



UNIVERSIDAD NACIONAL DE COLOMBIA

Video and Image Processing based on Kernel Representations

Santiago Molina Giraldo

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electric, Electronic and Computing Engineering
Manizales, Colombia
2014

Video and Image Processing based on Kernel Representations

Santiago Molina Giraldo

Thesis submitted as a partial requirement to receive the grade of:
Master in Engineering - Industrial Automation

Advisor:

Ph.D. Germán Castellanos Domínguez

Co-advisor:

MSc. Andrés Marino Álvarez Meza

Academic Research Group:

Signal Processing and Recognition Group - SPRG

Universidad Nacional de Colombia
Faculty of Engineering and Architecture
Department of Electric, Electronic and Computing Engineering
Manizales, Colombia

2014

Procesamiento de Video e Imágenes basado en Representaciones Kernel

Santiago Molina Giraldo

Tesis presentada como requisito parcial para optar al título de:
Magister en Ingeniería - Automatización Industrial

Director:

Ph.D. Germán Castellanos Domínguez

Co-director:

MSc. Andrés Marino Álvarez Meza

Grupo de trabajo académico:

Grupo de Procesamiento y Reconocimiento de Señales - GPRS

Universidad Nacional de Colombia
Facultad de Ingeniería y Arquitectura
Departamento de Ingeniería Eléctrica, Electrónica y Computación
Manizales, Colombia
2014

Dedico este trabajo a mi familia.

Acknowledgements

First of all, I want to thank my family for supporting my decisions and for being there in every moment of my life to share their love and comprehension.

I would like to express my gratitude to Prof. Germán Castellanos Domínguez for his orientation during this research. Besides, I thank to all my partners of the GC&PDS for their support, advisement and fellowship. Specially I would like to thank to Andrés Marino Álvarez, César Augusto Aguirre, Juan Sebastián Castaño and Andrés Eduardo Castro.

Furthermore, I recognize that this research would not have been possible without the financial assistance provided by the Universidad Nacional de Colombia through its project of outstanding postgraduate student scholarships. Besides, thank to project "Análisis del rendimiento competitivo de tenistas caldenses empleando técnicas de aprendizaje de máquina sobre mediciones antropométricas, biomecánicas, de condición física y psicológicas" code 20101007319 and project "Plataforma tecnológica para los servicios de teleasistencia, emergencias médicas, seguimiento y monitoreo permanente de pacientes y apoyo a los programas de prevención - Eje 2". Special thanks to the "Dirección de Investigación Sede Manizales - DIMA".

Santiago Molina Giraldo
2014

Abstract

In this work, different kernel-based feature representation frameworks are proposed. Our main goal is to properly reveal the most relevant information from high dimensional data for enhancing the performance of two different computer vision procedures: image and video segmentation. To this end, we propose to use Multiple Kernel Representations (MKR) to incorporate multiple image features, such as: color representations, pixel spatial information and optical flow-based information. Particularly, for image segmentation, we propose a new grouping-based image segmentation methodology, which using a MKR-based methodology, incorporate multiple image features. Regarding video segmentation, different Gaussian-based background modeling approaches are studied in order to cope with typical video surveillance artifacts recorded by static cameras. In this sense, different cost functions are analyzed, aiming to learn and model the temporal evolution of pixel dynamics. Furthermore, in order to incorporate spatial relationships among pixels, an object motion analysis stage is developed, which using an optical flow-based methodology is able to detect, model and track moving objects in a given video sequence. As a final result, a new video segmentation approach called STAL is developed. Proposed STAL couples both temporal and spatial information sources, generating an adaptive learning framework able to properly update the Gaussian-based background model parameters. Proposed image and video segmentation frameworks are tested by using real-world datasets, and attained results are measured by using supervised measures while comparing against ground-truth sets. From the obtained results, it is shown that the use of MKR to elaborate feature representations enhances the accuracy of the resulting segmentations. Moreover, it is demonstrated that proposed frameworks are efficient and competitive while comparing against top state of the art algorithms.

Keywords: kernel representations, relevance analysis, image segmentation, video segmentation, background subtraction, stationary/non-stationary dynamics, optical flow, particle filter.

Resumen

En este trabajo se proponen diferentes esquemas de representación de características basados en kernels. El objetivo principal es revelar de forma apropiada la información más relevante a partir de información de alta dimensionalidad, para mejorar el desempeño de dos procedimientos de visión por computador: Segmentación de imágenes y video. Con este fin, se propone usar Representaciones Multi-Kernel (MKR) para incorporar múltiples características de imágenes, como: representaciones de color, información espacial de píxeles y información basada en flujo óptico. Particularmente, para la segmentación de imágenes, se propone una nueva metodología de segmentación de imágenes basada en agrupamiento, la cual usando una metodología basada en MKR, incorpora múltiples características de imágenes. Con respecto a la segmentación de video, diferentes metodologías de modelado de fondo usando representaciones Gaussianas son estudiadas con el fin de enfrentarse a los artefactos típicos de video-vigilancia capturados por cámaras estáticas. En este sentido, diferentes funciones de costo son analizadas, buscando aprender y modelar la evolución temporal de la dinámica de los píxeles. Además, con el fin de incorporar relaciones espaciales entre píxeles, una etapa de análisis de movimiento de objetos es desarrollada, la cual usando una metodología basada en flujo óptico, es capaz de detectar, modelar y seguir objetos en movimiento en una secuencia de video dada. Como resultado final, una nueva metodología de segmentación de video llamada STAL es desarrollada. STAL acopla ambas fuentes de información (temporal y espacial), generando así un esquema de aprendizaje adaptativo capaz de actualizar adecuadamente los parámetros del modelo Gaussiano del fondo. Los esquemas de segmentación de imágenes y video son probados empleando bases de datos del mundo real y los resultados obtenidos son medidos usando medidas supervisadas mientras se compara contra conjuntos de *ground-truth*. A partir de los resultados obtenidos, se muestra que el uso de MKR para la elaboración de espacios de representación mejora la precisión de las segmentaciones resultantes. Además, se demuestra que los esquemas propuestos son eficientes y competitivos comparando contra algoritmos del estado del arte.

Palabras clave: representaciones kernel, análisis de relevancia, segmentación de imágenes, segmentación de video, *background subtraction*, dinámicas estacionarias/ no estacionarias, flujo óptico, filtro de partículas.

Contents

. Acknowledges	viii
. Abstract	ix
. Resumen	x
. List of Figures	xiv
. List of Tables	xv
I. Introduction	1
1. Motivation	2
2. State of the Art	3
3. Objectives	6
3.1. General objective	6
3.2. Specific objectives	6
II. Materials and Methods	7
4. A New Kernel-based Representation to Support Image and Video Segmentation	8
4.1. Kernel Representations Fundamentals	9
4.2. Multiple Kernel Learning	10
4.2.1. MKL Weight Selection based on Feature Relevance Analysis	10
4.3. Weighted Gaussian Kernel Image Segmentation	11
4.3.1. WGKIS Experiments	12
4.3.2. Discussion	16
4.4. Weighted Gaussian Kernel Video Segmentation	17
4.4.1. WGKVS Background Initialization	18
4.4.2. WGKVS Background Subtraction	18
4.4.3. WGKVS Experiments	19

4.4.4. Discussion	26
4.5. Conclusions	27
5. A New Approach to Incorporate Data Spatial Relations to Support Video Segmentation Tasks	29
5.1. Highlighting Spatial Relationships based on Optical Flow	30
5.1.1. Region Change Detection	30
5.1.2. Motion Modeling by Optical Flow	30
5.1.3. Object Movement Identification and Static Object Memory Computation	31
5.2. Background Modeling using Variability Constraints	32
5.3. Kernel-based Background Subtraction	33
5.3.1. Dealing with Illumination Changes	33
5.4. Experiments	34
5.5. Discussion	39
5.6. Conclusions	40
6. Kernel based Spatio-Temporal Adaptive Learning: a New Video Segmentation Approach	42
6.1. Spatio-Temporal Adaptive Learning	43
6.1.1. Adaptive learning for background modeling	43
6.1.2. Learning rate estimation based on object motion analysis	46
6.2. Experiments and Discussion	51
6.2.1. Pixel temporal relationship using Correntropy-based cost function . .	53
6.2.2. Improved object motion analysis using pixel spatio-temporal relationships	56
6.2.3. Background and foreground discrimination based on STAL	59
6.3. Conclusions	70
III. Conclusions and Future Work	72
7. Conclusions	73
7.1. Academic Discussion	75
8. Future work	76
IV. Appendix	77
A. MSE based cost function for background modeling	78
. Bibliography	86

List of Figures

4-1. Mapping function example.	9
4-2. WGKIS scheme.	12
4-3. Hand segmented ground-truth samples.	13
4-4. Including multiple information sources results. — RGB kmeans, — GKIS, — WGKIS.	14
4-5. Blond-girl segmentation results.	14
4-6. Weight selection by relevance feature analysis for the blond-girl image.	15
4-7. Image segmentation results. — EDISON1, — EDISON2, — EDISON3 and — WGKIS.	15
4-8. Image segmentation samples. Column 1: Original image. Column 2: WGKIS. Coulmn 3: EDISON.	16
4-9. WGKVS scheme.	19
4-10. DBa-Fountain segmentation results.	21
4-11. Relevance weights for sequence DBa-Fountain (Frame 1509 th).	22
4-12. DBa-WaterSurface segmentation results.	22
4-13. Relevance weights for sequence DBa-WaterSurface (Frame 1523 rd).	23
4-14. DBb-LeftBag video segmentation results.	24
4-15. DBa-ShoppingMall video segmentation results.	24
4-16. WGKVS segmented objects samples.	26
4-17. Geometrical features samples for objects of the two classes.	26
5-1. Proposed methodology scheme.	33
5-2. Motion detection results for DBb-LeftBag video	35
5-3. Detection of static object after movement (DBb-LeftBag video)	36
5-4. Segmentation results using the parameter configurations exposed in Table 5-2 for the illumination and shadow debugging stage. Column 1: Original frame, Column 2: 'No debug', Column 3: 'Debug 1', Column 4: 'Debug 2'	37
5-5. DBb-LeftBag video segmentation results.	38
5-6. DBa-WaterSurface video segmentation results.	39
6-1. Kernel bandwidth selection strategy. — pdf of error e_t , - - error standard deviation σ_{e_t} , - - kernel bandwidth ϕ_t	46
6-2. Proposed motion detection scheme	47

6-3. Object detection and modeling scheme	49
6-4. Particle filter based object tracking scheme	52
6-5. STAL scheme.	52
6-6. Correntropy-based stochastic gradient update.	54
6-7. Background modeling (μ_t parameter). — Pixel value, — MSE based model, — Correntropy based model.	55
6-8. Correntropy-based cost function evolution. — Normalized error, — Corren- tropy value.	56
6-9. Tracked objects obtained from DBd-StretLight and DBa-ShoppingMall sam- ple videos.	58
6-10. Object motion analysis performance against occlusions/crossing artifacts. . .	65
6-11. DBb-LeftBox frame 297. Dealing with bootstrapping and static objects . . .	66
6-12. Segmentation performance for different pixel dynamics. Column 1: Original frame. Column 2: STAL segmentation. Column 3: STAL-MSE segmentation. Column 4: Ground-truth.	66
6-13. Segmentation performance over challenging conditions. Column 1: Original frame with STAL tracking blobs. Column 2: STAL segmentation. Column 3: Z-GMM segmentation. Column 4: SC-SOBS segmentation. Column 5: PBAS segmentation. Column 6: Ground-truth.	68
6-14. Average F1 measure per video category: — i), — ii), — iii) varying the time window size	69
6-15. Average F1 measure per video category: — i), — ii), — iii) varying the number of particles	69
6-16. Average F1 measure per video category: — i), — ii), — iii) varying the segmentation threshold	70

List of Tables

4-1. Average number of groups and <i>PR</i> results for EDISON1, EDISON2, EDISON3 and WGKIS	14
4-2. <i>PR</i> results and number of groups for images of Figure 4-8	15
4-3. Segmentation performance for the three configurations of WGKVS	23
4-4. Segmentation performance for SOBS and WGKVS	24
4-5. Considered features for abandoned object classification.	25
4-6. Confusion matrix using the Knn classifier.	25
5-1. Proposed methodology parameters.	34
5-2. Parameter configurations for illumination and shadow debugging stage.	35
5-3. Segmentation performance using the three parameter configurations for the illumination and shadow debug stage.	36
5-4. Segmentation performance for the proposed approach and SOBS.	38
6-1. Object motion analysis parameters.	57
6-2. Estimated pixel-based F_1 measure for STAL and STAL-MSE approaches.	60
6-3. Estimated pixel-based F_1 measure for STAL, Z-GMM, SC-SOBS and PBAS approaches.	64
6-4. Estimated computational cost of the proposed STAL approach.	71

Part I.

Introduction

1. Motivation

Computer vision is a field of artificial intelligence which aims to electronically perceive and understand dynamics from an image and, in general, high-dimensional data obtained from the real world in order to produce numerical or symbolic information, e.g., in terms of decisions [1, 2]. Due to the fast spread of digital cameras and the impressive technical enhancements performed during the last decade, the development of new computer vision algorithms has become a research topic of great importance. Computer vision applications are widely used and they affect most of the daily human activities. Some applications of interest are: Medical imaging, industrial processes control, video surveillance, human computer interfaces, autonomous navigation, among others. Particularly, video surveillance applications are nowadays a topic of huge interest, since they facilitate the analysis of large amounts of information recorded by complex camera schemes in public places, which is not feasible to analyze online by human supervision in most of the cases[3, 4]. Some relevant video-based surveillance applications include pedestrian and car traffic monitoring, unusual human activity detection, people counting, among others.

In a local context, the research group *Grupo de Procesamiento y Reconocimiento de Seales* (GPRS) of the *Universidad Nacional de Colombia* has been working on the development of machine learning based methodologies to support different computer vision applications. As an example, medical imaging has been a research topic of interest in the group, particularly some works related to the segmentation of brain structures have been developed in [5]. Moreover, the use of thermographic images has been also explored to detect failures in rotating machines [6, 7]. Recently, in the GC&PDS, video-based computer vision applications have received major attention. In particular, some video segmentation [8, 9] frameworks have been developed to support video surveillance systems.

2. State of the Art

Traditionally, computer vision systems comprehend six main stages related to machine learning and pattern recognition, which are: Image acquisition, pre-processing, feature representation, detection/segmentation, high level processing and decision making. Particularly, feature representation is a crucial stage of computer vision systems, due to the high dimensionality of image and video data. Commonly, computer vision approaches tend to use raw feature representations, such as color spaces (i.e. RGB, HSV, YCbCr, etc.) to describe the process [10, 11]. Additionally, the use of local features such as edges, is also very common in image applications [12–14]. However, these feature representations might not be enough to describe the process if there exists non-linear information among data samples. In this regard, kernel methods aim to represent data samples into a new feature space by using non-linear mapping functions. Hence, kernel representations discover linear and non-linear similarities among samples, encoding them into a symmetric, positive defined matrix called kernel matrix [15]. Although, a single kernel helps to improve the representation of computer vision systems [16, 17], it might not be adequate to describe a process with multiple information sources. Particularly, for image and video segmentation tasks, it is desirable to enhance the separability and interpretability of the image regions of interest by including non-linear relations and moreover, by incorporating all possible information sources into the process such as: different color representations and spatial information [18].

After dealing with computer vision feature representation approaches, it is necessary to develop suitable segmentation methods in order to support further high level analysis. Regarding this, several image segmentation approaches have been proposed, their main goal consists in splitting an image into disjoint regions such that the pixels inside them have a high similarity according to a pre-set property and contrast high difference among regions. The main goal is to obtain a proper segmentation that can be used in processes such as: video object extraction [19], object recognition systems [20], and region-based tracking systems [21]. Mainly, image segmentation methods can be divided into three categories. The first category are the histogram-based algorithms, which considering the histogram of an image as a probability density function, perform the segmentation by classifying pixels using a threshold parameter [22, 23]. However, the proper selection of the threshold parameter represents a difficult task which could slant the algorithm, even more if the image is represented by multimodal histograms [24]. The second category are the boundary-based algorithms, which assume that value changes among neighbor pixels inside a region is not as significant as changes among pixel values on the boundary of a region, therefore regions

can be segmented when the boundaries are detected [13, 25]. The main drawback is that mostly the attained boundaries are not totally closed. Therefore, post-processing algorithms must be used to connect open boundaries [12]. Nevertheless, these algorithms always tend to attain over-segmented results. Finally, grouping-based algorithms aim to group pixels in the same cluster if they have similar patterns or characteristics, while pixels grouped into different clusters have different characteristics. Overall, these kind of algorithms achieve accurate results, nonetheless, traditional grouping algorithms, e.g., Kmeans and Expected Maximization, tend to fall into local optimal, whereas spectral clustering algorithms can converge in a global optimal [26].

In addition, as an extension of image segmentation methods, video segmentation also called background subtraction appears as a necessity to develop online systems to support video-based computer vision systems, e.g. video surveillance, people counting, traffic monitoring. Video segmentation consists in the extraction of moving objects (foreground) from video recordings. In this task, each new input frame is compared against a background model, in order to label as foreground those elements that deviate significantly from it [27, 28]. As a basic, a frame of the recording with no foreground objects could be considered as the background model, however, even though such image may be feasible, the problem arises due to the intrinsic time and spatial non-stationary fluctuations within considered recorded scenes, leading to an unsuitable background estimation [29, 30]. To deal with this issue, the background modeling stage is developed as an online learning task being robust to different video-based surveillance conditions, namely, illumination changes, shadows, camouflage, bootstrapping, static and moving foreground objects, etc. [28, 31, 32].

For this purpose, several dynamic background modeling methods based on updating one or multiple Gaussian representations are employed to model pixel intensity along the time. By instance, temporal pixel variations can be measured by updating both the mean and the standard deviation of assumed normal distributions, as proposed in [33, 34] where a single pixel Gaussian model is updated under a running average scheme. Yet, these approaches are reserved for video scenarios under highly stationary constraints, which are not always accomplished in real-world applications.

Instead, pixel models based on Gaussian Mixture Models (GMM) have also proposed to deal with more complex dynamical scenarios and gradual illumination fluctuations [35–37]. However, GMM based pixel models face the following drawbacks: they require for adding and pruning schemes that are far from being a straightforward task in real-world video scenarios, these models tend to have bootstrapping problems if there are foreground objects during initial learning segments [38], they assume that the background is most frequently visible than the foreground that is a not valid consideration for every video segment, and estimation of the model free parameters (the Gaussian mean and variance) becomes problematic in most of the real-world noisy environments that are non-Gaussian [39]. To deal with dynamical backgrounds with fast background variations (waving trees, water rippling, etc.), models that compute pixel probability density function using kernel estimators over recent samples

of intensity values are also discussed in [40, 41]. However, these multi-modality models have some computational disadvantages. Specifically, because of a commonly large fixed number of frames, they must store in memory a huge amount of information through the entire process [38]. Moreover, tuning of free kernel parameters is still an open issue.

Subspace learning techniques representing each background pixel in terms of the so called eigen-background model have also been studied as an alternative [42, 43]. Nonetheless, there are limitations relating to foreground dynamics (frequently provoking false segmentations) and computational burden of required eigen-based decompositions that become significantly expensive for real-time applications [38].

Recently, it has been demonstrated that codebook-based background subtraction techniques [3, 44–47], attain suitable results for video segmentation challenges [48]. Their main goal is to model each pixel by a sample set (called codebook) that is learned from a training video segment, which in some cases is required to be long. Afterwards, an updating scheme is carried out to adapt the codebook throughout the intrinsic scene dynamics. The main advantage of these kind of approaches is the low computational burden, which makes them suitable for real-time applications. However, depending on the established updating rules, these methods tend to have major segmentation drawbacks when background dynamics that were not present during the training phase are revealed afterwards (e.g. removed foreground objects, strong illumination changes) [32]. Other codebook-based background subtraction algorithms have problems when foreground elements remain motionless for long periods of times, leading to inaccurate segmentation results.

3. Objectives

3.1. General objective

To develop kernel representation frameworks able to support the analysis of image and video data, for both off-line and online modes, revealing the intrinsic non-linear data relationships, which can vary along space and time. Presented frameworks must be useful to facilitate the discrimination among data dynamics, enhancing the performance of both image and video segmentation tasks.

3.2. Specific objectives

- To develop a feature representation framework based on multiple kernel functions in order to incorporate different information sources about the process. Proposed framework should be useful to improve data interpretability and separability by properly weighting the relevance of each information source. Presented framework will be tested in image and video segmentation tasks comparing the attained results against state of the art algorithms by using supervised performance measures.
- To elaborate a multiple kernel-based feature representation methodology that allows to infer the main spatial dynamics from high dimensional data. Proposed methodology should be useful to support video segmentation tasks by revealing non-linear spatial relations among pixels. The performance of the developed methodology will be tested against state of the art approaches by using supervised measures given ground-truth data.
- To develop an online adaptive learning methodology using spatio-temporal kernel based representations to support the analysis of stationary/non-stationary data dynamics. Proposed methodology should be useful to discriminate, between intrinsic process behaviors which can be perturbed by Gaussian/non-Gaussian artifacts, enhancing the segmentation performance in video based surveillance systems.

Part II.

Materials and Methods

4. A New Kernel-based Representation to Support Image and Video Segmentation

Computer vision is a sub-field of artificial intelligence, which aims to build algorithms in order to give to a computer the capability to understand images and scenes. To this end, computer systems must perform some tasks and processes related to the fields of machine learning and pattern recognition. One of the most important tasks is the segmentation of images and video recordings, which main goal is to simplify and/or change the representation of an image or a recording into something that is more meaningful and easier to analyze [2].

In this chapter, we propose to develop a feature representation methodology that incorporates color intensity and spatial information based on Multiple Kernel Learning [49]. Moreover, the weighting factors of the MKL combination are automatically tuned according to an eigenvalue feature relevance analysis. Proposed scheme is used to support computer vision feature representations by incorporating multiple image features to describe each pixel. Particularly, developed methodology is employed to support two different motion analysis tasks: Image segmentation and video segmentation.

With this in mind, two different frameworks are developed. The first one is a grouping based image segmentation approach called Weighted Gaussian Image Segmentation (WGKIS), which incorporates different color representations and spatial information based on MKL to conform a weighted Gaussian kernel matrix, afterwards, an spectral clustering approach is employed to segment the image into different regions. The second developed framework is a Gaussian-based video segmentation approach called Weighted Gaussian Video Segmentation (WGKVS), which employing multiple kernel representations incorporates different color representations to conform an enhanced feature representation space. A grouping based algorithm is employed over the attained feature space to group pixels into foreground or background.

Experiments are performed aiming to test the performance of both WGKIS and WGKVS frameworks, when including multiple weighted image information sources. Resulting segmentations are evaluated by supervised measures attained when comparing against ground-truth sets. Furthermore, WGKIS and WGKVS performance is compared against the one attained by state of the art algorithms, while using real-world datasets.

This chapter presents the fundamentals of kernel representations and MKL in sections § 4.1

and 4.2. Then, the methodology to automatically set the MKL weighting factors is exposed in section § 4.2.1. After that, in section § 4.3, we present the developed image segmentation WGKIS. Experiments to assess the proposed WGKIS framework and the corresponding discussion are exposed in sections § 4.3.1 and § 4.3.2, respectively. The WGKVS framework for video segmentation is presented in section § 4.4. Sections § 4.4.3 and § 4.4.4 shows the experiments and discussion regarding to the proposed video segmentation framework. Finally, we conclude about the attained results in section § 4.5.

4.1. Kernel Representations Fundamentals

A kernel is a similarity function κ that embeds data samples from a Hilbert space to a Reproducing Kernel Hilbert Space (RKHS) by using a non-linear mapping function φ . Namely, the kernel function between data samples $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X} \subset \mathbb{R}^Z$, being Z the number of features, is defined as:

$$\begin{aligned} \kappa : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathbb{R}^+ \\ (\mathbf{x}, \hat{\mathbf{x}}) &\longmapsto \kappa(\mathbf{x}, \hat{\mathbf{x}}) = \langle \varphi(\mathbf{x}), \varphi(\hat{\mathbf{x}}) \rangle \end{aligned} \quad (4-1)$$

Given a feature space \mathcal{X} , the mapping function φ helps to discover non-linear relationships among samples, obtaining a new embedded RKHS space \mathcal{H} where the data can be separated by using linear operations (see Figure 4-1).

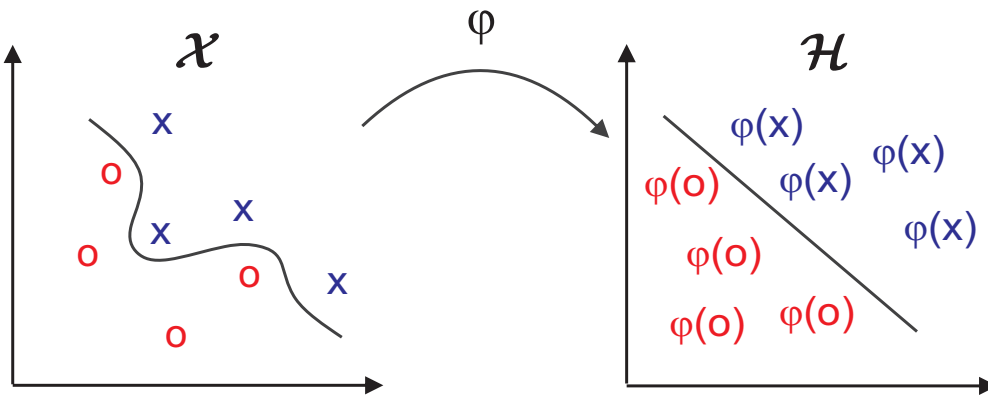


Figure 4-1.: Mapping function example.

However, the computation of the inner product in the mapped space seems to be inconvenient. In this sense, the so called *kernel trick* instead of mapping the data via φ and then computing the inner product, allows to perform both procedures by using just one operation, leaving the mapping completely implicit. In fact, there is not need to know the mapping function φ but the modified inner product which actually is the kernel function κ [50].

The resulting feature representation after applying κ is called the kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$, which holds the similarity measured by the kernel function for each pair of data samples, where N is the total amount of samples. It should be quoted that the kernel matrix \mathbf{K} must be symmetric positive semi-definite in order to fulfill the Mercer's conditions [50].

4.2. Multiple Kernel Learning

Recently, machine learning approaches have shown that the use of multiple kernels instead of only one, can be useful to improve the interpretation of data [49]. With this aim, multiple kernel learning (MKL) [51] allows to compute the similarity among data samples \mathbf{x} and $\hat{\mathbf{x}}$, using their Z feature representations as:

$$\kappa_{\vartheta}(x_z, \hat{x}_z) = \sum_{z=1}^Z \vartheta_z \kappa(x_z, \hat{x}_z), \quad (4-2)$$

being x_z and \hat{x}_z the z -th feature value of samples \mathbf{x} and $\hat{\mathbf{x}}$, respectively, with $z = 1, \dots, Z$. Restricting the weighting factors of Eq. (4-2) to $\vartheta_z \geq 0$, and $\sum_{z=1}^Z \vartheta_z = 1$ ($\forall \vartheta_z \in \mathbb{R}$). Thereby, the input data can be analyzed from different information sources by means of a convex combination of basis kernels. Regarding to image processing, each pixel of an image can be represented by combining Z image features based on MKL, enhancing the data separability and supporting further clustering and/or classification stages. Nonetheless, as can be seen in Eq. (4-2), it is necessary to fix the ϑ_z weighting factors, to take advantage, as well as possible, of each feature representation.

4.2.1. MKL Weight Selection based on Feature Relevance Analysis

Aiming to fix the MKL weighting factors, we propose to employ feature relevance analysis. Regarding this, traditional approaches such as the Principal Component Analysis (PCA), try to find low-dimensional representations by searching the feature space directions with the greatest variance in order to project the data. Although these approaches are commonly used as feature extraction methods, they are useful to quantify the relevance of the original features, providing weighting factors which take into account the best representations from an explained variance point of view [52]. In this sense, we propose to select the MKL weighting factors ϑ_z from Eq. (4-2) by means of a relevance analysis over the original feature space.

Given a feature representation matrix $\mathbf{F} \in \mathbb{R}^{N \times Z}$, the relevance of each feature z can be identified as the weighting factors ϑ_z , which are calculated as:

$$\boldsymbol{\vartheta} = \sum_{b_d=1}^{B_d} |\lambda_b \mathbf{v}_b|, \quad (4-3)$$

where $\boldsymbol{\vartheta} \in \mathbb{R}^{Z \times 1}$ is a vector containing the weighting factors for all Z features, and λ_b and \mathbf{v}_b are the eigenvalues and eigenvectors of the covariance matrix $\mathbf{F}^T \mathbf{F}$, respectively. It can

be inferred that the largest values of ϑ lead to the best input attributes, since they exhibit overall higher correlations with the principal components. The $B_d \in \mathbb{N}$ value is fixed as the number of dimensions needed to conserve a percentage of the input data variability.

The proposed methodology for building enhanced feature representations is tested to support the characterization stage of computer vision tasks. Particularly, we developed two different frameworks on the field of image segmentation and video segmentation, which include the proposed methodology to incorporate multiple image feature representations in the process. Following, in sections § 4.3, § 4.3.1 and § 4.3.2 the proposed framework for image segmentation and its corresponding results are presented and discussed. Furthermore, the developed framework for video segmentation with its corresponding results and discussion are exposed in sections § 4.4, § 4.4.3 and § 4.4.4.

4.3. Weighted Gaussian Kernel Image Segmentation

We propose to implement the proposed MKL-based representation methodology to construct enhanced feature representations in a new image segmentation framework called Weighted Gaussian Kernel Image Segmentation (WGKIS). Proposed WGKIS for image segmentation incorporates multiple color and spatial representations by means of a MKL approach. Moreover, the feature relevance analysis proposed in section § 4.2.1 is carried out to automatically tune the feature weighting factors. The main goal of WGKIS is to attain meaningful segmentations which could be used in further processes like object recognition.

