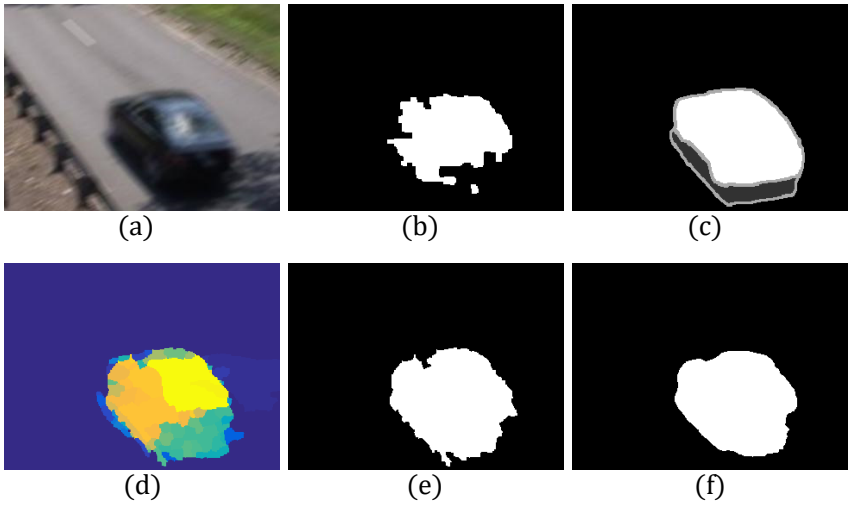


Figure 5.8: Example of foreground segmentation improvement when using or not the pairwise potential. An image $\mathcal{I}$ (a) with foreground segmentation mask $\mathcal{M}$ (b) has a ground-truth shown in (c). Maximum a posteriori inference over the foreground probability $\mathcal{F}$ (d) leads to the foreground segmentation shown in (e) when only $\mathcal{F}$ is used, whereas $\mathcal{M}^*$ (f), obtained through the CRF that considers spatial information, produces a better foreground mask.

quality $\mathcal{Q}$ (d) of the foreground segmentation mask $\mathcal{M}$ (b) with associated ground-truth (c). The foreground quality $\mathcal{Q}$ is transformed into a foreground probability $\mathcal{F}$ using the linear transformation shown in Figure 5.7(e) to compute the improved foreground mask $\mathcal{M}^*$ in Figure 5.7(f). Note that in the example presented in Figure 5.7 τ is set to 0.2. Furthermore, in Figure 5.8 an example image (a) is segmented with errors (b) compared to ground-truth (c) and has an estimated foreground probability $\mathcal{F}$ (d) that leads to different improved foreground masks (e)-(f) depending on the technique applied. The foreground mask presented in (e) is obtained through maximum a posteriori inference over $\mathcal{F}$ (i.e. thresholding over 0.5 without considering the pairwise potential), thus leading to errors in the object contours that are mostly solved in $\mathcal{M}^*$ (f) as it jointly considers the unary and pairwise potentials via the CRF framework.

5.4 Experimental work

5.4.1 Experimental methodology

We use real and synthetic sequences from four datasets: the well-known CDNET2014 dataset [Wang et al., 2014b], the recent LASIESTA dataset [Cuevas et al., 2016] and the synthetic datasets SABS [Brutzer et al., 2011] and BMC [Vacavant et al., 2013]. These datasets contain common BS challenges with their corresponding ground-truth data. For CDNET2014, we select eight of the eleven categories (*PTZ*, *Thermal* and *Turbulence* are excluded) as the pro-

Algorithm	Model type description	Features	Dataset			
			CDNET	LASIESTA	SABS	BMC
GMM	Mixture of Gaussians	C	✓	✓	✓	✓
KDE	Non-parametric kernel	C		✓	✓	✓
MLAYER	Layer-based	C, T		✓	✓	✓
FuzzySOM	Self-organized neural network	C		✓	✓	✓
SC-SOBS	Self-organized neural network	C	✓			
CwisarDH	Weightless neural network	C	✓			
Spectral-360	Dichromatic reflection model	C	✓			
FTSG	Flux tensor and mixture of Gaussians	C, M	✓			
LOBSTER	Non-parametric sample-based	C, T		✓	✓	✓
AMBER	Multi-resolution temporal templates	C, T	✓			
SharedModel	Mixture of Gaussians	C	✓			
SuBSENSE	Non-parametric sample-based	C, T	✓	✓	✓	✓
MBS	Single Gaussian of multiple features	C	✓			
PAWCS	Non-parametric sample-based	C, T	✓			
WeSamBE	Non-parametric sample-based	C	✓			

Table 5.1: Background subtraction algorithms selected to validate the improvement obtained by the proposed post-processing framework. Key: C: Color. T: Texture. M: Motion.

posed framework has been designed for color images in stationary camera scenarios, thus using 40 sequences (113848 frames). For LASIESTA, we select both indoor and outdoor sequences discarding those involving moving cameras (*MC Moving Camera* and the first three sequences of *SM Simulated Motion*), thus using 38 sequences (16250 frames). For the SABS synthetic dataset, we select 8 of the 12 sequences (6400 frames) and discard 4 out of 5 sequences with different compression qualities. For the BMC synthetic dataset, we use 10 sequences from the learning category (14990 frames). We do not use the rest due to the extremely low availability of ground-truth for long sequences. Note that we do not use unconstrained video object segmentation datasets [Prest et al., 2012][Perazzi et al., 2016] as they consider that moving objects may not be part of the foreground.

To apply the proposed post-processing framework, we analyze the datasets with several algorithms (see Table 5.1 for a brief summary) to demonstrate that the improvement achieved is generalizable: GMM [Stauffer and Grimson, 1999] KDE [Elgammal and Davis, 2000], MLAYER Yao and Odobez [2007], FuzzySOM [Maddalena and Petrosino, 2010], SC-SOBS [Maddalena and Petrosino, 2012], CwisarDH [Gregorio and Giordano, 2014], Spectral-360 [Sedky et al., 2014], FTSG [Wang et al., 2014a], LOBSTER [St-Charles and Bilodeau, 2014], AMBER [Wang and Dudek, 2014], SharedModel [Chen et al., 2015], SuBSENSE [St-Charles et al., 2015], MBS [Sajid and Samson Cheung, 2015], PAWCS [St-Charles et al., 2016] and WeSamBE [Jiang and Lu, 2017]. We have selected this set of algorithms to demonstrate the framework capability to improve results from low to top performance algorithms. We use the results provided in CDNET2014, whereas we employ the BGSlibrary [Sobral and Vacavant, 2014] to run selected

		m				Mean	$\%\Delta Fs$
		0.25	**0.5**	**0.75**	**1**		
L	**1**	.7852	.7852	.7852	.7852	.7852	-
	2	.7911	.7909	.7908	.7906	.7908	0.70
	4	.7958	.7954	.7952	.7950	.7953	0.57
	8	.8011	.8006	.8003	.8001	.8005	0.65
	12	.8044	.8040	.8038	.8035	.8039	0.42
	16	.8069	.8066	.8066	.8065	.8067	0.34
	32	.8005	.8015	.8027	.80394	.8021	-0.57
	64	.5062	.5331	.5555	.5705	.5413	-32.51

Table 5.2: Example of the effect of L and m in the F-score. The higher L, the better the performance until too coarse levels are used and foreground-background merging occurs (see L=32 and L=64). The selection of the parameter m has low impact in the F-score. $\%\Delta Fs$ denotes the improvement percentage in terms of average F-score. Note that $\tau = 0.25$ is used for the experiment.

algorithms in the remaining datasets. We do not consider recently emerged deep learning models [W. et al., 2017][Ang Lim and Yalim Keles, 2018] as they currently rely on the ground-truth data for training from the same sequences in which tests are performed. Also, we have selected a top (SuBSENSE) and a low (GMM) performing algorithm across all datasets to compare performance among databases.

To assess the algorithms performance, we use standard Precision (Pe), Recall (Re) and F-score (Fs) based on pixel-level comparisons between foreground segmentation masks and ground-truth. These measures are computed as:

$$Pe = TP/\left(TP + FP\right), \tag{5.11}$$

$$Re = TP/\left(TP + FN\right), \tag{5.12}$$

$$Fs = 2 \cdot Pe \cdot Re/\left(Pe + Re\right), \tag{5.13}$$

where TP, FP and FN are, respectively, correct, false and missed detection pixels (as compared to ground-truth ones).

5.4.2 Effect of parameters in performance improvement

5.4.2.1 Number of levels and optical flow bounding

The use of a hierarchy to extend over foreground objects is one of the main contributions of this chapter. This hierarchy has a predefined number of levels that are combined using a weighted average dependent on the upper bound m for $\|\mathcal{O}^{\mathbf{P}}\|$. We present in Table 5.2 the impact of these parameters values in the average F-score of six sequences from CDNET2014

dataset (*skating, highway, canoe, winterDriveway, tramCrossroad_1fps and cubicle*) segmented with SuBSENSE and GMM. Firstly, a higher number of hierarchy levels L leads to higher performance due to larger extensions of uncompleted foreground objects and the removal of more erroneous foreground pixels through low fitness-to-regions values. Secondly, the value of m is related to the optical flow magnitude and the importance given to coarse levels. The lower the value the better, but its value has little impact in the performance. Attending to Table 5.2, we have used L=8 due to its pick of performance increment ($\%\Delta L$) and $m = 0.25$ as it provides slightly better results than the rest of the values analyzed. Additionally, we have heuristically set the step s to 0.015, thus leading to the coarsest level L=8 using a threshold $\lambda^L = 0.105$. Note that heuristically setting the number of levels and using a step to threshold an UCM are common practices in the literature [Yan et al., 2013][Liu et al., 2014].

5.4.2.2 Linear mapping

The transformation of foreground quality to foreground probability is done through a linear mapping guided by parameter τ (see Eq. 5.9). Therefore, we sweep the value of $\tau \in [0, 1]$ to find out how its value affects the improvement capabilities (using $L = 8$, $m = 0.25$). In particular, we compare the original algorithm performance against the performance obtained by the proposed improvement framework when only a unary or both a unary and a pairwise potential are used in the CRF energy function.

We have performed this experiment in CDNET2014 (using CwisarDH, SuBSENSE, AMBER, MBS, FTSG, SC-SOBS and GMM) and in LASIESTA (using the six algorithms evaluated) datasets. For space constraints, we have selected SuBSENSE and AMBER and SuBSENSE and FuzzySOM as top and medium performance algorithms, respectively, in CDNET2014 and LASIESTA datasets. The remaining algorithm results are available online (`http://www-vpu.eps.uam.es/publications/HFI/`). Figure 5.9 presents the average performance achieved in terms of Pe, Re and Fe (columns) for each pair of algorithms (rows) in CDNET2014 and LASIESTA datasets, respectively. In general terms, using a unary potential alone (superscript *1 in the figures) improves recall for low values of τ (approximately between 0.1 and 0.5), thus supporting the capability to extend over foreground objects. However, this recall improvement comes with the reduction of the precision due to contour-inaccurate partitions in the motion-aware hierarchy that lead to an extension of foreground masks not fitted to objects contours. This precision reduction is overcome by including the pairwise potential in the CRF energy function (superscript *2 in the figures), which is able to fit foreground masks to object contours while keeping and improved recall (see Figure 5.8). Therefore, as shown in Figure 5.9, we can conclude that a good value of τ is approximately between 0.2 and 0.3 as both precision and recall are improved and the CRF with both unary and pairwise potentials outperforms the use of the unary potential alone, thus we select the unary and pairwise based CRF to present the results

