UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE S2IM

Laboratoire Mathématiques, Image et Applications (MIA)

THÈSE présentée par :

Caroline PACHECO DO ESPIRITO SILVA

soutenue le : **10/05/2017**
pour obtenir le grade de : **Docteur de l'université de La Rochelle**
Discipline : **Mathématiques, Image et Applications**

Extraction et sélection de caractéristiques pour la détection d'objets mobiles dans des vidéos

---

JURY :

Jenny BENOIS-PINEAU	Professeure, Univ. de Bordeaux, Président du jury.
Lyudmila MIHAYLOVA	Professeure, Univ. de Sheffield, UK, Rapporteur.
Marc VAN DROOGENBROECK	Professeur, Univ. de Liège (Belgium), Rapporteur.
Jordi GONZÀLEZ	Professeur associé, Univ. Autonome de Barcelone (Espagne), Examinateur.
Thierry BOUWMANS	Maître de conférences (HDR), Univ. de La Rochelle, Co-directeur de thèse.
Carl FRÉLICOT	Professeur, MIA, Univ. de La Rochelle, Directeur de thèse.

Feature extraction and selection for background modeling and foreground detection

Thesis submitted by **Caroline Pacheco do Espirito Silva** at Université de La Rochelle to fulfill the degree of **Doctor in Mathematics and Applications**.

La Rochelle, May 10, 2017

Director	**Pr. Carl Frélicot** Laboratoire Mathématiques, Image et Applications Université de La Rochelle (France)
Co-Director	**Dr. Thierry Bouwmans** Laboratoire Mathématiques, Image et Applications Université de La Rochelle (France)
Thesis committee	**Pr. Jenny Benois-Pineau** Laboratoire Bordelais de Recherche en Informatique Université de Bordeaux (France) **Dr. Jordi Gonzàlez** Centre de Visió per Computador Universitat Autònoma de Barcelona (Spain)
European evaluators	**Pr. Lyudmila Mihaylova** Department of Automatic Control and Systems Engineering University of de Sheffield (UK) **Pr. Marc Van Droogenbroeck** Department of Electrical Engineering and Computer Science Université de Liège (Belgium)

Acknowledgement

First I would like to express my gratitude to my supervisors, Dr. Thierry Bouwmans and Prof. Carl Frélicot for their invaluable guidance and advices. I would also like to take this opportunity to thank the University of La Rochelle for providing me a full Ph.D. scholarship. Thank you to the Computer Vision Centre (CVC) members for having welcomed me (Doctoral stage), especially Dr. Jordi Gonzàlez. Without the above supports, this thesis is impossible.

I would like to thank Prof. Lyudmila Mihaylova and Prof. Marc Van Droogenbroeck for their acceptances to be the reviewers of the this European thesis manuscript and for sharing interesting comments and criticism that helped improve this manuscript.

Sincere thanks to my dissertation examiner, Prof. Jenny Benois-Pineau from Université de Bordeaux for her interest in my work and for making her time available for me.

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

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

Abstract

In recent years, background subtraction has been one of the most active research topics in computer vision due to many potential applications including surveillance devices in public spaces, traffic monitoring and industrial machine vision. Background modeling methods have increased its efficiency for robust modeling of the background enabling the detection of moving objects in any visual scene. Despite several background subtraction and foreground detection approaches have been proposed recently, no traditional algorithm today still seem to be able to simultaneously address all the key challenges of illumination variation, dynamic camera motion, cluttered background and occlusion. This limitation can be attributed to the lack of systematic investigation concerning the role and importance of features within background modeling and foreground detection. In this thesis, we address this issue by proposing a novel and effective method to deal with the background subtraction problems focused on visual features.

Firstly, a comprehensive survey of the main features used in the context of background subtraction is introduced. In addition, the traditional approaches for feature selection including the recent works in this domain are discussed. Secondly, a robust descriptor for background subtraction which is able to describe texture from an image sequence is proposed. The descriptor is less sensitive to noisy pixels and produces a short histogram, while preserving robustness to illumination changes. Moreover, a descriptor for dynamic texture recognition is also proposed. This descriptor extracts not only color information, but also a more detailed information from video sequences.

Finally, we present an ensemble for feature selection approach that is able to select suitable features for each pixel to distinguish the foreground objects from the background ones. Our proposal uses a mechanism to update the relative importance of each feature over time. For this purpose, a heuristic approach is used to reduce the complexity of the background model maintenance while maintaining the robustness of the background model. However, this method only reaches the highest accuracy when the number of features is huge. In addition, each base classifier learns a feature set instead of individual features. To overcome these limitations, we extended our previous approach by proposing a novel methodology for selecting features based on wagging. We also adopted a superpixel-based approach instead of a pixel-level approach. This does not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance of moving objects.