Given an image $\mathbf{X} \in \mathbb{R}^{w_R \times w_C \times Z}$, with w_R rows, w_C columns, and Z color and spatial feature representations, a feature space matrix $\mathbf{F} \in \mathbb{R}^{N \times Z}$ is calculated, with $N = w_R \times w_C$. The attained \mathbf{F} is employed to compute the MKL feature weighting factors ϑ as explained in section § 4.2.1.

Afterwards, Z Gaussian kernel matrices $\mathbf{G}^z \in \mathbb{R}^{N \times N}$ are computed. To this, a Gaussian kernel between each pair of pixels (i, j) and (i^*, j^*) as:

$$\kappa^z(x_z^{i,j}, x_z^{i^*,j^*}) = \exp\left(\frac{-(x_z^{i,j} - x_z^{i^*,j^*})^2}{2\psi\phi^2}\right), \quad (4-4)$$

where $\phi \in \mathbb{R}^+$ is the kernel bandwidth and $\psi \in \mathbb{R}^+$ stands to normalize the bandwidth value. In case of color feature representations it is tuned as $\psi = |x_z^{i,j}|_1 |x_z^{i^*,j^*}|_1$, with $|\cdot|$ as the 1-norm operator. For spatial information it is set according to the image size.

Having the weighting factors ϑ and the kernel matrices \mathbf{G}^z , a weighted Gaussian kernel matrix $\mathbf{G} \in \mathbb{R}^{\hat{w} \times \hat{w}}$ can be attained as $\mathbf{G} = \sum_{z=1}^Z \vartheta_z \mathbf{G}^z$, as described in Eq. (4-2). The segmentation process is carried out by using a kernel K-means algorithm [53] over the weighted Gaussian kernel matrix \mathbf{G} . Besides, the number of groups $k_g \in \mathbb{N}$ of the spectral clustering algorithm is estimated via an eigenvalue analysis performed over a weighted linear kernel computed as $\mathbf{K}_L = \mathbf{F}\mathbf{W}\mathbf{F}^T$, where $\mathbf{W}_{p \times p} = \text{diag}(\vartheta_{p \times 1})$ [54]. Attained labels by the

kernel K-means algorithm are stored into matrix $\mathbf{S} \in \mathbb{N}^{w_c \times w_R}$ with $S_{i,j} \in 1, 2, \dots, k_g$. Such matrix describe the resulting segmented image. The proposed WGKIS scheme is shown in Figure 4-2.

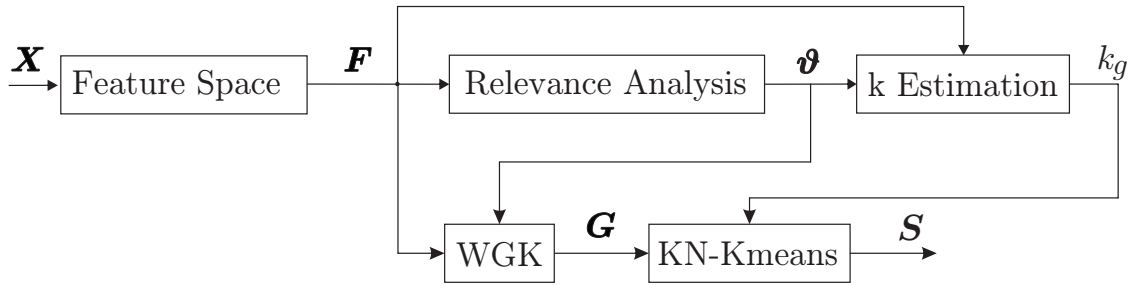


Figure 4-2.: WGKIS scheme.

4.3.1. WGKIS Experiments

The proposed methodology to build enhanced feature representations is tested as a supporting stage of the proposed WGKIS. In this regard, we aim to test how the weighted inclusion of multiple image sources improves the image segmentation performance. To this end, the Berkeley Image Segmentation Database [55] is employed. The database holds 300 color images of 481×321 depicting a large variety of real world scenarios. Hand-labeled segmentations made by 30 human subjects are available in order to perform objective evaluations. Original images are in *jpg* format and human segmentation results in *seg* format. For concrete testing, images are resized to 97×65 . The following color representations are employed to describe each pixel: RGB components, normalized rgb components, HSV components and YCbCr components. As spatial information, the row position i and the column position j of each pixel are used. Thus, for each image an input feature space $\mathbf{F} \in \mathbb{R}^{6305 \times 14}$ is obtained.

Since the image segmentation task does not have a unique solution, each human segmenter can provide a different segmentation for the same input image, as seen in Figure 4-3. Therefore, aiming to objectively evaluate the attained segmentation results, the Probabilistic Rand Index (PR) is employed, which allows to compare a segmentation against multiple hand labeled ground-truth images, through soft non-uniform weighting of pixel pairs as function of the variability in the ground-truth set [56]. Namely, the PR index aims to model the probability of a pair of pixels belonging to the same segmented region according to the human ground-truth set, therefore, it has the capability to perform comparisons even if the number of groups of each segmented image is different [57].

It is important to remark that for all the performed experiments, the WGKIS free parameter B_d is set in order to conserve a 90% of variability and the kernel bandwidth ϕ is heuristically set as 0.7.

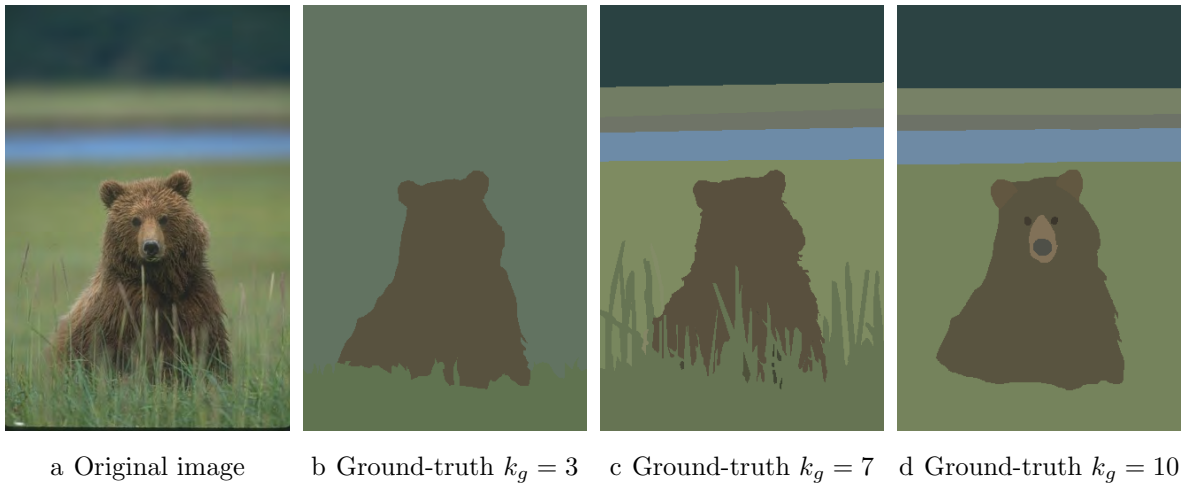


Figure 4-3.: Hand segmented ground-truth samples.

Two different kind of experiments are performed to test the developed WGKIS. The first kind experiment aims to prove the effectiveness of the proposed methodology to build enhanced feature representations for incorporating the multiple image features previously defined. To this end, 20 images are randomly selected from the Berkeley dataset. The WGKIS segmentation results over the 20 images are compared against GKIS (WGKIS with equal weighting factors), and against traditional Kmeans using only the RGB components. Figure 4-4 shows the attained results for each method. Furthermore, aiming to visually analyze attained results, the segmented images for each method are exposed in Figure 4-5 for the 388016 image of the dataset called blond-girl. The obtained WGKIS relevance weights for blond-girl are shown in Figure 4-6.

The second experiment is performed to compare the WGKIS algorithm against a traditional image segmentation algorithm named Edge Detection and Image Segmentation System (EDISON), which is a low-level feature extraction tool that integrates confidence-based edge detection and mean shift based image segmentation [58]. The EDISON system has been widely used as a reference to compare image segmentation approaches [59, 60]. For testing, the parameters of the EDISON system: scale bandwidth (b_s) and color bandwidth (b_c), are set as suggested in [60]. Thus, three different configurations are used: EDISON1 $b_s = 7, b_s = 7$, EDISON2 $b_s = 7, b_s = 15$ and EDISON3 $b_s = 20, b_s = 7$. 50 randomly selected images from the Berkeley database are used. The PR results for the second kind of experiments are presented in Figure 4-7. Furthermore, the average estimated number of groups and the average PR measure for all the 50 tested images are exposed in Table 4-1. Finally, some relevant results are shown in Table 4-2 and Fig. 4-8, the EDISON1 configuration is employed as baseline for the last .

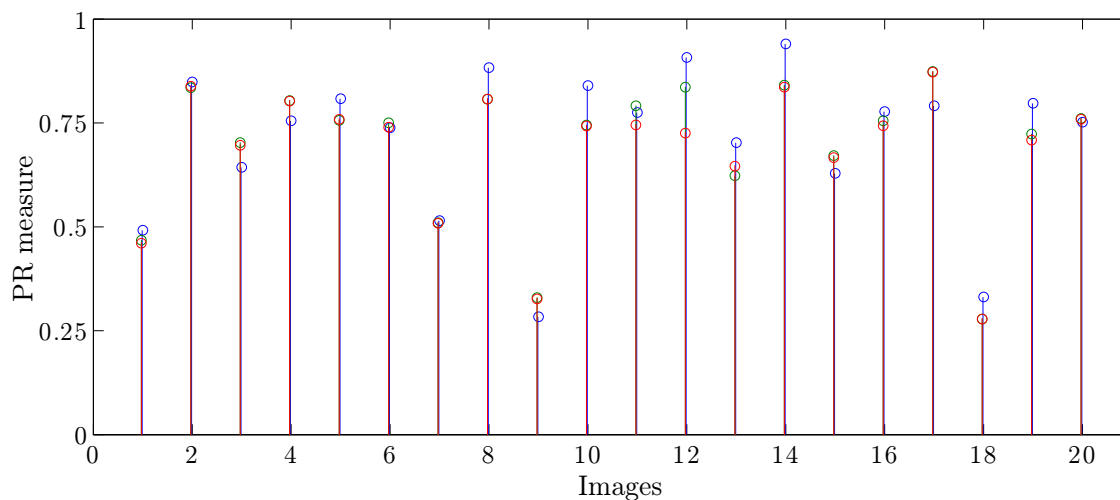


Figure 4-4.: Including multiple information sources results. — RGB kmeans, — GKIS, — WGKIS.

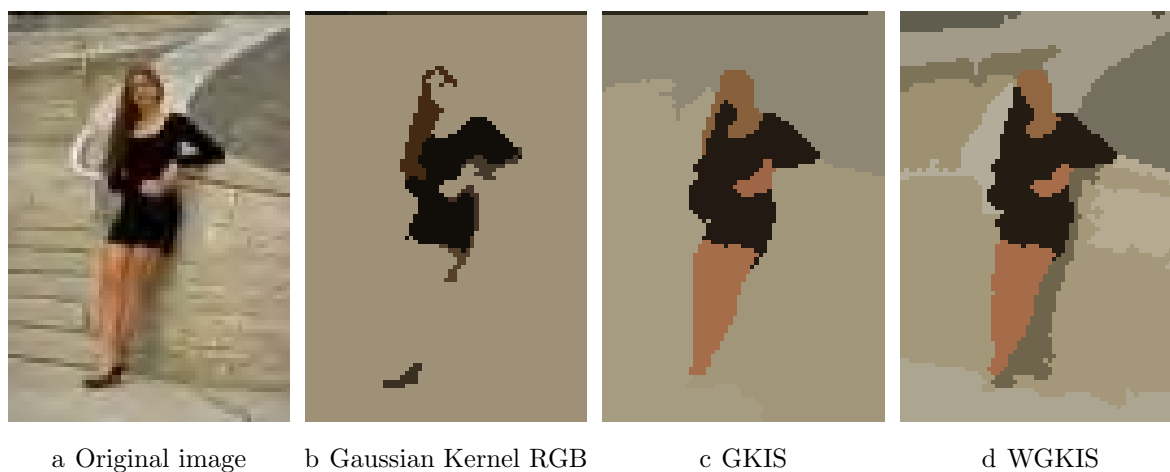


Figure 4-5.: Blond-girl segmentation results.

Table 4-1.: Average number of groups and PR results for EDISON1, EDISON2, EDISON3 and WGKIS

Method	kg	PR
EDISON1	79.68 ± 47.65	0.660 ± 0.191
EDISON2	21.68 ± 25.53	0.473 ± 0.208
EDISON3	54.68 ± 43.46	0.589 ± 0.205
WGKIS	9.862 ± 4.412	0.742 ± 0.142

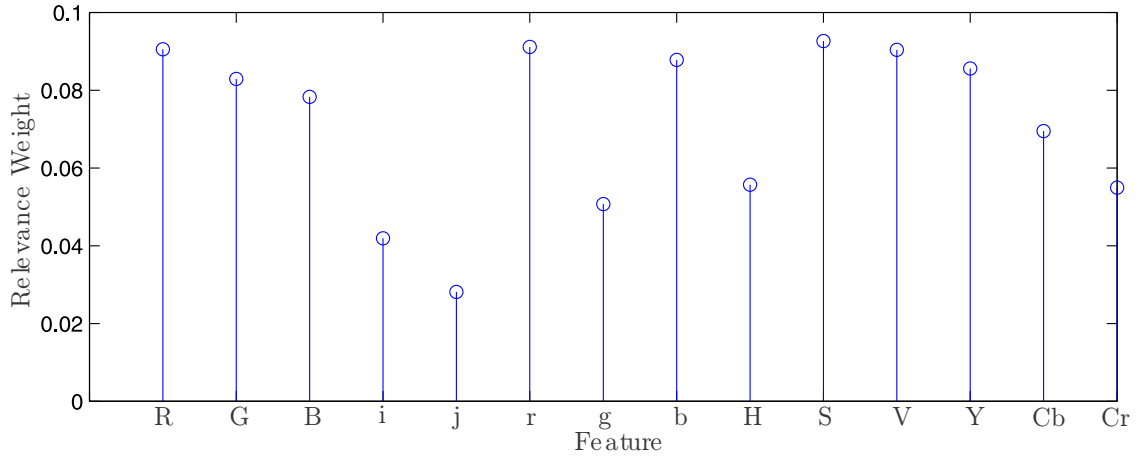


Figure 4-6.: Weight selection by relevance feature analysis for the blond-girl image.

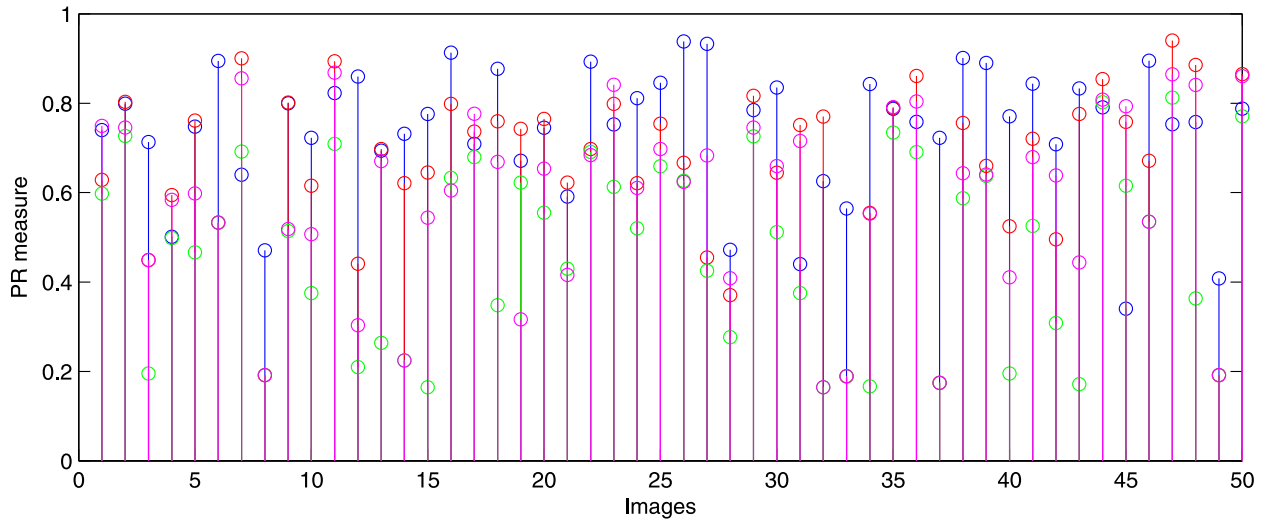


Figure 4-7.: Image segmentation results. — EDISON1, — EDISON2, — EDISON3 and — WGKIS.

Table 4-2.: *PR* results and number of groups for images of Figure 4-8

Method		a	b	c	d	e	f
EDISON1	<i>kg</i>	100	61	76	43	32	46
	<i>PR</i>	0.659	0.671	0.532	0.666	0.456	0.554
WGKIS	<i>kg</i>	8	6	4	4	9	10
	<i>PR</i>	0.890	0.894	0.897	0.936	0.932	0.842

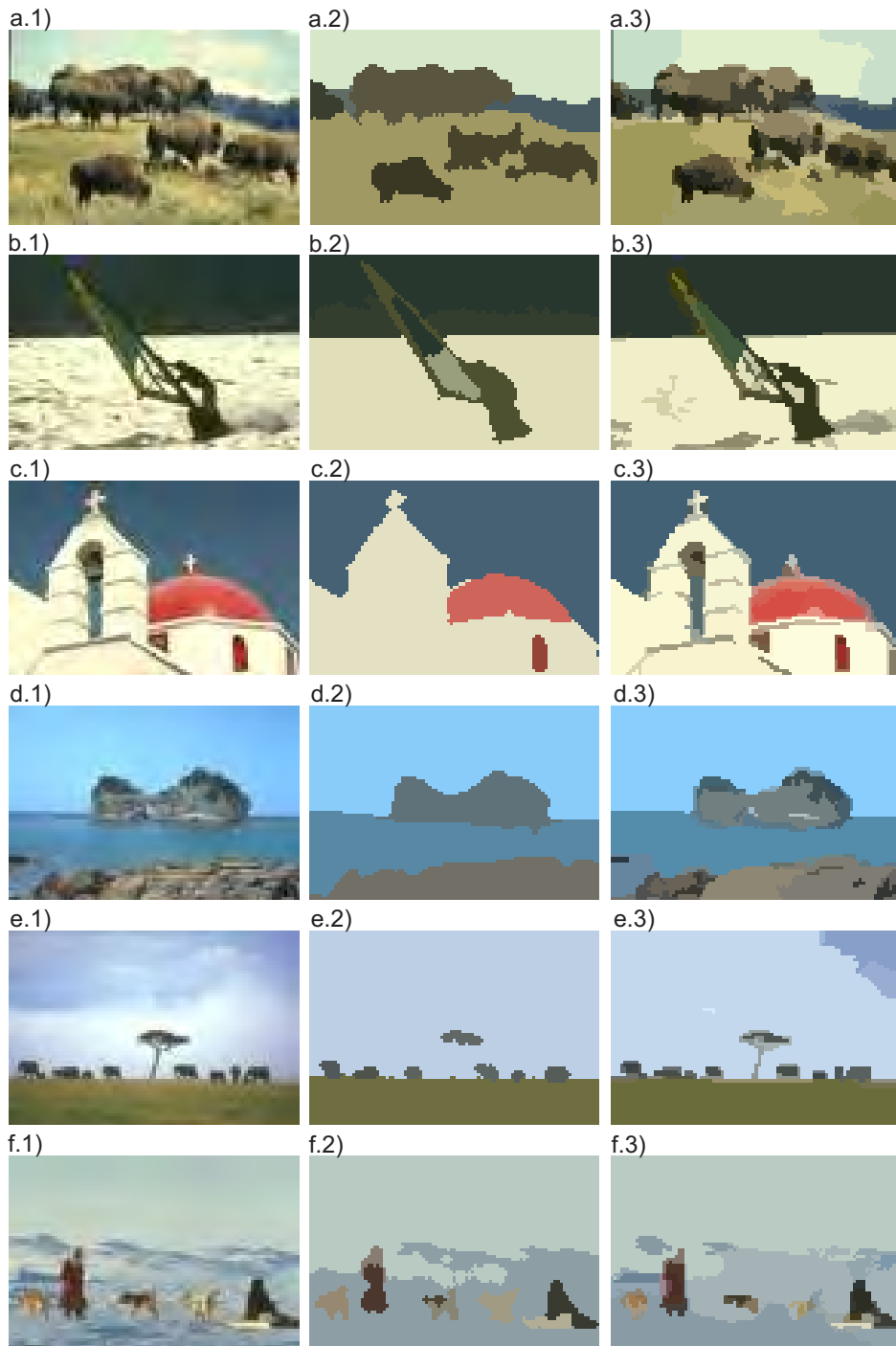


Figure 4-8.: Image segmentation samples. Column 1: Original image. Column 2: WGKIS. Column 3: EDISON.

4.3.2. Discussion

From the first experiment it can be seen that proposed WGKIS attain better results for most of the images (Figure 4-4), with an average PR measure of 0.716, while GKIS and RGB-

Kmeans obtain 0.699 and 0.688 PR values respectively. Demonstrating that the weighted inclusion of multiple image features enhances the pixel representation. A particular example are the results attained for the blond-girl image, it can be seen how the Gaussian kernel based segmentation using only RGB components poorly performs, lacking of extra information that could improve the estimation of the number of groups and the kernel K-means clustering (see Figure 4-5 b). The latter can be corroborated by a PR measure of 0.055. When the spatial and color spaces information are incorporated by means of the MKL approach, the performance improves significantly, obtaining a PR or 0.721 (see Figure 4-5 c). Finally, when including the proposed methodology to build enhanced feature representations, the best result is achieved obtaining a PR of 0.774. It can be explained by the estimated weights using the relevance analysis, which allows to identify the most relevant features, avoiding redundant information which could affect the pixel representation (see Figure 4-5 d).

When comparing the results of WGKIS against the state of the art EDISON system, it can be seen from Figure 4-7 that our methodology obtains the best results in most of the cases, obtaining the first place for 29 images, while EDISON1 for 16. In Table 4-1 are exposed the mean results for the 50 images from the database, it can be seen that the WGKIS algorithm obtains the best results with the highest mean PR measure 0.742, furthermore, obtains the best stability exposing the lowest standard deviation 0.142. It is also remarkable that the EDISON system tends to obtain over-segmented results, generating a large amount of groups for each image, whereas the proposed algorithm can correctly estimate the number of objects in the scene in most of the cases. The above can be explained by the correct estimation of the number groups made by the eigenvalue analysis of the weighted linear kernel.

Image segmentation results attained by WGKIS methodology, and EDISON1 shown in Figure 4-8 demonstrate that the proposed methodology produces more accurate segmentation regions than the EDISON system, which clearly generate over-segmented results. The results in Table 4-2 expose that according to the PR measure, WGKIS methodology generate very similar segmentations as those realized by each human subject, identifying the objects present in the scene. In contrast, EDISON system generates a large amount of groups for each image, hence, the PR measure penalizes the results. However, it is important to note that all the approaches based on spectral techniques require a high computational cost due to the similarity matrix estimation.

4.4. Weighted Gaussian Kernel Video Segmentation

Now, we aim to incorporate the proposed methodology to build enhanced feature representations as the characterization stage of a new video segmentation framework based on background subtraction called Weighted Gaussian Kernel Video Segmentation (WGKVS). Developed WGKVS aims to represent each pixel by incorporating multiple similarity measures against a background model for different color representations, moreover, pixel spatial

information is also included. Afterwards, the weighting factors of each Z feature are set according to the feature relevance analysis described in section § 4.2.1. Developed WGKVS purpose is to obtain meaningful segmentations which could be useful in processes like object recognition, without requiring the manual set of a large number of parameters.

4.4.1. WGKVS Background Initialization

Looking for a suitable video segmentation, a background model initialization approach is introduced. To this end, given a set of frames $\{X_t : t = 1, \dots, T\}$ with $T \in \mathbb{N}$, we propose to use a subsequence of frames $T_b \in \mathbb{N}$ ($T_b < T$), to initialize a background model as in [61]. This approach, using an optical flow algorithm is able to construct a statistical background model with the most likely static pixels during the subsequence for each color representation, and it also computes its standard deviation. Hence, a background model $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for each pixel is attained, where $\boldsymbol{\mu} \in \mathbb{R}^Z$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{Z \times Z}$ describe the mean vector and covariance matrix of considered features.

4.4.2. WGKVS Background Subtraction

Given a pixel \mathbf{x}_t from frame \mathbf{X}_t with $t > T_b$ and its corresponding background model obtained as described in section § 4.4.1, we propose to perform a background subtraction procedure by elaborating a feature space based on multiple kernel representations similar as in section § 4.2, but instead of adding all the kernel information sources, we propose to conform an enhanced feature space $\mathbf{F}_t \in \mathbb{R}^{N \times Z}$ for each frame \mathbf{X}_t . Namely, for pixel \mathbf{x}_t attained representation is defined as:

$$\mathbf{f}_t = [\vartheta_{t,1}\kappa_1(x_{t,1}, \mu_1) \dots \vartheta_{t,Z}\kappa_Z(x_{t,Z}, \mu_Z)], \quad (4-5)$$

where $x_{t,Z}$ corresponds to the Z -th representation of pixel \mathbf{x} at time t , and μ_Z is the mean background value for the same pixel in the Z th feature. With such combination, we aim to incorporate the kernel similarity measures between each input pixel and its corresponding background model value for all considered features. Overall, high values in \mathbf{f}_t will denote that pixel \mathbf{x}_t is very likely to belong to the background. The well-known Gaussian kernel is used as basis kernel κ as:

$$\kappa_z(x_{t,z}, \mu_z) = \exp\left(\frac{-(x_{t,z} - \mu_z)^2}{2\sigma_z^2}\right), \quad (4-6)$$

where the kernel bandwidth σ_z corresponds to a percentage of the standard deviation in the feature z computed from section § 4.4.1. Afterwards, the spatial information is included into \mathbf{F} as new characteristics, and using the relevance analysis described in section § 4.2.1, the weighting factors $\boldsymbol{\vartheta}$ are computed. Then, an enhanced feature representation space is computed as $\hat{\mathbf{F}} = \mathbf{F} \text{diag}(\boldsymbol{\vartheta})$.

To perform the segmentation, a K-means clustering algorithm with a fixed number of clusters $k_g = 2$ is employed over $\hat{\mathbf{F}}_t$. Hence, foreground pixels are grouped in a cluster and background pixels in the other one. Initially, the clusters are located at the coordinates given by the cluster initialization algorithm called *maxmin* described in [62], making the segmentation process more stable. As a result, a segmented image represented with matrix \mathbf{S}_t is obtained. Finally, using a post-process stage, groups of pixels detected as moving objects that do not surpass a value u_m of minimum size for an object are deleted from \mathbf{S}_t , obtaining a matrix $\hat{\mathbf{S}}_t \in [0, 1]^{w_r \times w_c}$. In Figure 4-9 is illustrated the general scheme for WGKVS.

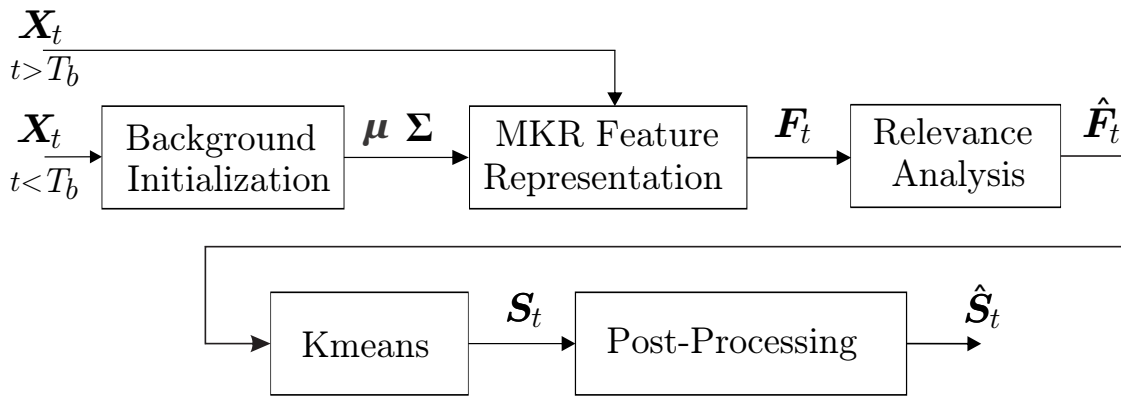


Figure 4-9.: WGKVS scheme.

4.4.3. WGKVS Experiments

Three different experiments are performed in order to test the proposed WGKVS. The first one aims to prove the effectiveness of the proposed WGKVS approach when incorporating multiple information sources into the segmentation process with an automatic weighting selection. The second experiment compares the attained segmentation results of WGKVS against a state of the art algorithm. Lastly, the purpose of the third experiment is to proof if the attained segmented regions by WGKVS could be used to support object classification stages. To this aim, three different databases are employed. Each database includes image sequences that represent typical surveillance scenarios. Following, the databases are described.

DBa - *A-Star-Perception*¹: Contains 9 image sequences recorded in both indoor and outdoor scenarios. Moreover, the image resolution vary from low image resolution sequences (160×120) to medium size resolution images (320×240). Hand-segmented ground-truths are available for random frames of the sequence. Each ground-truth pixel holds

¹<http://perception.i2r.astar.edu.sg>

two possible labels: 0 (black) for background pixels and 255 (white) for foreground pixels.

DBb - *Left-Packages*²: Holds 5 different image sequences recorded at an interior scenario which has several illumination changes. The main purpose of this database is the identification of abandoned objects (a box and a bag). For testing, hand-segmented ground-truths from randomly selected frames are made with two possible labels for each pixel: 0 (black) for background pixels and 255 (white) for foreground pixels. All videos have a medium size resolution (388×244).