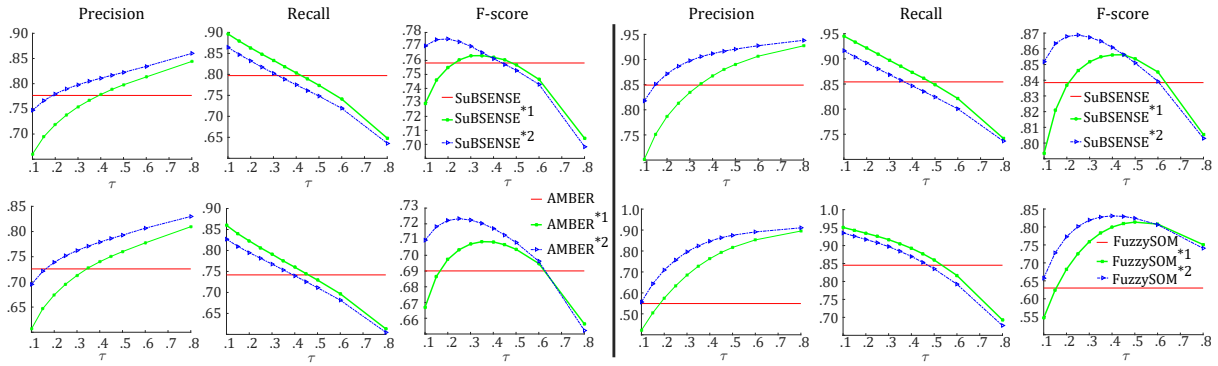


Figure 5.9: Examples of the effect of τ parameter in the performance of SuBSENSE [St-Charles et al., 2015] and AMBER [Wang and Dudek, 2014] and SuBSENSE [St-Charles et al., 2015] and FuzzySOM [Maddalena and Petrosino, 2010] algorithms for, respectively, CDNET2014 (left) and LASIESTA (right) datasets. Each row denotes an algorithm, whereas each column presents, respectively, the average Precision (Pe), Recall (Re) and F-score (Fs) in the dataset. In each figure, the red line denotes the performance of the algorithm in the dataset, the green line with dots is the performance achieved by applying maximum a posteriori inference only using the foreground probability $\mathcal{F}$ (*1) and the blue line with triangles is the performance using both $\mathcal{F}$ and the pairwise potential (*2).

	LASIESTA			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.8491	.8542	.8385	3.60
SuBSENSE*	.8867	.8801	.8687	
LOBSTER	.6899	.8204	.7159	6.90
LOBSTER*	.7416	.8534	.7650	
FuzzySOM	.5491	.8452	.6299	27.20
FuzzySOM*	.7572	.9077	.8011	
MLAYER	.6514	.8237	.6749	6.13
MLAYER*	.6916	.8541	.7163	
GMM	.3227	.9234	.4134	17.30
GMM*	.4001	.9784	.4849	
KDE	.3792	.9493	.5013	36.7
KDE*	.5947	.9626	.6853	

	SABS			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.7138	.7786	.6740	0.77
SuBSENSE*	.7125	.7961	.6792	
LOBSTER	.7429	.6834	.6555	5.00
LOBSTER*	.7483	.7511	.6883	
FuzzySOM	.4375	.5716	.4861	17.63
FuzzySOM*	.5813	.6618	.5718	
MLAYER	.5392	.7549	.6237	11.45
MLAYER*	.6012	.8315	.6951	
GMM	.5532	.6481	.5685	11.72
GMM*	.6690	.7020	.6351	
KDE	.3369	.7388	.4536	24.21
KDE*	.4658	.8000	.5634	

	BMC			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.8420	.8706	.8494	0.21
SuBSENSE*	.8568	.8597	.8512	
LOBSTER	.8024	.7757	.7359	2.51
LOBSTER*	.8222	.7835	.7544	
FuzzySOM	.7099	.8083	.7166	7.31
FuzzySOM*	.7544	.8434	.7690	
MLAYER	.7686	.8222	.7500	1.96
MLAYER*	.8013	.8171	.7647	
GMM	.6789	.8833	.7448	7.73
GMM*	.7518	.9005	.8024	
KDE	.4934	.7848	.5395	46.45
KDE*	.8123	.8222	.7901	

Table 5.3: Overall average performance for each analyzed algorithm and the proposed improvement in LASIESTA, SABS and BMC datasets. %ΔFs denotes the improvement percentage achieved for F-score.

in the following subsections.

5.4.3 Improvement over the original algorithms in CDNET2014, LASIESTA, SABS and BMC datasets

We present the improvement in all datasets results for a fixed configuration of $L = 8$, $m = 0.25$ and $\tau = 0.25$. In Table 5.3, we show the average performance results in terms of Pe, Re and Fs, together with the percentage increases of Fs for LASIESTA, SABS and BMC datasets. In these datasets, improvements are obtained for all algorithms on average and we present some

	Baseline			Bad Weather			Camera Jitter			Dynamic Background			Intermittent Object Motion		
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs
PAWCS	.9394	.9408	.9397	.9379	.7091	.8059	.8660	.7840	.8137	.9038	.8868	.8938	.8392	**.7487**	.7764
PAWCS*	**.9397**	**.9525**	**.9420**	.9370	**.7950**	**.8576**	**.8732**	**.8078**	**.8213**	**.9194**	**.9018**	**.9074**	**.9302**	.7200	**.8021**
FTSG	.9170	.9513	.9330	.9192	.7393	.8184	.7645	.7717	.7513	.9129	.8691	.8792	.8512	.7813	.7891
FTSG*	.9125	**.9606**	**.9352**	**.9413**	**.8244**	**.8769**	.7753	**.8204**	**.7664**	**.9303**	**.8824**	**.8974**	.8432	**.7873**	.7850
SuBSENSE	.9495	.9520	.9503	.9168	.8121	.8594	.8115	.8243	.8152	.8915	.7768	.8177	.7957	.6578	.6569
SuBSENSE*	.9430	**.9610**	**.9514**	**.9267**	**.8672**	**.8944**	**.8247**	**.8794**	**.8498**	**.9371**	**.7982**	**.8539**	**.8156**	.6270	.6414
SharedModel	.9502	.9545	.9522	.8559	.8387	.8439	.8377	.7960	.8141	.9198	.7597	.8222	.7587	.7182	.6727
SharedModel*	.9419	**.9669**	**.9541**	**.8649**	**.8840**	**.8706**	**.8474**	**.8376**	**.8377**	**.9400**	**.7768**	**.8384**	**.8002**	**.7252**	**.6930**
WeSamBE	.9422	.9422	.9413	.9184	.8017	.8531	.8395	.7777	.7976	.8933	.6796	.7440	.7888	.7472	.7392
WeSamBE*	.9356	**.9589**	**.9466**	**.9281**	**.8598**	**.8908**	**.8660**	**.8354**	**.8417**	**.9283**	**.6819**	**.7656**	**.8093**	.7254	.7381
Spectral-360	.9065	.9616	.9330	.8621	.7175	.7769	.8387	.6696	.7142	.8456	.7819	.7766	.7374	.5878	.5609
Spectral-360*	**.9105**	**.9709**	**.9395**	**.8756**	**.7878**	**.8242**	.8341	**.7306**	**.7471**	**.8906**	**.8085**	**.8317**	**.7804**	.5783	.5518
MBS	.9431	.9158	.9287	.7652	.8312	.7802	.8443	.8321	.8367	.8606	.7637	.7904	.8201	.6386	.7092
MBS*	.9389	**.9330**	**.9356**	**.8354**	**.8780**	**.8483**	**.8727**	**.8857**	**.8788**	**.8950**	**.8045**	**.8169**	**.9403**	**.6069**	**.7132**
AMBER	.8980	.8784	.8813	.9010	.6782	.7698	.8493	.6505	.7107	.7990	.9177	.8436	.7530	.7617	.7211
AMBER*	**.9067**	**.8913**	**.8925**	**.9297**	**.7854**	**.8460**	**.8636**	**.7230**	**.7579**	**.8373**	**.9358**	**.8740**	**.7891**	**.7706**	**.7366**
CwisarDH	.9337	.8972	.9145	.9173	.6697	.7477	.8516	.7437	.7886	.8499	.8144	.8274	.7417	.5549	.5753
CwisarDH*	.9322	**.9577**	**.9446**	**.9412**	**.7391**	**.8004**	**.8809**	**.832**	**.8513**	**.9248**	**.8878**	**.9019**	**.7923**	**.5905**	**.6008**
SC-SOBS	.9341	.9327	.9333	.8412	.5655	.6605	.6286	.8113	.7051	.6283	.8918	.6686	.5896	.7237	.5918
SC-SOBS*	**.9384**	**.9524**	**.9452**	**.8735**	**.6850**	**.7589**	**.7085**	**.8463**	**.7647**	**.6805**	**.9255**	**.7241**	**.8039**	.7065	**.6660**
GMM	.8461	.8180	.8245	.8285	.7152	.7662	.5126	.7334	.5969	.5989	.8344	.6330	.6688	.5142	.5207
GMM*	**.8670**	**.8581**	**.8569**	**.8951**	**.8245**	**.8572**	**.6188**	**.7972**	**.6759**	**.7180**	**.9020**	**.7301**	**.7345**	**.5488**	**.5503**

	Low Framerate			Night Videos			Shadows			Average			
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	%ΔFs
PAWCS	.6285	.7555	.6433	.5559	.3929	.4171	.8710	.9172	.8913	.8179	.7669	.7726	1.7
PAWCS*	.6285	**.7702**	**.6512**	**.5570**	**.3984**	.4044	.8628	**.9470**	**.9000**	**.8310**	**.7866**	**.7857**	
FTSG	.6996	.7547	.6563	.4179	.6873	.5043	.8535	.9214	.8832	.7920	.8095	.7768	2.0
FTSG*	**.7087**	**.7669**	**.6673**	**.4268**	**.7196**	**.5158**	.8503	**.9561**	**.8973**	**.7985**	**.8397**	**.7926**	
SuBSENSE	.6276	.8435	.6594	.4224	.6494	.4918	.8646	.9419	.8986	.7849	.8072	.7687	2.1
SuBSENSE*	**.6353**	**.8600**	**.6763**	**.4317**	**.6832**	**.5083**	.8602	**.9596**	**.9041**	**.7968**	**.8294**	**.7849**	
SharedModel	.7362	.8342	.7696	.4030	.5810	.4663	.8455	.9445	.8898	.7884	.8033	.7788	2.3
SharedModel*	**.7614**	**.8517**	**.7950**	**.4159**	**.6181**	**.4864**	.8442	**.9651**	**.8981**	**.8020**	**.8282**	**.7967**	
WeSamBE	.6459	.8768	.6884	.4683	.6429	.5335	.8686	.9401	.8999	.7956	.8010	.7746	2.4
WeSamBE*	**.6535**	**.8966**	**.7072**	**.4797**	**.6724**	**.5520**	.8596	**.9560**	**.9017**	**.8075**	**.8233**	**.7930**	
Spectral-360	.6666	.7349	.6977	.3605	.7113	.4553	.8187	.8898	.8519	.7545	.7568	.7208	3.1
Spectral-360*	**.7351**	**.7616**	**.7425**	.3485	**.7367**	.4491	**.8247**	**.9046**	**.8620**	**.7749**	**.7849**	**.7435**	
MBS	.8864	.6727	.6754	.4716	.5049	.4834	.8063	.7762	.7784	.7997	.7419	.7478	3.6
MBS*	**.9192**	**.6853**	**.6810**	**.5049**	**.5373**	**.5137**	.8015	**.8431**	**.8111**	**.8391**	**.7717**	**.7748**	
AMBER	.5943	.4727	.4338	.3149	.6498	.3593	.8098	.8297	.8128	.7399	.7298	.6916	5.1
AMBER*	**.6534**	**.4805**	**.4859**	**.3303**	**.7004**	**.3799**	**.8199**	**.8818**	**.8431**	**.7662**	**.7711**	**.7270**	
CwisarDH	.7421	.6659	.6986	.4442	.4511	.3753	.8476	.8786	.8581	.7910	.7094	.7232	6.6
CwisarDH*	**.8407**	**.7783**	**.7962**	**.5132**	**.4948**	**.3801**	**.8569**	**.9395**	**.8927**	**.8353**	**.7775**	**.7710**	
SC-SOBS	.5451	.7844	.5565	.3303	.6225	.3841	.7230	.8502	.7786	.6506	.7727	.6586	8.3
SC-SOBS*	**.6290**	**.8688**	**.6325**	**.3585**	**.6826**	**.4142**	**.7376**	**.9188**	**.8149**	**.7162**	**.8232**	**.7151**	
GMM	.6997	.5643	.5284	.3300	.5531	.3793	.7156	.7960	.7370	.6500	.6911	.6232	10.4
GMM*	**.7824**	**.6226**	**.6539**	**.3417**	**.6170**	**.4001**	**.7340**	**.8678**	**.7785**	**.7114**	**.7548**	**.6879**	