Résumé

Durant ces dernières années, la soustraction de l'arrière-plan a été l'un des sujets de recherche les plus actifs dans la vision par ordinateur en raison des nombreuses applications comme les dispositifs de surveillance dans les espaces publics, la surveillance du trafic et la vision industrielle. Les méthodes de modélisation du fond ont augmenté leur efficacité pour la modélisation robuste de l'arrière-plan permettant la détection d'objets mobiles dans n'importe quelle scène visuelle. Bien que plusieurs approches de soustraction du fond aient été proposées récemment, aucun algorithme traditionnel n'est aujourd'hui capable d'aborder simultanément tous les défis clés du domaine comme les variations lumineuses, les mouvements dynamiques de la caméra, du fond encombré et de l'occlusion. Cette limitation peut être attribuée à l'absence d'une recherche systématique sur le rôle et l'importance des caractéristiques dans la modélisa-tion de l'arrière-plan et la détection de premier plan. Dans cette thèse, nous abordons cette question en proposant une méthode nouvelle et efficace pour traiter les problèmes de soustraction du fond centrés sur les caractéristiques visuelles.

Tout d'abord, une étude exhaustive des principales caractéristiques utilisées dans le contexte de soustraction du fond est présentée. En outre, les approches traditionnelles pour la sélection des caractéristiques, y compris les travaux récents dans ce domaine, sont analysées. Deuxièmement, un descripteur robuste pour la soustraction d'arrière-plan qui est capable de décrire la texture à partir d'une séquence d'images est proposé. Ce descripteur est moins sensible aux bruits et produit un histogramme court, tout en préservant la robustesse aux changements d'éclairage. Un autre descripteur pour la reconnaissance dynamique des textures est également proposé. Le descripteur permet d'extraire non seulement des informations de couleur, mais aussi des informations plus détaillées provenant des séquences vidéo.

Enfin, nous présentons une approche de sélection de caractéristiques basée sur le principe d'apprentissage par ensemble qui est capable de sélectionner les caractéristiques appropriées pour chaque pixel afin de distinguer les objets de premier plan de l'arrière-plan. En outre, notre proposition utilise un mécanisme pour mettre à jour l'importance relative de chaque caractéristique au cours du temps. De plus, une approche heuristique est utilisée pour réduire la complexité de la maintenance du modèle d'arrière-plan et aussi sa robustesse. Par contre, cette méthode nécessite un grand nombre de caractéristiques pour avoir une bonne précision. De plus, chaque classificateur de base apprend un ensemble de caractéristiques au lieu de chaque caractéristique individuellement. Pour compenser ces limitations, nous avons amélioré cette approche en proposant une nouvelle méthodologie pour sélectionner des caractéristiques basées sur le principe du « wagging ». Nous avons également adopté une approche basée sur le concept de « superpixel » au lieu de traiter chaque pixel individuellement. Cela augmente non seulement l'efficacité en termes de temps de calcul et de consommation de mémoire,

mais aussi la qualité de la détection des objets mobiles.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter presents an introduction about the background subtraction (BS) task, describes its perspectives and challenges in scene modeling, and then we also detailed the main steps in a background subtraction algorithm. Moreover, an outline of the thesis is included in this chapter as well as a list of the main contributions.

1.1 Challenges in scene modeling

Background subtraction is an attractive research field in computer vision. It concerns a set of methods that aim to differentiates the moving objects (the foreground) in the scene from a robust model of the static environment (the background). BS has been fueled by many academic scientists and developers over the last twenty years. This is rooted in its numerous potential applications and the availability of surveillance cameras installed in security sensitive areas such as banks, train stations, highways, and borders. Background subtraction can be used for surveillance devices in public spaces (such as football stadiums, and big trade centers), in traffic monitoring (counting vehicles, detecting and tracking vehicles) and industrial machine vision (inspection and identification products and robot guidance). There are three main conditions which assure a good functioning of the background subtraction methods: the camera is fixed, the illumination is constant and the background is static, that is pixels have a unimodal distribution and no background objects are moved or inserted in the scene. In these ideal conditions, background subtraction gives good results. In practice, the appearance of an outdoor or indoor scene depends on a variety of changes that can occur over time. Usually, it is challenging to design a good background model able to tolerate these changes. There are various situations that may affect scene appearance, thus reducing the accuracy of the BS algorithms. To the best of our knowledge, the typical challenges of background subtraction are [25, 100, 172]:

- ***Camera jitter:*** Usually, the camera jitter occurs in outdoor scenes. For instance, strong winds may cause a fixed camera to sway back and forth, causing nominal motion in the video sequence. This nominal motion is usually indistinguishable from the motion of foreground objects, and this leads to undesirable detection results.