DBc - *MSA*³: Contains a single indoor sequence, with stable lighting conditions, nonetheless, strong shadows are casted by the moving objects. The purpose of this sequence is the detection of abandoned objects, in this case a briefcase. In order to attain quantitative measures, hand-segmented ground-truths from randomly selected frames are made with two possible labels: 0 (black) for background pixels and 255 (white) for foreground pixels. The image sequence has a medium size resolution (320×240).

For measuring the accuracy of the attained segmentation by the proposed methodology, three different pixel-based measures have been adopted: Recall r_θ , Precision p_θ and F1. All of them belong to the interval $\mathbb{R} \subset [0, 1]$ and are calculated as:

- $r_\theta = t_p / (t_p + f_n)$
- $p_\theta = t_p / (t_p + f_p)$
- $F1 = 2r_\theta p_\theta / (p_\theta + r_\theta)$,

where t_p (true positives), f_p (false positives) and f_n (false negatives) are obtained while comparing against a hand-segmented ground-truth. A method is considered good if it reaches high *Recall* measures, without sacrificing *Precision*. The *F1* measure is generally used to compare results attained by different methodologies, so, the higher its value the better the attained segmentation.

For concrete testing, the image features Z employed in section § 4.4.2 were the RGB components, and the the normalized RGB components (rgb) [63], which are commonly employed to suppress the shadows formed by the moving objects as described in [64]. It is important to remark that the same 6 features were employed in section § 4.4.1 to initialize the model. As spatial features, we propose to include the row i and column position j of each pixel. Hence, a resulting feature space $\hat{F}_t \in \mathbb{R}^{N \times 8}$ is obtained. Besides, for all the performed experiments, the size of the set of frames used for the background modeling is set as $T : b = 50$, the kernel bandwidth σ^z is heuristically set as 5 times the standard deviation of the corresponding pixel, and the minimum size of a detected moving object is set as a percentage of the total image size $0.005(w_R \times w_C)$.

²<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1>

³<http://cvprlab.uniparthenope.it>

Experiment 1 With the purpose of proving the effectiveness of the proposed WGKVS when incorporating multiple information sources into the segmentation process, we compare the segmentation results of WGKVS against GKVS (WGKVS with equal weighting factors), and traditional GKVS-RGB (GKVS using only RGB components). For concrete testing, the sequences: DBa-WaterSurface, DBa-Fountain, DBa-ShoppingMall, DBa-Hall, DBb-LeftBag, DBb-LeftBox and DBc-MSA are used. The first two sequences are recorded in outdoor scenarios which present high variations due to their nature, hence the segmentation process poses a considerable challenge. The DBa-ShoppingMall and DBa-Hall sequences are recorded in public halls, in which are present many moving objects generating strong shadows and crossing each other, hindering the segmentation task. The last three videos are recorded in indoor scenarios, however there are several illumination changes due to the sun light and the generated shadows.

Figures 4-10 and 4-12 show some sample segmentation results attained by the three different configurations above described for the DBa-Fountain sequence 1509th frame and for the DBa-WaterSurface sequence 1523th frame respectively. The corresponding weighting factors for each frame are exposed in Figures 4-11 and 4-13.

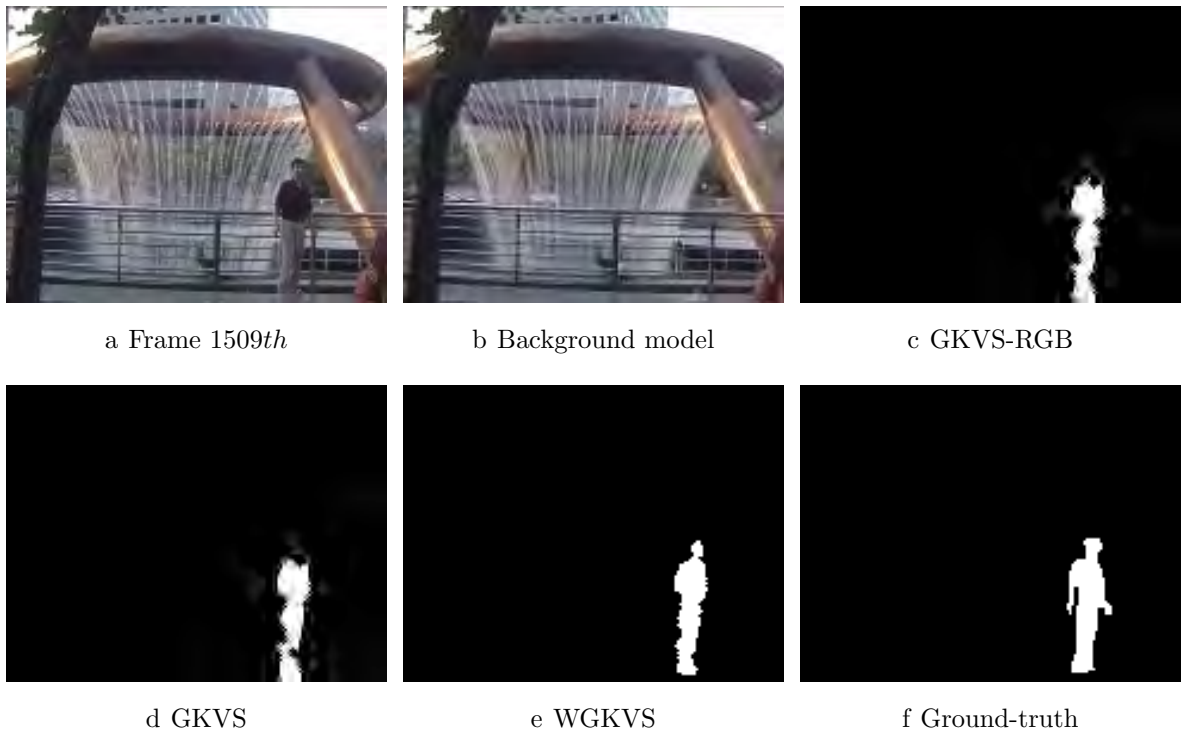


Figure 4-10.: DBa-Fountain segmentation results.

In Table 4-3 are exposed the supervised pixel measures attained by GKVS-RGB, GKVS and WGKVS. Highlighted are the best obtained measures for each video.

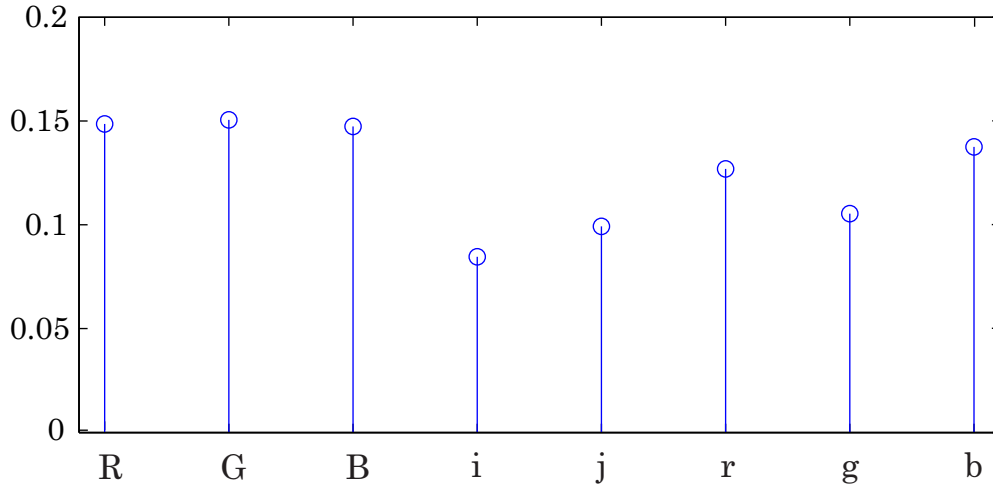


Figure 4-11.: Relevance weights for sequence DBa-Fountain (Frame 1509th).

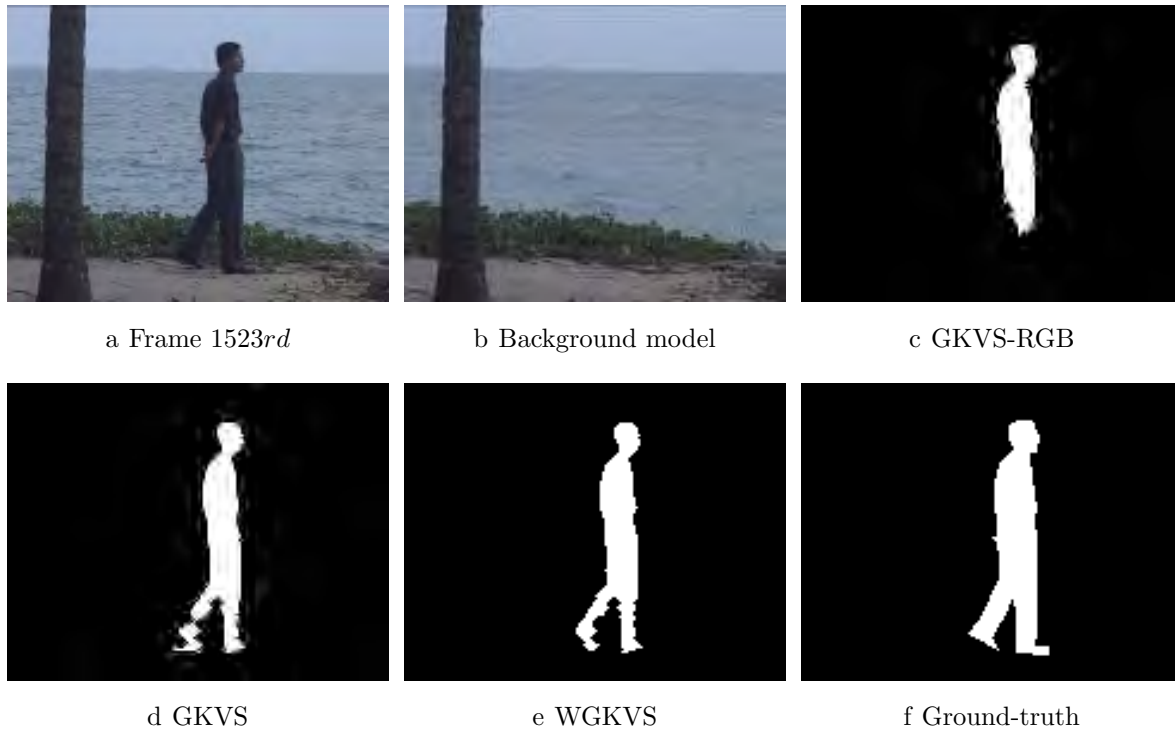


Figure 4-12.: DBa-WaterSurface segmentation results.

Experiment 2 Now, we compare the WGKVS algorithm against a traditional video segmentation algorithm named Self-Organizing Approach to Background Subtraction (SOBS) [46]. This algorithm builds a codebook during a training phase to model the background. The background subtraction task is performed by measuring the squared Mahalanobis distance between each new input pixel and their corresponding codebook values, if given a threshold

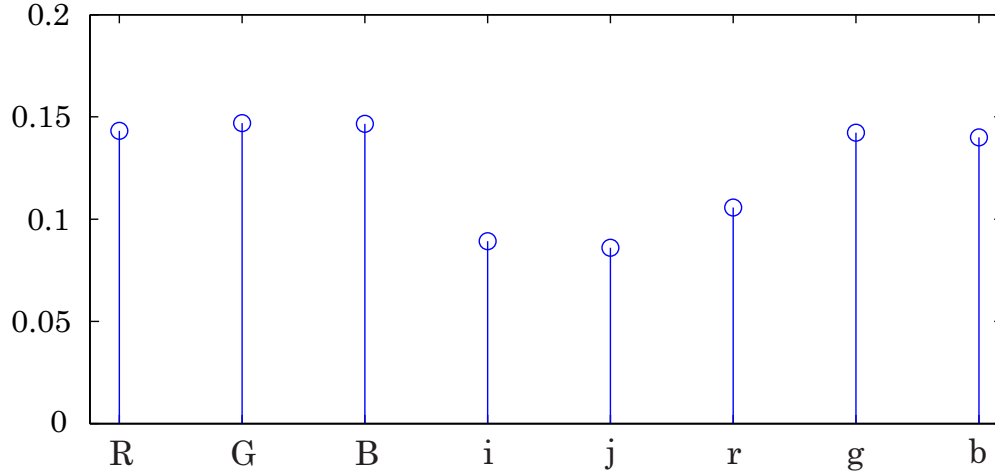


Figure 4-13.: Relevance weights for sequence DBa-WaterSurface (Frame 1523rd).

Table 4-3.: Segmentation performance for the three configurations of WGKVS

Video	GKVS-RGB			GKVS			WGKVS		
	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$
DBa-WaterSurface	0.657	0.995	0.791	0.692	0.995	0.816	0.701	0.994	0.822
DBa-Fountain	0.509	0.897	0.649	0.559	0.909	0.692	0.587	0.908	0.713
DBa-ShoppingMall	0.436	0.385	0.409	0.481	0.653	0.554	0.512	0.645	0.571
DBa-Hall	0.389	0.709	0.504	0.422	0.729	0.533	0.420	0.737	0.535
DBb-LeftBag	0.510	0.639	0.567	0.514	0.702	0.593	0.507	0.728	0.600
DBb-LeftBox	0.697	0.906	0.788	0.691	0.912	0.786	0.729	0.915	0.812
DBc-MSA	0.756	0.715	0.735	0.815	0.711	0.759	0.819	0.702	0.756

none of such values match the input pixel, then, it is labeled as foreground, otherwise, it is labeled as background, and the codebook for such pixel is updated in a self-organizing manner, aiming to learn the scene variations and to consider spatial relationships. The SOBS video segmentation approach has been used as a reference to compare video segmentation approaches, moreover, it has been included in recent video surveillance systems surveys [28, 38], besides, it was included in the Change Detection Challenge 2012 [48]. The SOBS algorithm requires the setting of 10 parameters, for testing, all of them are left as default except the SOBS K parameter, which is the number of frames used for the training phase. Such parameter is set as Tb in order to give the same amount of information to both SOBS and WGKVS for the background modeling.

In Figures 4-14 and 4-15 are the segmentation results using WGKVS and SOBS for the 996th frame of the sequence DBb-LeftBag and 1980th frame of the sequence DBa-ShoppingMall respectively. In order to compare, Table 4-4 expose the segmentation results

for the SOBS algorithm and again the WGKVS results.

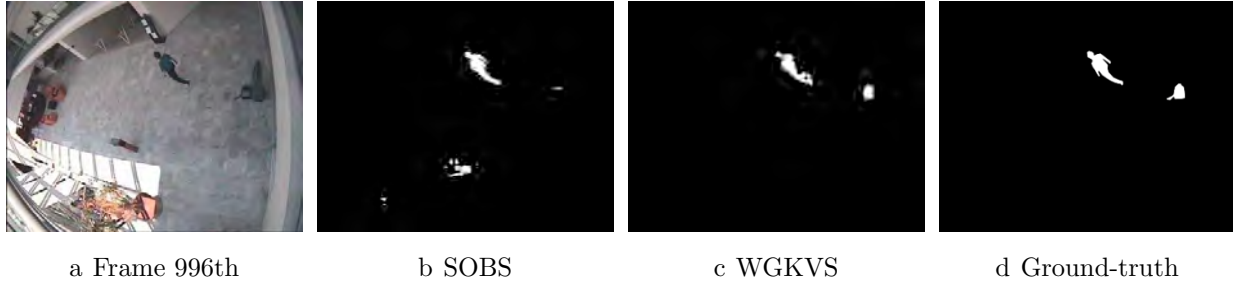


Figure 4-14.: DBb-LeftBag video segmentation results.

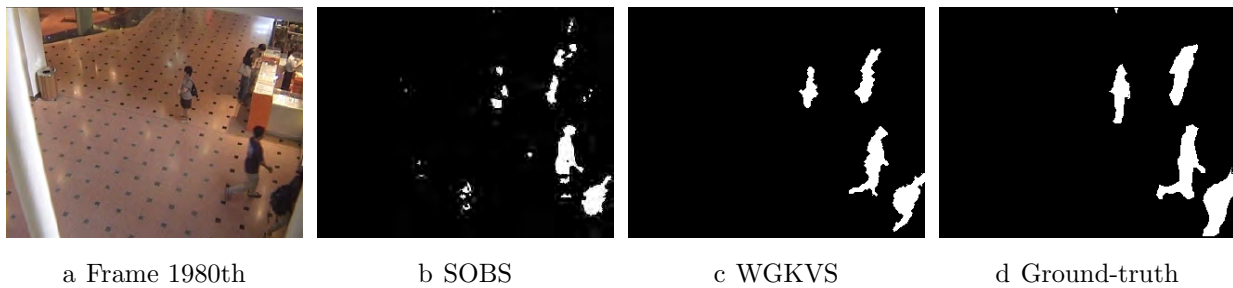


Figure 4-15.: DBa-ShoppingMall video segmentation results.

Table 4-4.: Segmentation performance for SOBS and WGKVS

Video	SOBS			WGKVS		
	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$
DBa-WaterSurface	0.709	0.998	0.829	0.701	0.994	0.822
DBa-Fountain	0.349	0.971	0.513	0.587	0.908	0.713
DBa-ShoppingMall	0.522	0.661	0.583	0.512	0.645	0.571
DBa-Hall	0.488	0.607	0.539	0.420	0.737	0.535
DBb-LeftBag	0.472	0.642	0.544	0.507	0.848	0.600
DBb-LeftBox	0.746	0.806	0.775	0.729	0.915	0.812
DBc-MSA	0.778	0.788	0.783	0.819	0.702	0.756

Experiment 3 Finally, the third type of experiment is made in order to test how the proposed WGKVS supports and object recognition system. Particularly, the recognition of unattended objects such as bags or boxes in public premises (i.e. airports, train stations

etc.) is a task of great importance in the computer vision field, since these abandoned objects could be considered as a mean of terrorist threats [65].

In this sense, using the segmented frame \mathbf{S}_t , the groups detected as moving objects that are spatially split, are relabelled as new independent regions and enclosed in bounding boxes. Afterwards, each bounding box is mapped to the original frame \mathbf{X}_t , attaining a set $\{Y_t^p : p = 1 \dots P_t\}$ of RGB matrices, where P_t is the total number of detected regions. Having the set Y_t , we propose to use the object characterization process described in [66], which represents each object by using the geometrical and statistical features presented in Table 4-5. Since the challenge is to detect unattended objects in places with moving people, a knn classifier is trained using a dataset of 70 images of people and 82 images of baggage objects. As validation, we propose to use the objects segmented by WGKVS. It is important to remark, that the objects from the dataset used for training are characterized by the same process.

Table 4-5.: Considered features for abandoned object classification.

Lines	Corners
<ul style="list-style-type: none"> - Percentage of lines: diagonal, horizontal and vertical. - Vertical lines: percentage of lines placed at the left, right and in the center. - Horizontal lines: percentage of lines placed at the top, bottom and in the middle. 	<ul style="list-style-type: none"> - Total of corners. - Percentage of corners: left, right, top and bottom. - Horizontal and vertical standard deviation.
Ellipses	Other Features
<ul style="list-style-type: none"> - Fitting ellipse aspect ratio. - Fitting ellipse area ratio. 	<ul style="list-style-type: none"> - Hu's moments.

For testing, the sequences: DBb-LeftBag, DBb-LeftBox and DBb-MSA are used. In Figure 4-16, are shown some resulting bounded objects. In total, 38 objects are used as validation, 11 belong to the baggage objects class and 27 to the people class. Figure 4-17 presents two samples of the characterization process for a person and a bag. The classification results are exposed in Table 4-6.

Table 4-6.: Confusion matrix using the Knn classifier.

	People	Baggage Objects
People	21	6
Baggage Objects	1	10

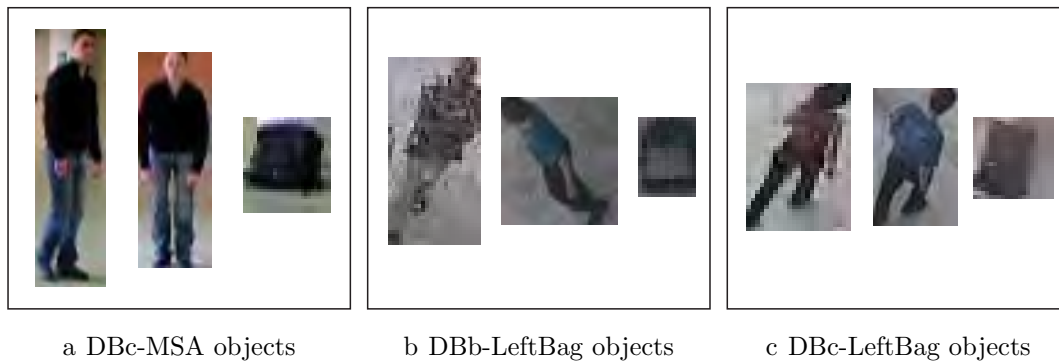


Figure 4-16.: WGKVS segmented objects samples.

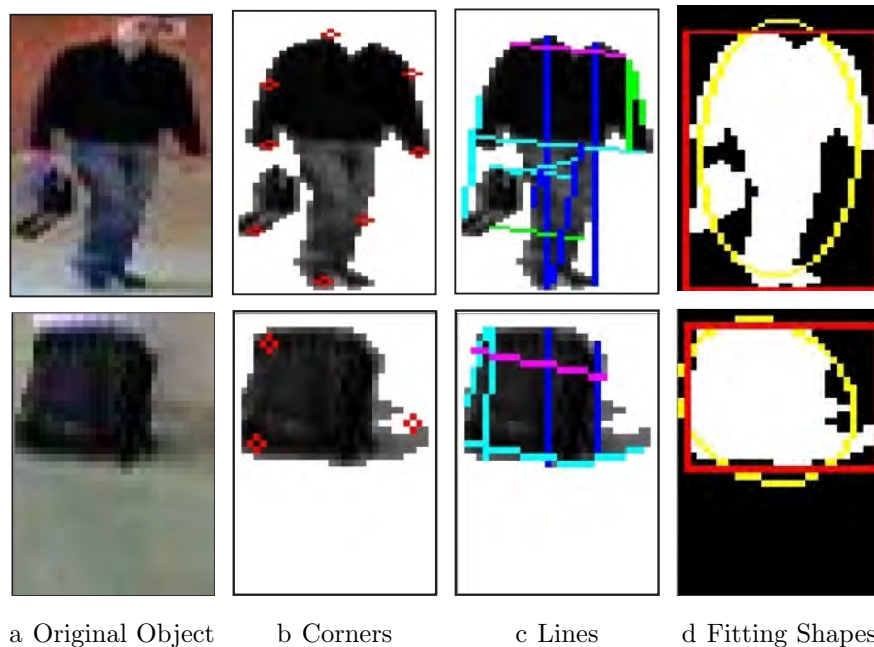


Figure 4-17.: Geometrical features samples for objects of the two classes.

4.4.4. Discussion

From the attained results for the first experiment, it can be seen that when working only with the RGB components, proposed framework does not achieve a suitable performance, lacking of extra information to enhance the clustering process (see Figure 4-12(c) and Table 4-3). When the rgb components and the spatial information are incorporated, the performance improves for most of the videos as presented in the Table 4-3. Besides, Figure 4-12(d) shows a visual example of the improvement in the segmentation. Finally, using the proposed WGKVS methodology the best results are achieved. Although the weighting factors are almost equal for the given frame samples(see Figures 4-11 and 4-13), an improvement in

the segmentation can be visually appreciated in Figures 4-10(e) and 4-12(e). The above can be corroborated by the attained $F1$ results exposed in Table 4-3, where the WGKVS expose the best $F1$ measure for 4 of the 6 videos.

The second experiment exhibits that overall the proposed WGKVS methodology results are competitive against the ones attained by the SOBS algorithm(see Table4-6). From Figures 4-14 and 4-15, it can be noticed that WGKVS framework achieves more reliable segmented regions which could be used in further stages like the classification of objects. It is worth saying that the low SOBS results could be explained by the SOBS K parameter which by default is set as 200, for the performed experiment such parameter was reduced to 40 frames.

The segmented objects by the WGKVS for the third experiment (see Figure 4-16), are accurate for the employed characterization process as it can be appreciated in Figure 4-17. The aforementioned can be corroborated with a classification performance of 84.21%, which is significantly high given that a simple knn classifier was used, and moreover, that the resolution of the segmented regions is quite low. The misclassified samples belonging to the people class, are objects where the complete body of the person is not in the scene, therefore, they not match any of the people class objects used in the training phase.

4.5. Conclusions

A new methodology for building enhanced feature representations to support machine learning processes was proposed. Using a multiple kernel learning approach, proposed methodology aims to incorporate multiple information sources into the process by means of kernel similarity measures. Moreover, the importance of each information source is automatically weighted by a feature relevance analysis. Proposed methodology is tested in image processing by including multiple image features to describe each pixel. Particularly, developed methodology was implemented to support two computer vision tasks: image segmentation and video segmentation. Regarding this, two different frameworks were proposed. The first one called WGKIS, aims to perform an image segmentation by using an spectral clustering based approach over a multiple Gaussian kernel matrix obtained from the proposed methodology. The purpose of the second developed framework named WGKVS, is to perform a video segmentation given a sequence of frames recorded by an static camera. WGKVS builds an enhanced feature space following based on multiple kernel representations, afterwards, the segmentation is performed by using a tuned kmeans algorithm, attaining in one cluster the pixels belonging to the foreground and the pixels belonging to the background in the other. Experiments for both segmentation frameworks exhibited that when including multiple color spaces information to describe each pixel by means of the proposed methodology, the segmentation results improved, furthermore, the attained segmented regions were more meaningful.

Regarding WGKIS, attained results also showed that the estimation of the number of

groups made by means of the eigenvalue analysis of the weighted linear kernel was accurate, supporting the performance of the spectral clustering algorithm. As future work, we will study alternatives to automatically estimate WGKIS free parameters. Furthermore, due to the complexity of the proposed WGKIS, a GPU computation scheme could be proposed in order to achieve a real-time application over full size images.

Experiments for WGKVS also showed that the proposed framework has stable results using the same parameters for all the experiments, and that is suitable for supporting real surveillance applications like the classification of abandoned objects. Moreover, attained results exposed that WGKVS performs equally as state of the art approaches such as the SOBS algorithm. However, there is still the challenge of including temporal and spatial dynamics into the background modelling process in order to be able to deal with more complex video surveillance scenarios.

5. A New Approach to Incorporate Data Spatial Relations to Support Video Segmentation Tasks

Video sequences recorded in real world scenarios present some conditions, in which elements of the foreground and background reach similar dynamics, (e.g., moving objects that become static for a time period, or static objects that start moving due to the own nature of the scene) therefore the background subtraction task becomes complex. Most of video segmentation approaches which focus their attention only in background modeling [35, 46], could tend to fail in the detection of such scene conditions, leading to low discrimination performances. Hence, learning only background behaviors is not enough to infer a suitable background model. Furthermore, aside from the intrinsic usefulness of being able to segment video streams into foreground and background components, detecting moving objects provides a focus of attention for recognition, classification, and activity analysis, making these later steps more efficient [67]. Therefore, it is desirable to identify, model, and track the dynamics of foreground elements. Afterwards, if a given pixel is identified as a foreground component (moving or static), then this information should be included into the background model. Moreover, due to the intrinsic dynamics of video recordings, the estimated background model can't be assumed as stationary, it must be considered as an online learning task which fluctuates along time, in this regard, the model parameters should be updated for each new input sample.

In this sense, we present in this chapter a new kernel-based video segmentation framework, which using an optical flow based methodology aims to identify spatial information related to the objects in the scene. Namely, proposed framework uses motion information to track moving objects, which can stop or change smoothly their movement along the video. Objects detected as static are stored into a memory, with the purpose of weighting such information into the background modeling process. Moreover, based on the assumption that background elements behavior tends to be static, we propose a background updating methodology which analyses the pixel variability over a time window, and afterwards decide if including the new information into the model. Experiments are performed to test the proposed framework over real world scenarios, particularly videos in which foreground and background dynamics become similar. Besides, attained results are compared against state of the art approaches by using supervised measures.

The remainder of this chapter is organized as follows. In section § 5.1 the proposed methodology to identify spatial information is described. The variability-based background updating approach is presented in section § 5.2. Experiments and results are shown in section § 5.4. Finally, in sections § 5.5 and 5.6 we discuss and conclude about the attained results.

5.1. Highlighting Spatial Relationships based on Optical Flow

5.1.1. Region Change Detection

Given a frame matrix set $\{\mathbf{X}_t\}$, we compute the mean intensity matrix $\mathbf{M}_t \in \mathbb{R}^{w_R \times w_C}$ holding elements $m_t^{i,j} = \mathbb{E} \{x_{t,z}^{i,j} : \forall z \in Z\}$, where $x_{t,z}^{i,j}$ is the color intensity of pixel (i, j) belonging to frame \mathbf{X}_t at color channel z . To detect informative changes between two consecutive frames, $t-1$ and t , matrix \mathbf{M}_t is split into several patches, each one of size $w_\Omega \times w_\Omega$, with $w_\Omega \ll w_C$. Then, to accentuate moving pixels from frame $t-1$ to t , the matrix \mathbf{H}_t of size $w_R \times w_C$ is introduced, containing elements in the form of the following *Sum of Absolute Differences* (SAD)[68, 69]:

$$h_t^{i,j} = \begin{cases} 1, & \|\Omega_t^{i,j} - \Omega_{t-1}^{i,j}\|_F^2 < \xi_H \\ 0, & \text{Otherwise} \end{cases} \quad (5-1)$$

where each couple of subsequent patches $\Omega_t^{i,j}, \Omega_{t-1}^{i,j} \subset \mathbb{R}^{w_\Omega \times w_\Omega}$ hold spatial pixels neighboring element (i, j) belonging to matrices \mathbf{M}_t and \mathbf{M}_{t-1} , respectively, $\xi_H \subset \mathbb{R}^+$ is an a priori threshold parameter, and notation $\|\cdot\|_F$ stands for the Frobenius norm operator. Thus, matrix \mathbf{H}_t in Eq. (5-1) highlights patches of pixels that considerably changed from frame $t-1$ to t .