Table 5.4: Per-category average foreground segmentation performance achieved by the proposed framework in CDNET2014 dataset. Bold denotes better performance of the proposed improvement (*). The Average column denotes the average performance across all categories, being $\%\Delta Fs$ the improvement percentage achieved in terms of average F-score.

examples of these improvements in Figures 5.10 and 5.11 for, respectively, LASIESTA and SABS and BMC datasets. Moreover, we present per-category and overall performance results for the CDNET2014 dataset in Table 5.4. Note that an improvement of around 2% for top algorithms in CDNET2014 (SuBSENSE, FTSG, WeSamBE or SharedModel) is a significant one as the

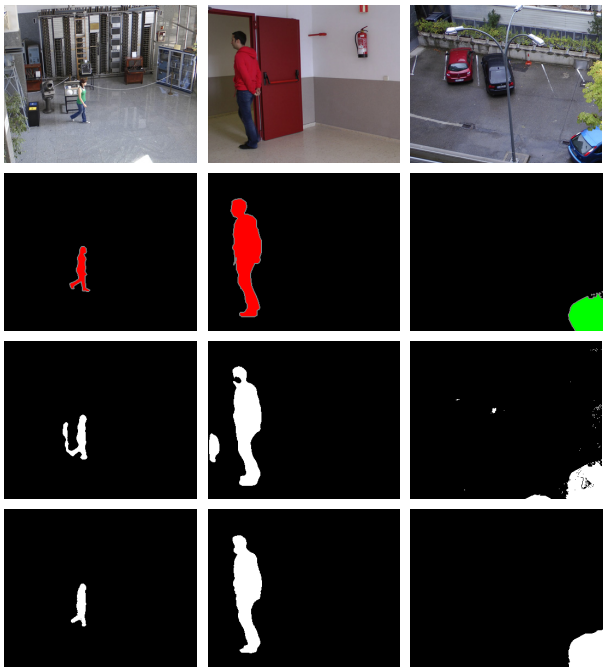


Figure 5.10: Example of foreground improvements in LASIESTA dataset. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. Form left to right, example for: SuBSENSE in frame 72 of *I_BGS_02* sequence (*Bootstrap* category), LOBSTER in frame 301 of *I_CA_01* sequence (*Camouflage* category) and FuzzySOM in frame 984 of *O_RA_01* sequence (*Rainy* category).

percentage between the first and fifth performing unsupervised algorithms in CDNET2014[4] is 2.6%. From the tables it can be observed that results are better than the original performance for almost all algorithms and categories in CDNET2014 (see Table 5.4) with the exception of the *Intermittent Object Motion* category for top-performing algorithms (SuBSENSE, FTSG and WeSamBE), where there are weak decreases in performance due to the static nature of most of the foreground objects in this category. This stationarity leads to no foreground-background merging prevention when performing the hierarchical image segmentation, thus foreground probabilities are easily expanded over background regions and foreground regions are less extended to undetected areas due to the lower importance of high levels in the hierarchy when combining the quality images. Additionally, Figure 5.12 presents some examples of improvements achieved in CDNET2014. Please, see online (`http://www-vpu.eps.uam.es/publications/HFI/`) all the foreground masks and complete performance results.

	Baseline			Camera jitter			Dynamic Background			Intermittent Object Motion			Shadows		
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs
SOBS	.9313	.9193	.9251	.6399	.8007	.7086	.5856	.8798	.6439	.5531	.7057	.5628	.7219	.8355	.7717
SOBS+	.9261	.9319	.9289	.7009	.8211	.7502	.6576	.8955	.6960	.5727	.7010	.5645	**.7281**	.8736	.7907
SOBS*	**.9382**	**.9527**	**.9453**	**.7474**	**.8446**	**.7834**	**.6983**	**.9303**	**.7463**	**.7146**	**.7147**	**.6128**	.7198	**.9202**	**.8022**

Table 5.5: Performance comparison of the proposed framework (*) against [Schick et al., 2012] (+) in categories with data available for [Schick et al., 2012].

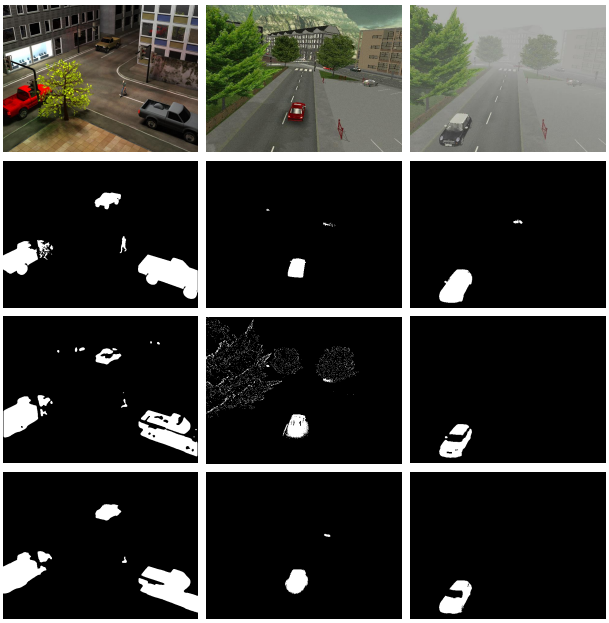


Figure 5.11: Example of foreground improvements in SABS and BMC datasets. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. Form left to right, example for: LOBSTER in frame 544 of *Bootstrap* sequence (SABS), FuzzySOM in frame 484 of *511* sequence (BMC) and LOBSTER in frame 918 of *411* sequence (BMC).

5.4.4 Comparison against the state-of-the-art

We compare our improvement capabilities against available similar approaches in the literature, i.e. approaches aiming to improve foreground masks from a model-independent perspective. In particular, we present in Table 5.5 the improvements over SOBS algorithm [Maddalena and Petrosino, 2008] (a previous version of the already evaluated SC-SOBS algorithm) for the state-of-the-art post-processing algorithm [Schick et al., 2012] (marked with +) and the proposed framework using $\tau = 0.25$ (marked with *). We use SOBS algorithm in 5 categories of CD-NET2014 as that are the categories and algorithm with available results. Despite the use of fitness-to-regions by [Schick et al., 2012], we achieve a superior improvement as we introduce a hierarchical approach that enables the extension of the segmented foreground masks to unde-

[4] `http://changedetection.net/`

	IUTIS-5			FusedQ		
	Pe	Re	Fs	Pe	Re	Fs
Baseline	.9464	.9680	.9567	.9197	.9793	.9484
Bad Weather	.9349	.7503	.8289	.9384	.8422	.8857
Camera Jitter	.8511	.8220	.8332	.8552	.8216	.8209
Dynamic Background	.9324	.8636	.8902	.9357	.9257	.9297
Intermittent Object Motion	.8501	.7047	.7296	.8304	.7590	.7532
Low Framerate	.7724	.8376	.7911	.7571	.8489	.7951
Night Videos	.4578	.6333	.5132	.3995	.7691	.4922
Shadows	.8766	.9492	.9084	.8545	.9651	.9039
Average	**.8300**	.8161	.8064	.8118	**.8643**	**.8161**

Table 5.6: Per-category average performance in CDNET2014 achieved by the proposed combination (FusedQ) compared to the combination strategy UITIS-5 [Bianco et al., 2017].

tected foreground areas while fitting to object contours. We have a higher improvement in all categories attending to Pe, Re and Fs and we only perform worse for Pe in Shadow category, where we decrease in a 0.3% the original Pe performance. Note that we only compare our post-processing improvement capabilities against [Schick et al., 2012] as other model-independent post-processing works, such as [Giordano et al., 2015] and [Raman et al., 2017], do not provide code to reproduce the complete post-processing technique.

5.4.5 Applying foreground quality to algorithm combination

Recently, combining BS algorithms results demonstrated to obtain substantially better results [Bianco et al., 2017]. Adopting this idea, we present here a potential use of the foreground quality image as the information to guide the algorithm combination. For each frame we average the foreground quality images from a set of algorithms and we use that image as the quality $\mathcal{Q}$ to feed to foreground improvement from Subsection 5.3.2.4. Despite being a simple combination, we outperform the IUTIS-5 algorithm [Bianco et al., 2017] as presented in Table 5.6. Note that IUTIS-5 combines the algorithms SuBSENSE, FTSG, CwisarDH, Spectral-360 and AMBER so we have used these five algorithms to average their foreground quality images.

5.4.6 Discussion

The results obtained in this section confirms that the proposed hierarchical fitness-to-regions strategy is effective for algorithm improvement. This capacity comes from its robustness to different challenges or distortions that typically affect background subtraction. Basically, regarding the distortions that produce false positives (e.g. dynamic backgrounds, camera jitter, shadows, illumination changes or ghost artifacts), a corresponding foreground quality image tends to produce low scores due to a low percentage of foreground pixels compared to the size

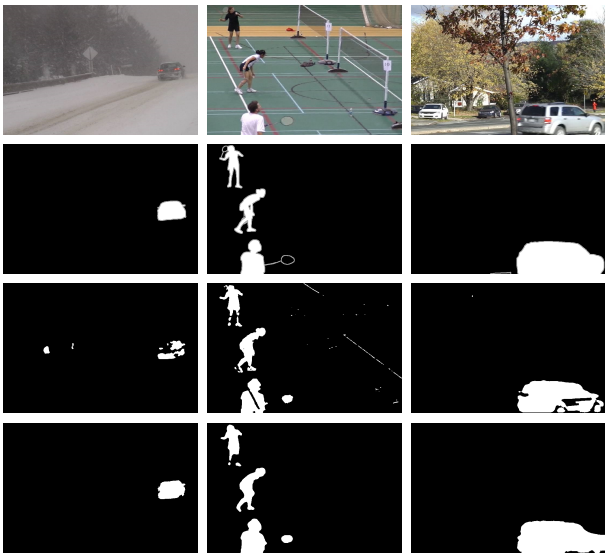


Figure 5.12: Example of foreground improvements in CDNET2014 dataset. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. From left to right, example for: FTSG in frame 956 of *snowFall* sequence (*Bad Weather* category), SC-SOBS in frame 1071 of *badminton* sequence (*Camera Jitter* category) and SuBSENSE in frame 2063 of *fall* sequence (*Dynamic Background* category).

of the corresponding segmented image regions in which that foreground is (i.e. low fitness). Furthermore, false negatives are typically induced by camouflages, challenge that the proposed framework overcomes using motion constraints to allow extending partially detected objects without merging with background regions. Moreover, a small image degradation like noise or compression should not substantially affect the proposed framework as modern optical flow is robust to these issues [Baker et al., 2011] and the key ingredient to keep the performance is to be able to delimit objects in the motion-aware color-based UCM $\mathcal{U}$, task supported by the strong boundaries extracted from the optical flow magnitude.

However, despite the aforementioned good results, fitness-to-regions has two main limitations. Firstly, foreground objects with weak foreground qualities are removed by our approach, which means that we cannot deal with extremely uncertain cases for reconstructing an entire object from few pixels. Secondly, fitness-to-regions may lead to errors when a complete background object is almost detected as foreground (i.e. a false positive), as high qualities may be obtained.

Moreover, the computational cost of the proposed approach is mainly due to the optical flow, the UCMs and the CRF optimization that require approximately 80% of processing time (see Figure 5.13, where relative computational cost is presented). Our un-optimized MATLAB implementation of the proposed approach has an average running time of 0.43 fps for color images of 320×240 in a standard laptop (i7-4600U @ 2.1GHz 2.7GHz and 8GB RAM).

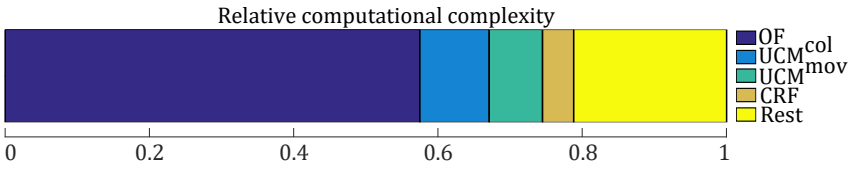


Figure 5.13: Relative computational complexity for the proposed approach. From left to right: optical flow (OF), ultrametric contour maps (UCMs), CRF and the rest of the operations.

5.5 Conclusions

In this chapter, we propose a framework for the improvement of foreground segmentation masks obtained by background subtraction algorithms that is independent of each algorithm characteristics. In particular, we use the foreground masks and the analyzed images to compute a foreground quality that is used to improve results through an optimization process. We obtain such foreground quality in a hierarchical manner by combining the fitness between the foreground mask and image segmentation partitions obtained at different degrees of detail that prevent foreground-background merging due to motion constraints. Experiments using fourteen algorithms and four large background subtraction datasets show that algorithms results can be improved analyzing the quality of their results. Current framework limitations are mainly related to a bad foreground probability estimation either when the original foreground segmentation is too bad for a segmented object or when complete foreground objects are not detected and, therefore, no fitness between foreground and segmented image regions can be estimated. Future work will explore the capabilities of semantic segmentation to improve foreground quality and the effects of temporal information in the energy function for foreground refinement.

Part IV

Conclusions

Chapter 6

Achievements, conclusions and future work

6.1 Summary of achievements and main conclusions

This thesis has addressed the improvement of background subtraction algorithms based on information that is independent of a particular algorithm. The goal was to improve algorithms without modifying inherent properties of them, i.e. the features and models used by background subtraction algorithms. To that end, we have studied two sources of information that are independent of the algorithms details: the input (color images) and output (foreground segmentation masks). The former involves background estimation (Chapter 2) and its application for stationary object detection (Chapter 3), while the latter involves stand-alone quality estimation (Chapter 4) and foreground segmentation improvement (Chapter 5).

Firstly, we have proposed a block-wise background estimation approach (Chapter 2), named RMR, to estimate the background of video sequences from their frames. This approach applies a clustering method without the need of thresholds over motion-filtered and dimension reduced data to determine candidate blocks to be background. Subsequently, a *Rejection based Multipath Reconstruction* based on background smoothness constraints selects the most suitable candidate. This multipath scheme represents the core of the approach and is based on the exploration of different paths or hypotheses to increase the robustness against errors in the background reconstruction process. We evaluated our approach against 13 algorithms in 29 real sequences selected from public datasets and showed that RMR outperforms state-of-the-art results due to robustness against crowds and a better handling of stationary objects. Additionally, experiments over the 7 videos of the SBMI2015 dataset were carried out confirming that RMR outperforms the related work. Moreover, we also participated in the Scene Background Modeling contest[1],

[1] `http://scenebackgroundmodeling.net/`

which proposed the SBMnet2016 dataset containing 79 videos for background estimation divided into 8 categories, obtaining top performance against stationary objects. To operate in this dataset, we had to adapt our algorithm slightly, which substantially degraded the performance in some categories.

Secondly, we have explored the application of the background estimation task to stationary object detection (Chapter 3). Unlike many state-of-the-art approaches that focus on post-processing foreground masks from background subtraction algorithms to determine stationarity, we exploit the spatio-temporal changes in the most stable image representations, i.e. background images computed following a temporal strategy, to detect the stationary objects. In particular, we perform a block-wise analysis that involves an *Online Clustering* stage to update clusters over time using non-moving image representations. This clustering provides robustness against illumination changes by considering pixel ratios at block level which groups blocks even if their illumination has changed. Then, a *Stationary Block Detection* stage computes the stationary objects by exploiting the new spatio-temporal stability changes between background images at different sampling instants. This stage introduces robustness to intermittent object motion related issues (i.e. stationary objects and ghost artifacts), that background subtraction algorithms usually suffer, by keeping a buffer of old stable clusters that allows the association of stability changes due to objects removal to a previously seen background, i.e. an old stable cluster. The fast updating of the proposed approach together with its capability to detect stationary objects makes it suitable for long-term operation as demonstrated in the experimental work. In particular, we evaluated our approach in 6 short sequences and 7 long sequences, thus testing over 364951 frames ($\sim$4.05 h) containing 51 abandoned objects and stopped vehicles as ground-truth. The proposed approach is able to detect all objects without any false positive in the short sequences, thus demonstrating the robustness in crowded situations; while for long-term sequences the proposed approach outperforms the state-of-the-art due to the techniques proposed.

Thirdly, the thesis has addressed the relatively unexploited field of stand-alone evaluation of foreground segmentation masks (Chapter 4), which involves using measures computed over connected components (i.e. blobs) for the estimation of the masks quality without making use of ground-truth data. We discuss 21 available measures in the literature to identify the properties of high-performance foreground segmentation masks. To compare these measures, the results of eight state-of-the-art background subtraction algorithms are analyzed using the CDNET2014 dataset [Wang et al., 2014b]. We first cluster these measures according to their linear and non-linear relations using the Pearson's correlation coefficient [Pearson, 1896] and Self-Organizing Maps [Kohonen, 1982]. Then, we select the most useful measures of each cluster to analyze their correlation with ground-truth based performance measures and their capabilities for discriminating low, medium and high performance (i.e. quality levels). Finally, we explore the application of these measures to rank algorithms as compared to rankings based on

ground-truth performance. To the best of our knowledge, this is the first attempt to provide a comprehensive study of stand-alone performance estimation for connected components in foreground segmentation masks (i.e. blobs) as previous works [Correia and Pereira, 2002][Erdem et al., 2004][SanMiguel and Martinez, 2010] are mainly focused on frame-level measures in simple scenarios. Such frame-level evaluations combine all blob qualities per frame, thus restricting a detailed analysis of relevant blob mask properties. The experiments performed reveal that, among all measures, fitness-to-regions can approximate ground-truth performance.

Finally, the thesis has concentrated in its main objective, the improvement of background subtraction algorithms performance exploiting information that does not depend on a particular algorithm (Chapter 5). We built on quality information through fitness-to-regions as it is independent of a particular algorithm and previous results in Chapter 4 indicated its potential for quality estimation. Therefore, we used quality estimation to improve the quality or performance. To that end, our approach estimates foreground quality maps and transforms them into foreground probabilities to obtain an improved mask through a Conditional Random Field. The novelty of our framework is related with both the use of quality to improve quality and the way in which the foreground quality is extracted. This extraction is based on exploiting the fitness between the foreground mask and a hierarchy of image segmentation partitions that are obtained at different degrees of detail and using motion constraints to prevent foreground-background merging. The intuition behind the potential of fitness to improve quality relies on the high (low) scores obtained for true positives and false negatives (false positives). For the former, when operating over foreground objects partially or entirely detected, a fitness measure reveals a certain score which helps to keep or recover the complete regions as foreground. For the latter, usually false positives belong to larger background areas (e.g. walls, floor, sky, etc), thus small fitness scores help to relabel that pixels as background. The extensive experiments performed demonstrate that algorithms results can be improved from the quality of their results. Furthermore, we demonstrate that foreground quality maps are a valuable mechanism for algorithm combination to further improve background subtraction results.

6.2 Future Work

Based on the results and discussions of this thesis, we propose the following future research lines:

- Background estimation for foreground segmentation improvement. In this thesis, we have worked over the input (image) and the output (foreground mask) of a background subtraction algorithm, but we have only used the latter to improve foreground segmentation performance. Therefore, as background estimation or initialization is an effective technique to capture a background representation and background subtraction algorithms commonly suffer problems when adapting to scene variations, introducing effective background ini-

tialization or re-initialization mechanisms could help to re-capture correct background representations and prevent foreground segmentation errors.

- Extension of the post-processing framework to consider semantic knowledge. The current limitations of our post-processing approach (Chapter 5) are mainly related to the low fitness (i.e. low foreground probabilities) obtained for weakly detected objects and the high fitness obtained for some false positives that do not fulfill the premise about a false positive being contained in a much larger image region. Therefore, current semantic segmentation algorithms [Zhao et al., 2017a] can provide a powerful information to both extend to complete entities and discover the category of the regions. For instance, while the current framework based on image segmentation provides multiple image regions in a waving tree area, semantic segmentation can go further and provide not only a unique tree region, but the valuable information about that regions being a tree (by definition a background region). In fact, a recently published paper [Braham et al., 2017] exploits semantic segmentation to improve background subtraction. However, we would like to explore and end-to-end system rather than a "hand-crafted" use of semantic segmentation labels.

- Background estimation and stationary object detection are closely related as seen in Chapters 2 and 3. Furthermore, foreground quality estimation (Chapter 4) and foreground segmentation improvement (Chapter 5) are close tasks as demonstrated by using a fitness-to-regions (a good quality estimator) for the improvement of background subtraction algorithms. Moreover, foreground segmentation and background initialization and modeling are closely related as background subtraction uses the background model to perform the segmentation. Therefore, it seems natural to overcome all this tasks at once through recent multi-task deep learning frameworks [Kendall et al., 2017][Cheng et al., 2017][Ranjan et al., 2017]. As seen in the literature, this techniques provide a powerful tool to address tasks simultaneously when tasks addressed are highly related. In particular, an architecture based on convolutional recurrent neural networks [Tokmakov et al., 2017] might be able to both understand the spatial continuities inherent to the background and the temporal patterns associated to foreground objects that have to be excluded from the background reconstructions but considered for the stationarity and the foreground segmentation.

- Exploration of deep learning frameworks for background subtraction. As a result of the recent success of deep learning for computer vision and in particular for segmentation related tasks, there are several recent approaches [Chen et al., 2017][Zhao et al., 2017b][Babaee et al., 2018] that tackle background subtraction using convolutional and recurrent neural networks. Despite achieving groundbreaking improvements in performance compared to previous approaches [Wang et al., 2014a][St-Charles et al., 2015], these improvements are

due to the fact that they train and test in the same environments, thus implicitly overfitting to one or a set of particular scenarios. Then, these approaches break the rules as one may expect the development of algorithms that are able to generalize. Conversely, we would like to explore which are the capabilities of convolutional recurrent neural networks to discover the spatio-temporal patterns that characterize foreground objects without overfitting to scene-specific appearances or overfitting in an unsupervised manner.

Part V

Appendixes

Appendix A

Publications

The following publications have been produced in association with this thesis (listed by chapters):

- Background estimation (Chapter 2)
  - D. Ortego and J.C. SanMiguel and J.M. Martínez, "Rejection based multipath reconstruction for background estimation in video sequences with stationary objects", *Computer Vision and Image Understanding*, vol. 147, pp. 23-37, 2016 (`https://doi.org/10.1016/j.cviu.2016.03.012`).
  - D. Ortego and J. C. SanMiguel and J. M. Martínez, "Rejection based multipath reconstruction for background estimation in SBMnet 2016 dataset", *in Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 114-119, 2016 (`https://doi.org/10.1109/ICPR.2016.7899618`).
- Stationary object detection (Chapter 3)
  - D. Ortego, J. C. SanMiguel and J. M. Martínez, "Long-Term Stationary Object Detection Based on Spatio-Temporal Change Detection" *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2368-2372, 2015. (`https://doi.org/10.1109/LSP.2015.2482598`).
- Stand-alone evaluation of background subtraction algorithms (Chapter 4)
  - D. Ortego, J. C. SanMiguel, J. M. Martínez, "Stand-alone quality estimation of background subtraction algorithms", *Computer Vision and Image Understanding*, vol. 162, pp. 87-102, 2017 (`https://doi.org/10.1016/j.cviu.2017.08.005`).
- Foreground segmentation improvement of background subtraction algorithms (Chapter 5)

– D. Ortego, J. C. SanMiguel, J. M. Martínez, "Hierarchical improvement of foreground segmentation masks in background subtraction", submitted to *IEEE Transactions on Circuits and Systems for Video Technology*, 2018 (in minor revision).