5.1.2. Motion Modeling by Optical Flow

Now the task is to estimate the movement direction of each patch from \mathbf{H}_t . To this end, we compute a vector field describing pixel velocities of image sequences. Though this vector field can be calculated based on optical flow, the majority of widely used procedures based on optical flow require high computational burden [70, 71]. To cope with this, we propose a selective optical flow approach that consists on searching each patch highlighted as moving at time instant $t-1$ into neighboring patches of the new input frame at time instant t .

To be precise, the patch $\Omega_t^{i^*,j^*} \in \mathbf{M}_t$ that best matches the patch highlighted as moving $\Omega_{t-1}^{i,j} \in \mathbf{M}_{t-1}$ according to the SAD measure, is used to estimate the optical flow field. So, to find the best matching patch, the searching process is carried out into a neighborhood

of size $k \subset \mathbb{N}$ around pixel (i, j) (center pixel of the patch highlighted as moving). Then, to encode the apparent movement measured between frames $t - 1$ and t , we calculate the relative coordinates $(\Delta_i = i - i^*, \Delta_j = j - j^*)$ of the best patch $\Omega_t^{i^*, j^*}$, which are afterwards stored in matrices $\mathbf{A}_t \subset \mathbb{Z}^{w_R \times w_C}$ and $\mathbf{B}_t \subset \mathbb{Z}^{w_R \times w_C}$, respectively. Lastly, to get a more compact description, object movement is directly computed from the optical flow angle matrix $\mathbf{Q}_t \subset \mathbb{R}^{w_R \times w_C}$ holding elements $q_t^{i,j} = \tan^{-1}(b_t^{i,j}/a_t^{i,j})$, with $q_t^{i,j} \in \mathbb{R}[0, 360]$.

5.1.3. Object Movement Identification and Static Object Memory Computation

At this stage, we aim to identify from the computed matrix \mathbf{H}_t the patches highlighted as moving which describe foreground regions. Afterwards, for each region, motion information based on optical flow matrix \mathbf{Q}_t is calculated to perform a tracking of the moving objects in the scene. Finally, the information obtained by the tracking is used to update the background model.

To this end, the first step is to identify foreground regions as groups of spatially connected patches highlighted as moving. Thus, a connectivity-based operation over \mathbf{H}_t is carried out, labeling spatially connected patches. Namely, we label the following regions of interest: label '1' makes up a first approximation about the background, label '2' makes up one connected region, and so on until label ' D_t ' that makes up the D_t -th connected region, at time instant t . Then, each d -th region ($d = 2, \dots, D_t$) region is mapped into matrix \mathbf{Q}_t to further compute a local optical flow angle histogram. From such histogram, the most frequent value is calculated and is stored in $\beta n_t^d \in \mathbb{R}[0, 360]$. Note that βn_t^d is an estimation of the motion direction trend of each region d . Additionally, the motion trend magnitude $\beta m_t^d \in \mathbb{R}$ is calculated as the average magnitude over all optical flow vectors in region d that satisfy βn_t^d trend.

To perform the tracking, each object d is enclosed by a bounding box, which is displaced at frame $t + 1$ according to βn_t^d and βm_t^d . In that way, we are predicting the new position of each object by taking into account its previous movement. After that, the displaced bounding box of each object d is mapped to \mathbf{H}_t aiming to check the number of moving pixels $\lambda n_d^{t+1} \in \mathbb{N}$ in region d at $t + 1$ instant. Object d is considered as moving from t to $t + 1$ if the following condition is fulfilled:

$$\frac{\lambda n_d^{t+1}}{\lambda n_d^t} > \xi_\lambda, \quad (5-2)$$

with $\xi_\lambda \in \mathbb{R}^+$. Otherwise, if Eq. (5-2) is not accomplished, object d is considered as static and the coordinates of all the pixels $C_t^d \in \mathbb{N}$ inside of the respective bounding box are stored into an static memory set $\Phi_t \in \mathbb{N}^{C_t^d \times 2}$, additionally, the number of moving pixels λn_d^t is also stored. Finally, with C_t an static memory matrix $\mathbf{L}_t \subset \{0, 1\}^{w_R \times w_C}$ is computed as:

$$l_t^{i,j} = \begin{cases} 1 & \text{if pixel } (i, j) \in \Phi_t \\ 0 & \text{Otherwise} \end{cases} . \quad (5-3)$$

Matrix \mathbf{L}_t is going to be further used to incorporate the spatial information of static objects in the scene. Note that, the static object d stored can only be considered as moving, if at a posterior time instant the condition of Eq. (5-2) corresponding is accomplished, then, all the information related to object d is removed from Φ_t .

5.2. Background Modeling using Variability Constraints

The main goal of our approach is to deal with real-world environment conditions, looking for a background model that allows to deal with non-stationary changes, which are inherent to video surveillance scenes. Hence, we propose to employ a Gaussian-based background model $\mathcal{N}(\mu_{t,z}, \sigma_{t,z})$ with mean $\mu_t^z \in \mathbb{R}^+$ and variance $\sigma_{t,z} \in \mathbb{R}^+$ to estimate the probability function of each pixel in each color channel z .

The model parameters $\mu_{t,z}^{i,j}$ and $\sigma_{t,z}^{i,j}$ for pixel i, j in each color channel z are updated based on the assumption that background elements tend to be more static than foreground elements. To this end, the mean $\gamma_{\mu,t,z}^{i,j} \in \mathbb{R}^+$ and standard deviation $\gamma_{\sigma,t,z}^{i,j} \in \mathbb{R}^+$ of pixel i, j are estimated from the last T pixel values as:

$$\gamma_{\mu,t,z}^{i,j} = \mathbb{E} \{ x_{t,z}^{i,j} : t \in T \} , \quad (5-4)$$

$$\gamma_{\sigma,t,z}^{i,j} = \sigma \{ x_{t,z}^{i,j} : t \in T \} , \quad (5-5)$$

where $\sigma\{\cdot\}$ is the standard deviation operator. It is important to note that $\gamma_{\mu,t,z}^{i,j}$ and $\gamma_{\sigma,t,z}^{i,j}$ are computed only by using the last T frames, then it is only needed to store these frames for such computation. Finally, using the estimated mean and standard deviation, and the spatial information encoded in matrix \mathbf{L}_t , the Gaussian-based background model parameters $\mu_t^{i,j,z}$ and $\sigma_t^{i,j,z}$ are updated as in (5-6) and (5-7)

$$\mu_t^{i,j,z} = \begin{cases} \gamma_{\mu,t,z}^{i,j} & (\gamma_{\sigma,t,z}^{i,j} < \sigma_{t-1,z}^{i,j}) \cap (l_t^{i,j} = 0) \\ \mu_{t-1}^{i,j,z} & \text{Otherwise} \end{cases} . \quad (5-6)$$

$$\sigma_{t,z}^{i,j} = \begin{cases} \gamma_{\sigma,t,z}^{i,j} & (\gamma_{\sigma,t,z}^{i,j} < \sigma_{t-1,z}^{i,j}) \cap (l_t^{i,j} = 0) \\ \sigma_{t-1,z}^{i,j} & \text{Otherwise} \end{cases} . \quad (5-7)$$

5.3. Kernel-based Background Subtraction

Having the background model estimated in section § 5.2, a clustering-based segmentation algorithm as described in section § 4.4.2 is used to extract the moving objects from the scene. The RGB components and the row and column position are used as information sources to construct the enhanced feature representation space. As a result, a segmented binary matrix $\mathbf{S}_t \subset \{0, 1\}^{w_R \times w_C}$ is attained.

5.3.1. Dealing with Illumination Changes

Nonetheless, some scene artifacts (e.g. light variations, shadows) perturb the quality of the segmented image. Regarding this, we propose to debug the segmented matrix \mathbf{S}_t by employing the shadow suppression approach explained in [64]. To this end, the first step is to represent each input pixel $\mathbf{x}_t^{i,j}$ and its corresponding background model mean value $\mu_t^{i,j}$ with normalized rgb components [63]. The use of such color representation is more insensitive to shadows perturbation, however, they have the disadvantage of losing lightness information [72]. To cope with this, each pixel and the corresponding background model are also represented with the sum of their respective RGB components. Afterwards, pixel i, j is compared against the corresponding background model value, by using the normalized RGB and the sum of RGB representations, such differences are compared against ξ_n and ξ_s thresholds respectively. Then, if both differences surpass the thresholds, the label $s_t^{i,j}$ of pixel i, j is still 1 (foreground), otherwise, it means that such pixel corresponds to a shadow, so $s_t^{i,j} = 0$ is relabeled as background.

Figure 5-1 expose the general scheme of proposed methodology for background subtraction. The feedback loop in Static Object Memory and Background Update stages represents that the task is an online learning task, which depends on the information previously learned.

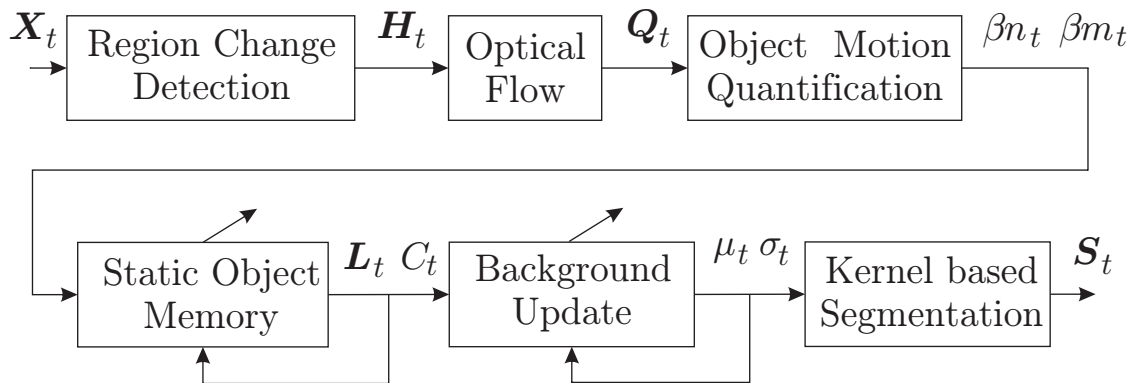


Figure 5-1.: Proposed methodology scheme.

5.4. Experiments

The proposed methodology for video segmentation is tested using the databases exposed in section § 4.4.3. Three different experiments are performed to evaluate the proposed methodology, the first one aims to visually analyze some particular results of the section § 5.1 to highlight spatial information based on optical flow. The second experiment is performed to analyze the behavior of the illumination and shadow debugging stage exposed in section § 5.3.1 while varying its parameters. Finally the third experiment aims to compare the obtained results by the proposed methodology against the ones attained by the SOBS algorithm explained in section § 4.4.3.

Considering the image resolution, proposed methodology free parameters are experimentally set as referred in Table 5-1 for all the experiments. Note that parameters k , ξ_λ and T could be set according to the FPS of the input video, since they are strictly related to the changes between frame to frame.

Table 5-1.: Proposed methodology parameters.

<i>Parameter</i>	<i>Value</i>
w_Ω	$0.02\sqrt{w_R \times w_C}$
ζ_H	$20(w_\Omega \times w_\Omega)$
k	$3w_\Omega$
ξ_λ	0.5
T	15
ξ_n *	0.03
ξ_s *	75

* The values for these two parameters is varied in Experiment 2.

Experiment 1 In order to make clear the contribution of the section § 5.1 to highlight spatial information based on optical flow, we aim to visually analyze some particular results using the DBb-LeftBag video sequence (see Figure 5-2). Particularly, Figure 5.2b shows the resulting \mathbf{H}_t matrix explained in section § 5.1.1. The obtained optical flow matrix \mathbf{Q}_t explained in section § 5.1.2 is shown in Figure 5.2c. Moreover, Figure 5.2d depicts the computed motion direction optical flow trend βn_t^d . Note that some morphological filters are applied to attain a smoother representation of the object.

Figure 6-11 shows an example of the proposed stage to detect static objects. Figures 5.3b and 5.3c shows the attained optical flow matrix \mathbf{Q}_t and static matrix \mathbf{L}_t respectively. The mean value of the generated background model is exposed in Figure 5.3d.

Experiment 2 Aiming to study the performance of the proposed illumination and shadow debugging stage explained in section § 5.3.1, three different configurations for parameters ξ_n and ξ_s are employed as depicted in Table 5-2. The first configuration, called 'No debug', corresponds as not using the proposed stage for illumination and shadow debugging. The

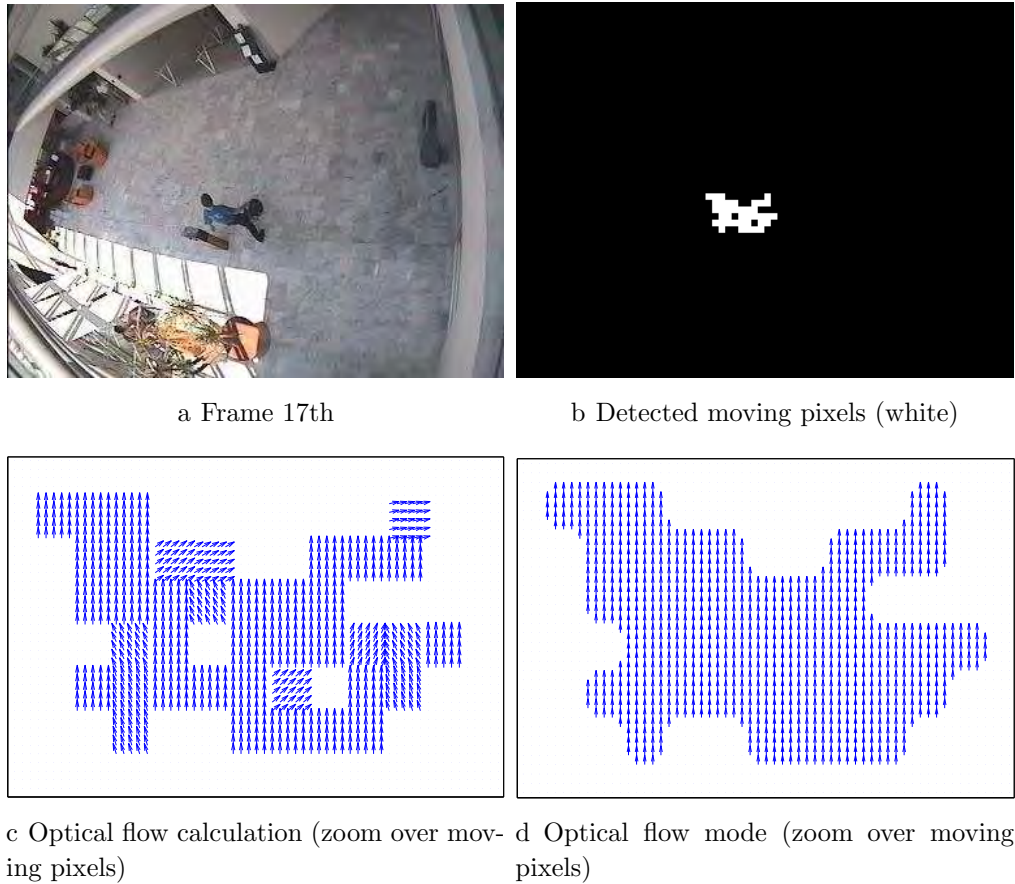


Figure 5-2.: Motion detection results for DBb-LeftBag video

second and the third configurations, 'Debug 1' and 'Debug 2' respectively, are used to see how the segmentation results are affected by the proposed stage. Intuitively, 'Debug 1' will be more strict than 'Debug 2', relabeling most of the segmented areas which correspond to shadows.

Table 5-2.: Parameter configurations for illumination and shadow debugging stage.

Configuration	ξ_n	ξ_s
No debug	0.00	255
Debug 1	0.05	75
Debug 2	0.03	75

The segmentation results for each category are assessed by using the supervised pixel-based measures exposed in section § 4.4.3. As database, the following image sequences are considered since they present strong shadows and illumination changes: DBa-ShoppingMall,

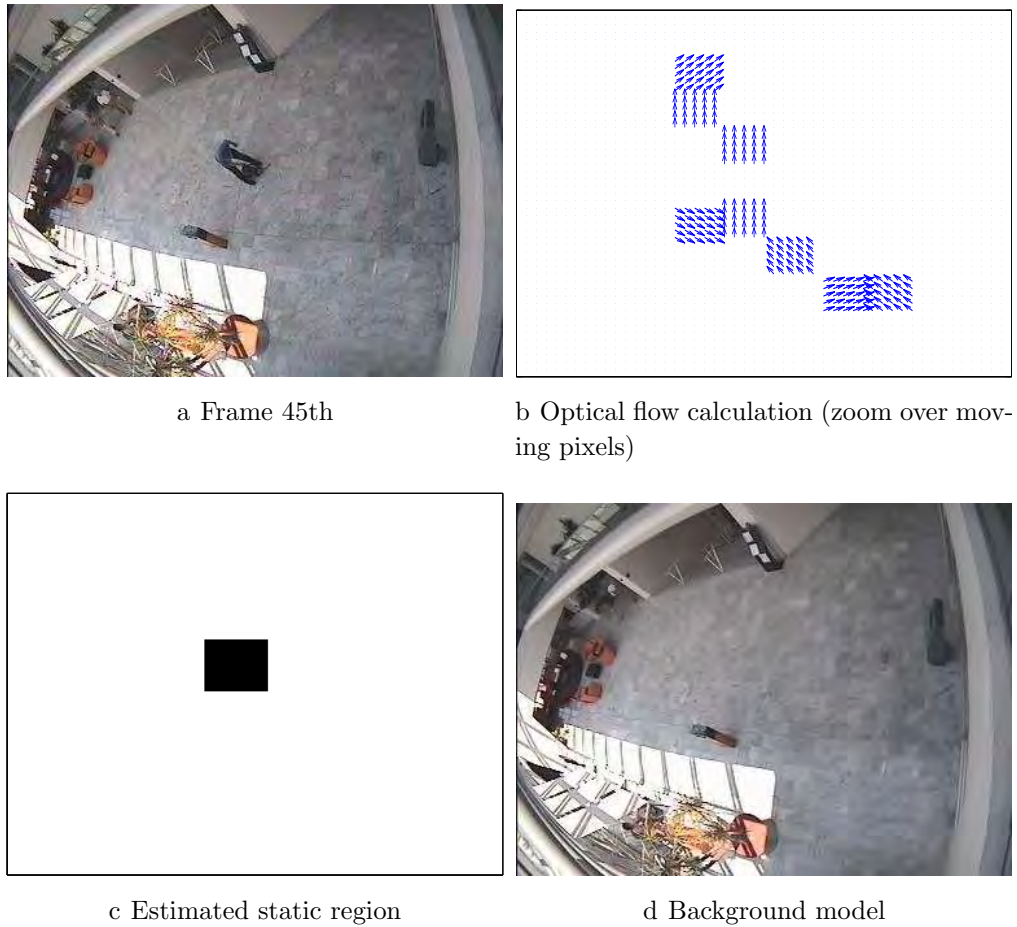


Figure 5-3.: Detection of static object after movement (DBb-LeftBag video)

DBb-LeftBag, DBb-LeftBox and DBc-MSA.

Table 5-3 exposes the attained segmentation results for each illumination and shadow debugging configuration. Moreover, in Figure 5-4, the segmentation results for some particular frames are exposed, with their respective supervised measures.

Table 5-3.: Segmentation performance using the three parameter configurations for the illumination and shadow debug stage.

Video	No debug			Debug 1			Debug 2		
	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$
DBa-ShoppingMall	0.823	0.281	0.424	0.427	0.745	0.543	0.602	0.645	0.622
DBb-LeftBag	0.962	0.462	0.624	0.417	0.994	0.587	0.531	0.764	0.629
DBb-LeftBox	0.962	0.456	0.614	0.743	0.992	0.849	0.822	0.889	0.854
DBc-MSA	0.988	0.664	0.794	0.709	0.993	0.827	0.762	0.989	0.860

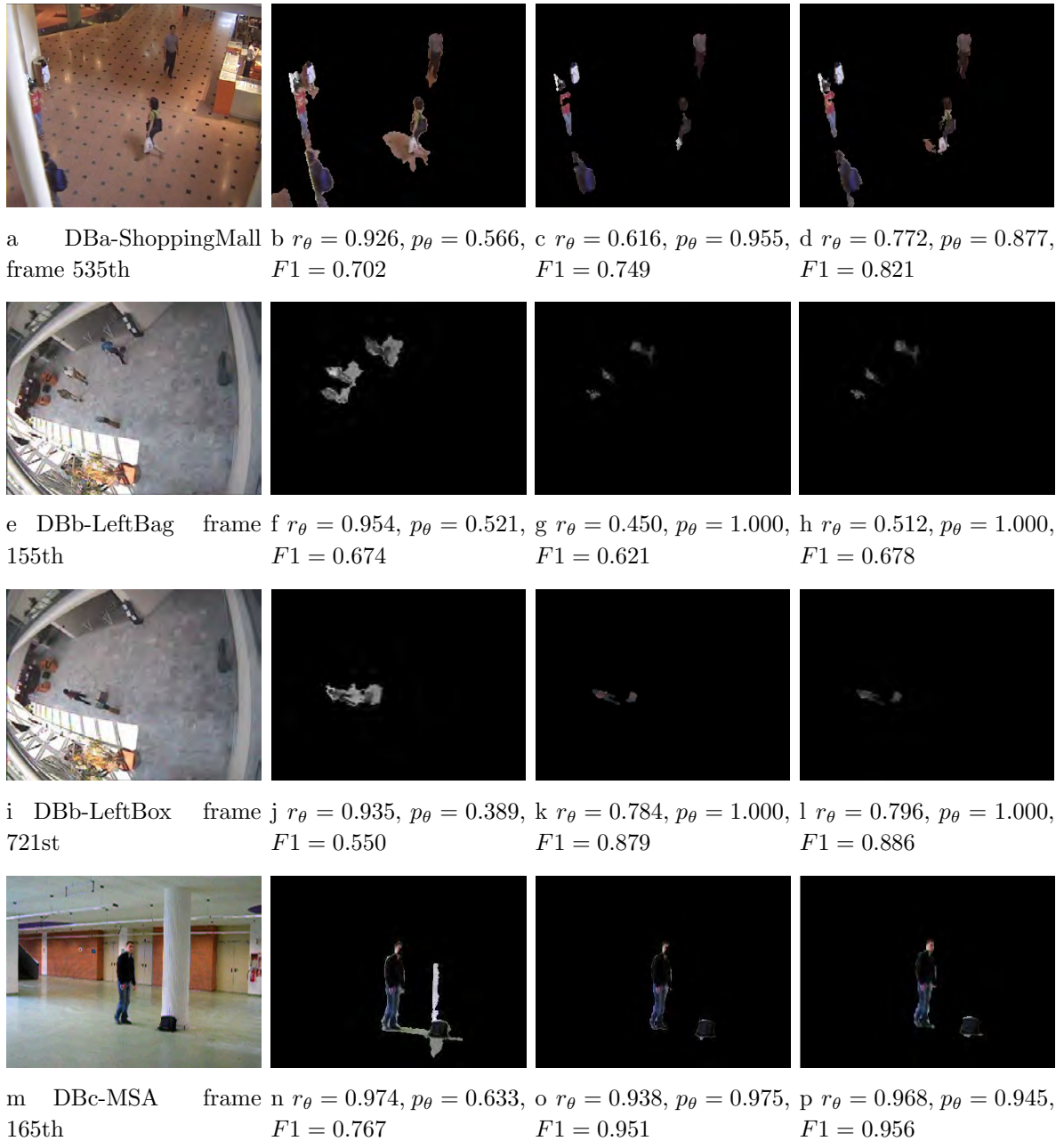


Figure 5-4.: Segmentation results using the parameter configurations exposed in Table 5-2 for the illumination and shadow debugging stage. Column 1: Original frame, Column 2: 'No debug', Column 3: 'Debug 1', Column 4: 'Debug 2'

Experiment 3 In this experiment, the segmentation results obtained by the proposed methodology are compared against the ones attained by the SOBS algorithm using the supervised pixel-based measures described in section § 4.4.3. To this aim, the following image sequences are employed: DBa-WaterSurface, DBa-ShoppingMall, DBb-LeftBag, DBb-

LeftBox and DBc-MSA. The parameters of the SOBS algorithm are left as default except the number of frames needed for the training phase which is adjusted as $2T$.

In Table 5-4 are presented the computed measures for the proposed approach and the SOBS algorithm. Additionally, in Figure 5-5 and 5-6 some relevant segmentation results related to the proposed algorithm and the SOBS method are presented.

Table 5-4.: Segmentation performance for the proposed approach and SOBS.

Video	SOBS			Proposed approach		
	r_θ	p_θ	$F1$	r_θ	p_θ	$F1$
DBa-WaterSurface	0.709	0.998	0.829	0.703	0.968	0.817
DBa-ShoppingMall	0.539	0.687	0.604	0.602	0.645	0.622
DBb-LeftBag	0.472	0.642	0.544	0.531	0.764	0.629
DBb-LeftBox	0.746	0.806	0.774	0.822	0.889	0.854
DBc-MSA	0.778	0.788	0.783	0.762	0.989	0.860

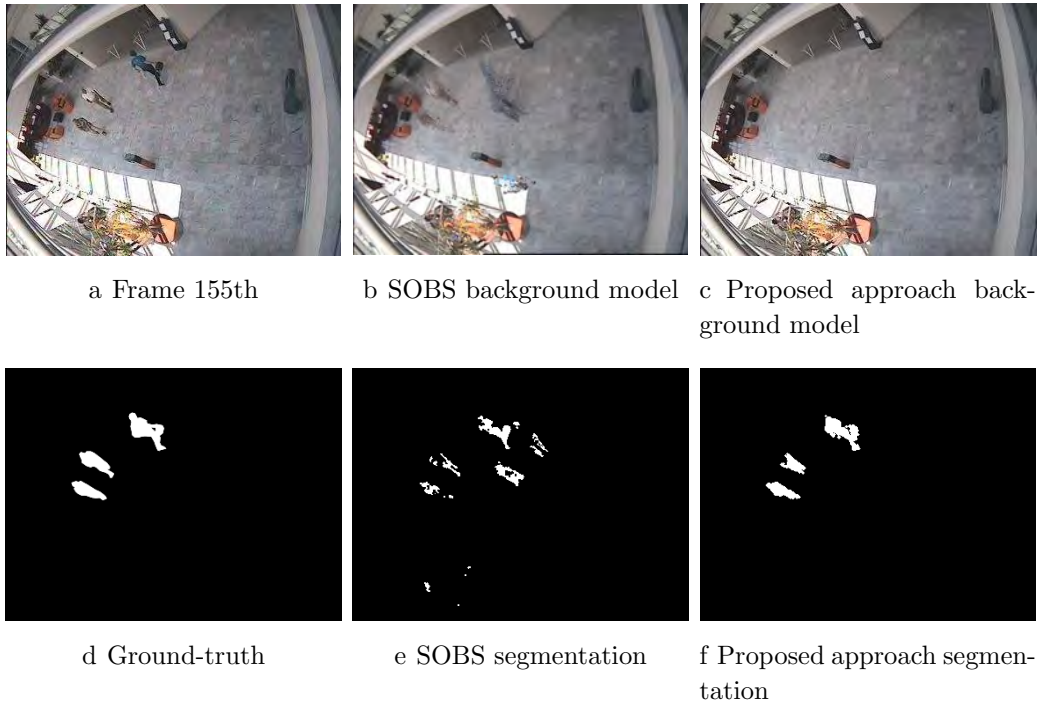


Figure 5-5.: DBb-LeftBag video segmentation results.

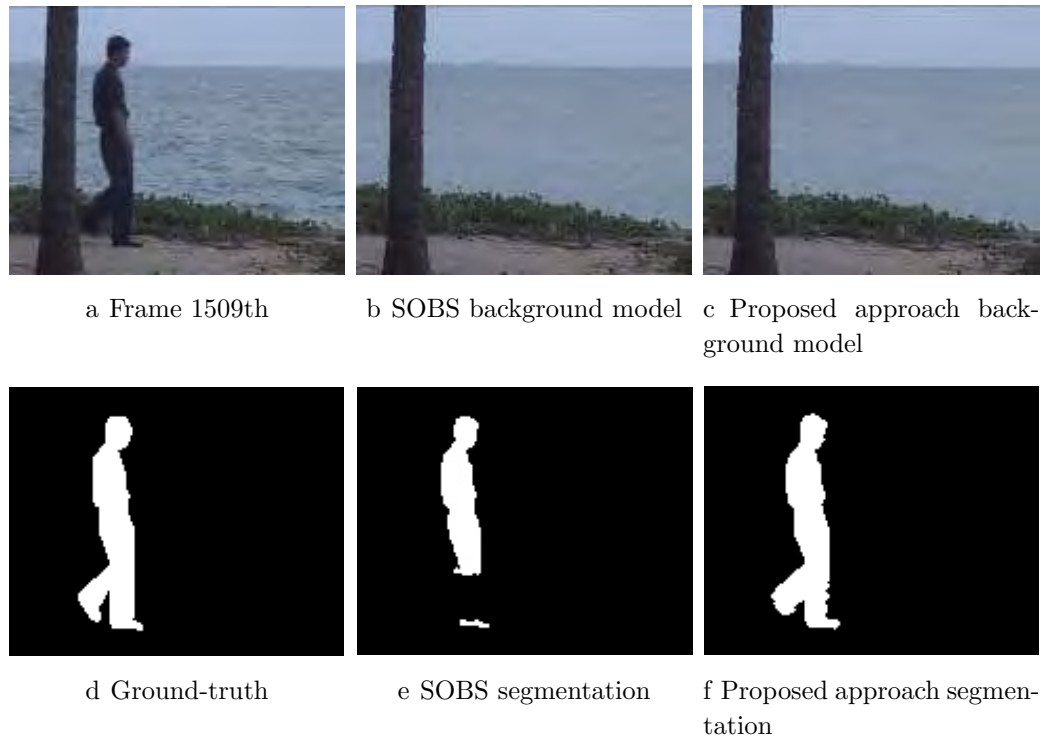


Figure 5-6.: DBa-WaterSurface video segmentation results.

5.5. Discussion

Results from the first experiment show that our approach is able to identify the movement direction of foreground objects in the scene, even if they become static for long periods of time. Particularly, it can be noticed from Figure 5.2a that our framework is able to identify the pixels that changed from the 16th to the 17th frame. Moreover, the computed optical flow over the patches highlighted as moving (see Figure 5.2b) has a visible direction trend of 90° , which is successfully identified by proposed approach, as it can be appreciated in Figure 5.2d. Now, from Figure 5-3 it can be seen that proposed methodology is able to detect the static object in the scene, in this case, a person who was previously walking and stops at the 45th frame. Computed optical flow for this frame doesn't show a clear motion trend (see Figure 5.3b), moreover, its corresponding λn is significantly lower than the previous one, therefore, such object is identified as static as shown in Figure 5.3c. Regarding this, the background model at the 45th frame is not altered by the presence of such object and is suitably inferred, as can be visually corroborated in Figure 5.3d.

From the second experiment it can be noticed that the employed illumination and shadow debugging stage has a big influence over the segmentation results. Table 5-3 exhibits that the parameter configuration 'No debug' attains for all the cases the highest *Recall* measure, however the *Precision* tends to be very low, this is explained since the attained segmented

regions for this configuration are big enough to cover most of the foreground objects in the scene, nevertheless, they also include big shadow regions (see Figure 5-4), lowering the F1 measure. On the other hand, 'Debug 1' reaches high *Precision* but low *Recall* values, since the attained segmented regions do not include any shadow region, nonetheless, when using a high value for parameter ξ_n , the illumination and shadow debugging stage also relabel the areas which have similar chromaticity to the background (see Figures 5.4c and 5.4g). Lastly, more balanced results are obtained for 'Debug 2' configuration, which is reflected in high F1 measure values as seen in Table 5-3. The aforementioned can be also visually corroborated in Figure 5-4, where the obtained segmented regions correspond mostly to the foreground objects in the respective scenes.

Attained segmentation results for the third experiment show that proposed methodology obtains more accurate segmentation results as the ones obtained by SOBS. From Figure 5-5 it can be observed that SOBS has bootstrapping problems when the amount of frames used for the training phase is small, therefore, the obtained background model is highly altered by moving and removed objects (see Figure 5.5b). In contrast, as shown in Figure 5.5c, proposed approach has the capability to infer a background model with a small amount of frames by means of a correct identification of moving/static foreground objects. Furthermore, Figure 5-6 shows that even though both methodologies estimated almost the same background model, the kernel-based background subtraction approach used by the proposed methodology, allows to attain a more suitable segmentation (see Figure 5.6f). The quantitative results exposed in Table 5-4 corroborate that proposed methodology is more accurate to deal with the conditions of the studied videos, while employing few frames for initialization.

5.6. Conclusions

A video segmentation framework by dynamic background modeling was presented. We developed an optical flow based methodology that takes into account object trajectories into the background model. Thus, it is possible to identify when a moving object stops or changes smoothly its motion along the video. Moreover, giving a memory property to our model, it is not altered by static objects even if they stay motionless for long time periods. Besides, proposed background updating approach allows to learn a suitable model under non-stationary conditions. An illumination and shadow debugging stage is employed, which allows to avoid confusions between illumination changes and objects in movement, moreover, different parameter configurations were employed, showing that they have a big influence over the attained segmentation, specially when the scene presents camouflage issues. Besides, obtained results for the third experiment showed that proposed framework outperforms the SOBS algorithm for most of the studied videos. Moreover it has the capability to infer an initial background model employing few frames. Nonetheless, if the variability assumption for the background model updating is not fulfilled (e.g. scenes with highly dynamical background elements), then, other updating rules have to be included to deal with such scenarios.

Moreover, a more elaborated tracking algorithm employing image features should be studied in order to deal with scenarios with complex foreground objects interactions. Regarding to computational burden, other segmentation approaches have to be studied, since the elaboration of the enhanced feature representation space and the employed cluster initialization approach, do not allow a real-time implementation.

6. Kernel based Spatio-Temporal Adaptive Learning: a New Video Segmentation Approach

Most of video surveillance systems are employed in non-stationary environments, where intrinsic foreground and background properties change over space and time, e.g., there are illumination changes, motion objects, occlusions, shadows, among others [31, 38]. In this sense, video segmentation must be considered as an online learning task, for which frame pixel samples are available one by one with each new input frame. Given each new input sample, the system must be able to weight its relevance, and using a set of predefined rules based on similarities with the background, it must decide how to include the information of the given sample into the generated background model.

Additionally, as remarked in chapter 5, the correct identification of foreground elements plays an important role of the background modeling task, specially when foreground dynamics exhibit similar behavior as the one of the background dynamics (mostly static).

This chapter develops a novel adaptive background subtraction approach, termed *Spatio-Temporal Adaptive Learning* or STAL, to support video-based surveillance systems. Our approach estimates the pixel spatio-temporal relationships as an adaptive learning task. The temporal statistical pixel distribution is inferred based on the stochastic gradient algorithm with a Correntropy-based cost function [73, 74], being able to measure relevance of both stationary and non-stationary pixel value fluctuations influencing the background model. Besides, we present an automatic tuning strategy of Correntropy-based cost function allowing to analyze the model error distribution shape within a fixed time window that is suitable for either Gaussian or non-Gaussian noise environments. Furthermore, based on the object detection framework developed in chapter 5, STAL carries out an object motion analysis to detect and track foreground dynamics by using a particle filter based tracking, enhancing the background modeling whenever foreground elements remain motionless. Detected moving objects are modeled by using color representations and optical flow based motion direction, improving the tracking performance under foreground complex conditions such as: crossing, occlusions and smooth scale and rotation changes. Performed accuracy results for well-known datasets show that overall STAL outperforms top state of the art algorithms. Besides, our algorithm can be useful for real-time applications since it requires an acceptable computational burden. The remainder of this chapter is organized as follows: In

section§ 6.1, we describe the theoretical background of proposed adaptive learning approach to support video-based surveillance systems. Experiments and Discussion are presented in section§ 6.2. Lastly, in section§ 6.3, we conclude about the achieved results.

6.1. Spatio-Temporal Adaptive Learning

6.1.1. Adaptive learning for background modeling

We will assume that considered video surveillance scenarios, recorded by a single static video camera, are an stochastic process composed by two main dynamics: foreground and background. So, our goal is to learn a mapping function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ based on pixel information extracted from a given sample sequence $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^N$, with N features and $t = 1, \dots, T$; being $T \in \mathbb{N}$ the number of captured frames. In real-world scenarios, value T tends to infinity. Basically, estimation of f can be seen as an unsupervised learning problem, where $\mathcal{Y} \in \{0, 1\}$ represents both the foreground and background labels. Besides, we are interested in online learning systems where frame pixel samples are available one by one, so that the developed learning algorithm produces the termed *sequence of hypothesis* $\{\mathcal{F}_1, \dots, \mathcal{F}_{t+1}\}$ [75, 76]. Thus, we aim to take advantage of new available information at each time instant to learn the discrimination function $\mathcal{F}_t \subset \mathbb{R}$, while adapting the system in accordance to provided non-stationary process conditions. Then, the sequence of hypothesis can be estimated based on classical stochastic gradient descent within the conventional online learning framework as follows [76, 77]:

$$\mathcal{F}_t := \mathcal{F}_{t-1} - \eta_t \partial_{\mathcal{F}_{t-1}} (l(\mathbf{x}_t, f_{t-1})) \quad (6-1)$$

being $\eta_t \in \mathbb{R}^+$ the learning rate, $l : \mathcal{X} \rightarrow \mathbb{R}$ is a given cost function, $\partial_f (\cdot)$ is the gradient with respect to f and f_0 is a given initial condition.

Owing to computational cost and model mathematical tractability, pixel statistical distribution is commonly assumed as either Gaussian or Gaussian mixture models [38]. So, given a pixel \mathbf{x}_t at time instant t , we state that the hypothesis f_t is ruled by a normal function $\mathcal{F}_t := \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, with mean $\boldsymbol{\mu}_t \in \mathbb{R}^N$ and covariance matrix $\boldsymbol{\Sigma}_t \in \mathbb{R}^{N \times N}$. Therefore, the gradient descent based algorithm in Eq. (6-1) can be used to adjust dynamically \mathcal{F} , taking into account measured pixel variations along the time. Thus, f can be adapted by updating simultaneously $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$. Particularly, we use the following updating rule:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} - \eta_t \partial_{\boldsymbol{\mu}_{t-1}} (l(\mathbf{x}_t, \boldsymbol{\mu}_{t-1})). \quad (6-2)$$

Although online learning models described as in Eq. (6-1) allow dealing with temporal dependencies, most of the sample pixels in video-based analysis are related through an unknown spatio-temporal relationship that must be estimated to improve accuracy of the

mapping function \mathcal{F} , in terms of discriminating between background and foreground. Bearing this in mind, we further propose to make clear the spatio-temporal pixel relationship in Eq. (6-2), by introducing the cost function $l(\mathbf{x}_t, \boldsymbol{\mu}_{t-1})$ as well as the learning rate factor η_t . Afterwards, we also update the parameter $\boldsymbol{\Sigma}_t$.

Correntropy-based cost function for adaptive learning: Correntropy is a localized measure estimating the probabilistic similarity between two given random variables [73, 78]. Particularly, if both variables are very close to each other, their Correntropy value yields the 2-norm distance, while it asymptotically evolves to the 1-norm distance when variables tend to get apart. Furthermore, Correntropy falls to the zero-norm as given variables are very far apart. Thus, provided two concrete random variables, $U \subset \mathbb{R}$ and $V \subset \mathbb{R}$, their Correntropy value is computed as:

$$c_\phi(U, V) = \mathbb{E} \{ \kappa_\phi(U - V) : n \in N \}, \quad (6-3)$$

where $\mathbb{E} \{ \cdot \}$ stands for the expectation operator, N is the random variable sample size, and $\kappa_\phi(\cdot)$ is a symmetric positive definite kernel (commonly assumed as Gaussian), which is scaled by the bandwidth parameter $\phi \in \mathbb{R}^+$ within the assessed similarity window [79].

Among Correntropy properties, the following are the most important [79]:

- Bounded positive definiteness, that is, $0 < c_\phi(U, V) \leq 1$ when using a normalized kernel, reaching its maximum at $U = V$.
- The existence of all even moments estimated from the Correntropy difference $E = U - V$, termed the *error*. Since the high-order values decay when ϕ increases, the second order moment dominates.
- Given the joint pdf $\zeta_{U,V}(u, v)$ of an i.i.d. data sample $\{(u_n \in U, v_n \in V)\}$, the value $c_\phi(U, V)$ tends to the Parzen estimation of the pdf $\hat{\zeta}_{E,\phi}$, at $e = 0$ ($e \in E$).
- Since the Correntropy depends on the kernel bandwidth ϕ , it is strictly concave within the range of $E \in [-\phi, \phi]$. Particularly, Correntropy concavity guarantees the existence and uniqueness of the optimal solution when used as optimization problem cost function, i.e., it can be employed in Eq. (6-1). However, the initial condition should be chosen carefully to make sure the current solution is near the global optimum.

Some works have demonstrated advantages of applying Correntropy to adaptive learning approaches [80]. Particularly, the Correntropy tends to be superior than minimum square error (MSE) measure if the residual of E is non-symmetric or with nonzero mean [73, 79]. Besides, MSE includes all samples in the input space to estimate the similarity/dissimilarity of two random variables, while the Correntropy is determined by the kernel function along $u = v$ line [50].

Considering the above properties, we introduce the Correntropy into the proposed adaptive learning algorithm in Eq. (6-2), so that the stochastic gradient descent maximizes the $c_\phi(\mathbf{x}_t, \boldsymbol{\mu}_{t-1})$. As in case of the MSE criterion [75, 76], we also search the optimal solution of Eq. (6-2) as follows:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \eta_t \partial_{\boldsymbol{\mu}_{t-1}} \{c_\phi(\mathbf{x}_t, \boldsymbol{\mu}_{t-1})\} \quad (6-4)$$

Taking into account Eqs. (6-3) and Eq. (6-4), the concrete gradient computation with respect to $\boldsymbol{\mu}_{t-1}$ yields

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \frac{\eta_t}{\phi^2} \mathbb{E} \{e_t^n \kappa_\phi(x_t^n, \mu_{t-1}^n) : n \in N\}$$

where $\mathbf{x}_t, \boldsymbol{\mu}_t \in \mathcal{X} \subset \mathbb{R}^N$, and $e_t^n = x_t^n - \mu_{t-1}^n$.

In online mode, we assume a single model for each pixel feature ($N=1$), e.g., each RGB color channel is considered aside assuming $\mathcal{F}_t := \mathcal{N}(\mu_t, \sigma_t^2)$, with a mean value $\mu_t \in \mathbb{R}$ and a variance $\sigma_t^2 \in \mathbb{R}^+$. Therefore, the stochastic gradient updating rule can be approximated as:

$$\mu_t = \mu_{t-1} + \frac{\eta_t}{\phi^2} e_t \kappa_\phi(x_t, \mu_{t-1}). \quad (6-5)$$

The above adaptive learning model infers the pixel temporal relationship between the actual sample, x_t , and the previous model, μ_{t-1} . At this point, the introduced Correntropy-based cost function can be useful to deal with non-Gaussian noise conditions related to process dynamics.

It is worth noting that cost function robustness is ruled by the parameter, ϕ , and, as explained in [74], its proper tuning influences the learning algorithm performance more than the choice of the same kernel, in terms of reaching local optimum, rate of convergence, and robustness to impulsive noise during adaption [73, 74, 80]. Mostly, the kernel bandwidth is selected as a compromise between outlier rejection and estimation efficiency. In a particular case of video background modeling tasks, the kernel value is fixed by estimating the sample standard deviation through a considered time window [29, 81]. When dealing with non-Gaussian conditions, however, this moment is not accurate enough. We rather use the adaptive kernel bandwidth selection algorithm based on Middleton's non-Gaussian interference models to adjust dynamically the Correntropy-based cost function in Eq (6-5) as well as the variance of the normal based pixel model $\mathcal{F}_t = \mathcal{N}(\mu_t, \sigma_t^2)$. Thus, given a time window of size T_e , the proposed algorithm aims to analyze the error distribution shape $e_t = x_t - \mu_{t-1}$ by estimating the Kurtosis of e_t at time t as $\beta_{e_r} = \mathbb{E} \{(e_r - \mu_{e_r}/\sigma_{e_r})^4\}$ with $\beta_{e_r} \in \mathbb{R}^+$, being μ_{e_r} and σ_{e_r} the mean and the standard deviation of e_r , respectively, with $r \in \{t - T_e, t - T_e + 1, \dots, t\}$. The Kurtosis value provides information about the distribution shape that may discriminate between Gaussian and non-Gaussian dynamics. Specifically, a distribution holding high kurtosis has a sharper peak and heavy tails, while a low kurtosis

implies shorter, thinner tails. Consequently, the Correntropy kernel bandwidth is updated as proposed in [74]:

$$\phi_t = \alpha\phi_{t-1} + (1 - \alpha)\sigma_{e_r}\sqrt{\beta_G/\beta_{e_r}}, \quad (6-6)$$

where $\alpha \in \mathbb{R}[0, 1]$ is a forgetting factor and β_G is the kurtosis of the Gaussian distribution. It must be quoted that we manage the tradeoff between robustness and convergence speed by properly fixing the initial kernel bandwidth ϕ_0 and α values in Eq. (6-6).

As a result, if the computed distribution error within the given window lasting T_e has longer and fatter tails (see Fig. 6.1a), the update rule in Eq. (6-6) tends to yield a bandwidth value smaller than σ_{e_t} , rejecting misleading error information (by instance because of burst-like samples). Otherwise, in case of light tail shapes (see Fig. 6.1b), ϕ_t value tends to be larger than σ_{e_t} , extracting more suitable updating information to improve algorithm speed convergence [74].

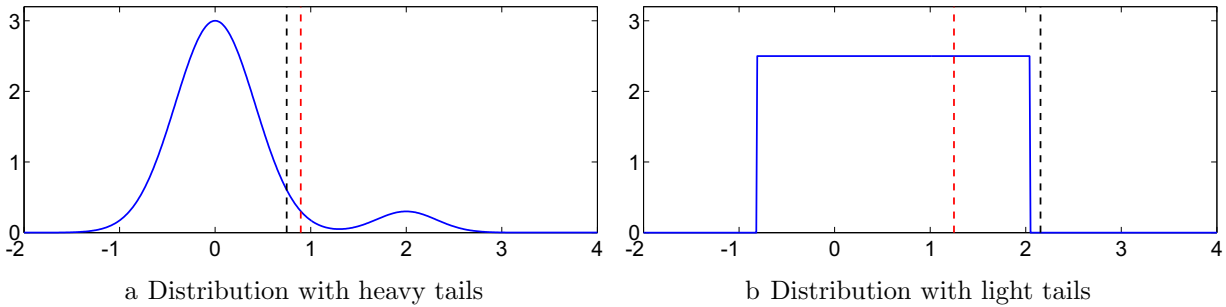


Figure 6-1.: Kernel bandwidth selection strategy. — pdf of error e_t , - - error standard deviation σ_{e_t} , - - kernel bandwidth ϕ_t

Due to the updating rule in Eq. (6-6) is computed over time windows and assuming pixel dynamics as stationary, the kernel bandwidth ϕ_t allows considering those main video backgrounds varying over time. Specifically, illumination changes and background motion objects can be modeled by the normal function $\mathcal{F}_t := \mathcal{N}(\mu_t, \sigma_t^2)$ by updating μ_t according to Eq. (6-5) and after fixing $\phi = \phi_t$, $\sigma_t^2 = \phi_t^2$. Hence, illumination variations (either Gaussian or non-Gaussian) as well as background motion objects influence directly on the error distribution shape that, in turn, rules the system adaptation.

6.1.2. Learning rate estimation based on object motion analysis

Most widely used video segmentation approaches center their attention only on background modeling, while implying outside elements as foreground [32, 36, 47]. However, these approaches may not distinguish between foreground and background dynamics if they become similar. For example, moving objects that become static for a while or, in contrast, static objects that start moving. Therefore, learning only based on background dynamics may not

be enough to infer a suitable mapping function \mathcal{F} . With this in mind, foreground components should also be identified, modeled, and tracked. As explained below, functional \mathcal{F} must be supplied with information about pixel dynamics, i.e., if a given pixel is identified as a foreground component, moving or static.

Motion detection: Given an image sequence described with the set $\{\mathbf{X}_t\}$, we propose to employ the Region Change Detection stage proposed in section § 5.1.1 to identify patches which changed between consecutive frames. So, with the attained matrix \mathbf{H}_t we highlight patches that changed between frame $t - 1$ and t . Afterwards, the optical flow approach described in section § 5.1.2 is used to find the motion direction of each patch highlighted as moving in \mathbf{H}_t . Thus, the optical flow matrices \mathbf{A}_t , \mathbf{B}_t and the angle direction matrix \mathbf{Q}_t are attained as result. Figure 6-2 illustrates the proposed motion detection procedure, the detected moving patches of \mathbf{H}_t emphasize in this case the walker's shape, while the computed optical flow, encoded in matrices \mathbf{A}_t and \mathbf{B}_t , shows the subject movement direction.

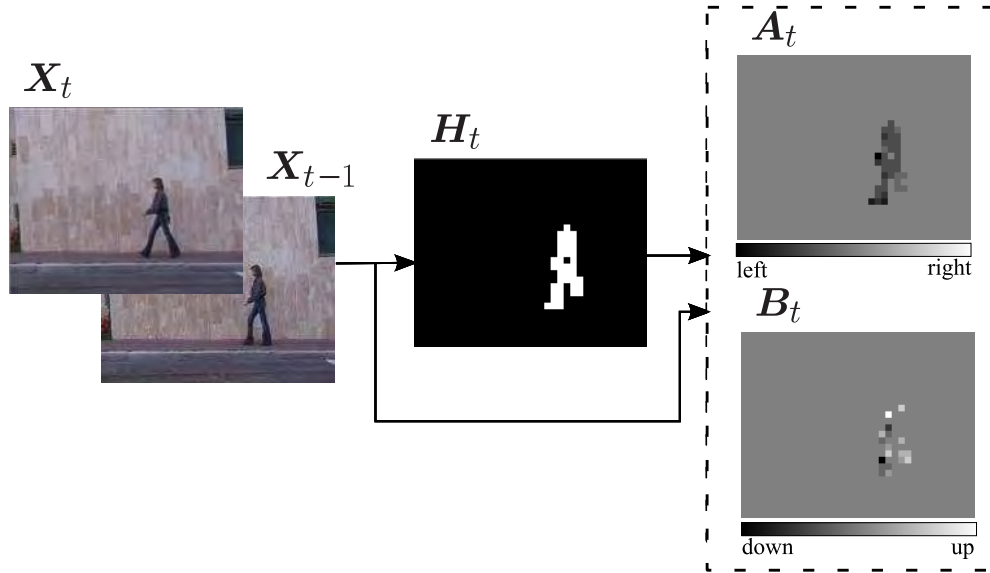


Figure 6-2.: Proposed motion detection scheme

Object detection and modeling: At this stage, we aim to identify from the computed matrix \mathbf{H}_t the patches highlighted as moving, which describe foreground objects in the scene. To this end, the first step is to identify regions as groups of spatially connected patches highlighted as moving. Thus, a connectivity-based operation over \mathbf{H}_t is carried out, labeling spatially connected patches. Namely, we label the following regions of interest: label '1' makes up a first approximation about the background, label '2' makes up one connected region, and so on until label ' D_t ' that makes up the D_t -th connected region, at time

instant t . There might be moving regions, which do not correspond to foreground objects in the scene. That is the case of regions which changed due to intrinsic scene behaviors, such as illumination changes or dynamical background elements. To avoid classifying such regions as foreground objects, we propose to impose a spatial smoothness constraint on the corresponding movement region information encoded in \mathbf{Q}_t . In particular, we search for regions having a low movement direction variability in terms of the estimated optical flow. Regarding this, the d_t -th region containing $N_{d_t} \subset \mathbb{N}$ pixels, is mapped into \mathbf{Q}_t to further compute a local optical flow angle histogram. As a result, the region d_t is assumed to be a moving object if the ratio $\theta_1^{d_t}/\theta_2^{d_t}$ is larger than $\rho \in \mathbb{R}^+$, being $\theta_1^{d_t}, \theta_2^{d_t} \in \mathbb{R}[0, 360]$ the first and second most frequent angle values, respectively. This assumption is made to avoid connected regions having high movement direction variability, which should rather be related to the intrinsic scenario artifacts above mentioned.

Afterwards, each one of the detected objects is enclosed into a bounding box generating the set $O_t = \{\boldsymbol{o}_t^p : p = 1, \dots, P_t\}$ where $\boldsymbol{o}_t^p \in \mathbb{N}^{1 \times 4}$ encodes the (i, j) up-left position, the width $w_C^p \in \mathbb{N}$, and the height $w_R^p \in \mathbb{N}$ of object p , being $P_t \in \mathbb{N}$ the number of detected moving objects at a time instant t . Lastly, to model the detected objects, we characterize both object shape (encoded in the color intensity matrix \mathbf{X}_t) and movement dynamics (in \mathbf{Q}_t). So, the sets $\Psi_t = \{\mathbf{Y}_t^p\}$ and $\mathcal{I}_t = \{\epsilon_t^p\}$ are built to model detected objects, where $\mathbf{Y}_t^p \in \mathbb{R}^{w_C^p \times w_R^p \times Z}$ is the matrix obtained by mapping the bounding box of object p into \mathbf{X}_t and $\epsilon_t^p \in \mathbb{R}[0, 360]$ corresponds to θ_1^p .

Figure 6-3 depicts proposed object detection and modeling methodology. The angle histogram of the moving region (colored in red) points out a clear motion direction trend, since ratio θ_1^1/θ_2^1 gets a high value. Therefore, the red labeled region is assumed as a foreground object and is modeled by both the intensity RGB matrix \mathbf{Y}_t^1 and the angle value ϵ_t^1 .

Object tracking: Once computed both object model sets (Ψ_t and \mathcal{I}_t) as well as the bounding box set O_t , we must keep tracking every single detected object to incorporate its spatial information into the mapping function f_t . Therefore, we search previously detected objects in each new input frame \mathbf{X}_{t+1} . Commonly, object video tracking approaches are based on particle filter algorithms assuming a dynamic model of the moving object [82]. Instead, we can reuse movement object information already estimated above. Namely, information about the position (bounding box) of moving object p at time instant t is encoded in the state representation \boldsymbol{o}_t^p . So, inspired on particle-based filter approaches, the following state transition function is defined:

$$\mathbf{g}_{t,p}^{n_p} = \mathcal{N}(\boldsymbol{o}_t^p, \mathbf{\Lambda}_t),$$

being $\mathbf{g}_{t,p}^{n_p} \in \mathbb{R}^4$ with $n_p = 1, \dots, N_p$, the estimated particle from \boldsymbol{o}_t^p , assuming additive Gaussian noise conditions with covariance matrix $\mathbf{\Lambda}_t \in \mathbb{R}^{4 \times 4}$. The number of particles $N_p \in \mathbb{N}$ and $\mathbf{\Lambda}_t$ can be selected according to some object motion constraints a priori imposed by the user. Thus, to model each particle, each $\mathbf{g}_{t,p}^{n_p}$ is mapped into \mathbf{X}_{t+1} , obtaining an RGB

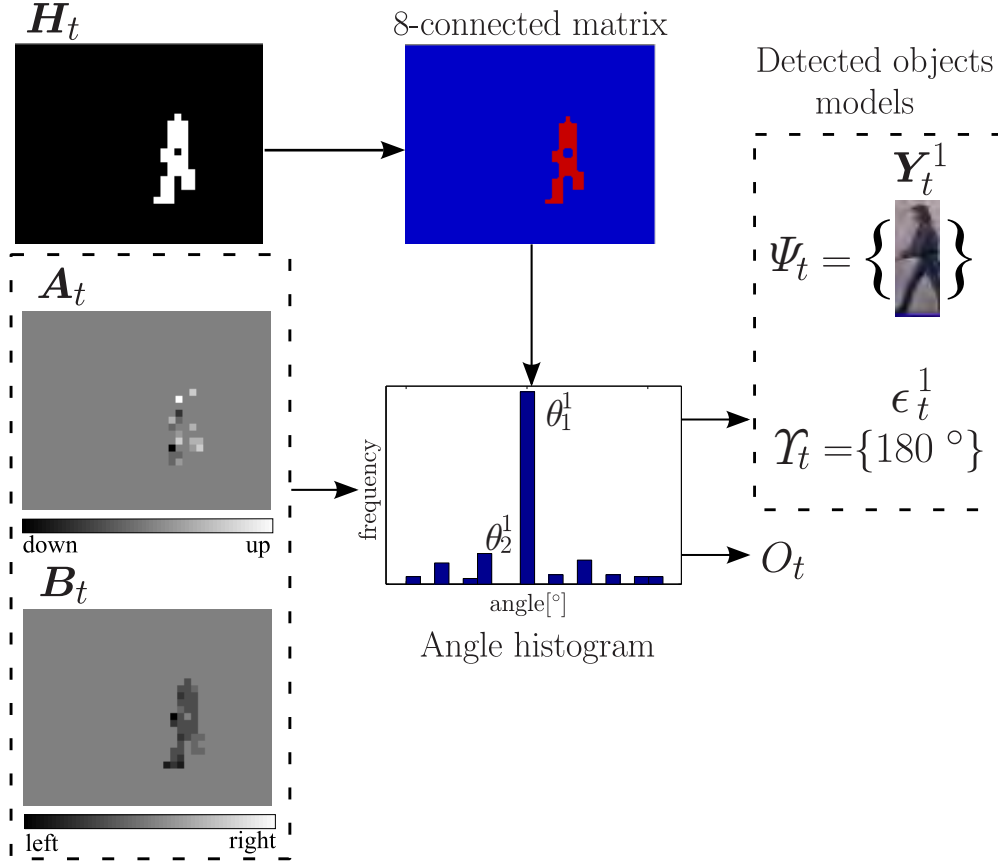


Figure 6-3.: Object detection and modeling scheme

intensity model $\mathbf{\Gamma}_{t,p}^{n_p} \in \mathbb{R}^{w_{Rt,p}^{n_p} \times w_{Ct,p}^{n_p} \times Z}$. Lastly, the particle movement direction $\gamma_{t,p}^{n_p} \in [0, 360]$ is estimated from the angle between the centroids of \mathbf{o}_t^p and $\mathbf{g}_{t,p}^{n_p}$.