Electronic versions are available at the following URL:
`http://www-vpu.ii.uam.es/webvpu/gti/user/99/`

Appendix B

Logros, conclusiones y trabajo futuro

B.1 Resumen de logros y principales conclusiones

Esta tesis ha estudiado la mejora de algoritmos de sustracción de fondo utilizando información que es independiente de las particularidades de cada algoritmo. El objetivo era mejorar los algoritmos sin modificar propiedades particulares de los mismos, es decir, sus características y modelos utilizados. Para lograrlo, hemos estudiado dos fuentes de información que son independientes de los algoritmos, su entrada (imagen de color) y su salida (máscara de segmentación de objetos de frente). La primera parte incluye un algoritmo de estimación de fondo de escena (Capítulo 2) y la aplicación de dicha tarea a la detección de objetos estáticos (Capítulo 3), mientras que la segunda se centra en la estimación de calidad sin utilizar datos anotados (Capítulo 4) y en la mejora de los resultados de segmentación de objetos de frente (Capítulo 5).

En primer lugar, se ha propuesto un algoritmo de estimación de fondo a nivel de bloque (Capítulo 2), llamado RMR, para estimar una imagen de fondo dados los *frames* de un vídeo. Esta técnica emplea un método de agrupamiento libre de umbrales que opera sobre bloques sin movimiento y de dimensionalidad reducida para determinar un conjunto de bloques candidatos a fondo. A continuación, se seleccionada el mejor candidato siguiendo una reconstrucción del fondo multi-camino basada en rechazo. Este esquema multi-camino es el núcleo del algoritmo y se basa en la exploración de varios caminos o hipótesis para incrementar la robustez frente a errores en el proceso de reconstrucción. Se ha evaluado nuestra propuesta frente a 13 algoritmos en 29 secuencias reales seleccionadas de conjuntos de datos públicos, mostrando que RMR mejora dichos algoritmos y que tiene robustez frente a multitudes y objetos estáticos. Además, se han realizado experimentos en el conjunto de datos SBMI2015 que confirman la mejora de RMR frente al estado del arte. Adicionalmente, se participó en el *Scene Background Modeling contest*[1], que propuso el conjunto de datos SBMnet2016 con 79 vídeos para la estimación de

[1] `http://scenebackgroundmodeling.net/`

fondo divididos en 8 categorías, obteniendo muy buenos resultados frente a objetos estáticos. No obstante, para garantizar la operatividad en SBMnet2016 hubo que adaptar ligeramente el algoritmo, lo que redujo considerablemente su rendimiento en algunas categorías.

En segundo lugar, se ha explorado la aplicación de la estimación de fondo a la detección de objetos estáticos (Capítulo 3). A diferencia de la mayoría de aproximaciones del estado del arte que se centran en pos-procesar las máscaras de objetos de frente generadas por algoritmos de sustracción de fondo, se han utilizado las variaciones espacio-temporales en las representaciones estables de la imagen (es decir, el fondo estimado mediante una técnica temporal), para detectar los objetos estáticos. En concreto, se ha desarrollado un algoritmo a nivel de bloque que comienza con una etapa de agrupamiento temporal que actualiza las representaciones de la escena empleando bloques sin movimiento. Este agrupamiento proporciona robustez frente a cambios de iluminación empleando el ratio entre píxeles para poder agrupar bloques que son similares pero con una iluminación diferente. A continuación, una etapa de detección de bloques estáticos determina los objetos estáticos basándose en las nuevas variaciones espacio-temporales que aparecen en distintos instantes temporales. Esta etapa introduce robustez frente a objetos con movimiento intermitente (es decir, objetos estáticos y detecciones fantasma), que suelen generar numerosos problemas en los algoritmos de sustracción de fondo, mediante la utilización de un *buffer* de antiguas representaciones estables de la escena que permite asociar la nueva estabilidad generada en la escena al quitar un objeto a una imagen de fondo previa, es decir, a una representación estable previa. La rápida actualización a los cambios espacio-temporales junto con la capacidad de detectar los objetos estáticos, convierten al algoritmo propuesto en una opción viable para operar a largo plazo tal y como se demuestra en la evaluación realizada. En particular, se ha evaluado nuestra propuesta en 6 secuencias cortas y 7 largas, que suponen 364951 *frames* ($\sim$4.05 h) con 51 objetos abandonados y vehículos estáticos en ellos. El algoritmo propuesto es capaz de detectar todos los objetos sin falsos positivos en las secuencias cortas, demostrando su robustez en multitudes; mientras que para las secuencias a largo plazo el algoritmo desarrollado mejora el estado del arte gracias a las técnicas propuestas.

En tercer lugar, la tesis se ha adentrado en la relativamente poco investigada temática de la evaluación sin datos anotados para la tarea de segmentación de objetos de frente (Capítulo 4), que se ha centrado en analizar medidas obtenidas sobre componentes conexas de la máscara de frente. En este sentido, se han analizado 21 medidas de la literatura relacionada para identificar propiedades de las máscaras que revelan su calidad o rendimiento. Para comparar estas medidas, se han utilizado los resultados de 8 algoritmos de sustracción de fondo en el conjunto de datos CDNET2014 [Wang et al., 2014b]. En primer lugar, se han agrupado las medidas siguiendo sus relaciones lineales y no lineales mediante la utilización del coeficiente de correlación de Pearson [Pearson, 1896] y *Self-Organizing Maps* [Kohonen, 1982]. A partir de dichas agrupaciones, se han seleccionado las medidas más útiles de cada grupo para analizar sus capacidades para aproximar

la calidad que reportan las medidas que hacen uso de datos anotados y para discriminar entre calidades baja, media y alta (niveles de calidad). Finalmente, se ha explorado la utilidad de dichas medidas para replicar la ordenación de algoritmos basada en calidad que obtienen las medidas basadas en datos anotados. Que nosotros sepamos, este es el primer estudio de medidas de evaluación sin datos anotados para evaluar máscaras de segmentación de objetos de frente, ya que otros trabajos previos [Correia and Pereira, 2002][Erdem et al., 2004][SanMiguel and Martinez, 2010] se centran en medidas a nivel de *frame* en escenarios sencillos. Los experimentos realizados revelan que, de entre todas las medidas, el ajuste a regiones puede aproximar el rendimiento medido mediante datos anotados.

Finalmente, la tesis se ha centrado en su objetivo principal, mejorar los algoritmos de sustracción de fondo sacando partido a información independiente de cada algoritmo (Capítulo 5). Por tanto, este capítulo se ha basado en la información de calidad que el ajuste a regiones es capaz de obtener, pues esta información es independiente de las particularidades de los algoritmos. Es decir, se ha utilizado una estimación de calidad para mejorar la calidad o rendimiento. Con este fin, se han calculado unos mapas de calidad de la máscara de frente que son transformados en probabilidades para poder calcular una máscara mejorada mediante la utilización de *Conditional Random Fields*. La novedad de la propuesta reside tanto en la utilización de la calidad para mejorar la calidad como en la manera en la que dicha calidad es extraída. Este proceso de extracción de calidad se basa en calcular el ajuste entre la máscara de objetos de frente y particiones con distinto nivel de detalle que son calculadas teniendo en cuenta información de movimiento para prevenir que las regiones de los objetos de frente y las zonas de fondo se fusionen en una misma región. La idea detrás del potencial del ajuste a regiones reside en los altos (bajos) valores obtenidos para los verdaderos positivos y los falsos negativos (falsos positivos). Respecto a los primeros, cuando hay un objeto de frente total o parcialmente detectado, una medida de ajuste obtiene un cierto valor que permite mantener o expandir el objeto. Mientras que para el segundo caso, los falsos positivos generalmente pertenecen a zonas extensas de fondo (p.ej. muros, suelo, cielo, etc), lo que permite obtener valores de ajuste pequeños que ayudan a re-detectar dichos píxeles como fondo. Los numerosos experimentos realizados sobre 4 conjuntos de datos y 15 algoritmos demuestran que los resultados de los algoritmos pueden mejorarse empleando la calidad de dichos resultados. Además, se ha demostrado que los mapas de calidad de segmentación de objetos de frente son un mecanismo efectivo de combinación de algoritmos que permite mejorar aún más el resultado.

B.2 Trabajo futuro

En base a los resultados y discusiones de esta tesis, proponemos las siguientes líneas de trabajo futuro:

- La estimación de fondo como mecanismo de mejora de los algoritmos de sustracción de fondo. En esta tesis se ha trabajado sobre las imágenes de entrada y sobre la máscara de frente de los algoritmos de sustracción de fondo, pero solo se ha explorado la mejora del rendimiento de los algoritmos mediante esta última. Por un lado, la estimación o inicialización de fondo es un técnica útil para capturar el fondo de escena. Por otro lado, los algoritmos de sustracción de fondo tienen problemas para adaptarse a las variaciones de la escena, lo cual afecta directamente al rendimiento de los algoritmos debido a que la representación del fondo con la que trabajan no está actualizada. Por lo tanto, la utilización de técnicas de inicialización y re-inicialización de fondo podría ayudar a re-capturar una buena representación del mismo y prevenir así los errores en la segmentación.

- La extensión del esquema de pos-procesado propuesto para considerar información semántica. Las limitaciones actuales del esquema propuesto (Capítulo 5) están relacionadas con el bajo ajuste (es decir, la baja probabilidad de frente) obtenido para objetos muy mal detectados y con el alto ajuste obtenido para algunos falsos positivos que no cumplen la premisa de que un falso positivo pertenece a una región de fondo mucho más grande. Por tanto, los algoritmos actuales de segmentación semántica [Zhao et al., 2017a] pueden aportar una información muy útil para, por un lado, extender las detecciones de objetos de manera completa y, por otro lado, descubrir la naturaleza de cada región de la imagen. Por ejemplo, mientras el esquema actual basado en segmentación de la imagen en regiones genera una partición en la que un árbol puede pertenecer a múltiples regiones, la segmentación semántica puede ir un paso más allá y no solo generar una única región, sino proporcionar la información de que esa región se corresponde con un árbol (por definición parte del fondo). De hecho, un trabajo publicado recientemente [Braham et al., 2017] utiliza segmentación semántica para mejorar los algoritmos de sustracción de fondo. No obstante, querríamos explorar una solución *end-to-end* en lugar de una aproximación basada en reglas empíricas.

- La estimación de fondo y la detección de objetos estáticos son tareas estrechamente relacionadas tal y como se ha visto en los Capítulos 2 y 3. Además, la estimación de calidad de máscaras de segmentación de frente (Capítulo 4) y la mejora de dichas máscaras (Capítulo 5) son también tareas cercanas, tal y como se ha demostrado mediante la utilización de la propiedad de ajuste a regiones (un buen estimador de calidad) para la mejora de las máscaras de frente. Por otro lado, la segmentación frente-fondo y la inicialización de fondo son tareas estrechamente relacionadas pues los algoritmos de sustracción de fondo se basan en un modelo de fondo para segmentar los objetos de frente. Por lo tanto, resulta natural pensar en abordar todas estas tareas conjuntamente mediante esquemas recientes de aprendizaje profundo multi-tarea [Kendall et al., 2017][Cheng et al., 2017][Ranjan et al.,

2017]. En concreto, una arquitectura basada en redes neuronales recurrentes convolucionales [Tokmakov et al., 2017] podría ser capaz de entender las continuidades espaciales inherentes al fondo y los patrones temporales asociados a los objetos de frente que tienen que ser excluidos en la estimación de fondo pero considerados en la detección de objetos estáticos.