Now, we search for the particle $\mathbf{g}_{t,p}^{n_p}$ that best matches the object \mathbf{o}_t^p . Thus, we compare their respective RGB and movement direction models based on kernel similarity functions [50]. Recently, some machine learning approaches have shown that the use of multiple kernels instead of just one may improve data interpretability [49, 51]. In fact, as stated in section § 4.2, input data can be analyzed from different information sources by a convex combination of some basis kernels. In particular, we propose a color intensity based similarity function $\kappa_{\sigma_c}(\cdot, \cdot)$ to be applied to the p object and its n_p -th particle as a combination of Gaussian kernels with bandwidth $\sigma_c \in \mathbb{R}^+$, yielding:

$$\kappa_{\sigma_c}(\mathbf{Y}_t^{p,z}, \hat{\mathbf{\Gamma}}_{t,p}^{n_p,z}) = \mathbb{E} \left\{ \exp \left(\frac{-\|\mathbf{Y}_t^{p,z} - \hat{\mathbf{\Gamma}}_{t,p}^{n_p,z}\|_F^2}{2\sigma_c^2 \|\mathbf{Y}_t^{p,z}\|_F \|\hat{\mathbf{\Gamma}}_{t,p}^{n_p,z}\|_F} \right) : z \in Z \right\}, \quad (6-7)$$

being $\hat{\mathbf{\Gamma}}_{t,p}^{n_p,z} \in \mathbb{R}^{w_R^p \times w_C^p}$ the resized version of $\mathbf{\Gamma}_{t,p}^{n_p,z}$ according to the p object bounding box size. The Kernel in Eq. (6-7) aims to reveal average similarity between color intensities

avoiding influence of different particle sizes. Additionally, a similarity measure based on relative movement direction is presented through the following Gaussian kernel:

$$\kappa_{\sigma_\epsilon}(\epsilon_t^p, \gamma_{t,p}^{n_p}) = \exp\left(\frac{-(180 - ||\epsilon_t^p - \gamma_{t,p}^{n_p}||_1 - 180|_1)^2}{2\sigma_\epsilon^2}\right), \quad (6-8)$$

with kernel bandwidth $\sigma_\epsilon \in \mathbb{R}^+$. Notation $|\cdot|_1$ stands for the 1-norm.

As said before, to merge the presented similarity functions in Eqs. (6-7) and (6-8), the following multiple kernel based approach is employed to compare the p object against n_p -th particle, yielding:

$$\kappa_{\vartheta}(\sigma_t^p, \mathbf{g}_{t,p}^{n_p}) = \nu_{t,p} \left(\vartheta_1 \kappa_{\sigma_c}(\mathbf{Y}_t^p, \hat{\mathbf{\Gamma}}_{t,p}^{n_p}) + \vartheta_2 \kappa_{\sigma_\epsilon}(\epsilon_t^p, \gamma_{t,p}^{n_p}) \right),$$

with $\vartheta_1 + \vartheta_2 = 1$ and $\vartheta_1, \vartheta_2 \in \mathbb{R}^+$.

On the other hand, along with the similarity information given by color intensity and movement direction representations, we also include the $\nu_{t,p} \in \{0, 1\}$ parameter to avoid matching particles related to background regions including the $\nu_{t,p} \in \{0, 1\}$ parameter that we estimate as the color intensity similarity between a given particle $\hat{\mathbf{\Gamma}}_{t,p}^{n_p}$ and the background region, $\hat{\mathbf{\Xi}}_{t,p}^{n_p} \subset \mathbb{R}^{w_R^p \times w_C^p \times Z}$, contained within the same particle, i.e.:

$$\nu_{t,p} = \begin{cases} 1, & \kappa_{\sigma_b}(\mathbf{\Gamma}_{t,p}^{n_p}, \hat{\mathbf{\Xi}}_{t,p}^{n_p}) < \zeta_b \\ 0, & \text{Otherwise} \end{cases}$$

where $\zeta_b \in \mathbb{R}^+$ and the Gaussian kernel function $\kappa_{\sigma_b}(\cdot, \cdot)$ is defined as follows:

$$\kappa_{\sigma_b}(\hat{\mathbf{\Gamma}}_{t,p}^{n_p,z}, \hat{\mathbf{\Xi}}_{t,p}^{n_p,z}) = \mathbb{E} \left\{ \exp \left(\frac{-\|\mathbf{\Gamma}_{t,p}^{n_p,z} - \hat{\mathbf{\Xi}}_{t,p}^{n_p,z}\|_F^2}{2\sigma_b^2 \|\hat{\mathbf{\Gamma}}_{t,p}^{n_p,z}\|_F \|\hat{\mathbf{\Xi}}_{t,p}^{n_p,z}\|_F} \right) : z \in Z \right\},$$

being $\sigma_b \in \mathbb{R}^+$.

As a result, the particle $\mathbf{g}_{t,p}^*$ that has the highest similarity value with object p , at time instant t , can be found as:

$$\mathbf{g}_{t,p}^* = \arg \max_{\mathbf{g}_{t,p}^{n_p}} \kappa_{\vartheta}(\sigma_t^p, \mathbf{g}_{t,p}^{n_p}). \quad (6-9)$$

Nonetheless, the particle $\mathbf{g}_{t,p}^*$ can be estimated with not enough accuracy since the tracked object may either leave the scene or be occluded. To cope with this withdraw, we impose that the optimization carried out in Eq. (6-9) (computing similarity between object p and $\mathbf{g}_{t,p}^*$) must be subject to the following restriction: $\kappa_{\vartheta}(\sigma_t^p, \mathbf{g}_{t,p}^*) < \zeta_{\kappa_{\vartheta}}$, with $\zeta_{\kappa_{\vartheta}} \in \mathbb{R}^+$. Therefore, it

holds that $\mathbf{o}_{t+1}^p = \mathbf{o}_t^p$, $\mathbf{Y}_{t+1}^p = \mathbf{Y}_t^p$, and $\epsilon_{t+1}^p = \epsilon_t^p$, otherwise the following updating rules are employed instead:

$$\mathbf{o}_{t+1}^p = \mathbf{g}_{t,p}^* \quad (6-10a)$$

$$\mathbf{Y}_{t+1}^p = \mathbb{E} \{ \mathbf{Y}_{t_o}^p + \mathbf{\Gamma}_{t,p}^* : t_o = t - T_o + 1, \dots, t \} \quad (6-10b)$$

$$\epsilon_{t+1}^p = \mathbb{E} \{ \epsilon_{t_o}^p + \gamma_{t,p}^* : t_o = t - T_o + 1, \dots, t \} \quad (6-10c)$$

where T_o is the size of the time window employed to infer object model variations.

Finally, to incorporate the estimated foreground dynamics into the background updating rule (see Eq. (6-5)), we calculate the set Φ_t holding the coordinates of all the $C_t \in \text{mathbb{N}}$ pixels of tracked objects according to the object bounding box set O_t . Thus, to avoid inclusion of false information into the background model generated by complex foreground object dynamics, the learning factor η_t of pixel x_t is set as follows:

$$\eta_t = \begin{cases} \lambda\phi_t, & x_t \notin \Phi_t \\ 0, & \text{Otherwise} \end{cases} \quad (6-11)$$

Figure 6-4 shows the general scheme of the proposed object tracking that determines the particle having the highest similarity with the object models \mathbf{Y}_t^1 and a_t^1 . Here, the new estimated models are estimated as result of the updating rules given in Eqs. (6-10b) and (6-10c).

The spatio-temporal adaptive learning (STAL) algorithm can be summarized, as shown in Figure 6-5, considering the proposed Correntropy based cost function for adaptive learning (section§ 6.1.1) and the learning rate estimation based on object motion analysis (section§ 6.1.2). It is worth saying that objects detected by the proposed object detection and modeling stage are compared at each time instant against all the previously detected and tracked objects. The above procedure is performed to achieve a unique representation for each motion object into the considered scenario. Namely, detected objects at a time instant t that occupy the same location in the scenario as previously detected and tracked objects are not included in sets O_t , Ψ_t and Υ_t , avoiding redundant models.

6.2. Experiments and Discussion

Generally, different real-world situations should be considered to assess quality of discrimination between foreground and background in video surveillance systems. Regarding this, we carry out the following three experiments to evaluate the performance of the main STAL stages: *i*) Presented Correntropy-based adaptive learning cost function is tested over both static and dynamic background scenarios. Here, we aim to analyze visually cost function temporal evolution and its contribution into the STAL-based background modeling. *ii*)

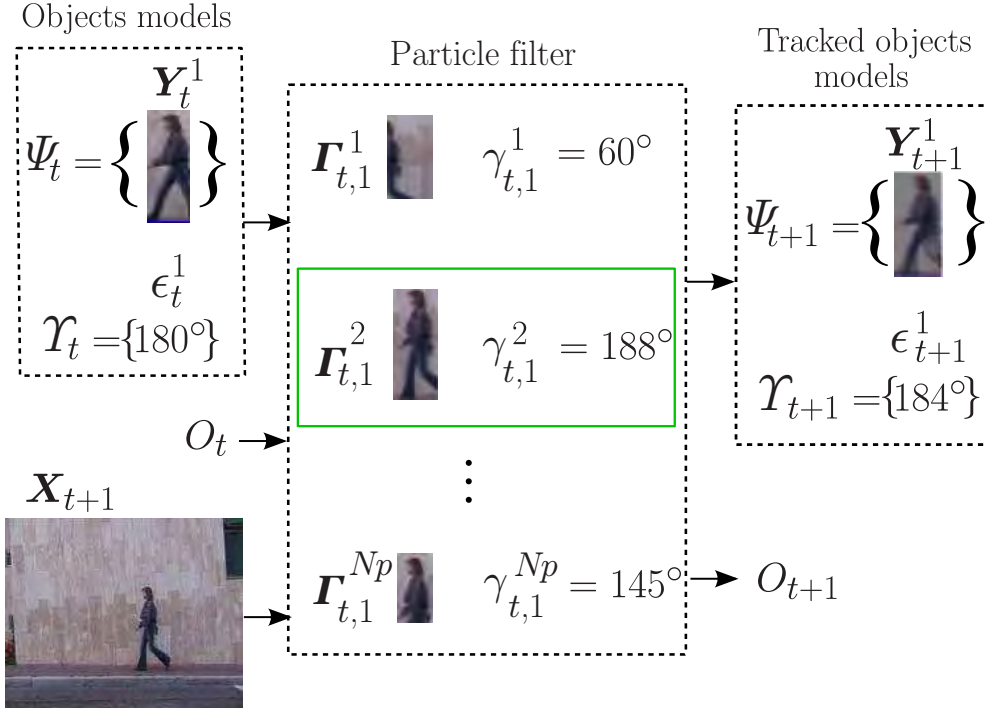


Figure 6-4.: Particle filter based object tracking scheme

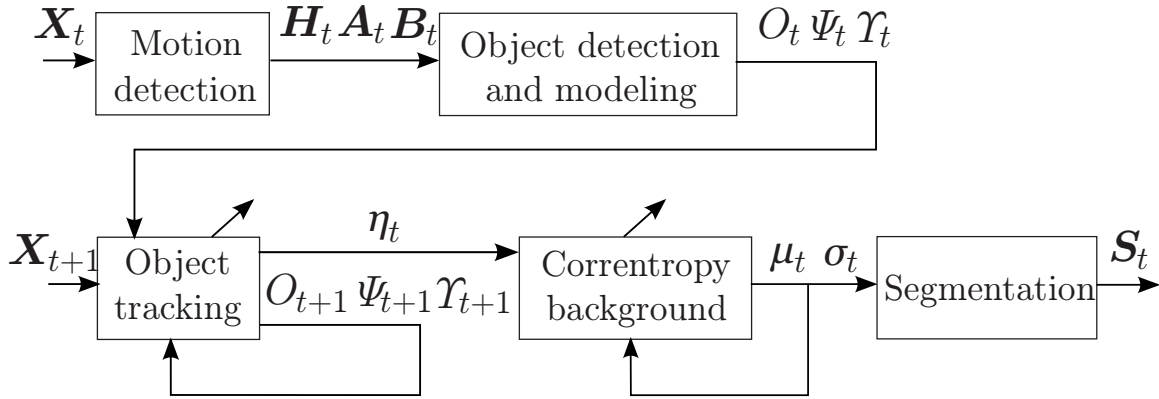


Figure 6-5.: STAL scheme.

The developed estimation strategy of learning rate is evaluated in cases with more complex foreground dynamics. Here, to show STAL tracking ability of complex dynamics and to encode properly pixel spatio-temporal relationships, we validate the STAL algorithm on scenarios presenting both static and moving objects, as well as, object occlusions and crossing cases. Lastly, *iii*) The STAL-based background and foreground discrimination model is tested against a ground-truth set related to video segmentation tasks for video surveillance systems.

All experiments are performed using video sequences recorded in real-world scenarios, including typical challenges for video surveillance systems such as: dynamical background, bootstrapping, static and moving objects, foreground object occlusions, shadows, and camouflages. Namely, the following four databases are employed:

DBa - *A-Star-Perception*: See section § 4.4.3 for dataset description.

DBb - *Left-Packages*: See section § 4.4.3 for dataset description.

DBd - *Change Detection 2012*:¹ Developed as part of the CVPR 2012 Change Detection Workshop challenge, holds 31 different video sequences, providing a wide range of detection challenges representative of typical indoor and outdoor visual data captured today in surveillance, smart environment, and video database scenarios. For each video sequence there is a spatial and a temporal region of interest (ROI). The spatial ROI is a mask which only considers the results for the pixels inside it. The temporal ROI states from which frame the segmentation results are considered. Hand segmented ground-truths are available. Each ground-truth pixel may have one of the 5 following labels: 0 background, 50 hard shadow, 85 outside region of interest, 170 unknown motion and 255 foreground.

DBe - *Activity Recognition*:² Contains several video recordings of people performing different activities. Although the main goal is to identify performed activities, it has been also used for testing of tracking algorithms [14]. Ground-truth images are available, where truth pixel holds one of the two following labels: 0 (black) for background pixels and 255 (white) for foreground pixels.

6.2.1. Pixel temporal relationship using Correntropy-based cost function

By visual inspection, we analyze the estimated temporal evolution of the μ_t parameter to make a clear contribution of the Correntropy-based adaptive learning cost function. To this end, we consider two scenes combining different background dynamics (mostly static and highly dynamical), which are part of two different video sequences: DBa-ShoppingMall and DBd-Fall, respectively. The former one is a common indoor surveillance scene with several foreground objects moving through, while the background remains mostly static. The latter one is an outdoor scenario with a highly dynamical background because of the tree leaf movement. For the sake of comparison, the learning rate factor η_t/ϕ_t is fixed at 0.5, as commonly used in adaptive learning approaches [74, 79]. In this case, we just focus only on the information provided by the cost function, but without considering the object motion

¹<http://www.changedetection.net/>

²<http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html>

analysis stage. To estimate the corresponding error moments, the forgetting factor α is also fixed at 0.5, while a non-overlapping time window of size $T_e = 25$ is employed [74].

Figure 6.6b shows the calculated gradient absolute value of the 378th frame taken from the DBa-ShoppingMall video (Figure 6.6a). As seen, the Correntropy based background modeling approach provides gradient values for foreground regions (i.e. moving people) close to zero, because of the ability of the introduced kernel bandwidth tuning criterion to learn pixel distributions along time. Hence, due to the static nature of this scene, the bandwidth ϕ_t becomes thinner, making the foreground pixels as outlier values, so that their gradient becomes close to zero. In contrast, in the 195th frame of DBd-Fall (Figure 6.6c), the kernel bandwidth becomes wider for those pixels located on the tree leaves, increasing significantly the computed gradient of the updating background modeling (see Figure 6.6d).

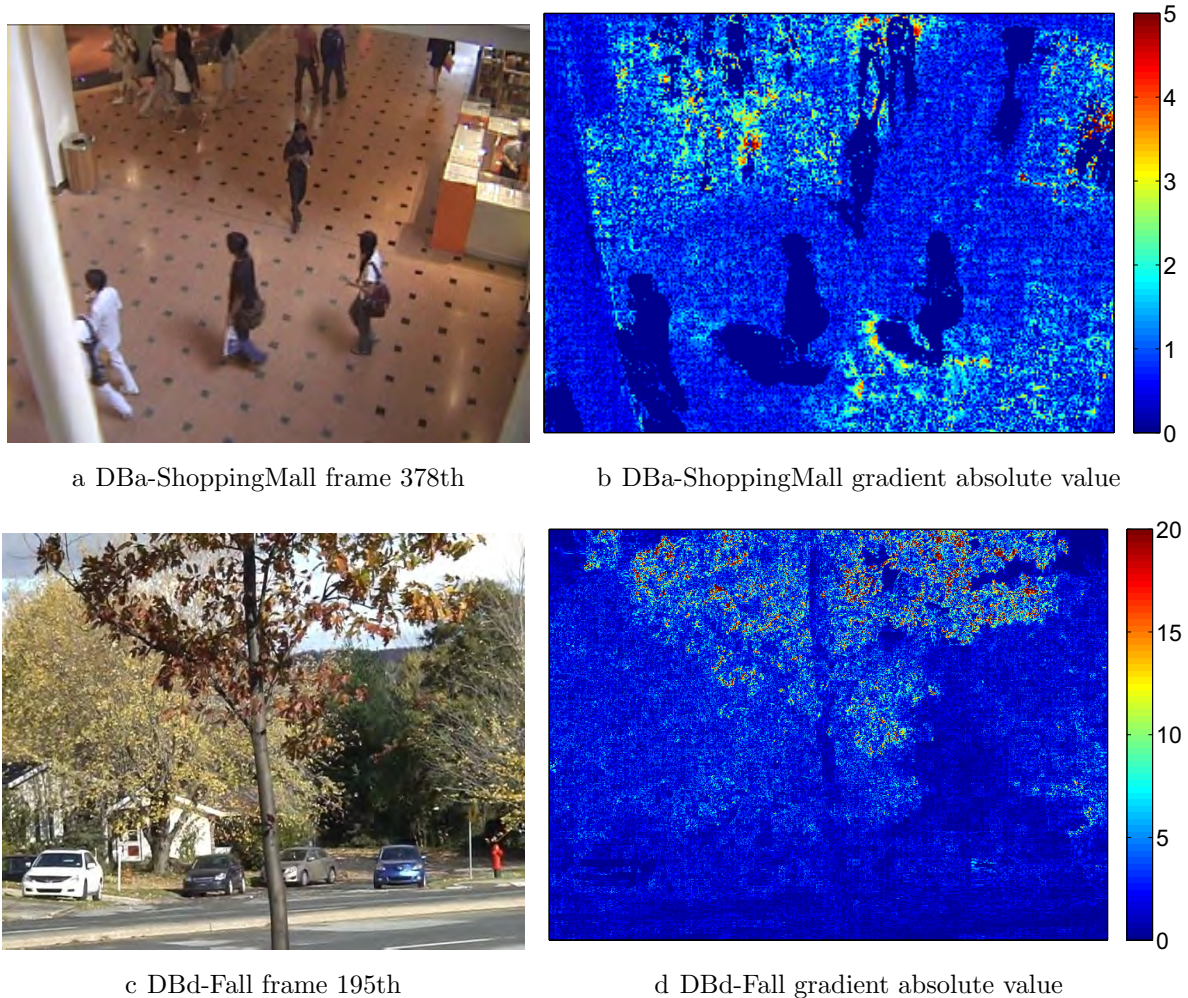


Figure 6-6.: Correntropy-based stochastic gradient update.

For the sake of comparison, the proposed background modeling approach using Correntropy based cost function is contrasted against the traditional MSE cost function algorithm,

as seen in Figure 6-7 showing an example of temporal evolution of the background model parameter μ_t estimated for a single pixel of the red color channel. Particularly for the DBa-ShoppingMall video (see Figure 6.7a), the given pixel value tends to be mostly static, although there are some abrupt changes (outliers) caused by moving objects. So, the MSE cost function does not consider the error distribution changes along time and turns to be sensitive to outlier values. Therefore, the MSE tends to generate noisy models leading to undesired information. Instead, the proposed Correntropy based model is able to discover the main sample dynamics by analyzing the error distribution shape along the time. That is, our approach models properly the pixel intensity by fixing a small Correntropy kernel bandwidth to better reject outlier values.

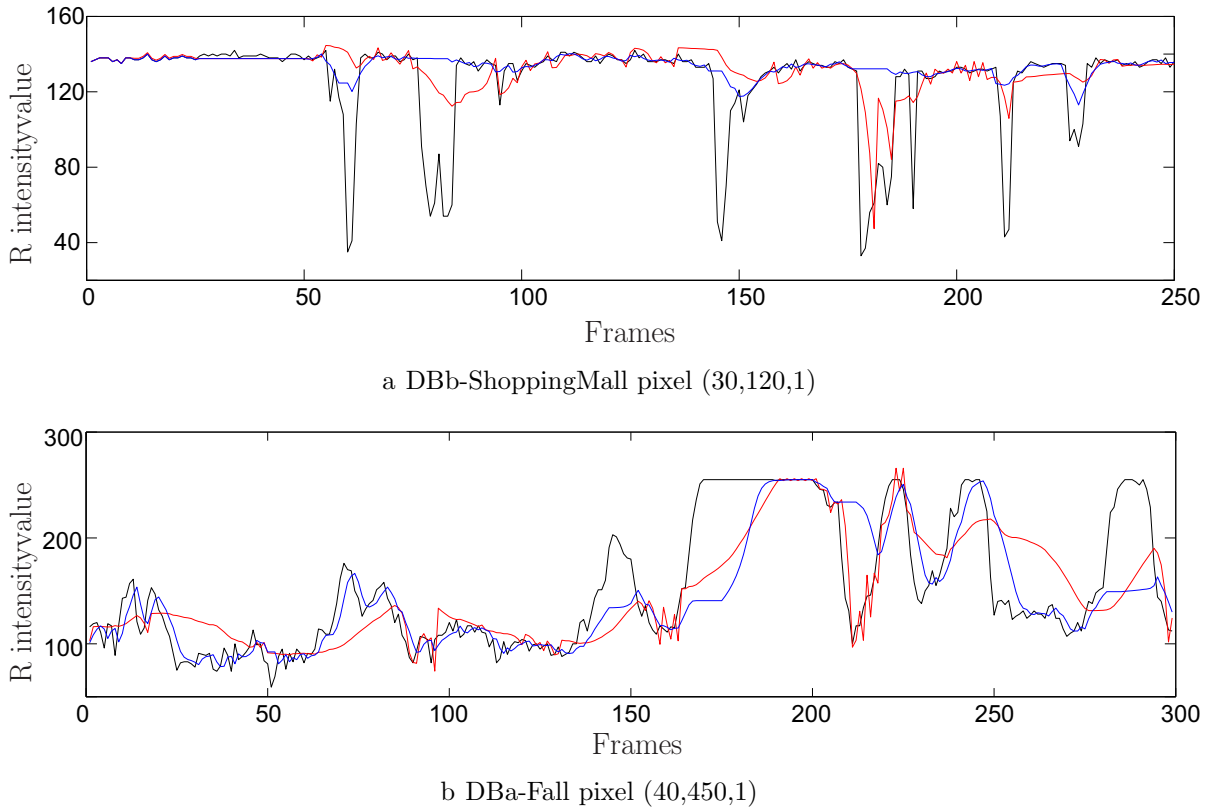


Figure 6-7.: Background modeling (μ_t parameter). — Pixel value, — MSE based model, — Correntropy based model.

As for the DBd-Fall video with highly dynamical along time pixel values (see Figure 6.7b), the MSE does not allow to follow abrupt changes, which might mislead into false segmentation results. In the Correntropy model, the kernel bandwidth becomes wider to include bigger model parameter changes. Therefore, proposed Correntropy cost function with automatic kernel bandwidth selection strategy makes easier to build a background model able to follow fast changes, according to the previously detected pixel dynamics. To make clear this situation, estimated Correntropy-based kernel temporal evolution is shown in Figure 6-8.

Particularly, it can be seen in Figure 6.8a that even for small errors, the kernel value for DBa-ShoppingMall so vanishes that significantly reduces stochastic gradient. On contrast, the estimated kernel allows following bigger changes, as seen in Figure 6.8b for DBd-Fall.

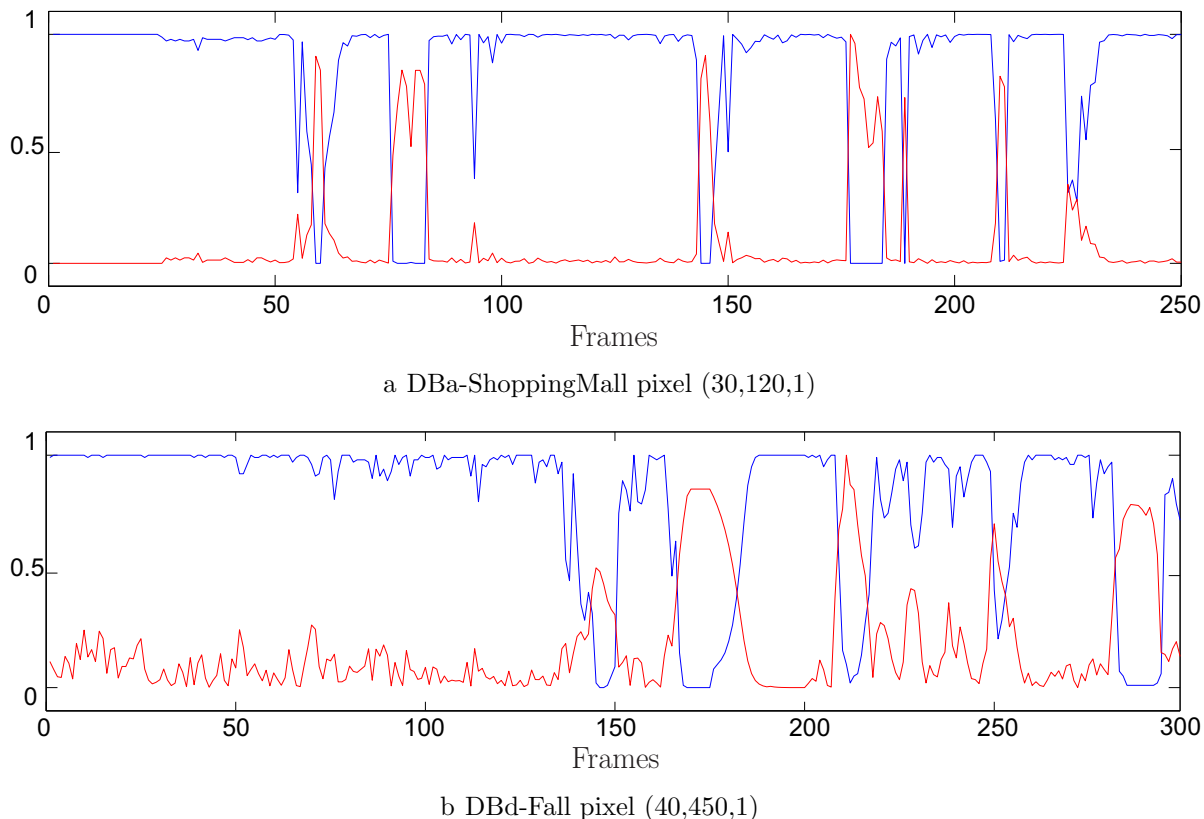


Figure 6-8.: Correntropy-based cost function evolution. — Normalized error, — Correntropy value.

6.2.2. Improved object motion analysis using pixel spatio-temporal relationships

As referred in chapter 5, to avoid false segmentations, we must carry out proper detection and tracking of moving objects. Our aim is to discover spatio-temporal relationships among pixels regarding foreground dynamics, which are incorporated in our background model through the learning rate factor. To this, the proposed object motion analysis stage is tested over three particular tracking challenging tasks, as described below. Free parameter values for this stage are experimentally fixed according to Table 6-1, considering both the input images resolution and the trade-off between system accuracy and computational burden. Note that we assume that detected objects follow small and medium sizes in comparison to the frame resolution.

Table 6-1.: Object motion analysis parameters.

<i>Parameter</i>	<i>Value</i>
w_Ω	$0.02\sqrt{w_R \times w_C}$
ζ_H	$20(w_\Omega \times w_\Omega)$
k	$3w_\Omega$
ρ	2
N_p	400
σ_c	0.2
σ_a	40
ϑ_1	0.7
ϑ_2	0.3
ζ_b	0.4
σ_b	0.2
ζ_{κ_ϑ}	0.6
λ	0.5

Besides, the first and second elements from the diagonal of covariance tracking matrix $\mathbf{\Lambda}_t$ are set as 0.02 the number of rows and columns, respectively. The third and fourth elements are fixed as 0.1 the previous width and height of the corresponding object.

Detection and tracking of moving objects: Figure 6-9 shows some tracked objects taken from the DBd-StretLight and DBa-ShoppingMall sample videos, respectively. As seen, the object labels (remarked by colored boxes) are properly achieved for considered tracked objects, even though both scenarios describe quite different situations. It should be quoted that proposed motion analysis approach is able to detect and track moving objects, considering both the object movement direction and the object color intensity into a particle filter based tracker.

Presence of crossing and/or occluding objects in the foreground: In some particular video surveillance conditions, foreground dynamics may be complex due to either presence of crossing objects and/or occlusions, making difficult to carry out a suitable moving object tracking, as it is the case for the DBd-Pedestrians and the DBd-PETS2006 videos. In the former video, objects appear with different dynamics, namely, bicycle and people crossing each other (see video examples in Figures 6.10a - 6.10d), while the latter video collection is a common surveillance scene holding multiple objects moving and crossing each other (see Figures 6.10f - 6.10h). For each shown frame, tracked objects using proposed object motion analysis methodology are delineated by colored boxes. Also, the estimated angle trend is displayed in Figures 6.10e and 6.10i, where the data-tips indicate the corresponding frames. So, inclusion of angle trend as foreground model improves separability among occluded objects. As seen in Figures 6.10a, 6.10c and 6.10g showing some objects crossed each other,



Figure 6-9.: Tracked objects obtained from DBd-StretLight and DBa-ShoppingMall sample videos.

the respective angle trajectory is discriminative and allows object motion analysis to track properly the label set.

Similar foreground and background dynamics: To consider object motion influence on the background model estimation, we carry out analysis of foreground objects that remain static during long time periods. Particularly, the DBb-LeftBox video sequence is employed having a person entering into the scene and stopping twice at the beginning of the video recording, making hard to accomplish background/foreground discrimination (see Figure 6-11). For comparison sake, we estimate the background model using the baseline SC-SOBS approach for which presence of static foreground objects degrades the needed neural network calibration (bootstrapping), producing false information in the background model, as seen in Figure 6.11a. In case of the STAL model, we firstly estimate the μ_t parameter model, but without taking into consideration any object motion information, that is, we set the learning rate as a constant value, $\eta_t/\phi_t^2 = 0.5$. Testing evidences that if foreground and background dynamics become similar, the Correntropy-based cost function is not able to properly detect the foreground pixels as outliers, introducing false information into μ_t . As a result, estimated model is also affected by static foreground objects, as shown in Figure 6.11b. Therefore, we recompute the μ_t parameter to include object motion information as

in Eq. (6-11) (Figure 6.11c shows a concrete object to be tracked). As a consequence, STAL-based estimated model allows distinguishing between considered foreground and background dynamics. Moreover, background model rejects foreground dynamics even if it remains static during a long time interval (see Figure 6.11d).

6.2.3. Background and foreground discrimination based on STAL

Lastly, the proposed STAL framework is tested as a video segmentation approach. Once both parameters (μ_t and σ_t) of the mapping function \mathcal{F} are tuned by STAL, discrimination between background and foreground dynamics can be carried out for video sequences. So, a given pixel x_t is mapped to matrix \mathbf{S}_t with dimension $w_R \times w_C$, holding elements expressed as follows:

$$s_t = \begin{cases} 1, & \mathcal{F}(x_t; \mu_t, \sigma_t) < \zeta_{\mathbf{S}} \\ 0, & \text{Otherwise} \end{cases}$$

where $\zeta_{\mathbf{S}} \in \mathbb{R}^+$ is a given segmentation parameter and $s_t \in \{0, 1\}$ corresponds to the (i, j) pixel label. Particularly, $s_t = 1$ stands for pixels labeled as foreground, otherwise, the label is background.

Two different tests are carried out to evaluate the performance of STAL as a video segmentation approach. The first one aims to analyze the contribution of the correntropy-based cost function on the segmentation output. To this, we propose to employ as an alternative cost function for STAL the MSE, naming it as STAL-MSE. Thus, the segmentation performance for both methodologies is compared using the supervised measures exposed in section § 4.4.3. The purpose of the second test is to compare STAL against state of the art methodologies by using several video sequences.

For both tests, STAL based video segmentation free parameters are fixed as follows: Correntropy-based cost function stage as in section § 6.2.1. Object motion analysis stage as in section § 6.2.2. Finally, the STAL-based background and foreground discrimination stage free parameter is set experimentally as $\zeta_{\mathbf{S}} = 0.6$. Finally, we propose to implement the shadow removal post-processing as explained in section § 5.3.1, aiming to avoid false segmented regions due to illumination changes, fixing the free parameter values as $\xi_n = 0.03$ and $\xi_s = 75$.

Test 1: Comparison of cost functions to support the segmentation process Using as cost function the MSE and based on the LMS adaptive filtering algorithm, we derive an updating rule for model parameter μ_t as described in Appendix A. To compare the segmentation results for both STAL and STAL-MSE, representative videos from the four datasets aforementioned are selected.

Table 6-2 presents the attained results for both STAL and STAL-MSE. Overall results demonstrate that STAL using the correntropy as cost function attains a higher F_1 measure with a lower standard deviation. Particularly, for videos DBe-MosheWalk and DBe-DariaWalk, results are almost the same since those videos present mostly stationary dynamics (see Figure 6.12c). Now, when pixel temporal evolution presents more complex dynamics (e.g. non-stationarity, non-Gaussianity), it is clear that STAL is able to deal with such dynamics, properly adjusting the background model. The above can be evidenced for DBd-Overpass and DBd-Canoe videos, where noisy segmentations are attained by STAL-MSE as seen in Figures 6.12g and 6.12k.

Table 6-2.: Estimated pixel-based F_1 measure for STAL and STAL-MSE approaches.

<i>Video sequence</i>	<i>STAL</i>	<i>STAL-MSE</i>
DBe-MosheWalk	0.883	0.868
DBe-DariaWalk	0.917	0.928
DBe-Limp	0.889	0.823
DBd-Pedestrians	0.933	0.872
DBd-BackDoor	0.846	0.683
DBa-WaterSurface	0.839	0.560
DBd-Fountain02	0.779	0.591
DBd-Overpass	0.742	0.683
DBd-Canoe	0.764	0.308
DBd-Fountain01	0.518	0.347
DBd-Office	0.952	0.929
DBb-LeftBag	0.625	0.541
DBa-Hall	0.561	0.486
DBa-ShoppingMall	0.656	0.592
Average	0.778±0.141	0.665 ± 0.204

Test 2: Comparison against state of the art approaches To include a wide variety of dynamics to test on, 23 video sequences (extracted from the four above described datasets) are so selected to exclude those scenes holding either camera jitter or thermal images. During comparison, videos are ranked in categories depending on the challenging level to discriminate between background and foreground dynamics. Particularly, all video set is split into the following categories: *a*) Foreground and background dynamics are clearly distinguishable *b*) Background dynamics tends to be similar to the foreground dynamics, e.g., moving leaves, water flowing, etc. *c*) On the opposite, foreground dynamics is similar to the background dynamics, e.g., static people. Besides, some video sequences also included different kind of artifacts, e.g., illumination changes, shadows, occlusions, among others. As a benchmark, the following three different state of the art algorithms are employed: Zivkovic Gaussian Model Mixture (Z-GMM) [36], Spatial Coherence Self-Organizing Background Subtraction

(SC-SOBS) [32], and the Pixel Based Adaptive Segmenter (PBAS)[47]. The Z-GMM is an improvement of the traditional Gaussian Model Mixture GMM [35], which is able to automatically tune the number of models needed to describe each pixel. The Z-GMM has been widely included in background subtraction and computer vision reviews [38, 83, 84]. Furthermore, it is included in the Change Detection Challenge 2012 [48].

The SC-SOBS algorithm, which is an enhancement of the SOBS algorithm described in the Experiment 2 of section § 4.4.3, integrates an spatial coherence post-processing stage to provide robustness against false foreground detection. The SC-SOBS algorithm is one of the top state of the art algorithms for background subtraction considering the results of the Change Detection Challenge 2012 [48]. The algorithm is publicly available³ and for our concrete testing all its parameters are left as default.

Lastly, the PBAS algorithm models the background by constructing a codebook with a fixed number of observed values. The foreground/background discrimination is made by comparing the difference between each new input pixel and the corresponding values in the codebook against a given threshold, if such threshold is exceeded for a minimum number of times, then it is labeled as background, otherwise it is labeled as foreground. The codebook updating is made by considering a learning rate parameter which is tuned by analyzing each pixel dynamics. Neighbor pixels are also updated in order to vanish false detected foreground regions. The PBAS is also considered by the Change Detection Challenge 2012 [48] as a top state of the art algorithm.

For concrete implementation of the Z-GMM and the PBAS algorithms, the BGS library [85] is employed. This library provides a C++ framework to perform 34 background subtraction techniques and is publicly available⁴. Parameters of the Z-GMM and PBAS were left as default. All obtained results are compared by using the $F1$ measure described in section § 4.4.3

Table 6-3 shows the estimated F_1 values for the studied videos. In case of the category a , the four algorithms attain high $F1$ measures, however, the PBAS algorithm attains in average the best results. Figures 6.13b- 6.13e reveal that all the algorithms are able to infer the main pixel dynamics of DBd-Pedestrians video, since this video sequence do not present a complex challenge for the discrimination between foreground and background elements. Nonetheless, in case of more complex dynamics as presented in DBd-Backdoor video, where a background illumination change occurs at the middle of the record, the Z-GMM and the SC-SOBS generate noisy segmentation results as seen in Figure 6.13i and 6.13j respectively. The above can be explained since the Z-GMM is not able to respond to fast illumination changes, and the SC-SOBS algorithm does not update its model for dynamics that have not been present in the initial learning stage, therefore the false segmented area is going to be present along the whole video sequence. In contrast, both STAL and PBAS are able to learn the new dynamics of the non-stationary illumination change (see Figures 6.13h and 6.13k

³<http://www.na.icar.cnr.it/~maddalena.l/>

⁴<https://code.google.com/p/bgslibrary/>

respectively).

For category *b*, the STAL and SC-SOBS approaches obtained in average the best results. However, as seen in Table 6-3, STAL attained more stable results than SC-SOBS according to the average performance standard deviation. As case of interest, it can be noticed from Figures 6.13n, 6.13o and 6.13q that there are some false segmentation results for DBd-Overpass video. In the case of STAL it can be explained since when foreground objects stay in front of a highly dynamical background region, STAL algorithm highlights spatio-temporal pixel relations given by the object motion analysis stage (see Figure 6.13m). Hence, in some cases, it can lead noisy segmentation inside the blobs. For Z-GMM the problem appears when a foreground object stays in the same location for a long time, therefore, given that the algorithm generates new models if any of the existing models matches the new input pixel, the foreground object is slowly incorporated into the model. A similar problem appears for the PBAS algorithm, since its property to remove false segmented foreground regions has the side effect of incorporating foreground values into the codebook if they stay static for a long period of time. Now, for DBd-Fountain02 video (see Figure 6.13s), we aim to highlight that proposed STAL needs a minimum time period to properly model the foreground objects to track. Then, some information about the object model is lost until it has completely entered into the scene (Figure 6.13t). As a final remark regarding category *b* videos, if background presents highly dynamical regions that can be taken for foreground elements, then the Z-GMM and SC-SOBS based video segmentations are biased, as shown in Figures 6.13aa and 6.13ab for DBd-Fountain01 video. In contrast, STAL and PBAS are able to significantly avoid such confusion, given that both take into account the pixel variability into their updating rules (see Figures 6.13z and 6.13ac). Particularly, STAL object motion analysis stage is able to determine whether an object is considered as foreground if it fulfills the ratio object detection condition described in section § 6.1.2, allowing the Conrrentropy-based cost function to include highly dynamical background information. Hence, a proper discrimination between foreground and background is attained using STAL.

In case of category *c*, STAL outperforms the other three algorithms in most of the cases. Table 6-3 shows that STAL attains the highest F1 values in average with a low standard deviation. It is important to remind that category *c* is challenging because foreground dynamics are similar to the background ones. Particularly, for DBd-CopyMachine and DBb-LeftBag videos the challenge is on the tracking of foreground objects that become static for a long period of time, exhibiting similar behavior to background dynamics. The proposed STAL can successfully track such objects (see Figures 6.13ae and 6.13ak), facilitating the discrimination between foreground and background, as shown in Figures 6.13af and 6.13al, while the other three algorithms have troubles discriminating between them (see Figures 6.13ag-6.13ai, and 6.13am-6.13ao). Now, the challenge of the DBd-StreetLight video, is the segmentation of very small foreground objects as it can be appreciated in the ground-truth image (Figure 6.13av). The SC-SOBS ist the only algorithm that is able to successfully identified such objects. Finally, when a foreground object remains static for a while at the

beginning of the recording and then starts moving (e.g. DBd-WinterDriveway and DBa-ShoppingMall), the ZGMM and the PBAS algorithms are not able to quickly include such non-stationary change into the background model, leading to false segmentations (see Figures 6.13ay, 6.13ba and 6.13be, 6.13bg). The same problem appears for SC-SOBS, although it is even worst since such uncovered area will not be incorporated into the background model (see Figures 6.13az and 6.13bf). In contrast, due to the capability of STAL to detect moving objects as described in section § 6.1.2, an object that starts moving is properly identified regardless the time that stayed static as seen in Figure 6.13aw. Moreover, STAL Correntropy-based cost function is able to adapt the model to the object absence, attaining a proper segmentation (see Figures 6.13ax and 6.13bd). Overall, attained results show that the coupling of Correntropy-based background modeling and object motion analysis stages allows to discriminate between foreground and background dynamics by our proposed STAL approach.

Table 6-3.: Estimated pixel-based F_1 measure for STAL, Z-GMM, SC-SOBS and PBAS approaches.

	<i>Video sequence</i>	<i>STAL</i>	<i>Z-GMM</i>	<i>SC-SOBS</i>	<i>PBAS</i>
Category a	DBe-MosheWalk	0.883	0.864	0.728	0.914
	DBe-DariaWalk	0.917	0.883	0.921	0.876
	DBe-Limp	0.889	0.819	0.833	0.897
	DBe-KneesUp	0.837	0.784	0.849	0.911
	DBd-Pedestrians	0.933	0.872	0.949	0.926
	DBd-BackDoor	0.846	0.883	0.840	0.957
	Average	0.884 ± 0.035	0.832 ± 0.072	0.853 ± 0.071	0.913 ± 0.027
Category b	DBa-WaterSurface	0.839	0.654	0.856	0.741
	DBd-Fountain02	0.779	0.782	0.886	0.904
	DBd-Overpass	0.742	0.392	0.883	0.389
	DBd-Boats	0.718	0.325	0.895	0.544
	DBd-Canoe	0.764	0.373	0.952	0.399
	DBd-Fall	0.425	0.384	0.277	0.844
	DBd-Fountain01	0.518	0.292	0.116	0.674
	Average	0.684 ± 0.141	0.468 ± 0.186	0.695 ± 0.319	0.642 ± 0.256
Category c	DBd-Office	0.952	0.253	0.970	0.379
	DBd-CopyMachine	0.890	0.425	0.571	0.554
	DBb-LeftBag	0.625	0.715	0.566	0.557
	DBd-StreetLight	0.616	0.191	0.972	0.413
	DBd-TramStop	0.652	0.463	0.841	0.532
	DBd-WinterDriveway	0.660	0.405	0.125	0.627
	DBd-PETS2006	0.896	0.578	0.868	0.625
	DBa-Hall	0.561	0.615	0.534	0.648
	DBa-ShoppingMall	0.656	0.627	0.584	0.653
	DBa-Bootstrap	0.497	0.557	0.326	0.444
	Average	0.701 ± 0.147	0.523 ± 0.172	0.636 ± 0.265	0.567 ± 0.118
Total average	0.743 ± 0.151	0.606 ± 0.221	0.711 ± 0.266	0.708 ± 0.216	

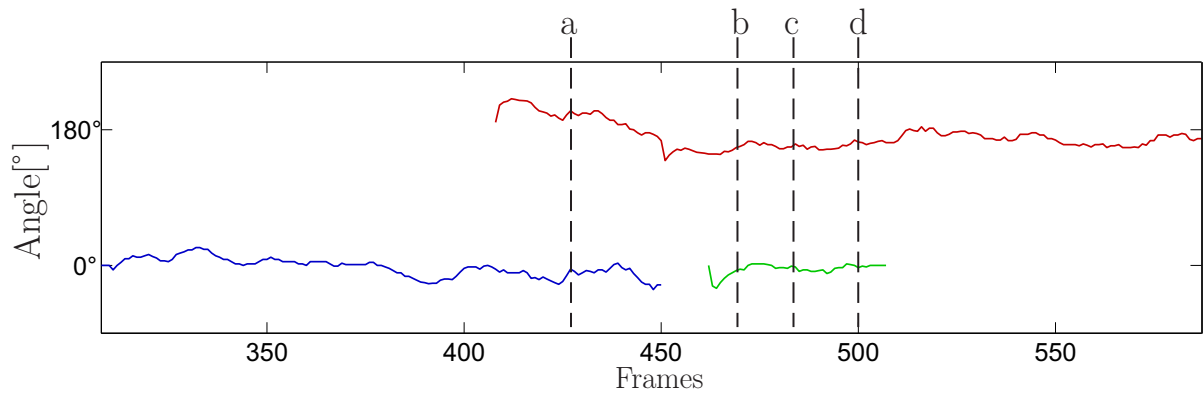


a Frame 427th

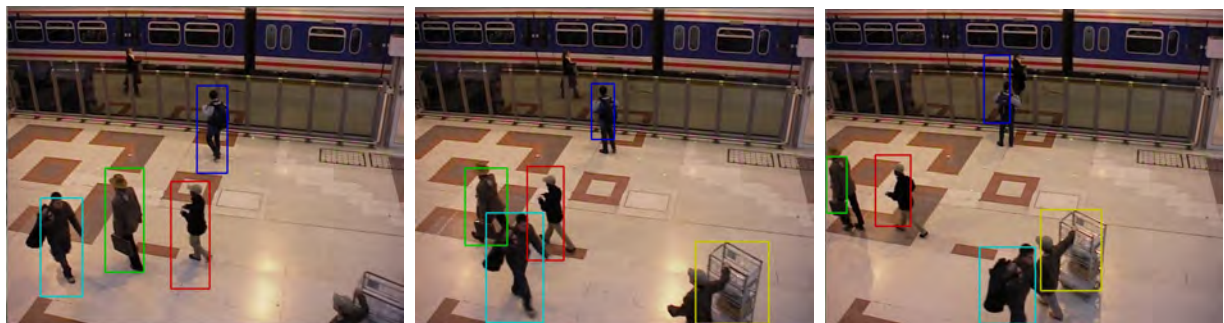
b Frame 468th

c Frame 482th

d Frame 500th



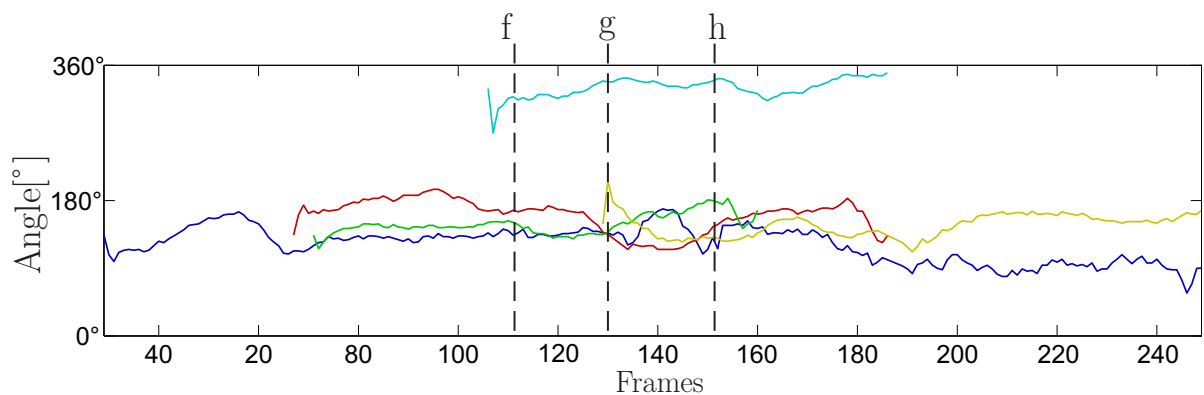
e Object trajectory angle for DBd-Pedestrians video.



f Frame 112th

g Frame 130th

h Frame 152th



i Object trajectory angle for DBd-PETS2006 video.

Figure 6-10.: Object motion analysis performance against occlusions/crossing artifacts.

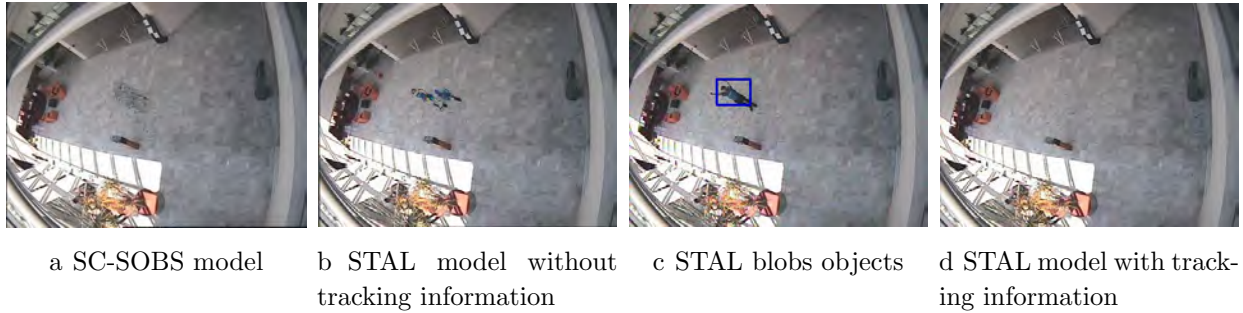


Figure 6-11.: DBb-LeftBox frame 297. Dealing with bootstrapping and static objects

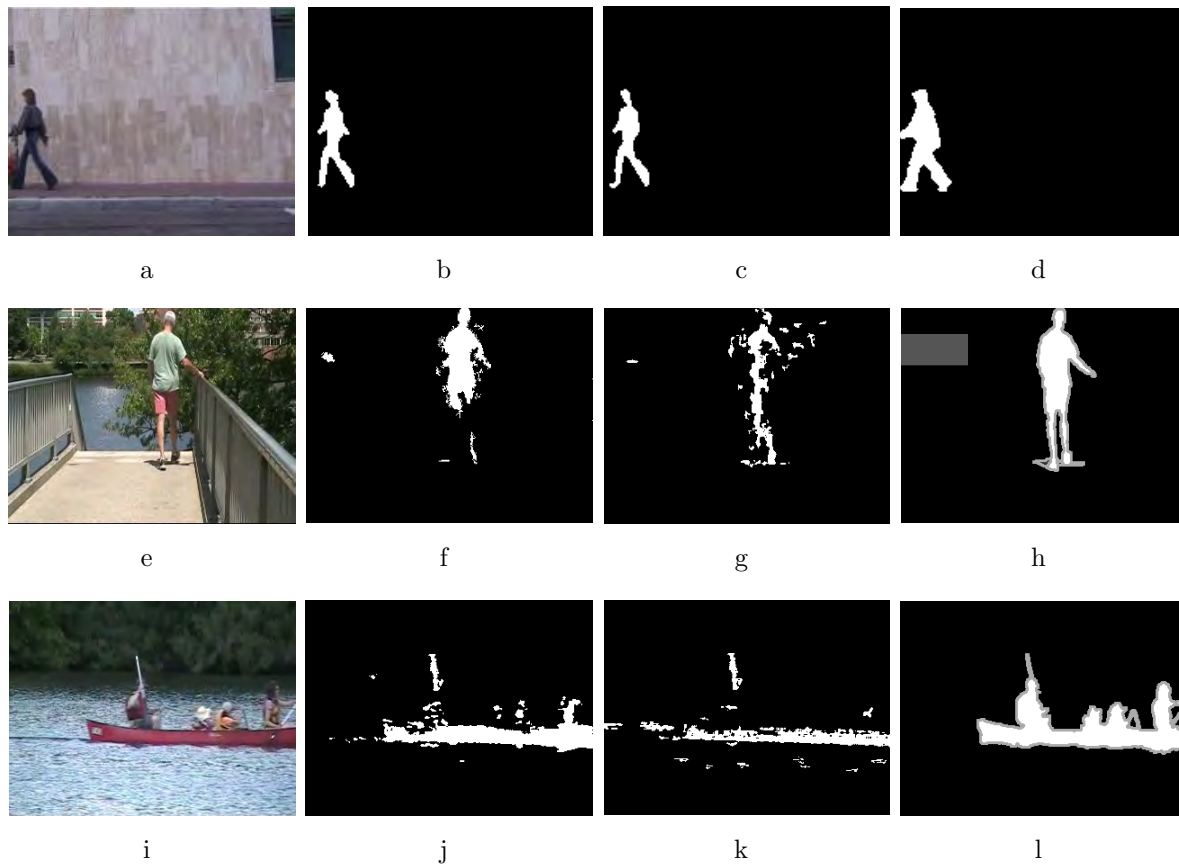
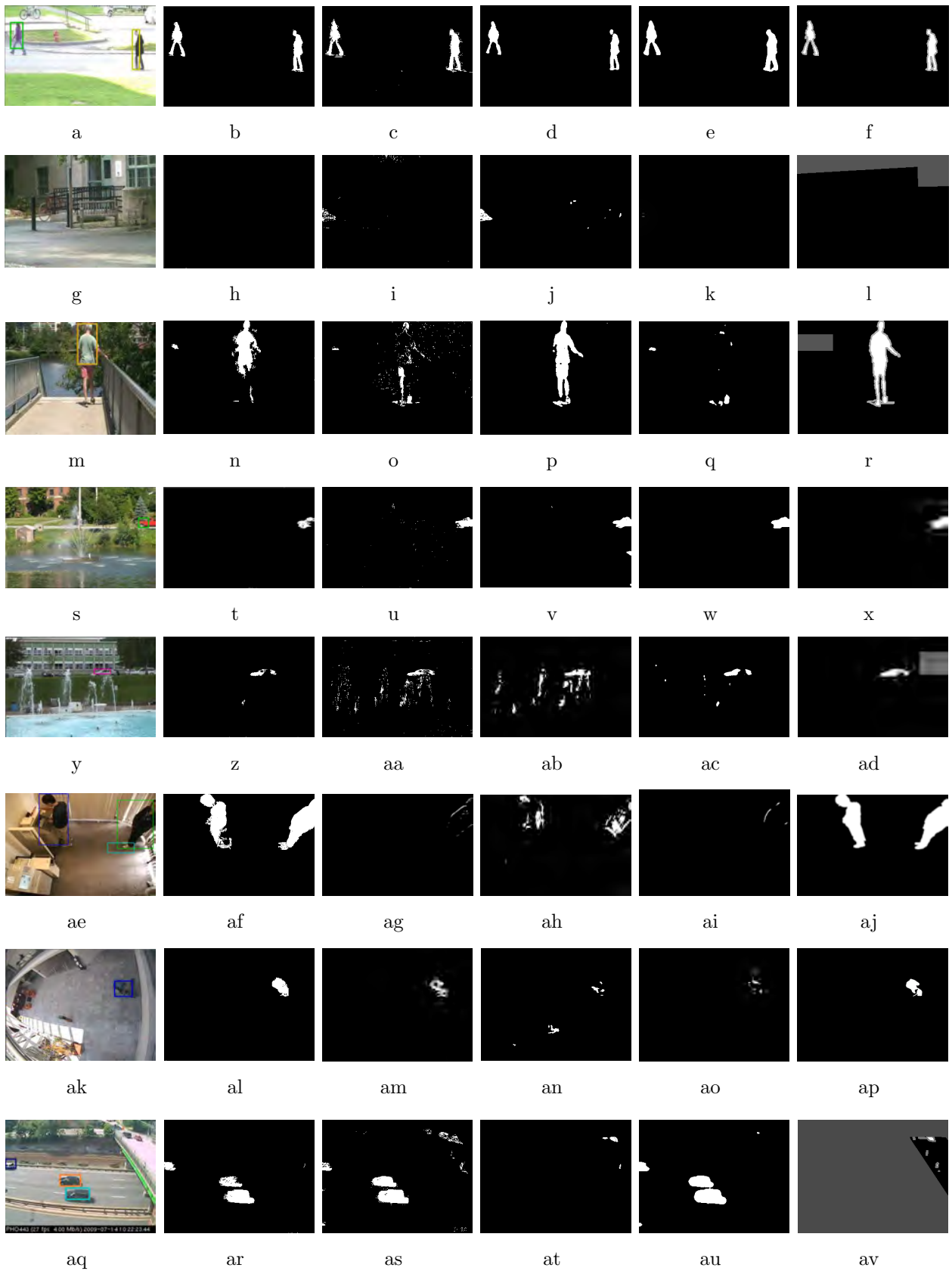


Figure 6-12.: Segmentation performance for different pixel dynamics. Column 1: Original frame. Column 2: STAL segmentation. Column 3: STAL-MSE segmentation. Column 4: Ground-truth.



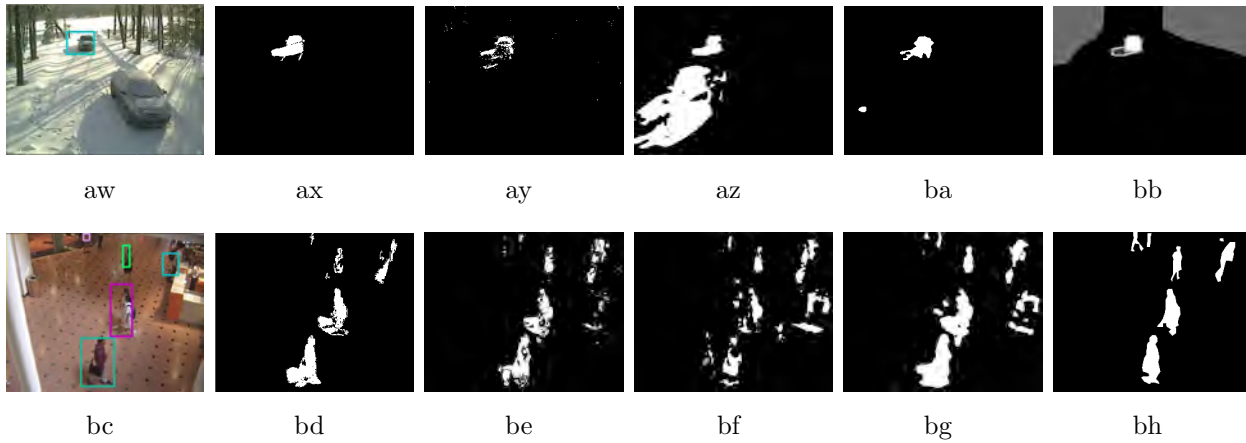


Figure 6-13.: Segmentation performance over challenging conditions. Column 1: Original frame with STAL tracking blobs. Column 2: STAL segmentation. Column 3: Z-GMM segmentation. Column 4: SC-SOBS segmentation. Column 5: PBAS segmentation. Column 6: Ground-truth.

Main STAL free parameter analysis: The main parameters of each STAL stage are studied to make clear their contribution to the final background and foreground discrimination performance. In this sense, the F1 average measure is calculated for each database category (a , b , and c) in order to make clear influence of studied STAL parameters against different background and foreground dynamics. First, for the Correntropy-based cost function in the adaptive learning stage (section § 6.1.1), the time window size T_e is ranged within values $\{5, 15, 25, 35, 50\}$.