- La exploración de esquemas basados en aprendizaje profundo para sustracción de fondo. Como resultado del éxito de las técnicas de aprendizaje profundo para visión por computador y en particular para las tareas de segmentación, han surgido recientemente varios algoritmos [Chen et al., 2017][Zhao et al., 2017b][Babaee et al., 2018] que abordan la sustracción de fondo usando redes neuronales convolucionales y recurrentes. A pesar de las sustanciales mejoras de rendimiento en comparación con los algoritmos tradicionales [Wang et al., 2014a][St-Charles et al., 2015], estos avances se deben a que los algoritmos entrenar y testar con datos de los mismos vídeos, hecho que lleva a un sobreajuste a los escenarios particulares de entrenamiento. Por lo tanto, estas técnicas rompen las reglas comparándose con los algoritmos tradicionales, pues no demuestran que sus algoritmos sean capaces de generalizar. Por el contrario, nos gustaría estudiar las capacidades de las redes neuronales recurrentes convolucionales para descubrir patrones espacio-temporales característicos de los objetos de frente sin sobreajustarnos a escenarios particulares.

Glossary

AHC *Aglomerative Hierarchical Clustering*

BE *Background Estimation*

BS *Background Subtraction*

CRF *Conditional Random Field*

FPB *False Positive Blob*

GT *Ground-Truth*

MRF *Markov Random Field*

PCA *Principal Component Analysis*

RGB *Red Green Blue color model*

RMR *Rejection based Multipath Reconstruction*

SMR *Sequential Multipath Reconstruction*

SOD *Stationary Object Detection*

SOM *Self-Organizing Map*

SSI *Sub-intervals of Stable Intensity*

SVM *Support Vector Machine*

TPB *True Positive Blob*

UCM *Ultrametric Contour Map*

Bibliography

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (11):2274–2282, 2012. [Cited on pages 3 and 98.]

N. Ahmed, T. Natarajan, and K. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. [Cited on page 30.]

N. Al-Najdawi, H. Bez, J. Singhai, and E. Edirisinghe. A survey of cast shadow detection algorithms. *Pattern Recognition Letters*, 33(6):752–764, 2012. [Cited on pages 71 and 95.]

A. Albiol, L. Sanchis, A. Albiol, and J. Mossi. Detection of parked vehicles using spatiotemporal maps. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1277–1291, 2011. [Cited on pages 55 and 56.]

B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012. [Cited on pages 4 and 78.]

L. Ang Lim and H. Yalim Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *CoRR*, abs/1801.02225, 2018. [Cited on pages 94 and 109.]

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2294–2301, 2009. [Cited on pages 77 and 98.]

P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 328–335, 2014. [Cited on pages 96 and 97.]

M. Babaee, D. T. Dinh, and G. Rigoll. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, 76:635–649, 2018. [Cited on pages 4, 124, and 135.]

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. [Cited on page 116.]

M. Balcilar and A. Sonmez. Background estimation method with incremental iterative re-weighted least squares. *Signal, Image and Video Processing*, pages 1–8, 2015. [Cited on page 16.]

D. Baltieri, R. Vezzani, and R. Cucchiara. Fast background initialization with recursive hadamard transform. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 165–171, 2010. [Cited on pages 16, 17, 18, 24, and 36.]

A. Bayona, J. SanMiguel, and J. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 25–30, 2009. [Cited on page 55.]

A. Bayona, J. SanMiguel, and J. Martinez. Stationary foreground detection using background subtraction and temporal difference in video surveillance. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 4657–4660, 2010. [Cited on pages 55, 62, and 63.]

M. Benalia and S. Ait-Aoudia. An improved basic sequential clustering algorithm for background construction and motion detection. In *Proceedings of International Conference on Image Analysis and Recognition (ICIAR)*, volume 7324, pages 216–223. 2012. [Cited on page 17.]

S. Bianco, G. Ciocca, and R. Schettini. Combination of video change detection algorithms by genetic programming. *IEEE Transactions on Evolutionary Computation*, 21(6):914–928, 2017. [Cited on page 115.]

D. Bloisi, A. Pennisi, and L. Iocchi. Background modeling in the maritime domain. *Machine Vision and Applications*, 25(5):1257–1269, 2014. [Cited on pages 39 and 40.]

A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015. [Cited on pages 4, 67, 69, 70, and 93.]

T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31–66, 2014. [Cited on pages 3, 4, 5, 7, 15, 55, 67, and 93.]

T. Bouwmans and E. H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014. [Cited on pages 17 and 45.]

T. Bouwmans, C. Silva, C. Marghes, M. S. Zitouni, H. Bhaskar, and C. Frélicot. On the role and the importance of features for background modeling and foreground detection. *CoRR*, abs/1611.09099, 2016. [Cited on pages 7 and 94.]

T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E.-H. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23:1–71, 2017. [Cited on page 94.]

M. Braham and M. V. Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In *Proceedings of International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, 2016. [Cited on pages 7 and 94.]

M. Braham, S. Pierard, and M. Van Droogenbroeck. Semantic background subtraction. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 4552–4556, 2017. [Cited on pages 7, 124, and 134.]

W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 833–840, 2009. [Cited on page 3.]

T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, pages 25–36. 2004. [Cited on page 99.]

S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1937–1944, 2011. [Cited on pages 8, 68, 95, and 107.]

S. Calarasanu, J. Fabrizio, and S. Dubuisson. What is a good evaluation protocol for text localization systems? concerns, arguments, comparisons and solutions. *Image and Vision Computing*, 46:1–17, 2016. [Cited on page 70.]

L. C. Campos, J. SanMiguel, and J. Martínez. Discrimination of abandoned and stolen object based on active contours. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 101–106, 2011. [Cited on page 95.]

L. Cehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *IEEE Transactions on Image Processing*, 25(3):1261–1274, 2016. [Cited on pages 67 and 83.]

M. Chacon-Murguia, J. Ramirez-Quintana, and D. Urias-Zavala. Segmentation of video background regions based on a dtcnn-clustering approach. *Signal, Image and Video Processing*, pages 1–10, 2014. [Cited on page 18.]

X. Chen, Y. Shen, and Y. Yang. Background estimation using graph cuts and inpainting. In *Proceedings of Graphics Interface (GI)*, pages 97–103, 2010. [Cited on pages 15 and 18.]

Y. Chen, J. Wang, and H. Lu. Learning sharable models for robust background subtraction. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2015. [Cited on page 108.]

Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu. Pixel-wise deep sequence learning for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. [Cited on pages 124 and 135.]

Z. Chen and T. Ellis. A self-adaptive gaussian mixture model. *Computer Vision and Image Understanding*, 122:35–46, 2014. [Cited on pages 8, 17, 63, 71, 94, and 95.]

F.-C. Cheng, S.-C. Huang, and S.-J. Ruan. Illumination-sensitive background modeling approach for accurate moving object detection. *IEEE Transactions on Broadcasting*, 57(4):794–801, 2011. [Cited on page 71.]

F.-C. Cheng, B.-H. Chen, and S.-C. Huang. A background model re-initialization method based on sudden luminance change detection. *Engineering Applications of Artificial Intelligence*, 38:138–146, 2015. [Cited on page 71.]

J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. [Cited on pages 124 and 134.]

C. Chia-Chih and J. Aggarwal. An adaptive background model initialization algorithm with objects moving at different depths. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2664–2667, 2008. [Cited on pages 17, 18, 20, and 22.]

A. Colombari and A. Fusiello. Patch-based background initialization in heavily cluttered video. *IEEE Transactions on Image Processing*, 19(4):926–933, 2010. [Cited on pages 16, 17, 18, 34, and 45.]

R. Colque and G. Camara-Chavez. Progressive background image generation of surveillance traffic videos based on a temporal histogram ruled by a reward/penalty function. In *Proceedings of Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 297–304, 2011. [Cited on page 16.]

C. Conaire, E. Cooke, and A. Smeaton. Detection thresholding using mutual information. In *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP)*, 2006. [Cited on page 71.]

C. O. Conaire, N. E. O'Connor, and A. F. Smeaton. Detector adaptation by maximising agreement between independent data sources. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007. [Cited on page 71.]

P. Correia and F. Pereira. Stand-alone objective segmentation quality evaluation. *EURASIP Journal on Advances in Signal Processing*, (4):1–12, 2002. [Cited on pages 8, 68, 72, 73, 75, 76, 123, and 133.]

T. Crivelli, P. Bouthemy, B. Cernuschi-Frías, and J.-f. Yao. Simultaneous motion detection and background reconstruction with a conditional mixed-state markov random field. *International Journal of Computer Vision*, 94(3):295–316, 2011. [Cited on pages 16 and 18.]

C. Cuevas, E. Yáñez, and N. García. Tool for semiautomatic labeling of moving objects in video sequences: Tslab. *Sensors*, 15(7):15159–15178, 2015. [Cited on pages 8 and 68.]

C. Cuevas, E. Yáñez, and N. García. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. *Computer Vision and Image Understanding*, 152:103–117, 2016. [Cited on pages 68, 95, and 107.]

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 233–240, 2006. [Cited on page 70.]

B. Dey and M. K. Kundu. Enhanced macroblock features for dynamic background modeling in h.264/avc video encoded at low-bitrate. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016. [Cited on pages 7 and 94.]

P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1841–1848, 2013. [Cited on pages 98 and 100.]

R. Dony and S. Wesolkowski. Edge detection on color images using rgb vector angles. In *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, volume 2, pages 687–692, 1999. [Cited on page 30.]

E. Dougherty. *An introduction to morphological image processing.* 1992. [Cited on pages 7, 94, and 96.]

F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. In *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1729–1736, 2008. [Cited on page 39.]

D. Elgammal, A.and Harwood and L. Davis. Non-parametric model for background subtraction. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 751–767, 2000. [Cited on pages 80 and 108.]

A. Ellis, A. Shahrokni, and J. Ferryman. PETS2009 and winter-PETS 2009 results: A combined evaluation. In *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 1–8, 2009. [Cited on page 34.]

H. Eng, K.-A. Toh, A. Kam, J. Wang, and W.-Y. Yau. An automatic drowning detection surveillance system for challenging outdoor pool environments. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 532–539, 2003. [Cited on page 17.]

C. Erdem, B. Sankur, and A. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Transactions on Image Processing*, 13(7):937–951, 2004. [Cited on pages 8, 67, 68, 71, 72, 73, 74, 123, and 133.]

S. Erfanian Ebadi and E. Izquierdo. Foreground segmentation via dynamic tree-structured sparse rpca. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 314–329, 2016. [Cited on page 94.]

R. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 71–76, 2011. [Cited on page 55.]

R. H. Evangelio, M. Patzold, I. Keller, and T. Sikora. Adaptively splitted gmm with feedback improvement for the task of background subtraction. *IEEE Transactions on Information Forensics and Security*, 9 (5):863–874, 2014. [Cited on page 40.]

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. [Cited on page 70.]

A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proceedings of British Machine Vision Conference (BMVC)*, 2014. [Cited on pages 5 and 67.]

Q. Fan, P. Gabbur, and S. Pankanti. Relative attributes for large-scale abandoned object detection. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2736–2743, 2013. [Cited on pages 3, 55, and 56.]

Q. Fan, S. Pankanti, and L. Brown. Long-term object tracking for parked vehicle detection. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 223–229, 2014. [Cited on page 56.]

P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. [Cited on pages 3, 77, and 85.]

K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4083–4090, 2015. [Cited on page 100.]

B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814): 972–976, 2007. [Cited on page 83.]

F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 760–774, 2012. [Cited on page 3.]

X. Gao, T. Boult, F. Coetzee, and V. Ramesh. Error analysis of background adaption. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 503–510, 2000. [Cited on page 69.]

A. García and J. Bescós. Video object segmentation based on feedback schemes guided by a low-level scene ontology. In *Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 322–333, 2008. [Cited on page 71.]

E. Gelasca and T. Ebrahimi. On evaluating video object segmentation quality: A perceptually driven objective metric. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):319–335, 2009. [Cited on page 69.]

A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Coupling video segmentation and action recognition. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 618–625, 2014. [Cited on page 3.]

D. Giordano, I. Kavasidis, S. Palazzo, and C. Spampinato. Rejecting false positives in video object segmentation. In *Proceedings of International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 100–112, 2015. [Cited on pages 7, 8, 72, 73, 74, 75, 76, 77, 78, 94, 95, 96, and 115.]

N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. A novel video dataset for change detection benchmarking. *IEEE Transactions on Image Processing*, 23(11):4663–4679, 2014. [Cited on pages 67 and 70.]

M. Granados, H. Seidel, and H. Lensch. Background estimation from non-time sequence images. In *Proceedings of Graphics Interface (GI)*, pages 33–40, 2008. [Cited on page 15.]

M. D. Gregorio and M. Giordano. Change detection with weightless neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 409–413, 2014. [Cited on pages 80, 94, and 108.]

S. Guler, J. Silverstein, and I. Pushee. Stationary objects in multiple object tracking. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 248–253, 2007. [Cited on pages 55, 62, and 63.]

C. Guo, S. Gao, and D. Zhang. Belief propagation algorithm for background estimation based on local maximum weight matching. In *Proceedings of International Congress on Image and Signal Processing (CISP)*, pages 82–85, 2012. [Cited on page 18.]

R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967, 2013. [Cited on page 71.]

D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. K. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 733–740, 2001. [Cited on pages 17, 18, and 22.]

J. Hartigan. *Clustering Algorithms*. John Wiley & Sons Inc., 1975. [Cited on page 23.]

M. Hassan, A. Malik, W. Nicolas, and I. Faye. Adaptive foreground extraction for crowd analytics surveillance on unconstrained environments. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 390–400, 2015. [Cited on page 96.]

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. [Cited on page 3.]

H. Hsiao and J. Leou. Background initialization and foreground segmentation for bootstrapping video sequences. *EURASIP Journal on Image and Video Processing*, 12:1–19, 2013. [Cited on pages 16, 17, and 36.]

Y.-T. Hu, J.-B. Huang, and A. Schwing. Maskrnn: Instance level video object segmentation. In *Proceedings of International Conference on Neural Information Processing Systems (NIPS)*, pages 324–333, 2017. [Cited on pages 3 and 4.]

Z. Hu, G. Ye, G. Jia, X. Chen, Q. Hu, K. Jiang, Y. Wang, L. Qing, Y. Tian, X. Wu, and W. Gao. Pku@ trecvid2009: Single-actor and pair-activity event detection in surveillance video. In *Proceedings of TRECVID Workshop*, 2009. [Cited on page 17.]

I. Huerta, M. Holte, T. Moeslund, and J. Gonzalez. Chromatic shadow detection and tracking for moving foreground segmentation. *Image and Vision Computing*, 41:42–53, 2015. [Cited on pages 8, 71, 94, and 95.]

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999. [Cited on page 20.]

S. Jain, B. Xiong, and K. Grauman. Pixel objectness. *CoRR*, abs/1701.05349, 2017. [Cited on page 4.]

S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 656–671, 2014. [Cited on page 5.]

S. Jiang and X. Lu. Wesambe: A weight-sample-based method for background subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. [Cited on page 108.]

I. Jolliffe. *Principal Component Analysis*. John Wiley & Sons, Ltd, 2005. [Cited on page 20.]

J. Kapur, P. Sahoo, and A. Wong. A new method for graylevel picture thresholding using the entropy of the histogram. *Computer Graph and Image Process*, 29(3):273–285, 1985. [Cited on pages 20 and 24.]

A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115, 2017. [Cited on pages 124 and 134.]

S. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Automatic shadow detection and removal from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):431–446, 2016. [Cited on page 71.]

J. Kim and D. Kim. Accurate static region classification using multiple cues for ARO detection. *IEEE Signal Processing Letters*, 21(8):937–941, 2014. [Cited on pages 55, 61, and 62.]

J. Kim, A. R. Rivera, B. Ryu, K. Ahn, and O. Chae. Unattended object detection based on edge-segment distributions. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 283–288, 2014. [Cited on page 95.]

A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018. [Cited on page 4.]

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 (1):59–69, 1982. [Cited on pages 68, 83, 122, and 132.]

P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proceedings of International Conference on Neural Information Processing Systems (NIPS)*, pages 109–117, 2011. [Cited on page 105.]

M. Kristan, A. Leonardis, J. Matas, and M. e. a. Felsberg. The visual object tracking vot2016 challenge results. In *Proceedings of European Conference on Computer Vision Workshops (ECCVW)*, pages 777–823, 2016. [Cited on page 93.]

C. Lallier, E. Reynaud, L. Robinault, and L. Tougne. A testing framework for background subtraction algorithms comparison in intrusion detection context. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 314–319, 2011. [Cited on page 70.]

N. Lazarevic-McManus, J. Renno, D. Makris, and G. Jones. An object-based comparative methodology for motion detection based on the f-measure. *Computer Vision and Image Understanding*, 111(1): 74–85, 2008. [Cited on page 70.]

Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1995–2002, 2011. [Cited on page 5.]

H. Li, F. Meng, B. Luo, and S. Zhu. Repairing bad co-segmentation using its quality evaluation and segment propagation. *IEEE Transactions on Image Processing*, 23(8):3545–3559, 2014. [Cited on pages 71, 72, 73, 74, 76, 77, and 78.]

H. H. Lin, T. L. Liu, and J. H. Chuang. Learning a scene background model via classification. *IEEE Transactions on Signal Processing*, 57(5):1641–1654, 2009. [Cited on pages 17 and 93.]

K. Lin, S. C. Chen, C. S. Chen, D. T. Lin, and Y. P. Hung. Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance. *IEEE Transactions on Information Forensics and Security*, 10(7):1359–1370, 2015. [Cited on page 55.]

Q. Ling, J. Yan, F. Li, and Y. Zhang. A background modeling and foreground segmentation approach based on the feedback of moving objects in traffic surveillance systems. *Neurocomputing*, 133:32–45, 2014. [Cited on page 71.]

L. Liu and N. Sang. Metrics for objective evaluation of background subtraction algorithms. In *Proceedings of International Conference on Image and Graphics (ICIG)*, pages 562–565, 2011. [Cited on page 70.]

Z. Liu, W. Zou, and O. L. Meur. Saliency tree: A novel saliency detection framework. *IEEE Transaction on Image Processing*, 23(5):1937–1952, 2014. [Cited on page 110.]

C. Lopez-Molina, H. Bustince, and B. De Baets. Separability criteria for the evaluation of boundary detection benchmarks. *IEEE Transactions on Image Processing*, 25(3):1047–1055, 2016. [Cited on page 67.]

F. López-Rubio and E. López-Rubio. Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding*, 133:30–50, 2015a. [Cited on pages 7 and 94.]

F. López-Rubio and E. López-Rubio. Local color transformation analysis for sudden illumination change detection. *Image and Vision Computing*, 37:31–47, 2015b. [Cited on pages 8, 71, 94, and 95.]

K. Ma, T. Zhao, K. Zeng, and Z. Wang. Objective quality assessment for color-to-gray image conversion. *IEEE Transactions on Image Processing*, 24(12):4673–4685, 2015. [Cited on page 67.]

L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008. [Cited on page 114.]

L. Maddalena and A. Petrosino. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Computing and Applications*, 19(2):179–186, 2010. [Cited on pages 108 and 111.]

L. Maddalena and A. Petrosino. The SOBS algorithm: What are the limits? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 21–26, 2012. [Cited on pages 16, 17, 39, 80, and 108.]

L. Maddalena and A. Petrosino. The 3dsobs+ algorithm for moving object detection. *Computer Vision and Image Understanding*, 122:65–73, 2014a. [Cited on pages 17, 39, 40, and 93.]

L. Maddalena and A. Petrosino. Background model initialization for static cameras. In *Background Modeling and Foreground Detection for Video Surveillance (Eds. T. Bouwmans, F. Porikli, B. Höferlin and A. Vacavant)*, chapter 3, pages 1–16. Chapman and Hall/CRC 2014, 2014b. [Cited on pages 16 and 17.]

L. Maddalena and A. Petrosino. Towards benchmarking scene background initialization. In *Proceedings of International Conference on Image Analysis and Processing (ICIAP)*, volume 9281, pages 469–476. 2015. [Cited on pages 35 and 36.]

L. Mai and F. Liu. Comparing salient object detection results without ground truth. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 76–91, 2014. [Cited on page 67.]

K. Maninis, S. Caelles, J. Pont-Tuset, and L. V. Gool. Deep extreme cut: From extreme points to object segmentation. *CoRR*, abs/1711.09081, 2017. [Cited on page 5.]

R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2014. [Cited on page 70.]

M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. [Cited on page 93.]

C. Min, J. Zhang, B. Chang, B. Sun, and Y. Li. Spatio-temporal segmentation of moving objects using edge features in infrared videos. *Optik - International Journal for Light and Electron Optics*, 125(7): 1809–1815, 2014. [Cited on pages 67, 72, 73, 74, 77, and 78.]

S. Minaee and Y. Wang. Masked signal decomposition using subspace representation and its applications. *CoRR*, abs/1704.07711, 2017. [Cited on pages 4 and 93.]

Y. Nakashima, N. Babaguchi, and F. Jianping. Automatic generation of privacy-protected videos using background estimation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2011. [Cited on page 15.]

J. C. Nascimento and J. S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, 2006. [Cited on pages 8 and 68.]

N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000. [Cited on page 17.]

D. Ortego and J. SanMiguel. Multi-feature stationary foreground detection for crowded video-surveillance. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2403–2407, 2014. [Cited on pages 55, 62, and 63.]

D. Ortego, J. SanMiguel, and J. Martínez. Long-term stationary object detection based on spatio-temporal change detection. *IEEE Signal Processing Letters*, 22(12):2368–2372, 2015. [Cited on pages 10 and 55.]

D. Ortego, J. SanMiguel, and J. Martínez. Rejection based multipath reconstruction for background estimation in video sequences with stationary objects. *Computer Vision and Image Understanding*, 147:23–37, 2016a. [Cited on pages 10, 15, and 48.]

D. Ortego, J. C. SanMiguel, and J. M. Martínez. Rejection based multipath reconstruction for background estimation in sbmnet 2016 dataset. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 114–119, 2016b. [Cited on pages 10 and 15.]

D. Ortego, J. C. SanMiguel, and J. M. Martínez. Stand-alone quality estimation of background subtraction algorithms. *Computer Vision and Image Understanding*, 162:87–102, 2017. [Cited on pages 10, 67, 94, 96, and 101.]

J. Pan, Q. Fan, and S. Pankanti. Robust abandoned object detection using region-level analysis. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 3597–3600, 2011. [Cited on pages 55, 61, and 62.]

A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1777–1784, 2013. [Cited on page 5.]

D. Park and H. Byun. A unified approach to background adaptation and initialization in public scenes. *Pattern Recognition*, 46(7):1985–1997, 2013. [Cited on pages 15, 16, 17, and 18.]

J. Park, K. Seshadrinathan, S. Lee, and A. Bovik. Video quality pooling adaptive to perceptual distortion severity. *IEEE Transactions on Image Processing*, 22(2):610–620, 2013. [Cited on page 67.]

D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 192–199, 2008. [Cited on pages 7, 94, and 96.]

M. Paul. Efficient video coding using optimal compression plane and background modelling. *IET Image Processing*, 6(9):1311–1318, 2012. [Cited on page 15.]

K. Pearson. On a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of Royal Society of London*, 60:489–498, 1896. [Cited on pages 68, 82, 83, 122, and 132.]

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [Cited on pages 4, 67, 69, 70, 93, and 108.]

D.-S. Pham, O. Arandjelovic, and S. Venkatesh. Detection of dynamic background due to swaying movements from motion features. *IEEE Transactions on Image Processing*, 24(1):332–344, 2015. [Cited on pages 8, 71, 94, and 95.]

J. Pilet, C. Strecha, and P. Fua. Making background subtraction robust to sudden illumination changes. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume 5305, pages 567–580, 2008. [Cited on pages 58 and 62.]