It can be seen in Figure 6-14 that attained segmentation results of category a are quite stable for all values of T_e ; obtained results can be explained by the fact that those videos include simple and almost stationary dynamics. Nevertheless, for categories b and c , small T_e values decrease the final STAL segmentation, namely, proposed Correntropy-based cost function with automatic bandwidth kernel selection has not enough information to learn pixel dynamics. As the T_e value increases, proposed approach can properly infer pixel changes along time improving the attained $F1$ measure, however, if recorded environment presents non-stationary foreground dynamics (category c), high T_e values do not allow to rightly describe them. Overall, $T_e = 25$ seems to be an adequate time window size value.

As regards the object motion analysis stage (section § 6.1.2), we study the number of particles N_p while ranging within the value set $\{10, 50, 100, 200, 400, 600, 800\}$. As shown in Figure 6-15, a low number of particles decreases assessed object tracking accuracy, which affects directly the final segmentation results. Looking for a trade-off between video segmentation performance and low computational burden, $N_p = 400$ is an adequate value. Finally, STAL based video segmentation is tested varying the threshold segmentation parameter

ζ_S from the set $\{0.05, 0.2, 0.4, 0.6, 0.8, 0.95\}$. Particularly, for category *a*, the higher ζ_S the better *F1* average. On the other hand, a ζ_S value from 0.6 to 0.8 is suitable for complex background and foreground dynamics (categories *b* and *c*), as it can be seen in Figure 6.2.3. Indeed, low ζ_S values tend to estimate under-segmented results, while high ζ_S values attain over-segmented outcomes.

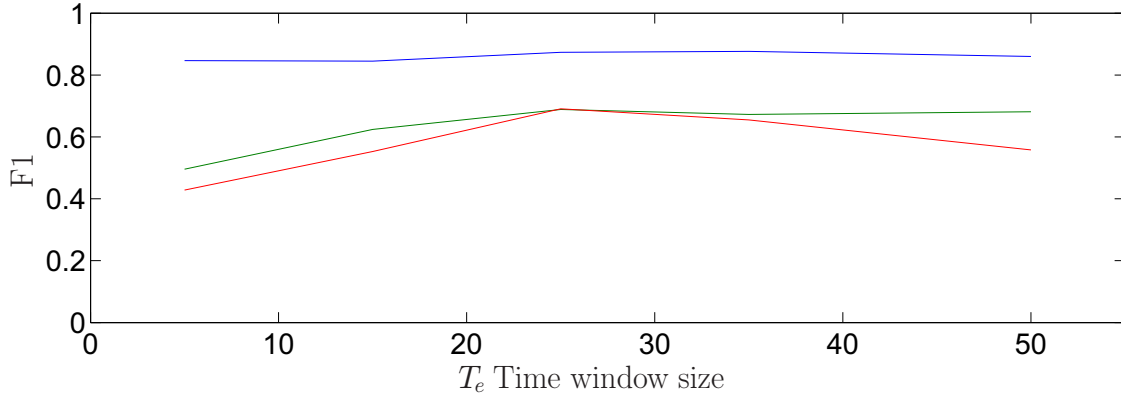


Figure 6-14.: Average F1 measure per video category: — i), — ii), — iii) varying the time window size

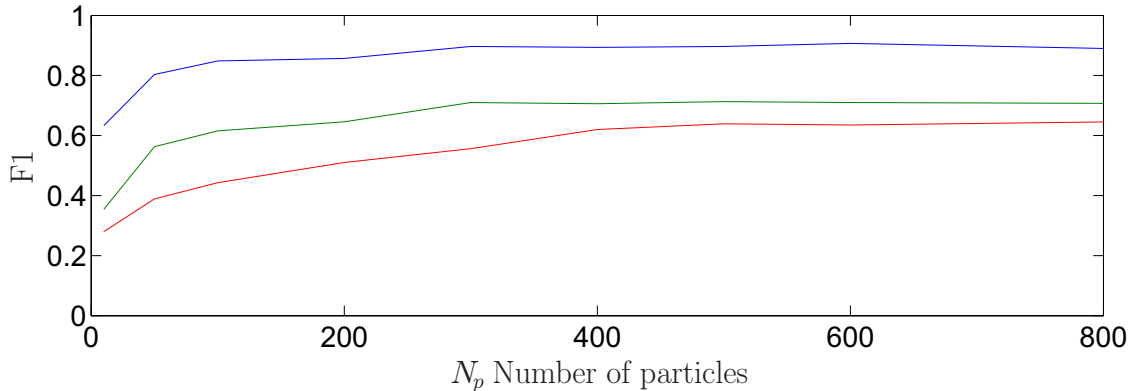


Figure 6-15.: Average F1 measure per video category: — i), — ii), — iii) varying the number of particles

STAL computational burden analysis: Computational cost is calculated for a concrete prototype development of STAL using a single-thread C++ implementation on an Intel Xeon 3.4 GHz processor. Table 6-4 shows the average FPS for the 23 videos and their respective resolution. It can be seen that the computational cost of STAL highly depends on the image resolution, since for the highest resolution videos DBd-Fall, DBd-CopyMachine and DBd-PETS2006, the lowest FPS is attained, otherwise, the highest FPS is obtained for the lowest

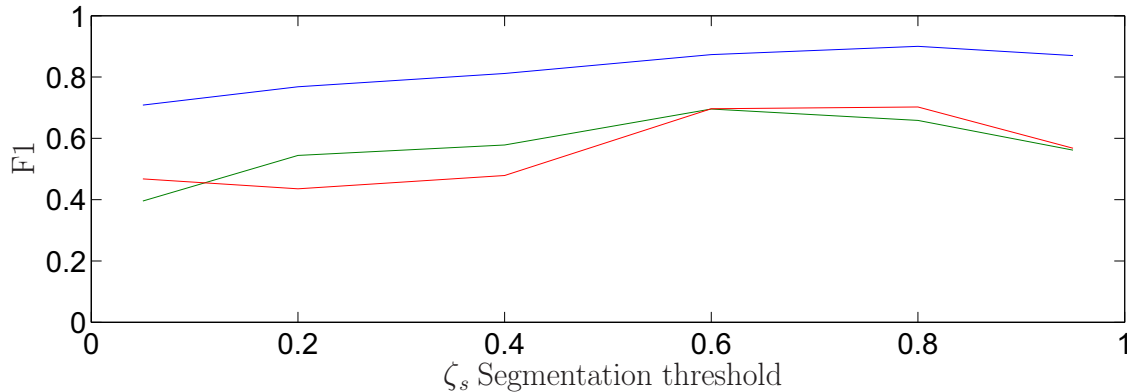


Figure 6-16.: Average F1 measure per video category: — i), — ii), — iii) varying the segmentation threshold

resolution video DBa-WaterSurface. It also can be appreciated that for videos with simple background and foreground dynamics such as DBe-MosheWalk, DBe-DariaWalk, the FPS is high. However, for DBd-Canoe, DBd-Fountain01, DBa-Bootstrap, DBa-ShoppingMall videos, when the relationship between foreground and background dynamics become more complex, the FPS decreases significantly, due to the high computational load over the object motion analysis stage. In general, the proposed STAL exhibits an affordable computational cost for medium resolution videos (aprox. 320×240). Some STAL based video segmentation results are online available⁵.

6.3. Conclusions

A methodology for background and foreground discrimination in video surveillance scenarios is proposed. Our approach, called STAL analyzes pixel spatio-temporal relations as an adaptive learning algorithm. STAL infers the temporal statistical distribution of each pixel by using a Correntropy-based cost function, which is able to weight the relevance of each new input pixel to build a background model. Moreover, an automatic tuning algorithm is proposed in order to suitably set the Correntropy kernel bandwidth by analysing the error distribution shape, achieving good performance under both Gaussian and non-Gaussian pixel distribution conditions. By including an object motion analysis stage, proposed STAL is able to properly detect, model, and track foreground objects. Such information is weighted afterwards into the background model, avoiding false segmentations when working with the presence of moving and static foreground objects in the scene. Besides, STAL object motion analysis stage is able to give extra information about the foreground object dynamics which could be further used to solve other video surveillance tasks.

Attained results expose that proposed STAL is able to handle well known issues from

⁵<https://vimeo.com/channels/stalsm>

Table 6-4.: Estimated computational cost of the proposed STAL approach.

Video	FPS	Resolution
DBe-MosheWalk	48.65	180 × 144
DBe-DariaWalk	46.75	180 × 144
DBe-Limp	39.32	180 × 144
DBe-KneesUp	40.64	180 × 144
DBd-Pedestrians	17.18	360 × 240
DBd-BackDoor	18.46	320 × 240
DBa-WaterSurface	51.83	160 × 128
DBd-Fountain02	15.72	432 × 288
DBd-Overpass	19.32	320 × 240
DBd-Boats	14.73	320 × 240
DBd-Canoe	4.78	320 × 240
DBd-Fall	2.29	720 × 480
DBd-Fountain01	8.91	432 × 288
DBd-Office	13.06	360 × 240
DBd-CopyMachine	3.15	720 × 480
DBb-LeftBag	13.37	388 × 284
DBd-StreetLight	11.77	320 × 240
DBd-TramStop	9.29	432 × 288
DBd-WinterDriveway	18.62	320 × 240
DBd-PETS2006	4.76	720 × 576
DBa-Hall	25.05	176 × 144
DBa-ShoppingMall	9.33	320 × 256
DBa-Bootstrap	11.09	160 × 120

video surveillance systems, furthermore, achieved performance is comparable with state of the art algorithms, obtaining even better results when the challenge of the segmentation relies on the complexity of the relations between foreground and background dynamics. Moreover, when comparing the segmentation results against the ones based on the MSE, it was shown that the correntropy cost function is able to properly model the background dynamics. Main STAL free parameters are experimentally studied, to validate algorithm's performance and stability under different video conditions. Proposed approach, is a suitable alternative to support video surveillance tasks achieving good segmentation results, while having an affordable computational burden.

As future work, it would be interesting to consider more robust approaches to model and track the detected foreground objects. Moreover, alternatives to incorporate the spatial information into the background model could be also studied. Finally, an STAL extension should be developed to deal with moving cameras and an optimized implementation will be carried out in order to improve STAL scalability.

Part III.

Conclusions and Future Work

7. Conclusions

In the present work, we have studied the use of kernel representations to support the analysis of image and video data by revealing the intrinsic non-linear data relationships along space and time. In chapter 4, we have presented a short review of the basics of kernel representations and how the inclusion of multiple information sources is possible by means of a convex combination of basis kernels. Afterwards, a methodology for the automatic tuning of the relevance weights of each information source based on data variability was proposed. Taking the above into account, two different computer vision methodologies were proposed. The first one called WGKIS, is an image segmentation framework, which allows to incorporate different color representations and some spatial information to enhance the data separability, facilitating the work of the further spectral clustering-based approach, which provides the segmented image. The second proposed framework, called WGKVS, using a video sequence recorded by an static camera, performs a video segmentation while comparing against a Gaussian-based background model. Kernel similarity measures based on color representations and some spatial information are included into the conformation of an enhanced feature representation space, which afterwards is used by a tuned kmeans algorithm to perform the discrimination between foreground and background elements. Attained results for both WGKIS and WGKVS methodologies, revealed that the incorporation of multiple image information sources, improves the accuracy of the resulting segmentations. Additionally, it was shown that the performance of both frameworks is comparable against state of the art algorithms. As a final remark regarding to WGKVS experiments, it was demonstrated that it could be coupled in a full computer vision task such as the detection of abandoned objects in scenarios with mainly stationary conditions.

In order to extend proposed WGKVS to non-stationary scene conditions and to make it possible to work with more complex foreground dynamics, a methodology which highlights spatial relationships based on optical flow was proposed in chapter 5. Proposed methodology is able to identify the motion direction of foreground objects, avoiding false segmentation results with the presence of static foreground objects in the scene, which is a common drawback of traditional background subtraction techniques. Besides, a variability-based updating framework was developed to adapt the Gaussian model parameters against non-stationary conditions such as illumination changes or intrinsic scenario conditions. Performed experiments showed that proposed optical flow-based framework is able to identify the motion direction trend of moving objects and static foreground objects. The supervised measures results demonstrated that proposed methodology outperforms the state of the art SOBS

algorithm when employing few frames for the initialization process.

After checking the effectiveness of the optical flow-based methodology to highlight spatial relationships, we presented in chapter 6 an adaptive learning framework to background modeling called STAL, which using spatio-temporal kernel based representations supports the video segmentation task. Proposed STAL uses a Correntropy-based cost function to learn each pixel dynamics and with such information the Gaussian-based background model parameters are updated. This updating methodology allows to model either mostly static and highly dynamical scenarios unlike the variability-based updating criterion proposed in chapter 5. Furthermore, based on the operation of the optical flow-based methodology to highlight spatial relationships explained in chapter 5, an object motion analysis stage is developed to fix the learning rate of the cost function. The segmentation process is made by comparing the difference between each input pixel and the corresponding background model against a threshold. Several experiments were performed to expose the particular results of each stage of STAL. Firstly, obtained results expose that the Correntropy-based cost function is able to properly learn the pixel dynamics of static and dynamical scenarios, furthermore, is able to weight the relevance of a new input pixel given the previously observed evolution. Regarding to the object motion analysis stage results, it was demonstrated that proposed methodology is able to identify moving objects in the scene regardless the nature of the recorded scenario. Besides, it was shown that the inclusion of both color and motion direction trend information, make possible to track moving foreground objects in complex conditions such as crossing, occlusions and static objects. It is also important to remark that the information given by the object motion analysis stage could be used to support other computer vision tasks (i.e. object classification, object path tracking, activity recognition). From the segmentation results, it was shown that proposed STAL in average outperforms the three studied state of the art algorithms and that is capable to obtain accurate results for most of the studied videos, demonstrating that overall it can handle the common video surveillance conditions described along the presented work. An additional experiment was performed to analyze the performance of STAL when varying the value of its most important parameters, attained results exposed that parameter selection of STAL alter the segmentation accuracy of mostly the videos with the most complex dynamics. Finally, the computational burden results exhibit that STAL could be used to support real-time applications for medium resolution videos.

7.1. Academic Discussion

- S. Molina-Giraldo, J. Valencia-Aguirre, A. M. Álvarez-Meza, C. D. Acosta-Medina, and G. Castellanos-Domínguez. Dimensionality Reduction Methods to Support Motion Analysis in Computer Vision Systems. In *16th Simposio de Tratamiento de señales, Imágenes y Visión Artificial - STSIVA*, 2011.
- S. Molina-Giraldo, J. Valencia-Aguirre, A. M. Álvarez-Meza, C. D. Acosta-Medina, and G. Castellanos-Domínguez. Medical Information Analysis by Nonlinear Dimensionality Reduction Incorporating Prior Knowledge. In *7th International Seminar on Medical Information Processing and Analysis - SIPAIM 2011*.
- S. Molina-Giraldo, A. M. Álvarez-Meza, D. H. Peluffo-Ordoez, and G. Castellanos-Domínguez. Image Segmentation based on Multi-Kernel Learning and Feature Relevance Analysis. In *13th Ibero-American Conference on Artificial Intelligence - IBERAMIA*, 2012.
- D. Ramirez-Giraldo, S. Molina-Giraldo, A. M. Álvarez-Meza, G. Daza-Santacoloma, and G. Castellanos-Domínguez. Kernel Based Hand Gesture Recognition Using Kinect Sensor. In *17th Simposio de Tratamiento de señales, Imágenes y Visión Artificial - STSIVA*, 2012.
- S. Molina-Giraldo, J. Carvajal-González, A. M. Álvarez-Meza, and G. Castellanos-Domínguez. Video Segmentation based on Multi-Kernel Learning and Feature Relevance Analysis for Object Classification. In *3rd International Conference on Pattern Recognition and Methods - ICPRAM*, 2013.
- S. Molina-Giraldo, A. M. Álvarez-Meza, J. C. García-Álvarez, and G. Castellanos-Domínguez. Video Segmentation Framework by Dynamic Background Modelling. In *17th International Conference on Analysis and Processing - ICIAP*, 2013.
- A. M. Álvarez-Meza, S. Molina-Giraldo, and G. Castellanos-Domínguez. Correntropy-based Adaptive Learning to Support Video Surveillance Systems. In *22nd International Conference on Pattern Recognition - ICPR*, 2014. (*Accepted*)
- A. M. Álvarez-Meza, S. Molina-Giraldo, and G. Castellanos-Domínguez. Spatio-Temporal Adaptive Learning for Non-Stationary Video-based Surveillance Analysis. *International Journal of Computer Vision*, Springer, 2014. (*Under review*)

8. Future work

From the attained results and the drawbacks found along the process, the following theoretical and experimental topics could be explored:

- Regarding to the proposed WGKIS framework, a free parameter optimization strategy should be studied in order to automatically tune the Gaussian kernel bandwidth. Furthermore, due to computational burden a more efficient calculation of each kernel matrix should be explored. Besides, for a real-time application, it would be desirable to implement the proposed WGKIS using real-time programming languages with a multithread paradigm.
- The information given by the optical flow computation could be used to develop a camera motion compensation approach, in order to allow STAL working under camera jitter conditions.
- Aiming to enhance the object motion analysis stage of proposed STAL, a more robust moving object modeling should be studied in order to improve the tracking performance under difficult foreground objects rotation and scale changing conditions.
- Alternatives to incorporate the spatial information into the learning rate of the background model updating should be studied in order to avoid noisy segmentations when foreground objects stay in front highly dynamical background areas.
- The segmentation and tracking information provided by STAL algorithm should be used to support a complete computer vision system.
- A post-processing stage, such as a median filter, should be implemented aiming to enhance STAL attained segmented regions shapes.
- In regard of computational burden, a more efficient implementation could be developed. Moreover, the usage of a multithread paradigm will be desirable in order to improve STAL scalability.

Part IV.

Appendix

A. MSE based cost function for background modeling

The LMS algorithm, is an adaptive filtering approach, which aims to attain the minima of the squared error defined as the difference between the signal produced by the algorithm and the desired signal. Regarding to background modeling, the LMS algorithm updating rule can be formulated as:

$$\mu_t = \mu_{t-1} - \eta \nabla \xi_{\mu_t}, \quad (\text{A-1})$$

where μ_t is the mean model parameter, η is the updating step, ∇ is the gradient operator and ξ_{μ_t} is the MSE. The negative sign indicates that we need to update μ_t towards an opposite direction of the MSE gradient slope. Now, in this case, the MSE can be considered as:

$$\xi_{\mu_t} = \mathbb{E} \{ (x_n - \mu_t)^2 : n = t - T, \dots, t \}, \quad (\text{A-2})$$

where T corresponds to a window of frames to evaluate the expected value. Taking derivatives, the updating rule from Eq. A-1 can be reformulated as:

$$\mu_t = \mu_{t-1} + \frac{2\eta}{T} \sum_{n=t-T}^t x_n - \mu_{t-1}, \quad (\text{A-3})$$

The convergence of the LMS algorithm highly depends on the updating step size η , if it is chosen too big, the algorithm could reach an oscillating behavior, otherwise, if it is chosen too small, the algorithm would take too long to reach the optimum. In this sense, the updating step is bounded as in [86]:

$$0 < \eta < \frac{2}{\text{tr}(\Sigma)}, \quad (\text{A-4})$$

being Σ a covariance matrix computed for x_t over a time window T . For concrete testing, the updating step was set as $1/\text{tr}(\Sigma)$ and was computed along sliding windows. Moreover, the model was initialized as the expected value computed over the first T frames.

Bibliography

- [1] Oliver Faugeras. *Three dimensional computer vision: A geometric viewpoint*. the MIT Press, 1993.
- [2] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.
- [3] Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6): 1709–1724, 2011.
- [4] Shireen Y Elhabian, Khaled M El-Sayed, and Sumaya H Ahmed. Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32–54, 2008.
- [5] D Cardenas-Pena, JD Martinez-Vargas, and G Castellanos-Dominguez. Local binary fitting energy solution by graph cuts for mri segmentation. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5131–5134. IEEE, 2013.
- [6] TJ Ramirez-Rozo, JC Garcia-Alvarez, and CG Castellanos-Dominguez. Infrared thermal image segmentation using expectation-maximization-based clustering. In *Image, Signal Processing, and Artificial Vision (STSIVA), 2012 XVII Symposium of*, pages 223–226. IEEE, 2012.
- [7] Benhur Ortiz-Jaramillo, HA Fandiño Toro, Hernan Darío Benitez-Restrepo, Sergio Alejandro Orjuela-Vargas, German Castellanos-Domínguez, and Wilfried Philips. Multi-resolution analysis for region of interest extraction in thermographic nondestructive evaluation. In *IS&T/SPIE Electronic Imaging*, pages 82951J–82951J. International Society for Optics and Photonics, 2012.
- [8] Santiago Molina-Giraldo, Andres M Álvarez-Meza, Julio C García-Álvarez, and Cesar G Castellanos-Domínguez. Video segmentation framework by dynamic background modelling. In *Image Analysis and Processing-ICIAP 2013*, pages 843–852. Springer, 2013.

-
- [9] Santiago Molina-Giraldo, Johanna Carvajal González, Andrés Marino Álvarez-Meza, and Germán Castellanos-Domínguez. Video segmentation based on multi-kernel learning and feature relevance analysis for object classification. In *ICPRAM*, pages 396–401, 2013.
- [10] Karin Sobottka and Ioannis Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal processing: Image communication*, 12(3): 263–281, 1998.
- [11] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.
- [12] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 679–698, 1986.
- [13] Thomas Leung and Jitendra Malik. Contour continuity in region based image segmentation. In *Computer Vision?ECCV’98*, pages 544–559. Springer, 1998.
- [14] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 833–840, 2009. doi: 10.1109/ICCV.2009.5459242.
- [15] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.
- [16] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *Neural Networks, IEEE Transactions on*, 14(1):117–126, 2003. ISSN 1045-9227. doi:10.1109/TNN.2002.806629.
- [17] Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features: the kernel recipe. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 257–264. IEEE, 2003.
- [18] S. Molina-Giraldo, A.M. Álvarez-Meza, D.H. Peluffo-Ordonez, and G. Castellanos-Domínguez. Image segmentation based on multi-kernel learning and feature relevance analysis. In *Advances in Artificial Intelligence-IBERAMIA 2012*, volume 7637 of *Lecture Notes in Computer Science*, pages 501–510. Springer Berlin Heidelberg, 2012.
- [19] D. Wang. Unsupervised video segmentation based on watersheds and temporal tracking. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):539–546, 1998.
- [20] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys (CSUR)*, 17(1):75–145, 1985.

-
- [21] E. Ozyildiz, N. Krahnstöver, and R. Sharma. Adaptive texture and color segmentation for tracking moving objects. *Pattern recognition*, 35(10):2013–2029, 2002.
- [22] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [23] Chris A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical models and image processing*, 55(6):532–537, 1993.
- [24] Orlando José Tobias and Rui Seara. Image segmentation by histogram thresholding using fuzzy sets. *Image Processing, IEEE Transactions on*, 11(12):1457–1465, 2002.
- [25] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.
- [26] C. Jung, LC Jiao, J. Liu, and Y. Shen. Image segmentation via manifold spectral clustering. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–6. IEEE, 2011.
- [27] Alan M McIvor. Background subtraction techniques. *Proc. of Image and Vision Computing*, 1(3):155–163, 2000.
- [28] T.D. Raty. Survey on contemporary remote surveillance systems for public safety. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(5):493–515, 2010.
- [29] Thierry Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey. *RPCS*, 4(3):147–176, 2011.
- [30] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4, Oct 2004. doi: 10.1109/ICSMC.2004.1400815.
- [31] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261 vol.1, 1999. doi: 10.1109/ICCV.1999.791228.
- [32] L. Maddalena and A. Petrosino. The sobs algorithm: What are the limits? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26, 2012. doi: 10.1109/CVPRW.2012.6238922.

-
- [33] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.
- [34] Alexandre RJ François and Gérard G Medioni. Adaptive color background modeling for real-time segmentation of video streams. In *Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pages 227–232, 1999.
- [35] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.
- [36] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.
- [37] Yu-Ting Chen, Chu-Song Chen, Chun-Rong Huang, and Yi-Ping Hung. Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40(10):2706–2715, 2007.
- [38] Thierry Bouwmans, Fida El Baf, Bertrand Vachon, et al. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.
- [39] Anuj Srivastava, Ann B Lee, Eero P Simoncelli, and S-C Zhu. On advances in statistical modeling of natural images. *Journal of mathematical imaging and vision*, 18(1):17–33, 2003.
- [40] Alireza Tavakkoli, Mircea Nicolescu, and George Bebis. Automatic statistical object detection for visual surveillance. In *proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 144–148, 2006.
- [41] Peng Tang, Lin Gao, and Zhifang Liu. Salient moving object detection using stochastic approach filtering. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 530–535. IEEE, 2007.
- [42] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.
- [43] J Rymel, J Renno, Darrel Greenhill, James Orwell, and Graeme A Jones. Adaptive eigen-backgrounds for object detection. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 3, pages 1847–1850. IEEE, 2004.

-
- [44] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 5, pages 3061–3064. IEEE, 2004.
- [45] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185, 2005.
- [46] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, 17(7):1168–1177, 2008.
- [47] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43. IEEE, 2012.
- [48] Nil Goyette, P Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.
- [49] Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.
- [50] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, USA, 2002.
- [51] Mehmet Gonen and Ethem Alpaydin. Localized multiple kernel regression. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pages 1425–1428, 2010.
- [52] Genaro Daza-Santacoloma, Julián D. Arias-Londoño, Juan I. Godino-Llorente, Nicolás Sáenz-Lechón, Víctor Osma-Ruíz, and Germán Castellanos-Domínguez. Dynamic feature extraction: An application to voice pathology detection. *Intelligent Automation and Soft Computing*, 2009.
- [53] I.S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.
- [54] P. Perona and L. Zelnik-Manor. Self-tuning spectral clustering. *Advances in neural information processing systems*, 17:1601–1608, 2004.

-
- [55] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [56] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):929–944, 2007.
- [57] R. Unnikrishnan, C. Pantofaru, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, page 34, june 2005. doi: 10.1109/CVPR.2005.390.
- [58] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.
- [59] H. Zhang, J.E. Fritts, and S.A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.
- [60] C. Pantofaru and M. Hebert. A comparison of image segmentation algorithms. *Robotics Institute*, page 336, 2005.
- [61] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A.K. Jain. A background model initialization algorithm for video surveillance. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 733–740. IEEE, 2001.
- [62] D. Cuesta-Frau, J.C. Pérez-Cortés, and G. Andreu-García. Clustering of electrocardiograph signals in computer-aided holter analysis. *Computer methods and programs in Biomedicine*, 72(3):179–196, 2003.
- [63] Martin D Levine and Martin D Levine. *Vision in man and machine*, volume 574. McGraw-Hill New York, 1985.
- [64] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.
- [65] Fatih Porikli, Yuri Ivanov, and Tetsuji Haga. Robust abandoned object detection using dual foregrounds. *EURASIP Journal on Advances in Signal Processing*, 2008:30, 2008.

- [66] Johanna Carvajal-González, AndrésM Álvarez-Meza, and German Castellanos-Domínguez. Feature selection by relevance analysis for abandoned object classification. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 837–844. Springer, 2012.
- [67] Robert T Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsing, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. *A system for video surveillance and monitoring*, volume 102. Carnegie Mellon University, the Robotics Institute Pittsburg, 2000.
- [68] S. Vassiliadis, E.A. Hakkennes, J. S S M Wong, and G.G. Pechanek. The sum-absolute-difference motion estimation accelerator. In *Euromicro Conference, 1998. Proceedings. 24th*, volume 2, pages 559–566 vol.2, 1998. doi: 10.1109/EURMIC.1998.708071.
- [69] S. Wong, S. Vassiliadis, and S. Cotofana. A sum of absolute differences implementation in fpga hardware. In *Euromicro Conference, 2002. Proceedings. 28th*, pages 183–188, 2002. doi: 10.1109/EURMIC.2002.1046155.
- [70] Miguel V Correia and Aurélio C Campilho. Real-time implementation of an optical flow algorithm. In *Pattern Recognition, 2002. Proceedings.*, volume 4, pages 247–250. IEEE, 2002.
- [71] Dongxiang Zhou and Hong Zhang. Modified gmm background modeling and optical flow for detection of moving objects. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 3, pages 2224–2229. IEEE, 2005.
- [72] Ernest L Hall, JJ Hwang, and FA Sadjadi. Computer image processing and recognition. In *1980 Los Angeles Technical Symposium*, pages 2–10. International Society for Optics and Photonics, 1980.
- [73] Ignacion Santamaría, Puskal Pokharel, and Jose Principe. Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Trans. on Signal Processing*, 54(6):2187–2197, 2006.
- [74] Songlin Zhao, Badong Chen, and J.C. Principe. An adaptive kernel width update for correntropy. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–5, 2012.
- [75] Weifeng Liu, P.P. Pokharel, and J.C. Principe. The kernel least-mean-square algorithm. *Signal Processing, IEEE Transactions on*, 56(2):543–554, 2008. ISSN 1053-587X. doi: 10.1109/TSP.2007.907881.
- [76] Jyrki Kivinen, Alexander J. Smola, and Robert C. Williamson. Online learning with kernels. *IEEE Trans. on Signal Processing*, 100(10):1–11, 2010.

-
- [77] Weifeng Liu, José C. Príncipe, and Simon Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*. John Wiley & Sons, Inc., 2010.
- [78] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11):5286–5298, November 2007.
- [79] Songlin Zhao, Badong Chen, and J.C. Principe. Kernel adaptive filtering with maximum correntropy criterion. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2012–2017, 2011.
- [80] A. Singh and J.C. Principe. Using correntropy as a cost function in linear adaptive filters. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 2950–2955, 2009.
- [81] Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 627–630. IEEE, 2000.
- [82] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 50:174–188, 2002.
- [83] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.
- [84] Marco Cristani, Michela Farenzena, Domenico Bloisi, and Vittorio Murino. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP Journal on Advances in signal Processing*, 2010:43, 2010.
- [85] Andrews Sobral. BGSLibrary: An opencv c++ background subtraction library. In *IX Workshop de Viso Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun 2013.
- [86] Simon S Haykin and Bernard Widrow. *Least-mean-square adaptive filters*, volume 31. John Wiley & Sons, 2003.