J. Pont-Tuset and F. Marques. Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1465–1478, 2015. [Cited on page 67.]

F. Porikli, Y. Ivanov, and T. Haga. Robust abandoned object detection using dual foregrounds. *EURASIP Journal on Advances in Signal Processing*, (1):1–11, 2008. [Cited on pages 55, 61, 62, and 63.]

A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3282–3289, 2012. [Cited on page 108.]

Z. Qiu, T. Yao, and T. Mei. Learning deep spatio-temporal dependency for semantic video segmentation. *IEEE Transactions on Multimedia*, 2017. [Cited on page 4.]

S. Ramadan. Using time series analysis to visualize and evaluate background subtraction results for computer vision applications. Master's thesis, 2006. [Cited on pages 8 and 68.]

R. Raman, S. Choudhury, and S. Bakshi. Spatiotemporal optical blob reconstruction for object detection in grayscale videos. *Multimedia Tools and Applications*, 77(1):741–762, 2017. [Cited on pages 96 and 115.]

J. Ramirez-Quintana and M. Chacon-Murguia. Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios. *Pattern Recognition*, 48(4):1137–1149, 2015. [Cited on pages 71 and 95.]

R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. [Cited on pages 124 and 134.]

V. Reddy, C. Sanderson, and B. Lovell. An efficient and robust sequential algorithm for background estimation in video surveillance. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 1109–1112, 2009. [Cited on pages 16, 17, 24, and 36.]

V. Reddy, C. Sanderson, and B. Lovell. A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. *EURASIP Journal on Image and Video Processing*, pages 1–14, 2011. [Cited on pages 16, 17, 20, 24, 36, 37, 38, and 39.]

P. Rodriguez and B. Wohlberg. Fast principal component pursuit via alternating minimization. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 69–73, 2013. [Cited on page 40.]

H. Sajid and S.-C. Samson Cheung. Background subtraction for static & moving camera. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 4530–4534, 2015. [Cited on pages 77, 80, and 108.]

A. Salti, S. Lanza and L. Stefano. Synergistic change detection and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(4):609–622, 2015. [Cited on page 71.]

A. Sánchez Rodríguez, J. González Castolo, and Ó. Déniz Suárez. Timeviewer, a tool for visualizing the problems of the background subtraction. In *Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 372–384, 2014. [Cited on pages 8 and 68.]

A. Sanin, C. Sanderson, and B. Lovell. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern Recognition*, 45:1684–1695, 2012. [Cited on pages 8, 71, 94, and 95.]

J. SanMiguel and A. Cavallaro. Temporal validation of particle filters for video tracking. *Computer Vision and Image Understanding*, 131:42–55, 2015. [Cited on page 67.]

J. SanMiguel and J. Martinez. On the evaluation of background subtraction algorithms without ground-truth. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 180–187, 2010. [Cited on pages 8, 9, 68, 69, 123, and 133.]

A. Schick, M. Bauml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 27–31, 2012. [Cited on pages 7, 94, 95, 96, 106, 114, and 115.]

M. Sedky, M. Moniri, and C. C. Chibelushi. Spectral-360: A physics-based technique for change detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 405–408, 2014. [Cited on page 108.]

J.-W. Seo and S. Kim. Recursive on-line $(2D)^2$PCA and its application to long-term background subtraction. *IEEE Transactions on Multimedia*, 16(8):2333–2344, 2014. [Cited on page 63.]

K. Seshadrinathan and J. Caviedes. Control of video processing algorithms based on measured perceptual quality characteristics. In *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 177–180, 2012. [Cited on page 67.]

R. Shi, K. Ngan, S. Li, R. Paramesran, and H. Li. Visual quality evaluation of image object segmentation: Subjective assessment and objective measure. *IEEE Transactions on Image Processing*, 24(12):5033–5045, 2015. [Cited on pages 9 and 69.]

A. Shrotre and L. Karam. Background recovery from multiple images. In *Proceedings of IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 135–140, 2013. [Cited on page 18.]

M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. *CoRR*, abs/1707.02432, 2017. [Cited on page 3.]

A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21, 2014. [Cited on pages 15, 40, and 108.]

A. Sobral, C. Baker, T. Bouwmans, and E.-h. Zahzah. Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction. In *Proceedings of International Conference on Image Analysis and Recognition (ICIAR)*, pages 94–103, 2014. [Cited on page 94.]

A. Sobral, T. Bouwmans, and E.-h. Zahzah. Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos. In *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor and Francis Group., 2015. [Cited on page 40.]

Y. Song, S. Noh, and M. Jeon. A new performance evaluation software for background subtraction algorithms. In *Proceedings of IEEE International Symposium on Consumer Electronics (ISCE)*, pages 1–2, 2014. [Cited on pages 8 and 68.]

P. St-Charles and G. Bilodeau. Improving background subtraction using local binary similarity patterns. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV*, pages 509–515, 2014. [Cited on pages 40 and 108.]

P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2015. [Cited on pages 4, 7, 8, 17, 40, 71, 80, 94, 95, 96, 108, 111, 124, and 135.]

P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. Universal background subtraction using word consensus models. *IEEE Transactions on Image Processing*, 25(10):4768–4781, 2016. [Cited on page 108.]

C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 246–252, 1999. [Cited on pages 4, 75, 77, 79, 80, and 108.]

G. Szwoch. Extraction of stable foreground image regions for unattended luggage detection. *Multimedia Tools and Applications*, 75(2):1–26, 2014. [Cited on page 56.]

A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu. Efficient background modeling through incremental support vector data description. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. [Cited on page 93.]

Y. Tian, Y. Wang, Z. Hu, and T. Huang. Selective eigenbackground for background modeling and subtraction in crowded scenes. *IEEE Transactions on Circuits and Systems for Video Technology*, 23 (11):1849–1864, 2013. [Cited on pages 17 and 93.]

P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. [Cited on pages 124 and 135.]

K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 255–261, 1999. [Cited on page 34.]

D. M. Tsai and S. C. Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE Transactions on Image Processing*, 18(1):158–167, 2009. [Cited on page 93.]

Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 760–775, 2016. [Cited on page 4.]

A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. [Cited on page 70.]

J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. [Cited on pages 96 and 97.]

A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre. A benchmark dataset for outdoor foreground/background extraction. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 291–300, 2013. [Cited on pages 95 and 107.]

B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013. [Cited on page 40.]

J. Varadarajan and J. Odobez. Topic models for scene analysis and abnormality detection. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1338–1345, 2009. [Cited on page 34.]

T. Vatanen, M. Osmala, T. Raiko, K. Lagus, M. Sysi-Aho, M. Oresic, T. Honkela, and H. Lähdesmäki. Self-organization and missing values in SOM and GTM. *Neurocomputing*, 147:60–70, 2015. [Cited on pages 83 and 84.]

A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *CoRR*, abs/1601.07140, 2016. [Cited on page 69.]

J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000. [Cited on pages 83 and 84.]

P. Villegas and X. Marichal. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 13(8):1092–1103, 2004. [Cited on page 69.]

H. Vojodi, A. Fakhari, and A. Moghadam. A new evaluation measure for color image segmentation based on genetic programming approach. *Image and Vision Computing*, 31(11):877–886, 2013. [Cited on pages 9 and 69.]

Y. W., Z. Luo, and P.-M. Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 96:66–75, 2017. [Cited on pages 7, 94, and 109.]

B. Wang and P. Dudek. A fast self-tuning background subtraction algorithm. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 401–404, 2014. [Cited on pages 80, 108, and 111.]

H. Wang and D. Suter. A novel robust statistical method for background initialization and visual surveillance. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, volume 3851, pages 328–337. 2006. [Cited on pages 17 and 39.]

K. Wang, B. Wang, and L. Peng. Cvap: Validation for cluster analyses. *Data Science Journal*, 8:88–93, 2009. [Cited on page 22.]

M. Wang, W. Li, and X. Wang. Transferring a generic pedestrian detector towards specific scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3274–3281, 2012. [Cited on page 34.]

R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 420–424, 2014a. [Cited on pages 75, 80, 108, 124, and 135.]

W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3395–3402, 2015a. [Cited on page 5.]

W. Wang, J. Shen, and L. Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Transactions on Image Processing*, 24(11):4185–4196, 2015b. [Cited on page 4.]

Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: An expanded change detection benchmark dataset. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 393–400, 2014b. [Cited on pages 8, 34, 68, 70, 79, 93, 95, 107, 122, and 132.]

L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. JOTS: Joint Online Tracking and Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2226–2234, 2015. [Cited on page 71.]

M. Wertheimer. *Laws of Organization in Perceptual Forms (partial translation)*, pages 71–88. 1938. [Cited on page 89.]

P. J. Withagen, K. Schutte, and F. C. A. Groen. Global intensity correction in dynamic scenes. *International Journal of Computer Vision*, 86(1):33–47, 2009. [Cited on page 71.]

C. Wolf, E. Lombardi, J. Mille, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandréa, C. Bichot, C. Garcia, and B. Sankur. Evaluation of video activity localizations integrating quality and quantity measurements. *Computer Vision and Image Understanding*, 127:14–30, 2014. [Cited on page 34.]

Q. Wu, H. Cheng, and B. Jeng. Motion detection via change-point detection for cumulative histograms of ratio images. *Pattern Recognition Letters*, 26(5):555–563, 2005. [Cited on page 58.]

Y. Xiao, C. Lu, E. Tsougenis, Y. Lu, and C.-K. Tang. Complexity-adaptive distance metric for object proposals generation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 778–786, 2015. [Cited on pages 96 and 97.]

C. Xu and J. J. Corso. Actor-action semantic segmentation with grouping process models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3083–3092, 2016. [Cited on page 4.]

J. Xu, S. Denman, S. Sridharan, and C. Fookes. Activity analysis in complicated scenes using dft coefficients of particle trajectories. In *Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 82–87, 2012. [Cited on page 34.]

X. Xun and T. Huang. A loopy belief propagation approach for robust background estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2008. [Cited on pages 18 and 45.]

Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013. [Cited on page 110.]

M. H. Yang, C. R. Huang, W. C. Liu, S. Z. Lin, and K. T. Chuang. Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(4):595–608, 2015. [Cited on pages 5 and 93.]

J. Yao and J. Odobez. Multi-layer background subtraction based on color and texture. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. [Cited on page 108.]

X. Yao, J. Han, D. Zhang, and F. Nie. Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering. *IEEE Transactions on Image Processing*, 26(7): 3196–3209, 2017. [Cited on page 4.]

T. YingLi, R. Feris, L. Haowei, A. Hampapur, and S. Ming-Ting. Robust detection of abandoned and removed objects in complex surveillance videos. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 41(5):565–576, 2011. [Cited on pages 55 and 61.]

Y. Yuan, S. Emmanuel, Y. Fang, and W. Lin. Visual object tracking based on backward model validation. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1898–1910, 2014. [Cited on page 67.]

D. Zhang, D. Meng, and J. Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):865–878, 2017a. [Cited on page 4.]

D. Zhang, L. Yang, D. Meng, D. Xu, and J. Han. Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5340–5348, 2017b. [Cited on page 5.]

H. Zhang, J. Fritts, and S. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008. [Cited on pages 67 and 71.]

R. Zhang, W. Gong, A. Yaworski, and M. Greenspan. Nonparametric on-line background generation for surveillance video. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1177–1180, 2012. [Cited on pages 16 and 17.]

W. Zhang, A. Borji, Z. Wang, P. Le Callet, and H. Liu. The application of visual saliency models in objective image quality assessment: A statistical evaluation. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1266–1278, 2016. [Cited on page 67.]

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017a. [Cited on pages 4, 124, and 134.]

X. Zhao, Y. Chen, M. Tang, and J. Wang. Joint background reconstruction and foreground segmentation via A two-stage convolutional neural network. *CoRR*, abs/1707.07584, 2017b. [Cited on pages 124 and 135.]

C. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 391–405, 2014. [Cited on page 71.]