- ***Camera automatic adjustments:*** Automatic exposure (means the amount of light that falls onto the sensor in a digital camera) is a setting available on most cameras today. The camera captures the light reflected by objects with homogeneous characteristics (e.g. intensity, texture) in the environment making the task of segmentation difficult. The foreground aperture occurs when parts of large moving homogeneous regions become part of the background instead of being considered as moving pixels.
- ***Pan-Tilt-Zoom (PTZ):*** The most research in background subtraction has been on stationary cameras, whereas PTZ cameras have become increasingly popular because of their ability to cover a wide field of view. Existing BS algorithms fail in the case of moving cameras as neither foreground objects nor background pixels are stationary.
- ***Video noise:*** Normally, a video signal is covered with noise caused by acquisition, coding, processing steps and transmission. This noise appearance disturbs the original information producing undesirable effects on the background scene, such as artifacts, unrealistic edges, unseen lines, and corners.
- ***Intermittent object motion:*** The intermittent motion happens when a moving object stops for a long period of time or a background object starts moving. This situation results in a "ghost" or "hole" in the background that is interpreted as part of the foreground. Some examples include objects that suddenly start moving (e.g. parked vehicle driving away, and abandoned objects). How to manage this situation depends on the context. Indeed in some applications, motionless foreground objects must be incorporated to the background model, and in others not.
- ***Dynamic backgrounds:*** In a dynamic environment, the state of the scene can changes continually. In other words, the transformation from one temporal stable to another is generally the outcome of an external event, or a chain of events (i.e. flowing water, moving leaves or shrubs). In such environment, it is challenging to have a good representation of the background model since even some part of the scene containing moving elements may be regarded as foreground.
- ***Presence of shadows:*** The detection of cast shadows as moving object is very common, producing undesirable results. For example, the shadows are so different from background that may mistakenly be detected as foreground.
- ***Illumination changes:*** In indoor or outdoor environment, illumination changes often occur over time and may cause false detections. For instance, in outdoor environments the gradual changes in appearance can be caused by a wide range of illumination conditions, in particular those encountered during a typical 24-hour day-night cycle. Moreover, sudden illuminations can occur due to turning on/off the light switch in an indoor scene. It is important that the background model be invariant or adaptable to these kind of changes.
- ***Bootstrapping:*** The initial video data without moving objects is not always available, then the representative background model cannot be produced. Thus, an initialization process is necessary to learn the correct background model over time.
- ***Camouflage:*** Some moving object can looks like the background, or some portion of it is camouflaged with the background (the so-called camouflage effect). This leads to an erroneous distinguish between foreground and background.

(a) Large shadows. (b) Variations lighting. (c) Large reflections.

Figure 1.1: Scenes from the same avenue under different conditions.

- ***Foreground aperture:*** The presence of moving objects can have the same motion features. Consequently, shadows usually make the geometrical shape of the moving objects distorted, and sometimes causing the fusion of moving objects.
- ***Night scenes:*** The videos captured at night are still a challenging task. Night scenes usually cause high false detections due to dramatic lighting change and low contrast between foreground and background.
- ***Challenging weather:*** In some cases, the background subtraction algorithm should adapt to adverse weather condition such as air turbulence or snow storm that modifies the background scene.

To address the above challenges, several researchers have proposed diversified methods and its evaluation results have often been available by Change Detection web site[1]. Recent experimental results have shown that the biggest problem is the distinction between the background and the foreground when the scene comes from night videos and videos captured by PTZ cameras [100]. Another great challenge is when different challenges occur in the same scene. Figure 1.1 shows three situations at the same avenue. While Figures 1.1a and 1.1b show shadows and different light variations, the Figure 1.1c displays large reflections. Despite all these situations are handled quietly nowadays [20,58,150,185,205], they still disturb the foreground detection process. Note that Figure 1.1 shows different situations, such as large shadows, light variations, and also large reflections. It is important to note that, until now, there is no background subtraction algorithm that is able to solve all of these challenges at the same time, making the BS field even more challenging.

1.2 Background subtraction steps

This section discusses the different steps related to background subtraction. Figure 1.2 shows an overview of these steps. In essence, background subtraction consists to output a binary segmentation map by initializing and updating a model of the static scene, which is named the background (BG) model, and comparing this model with the input image. Pixels or regions with a noticeable difference are assumed to belong to moving objects (they constitute the

[1] http://wordpress jodoin.dmi.usherb.ca/results2014/

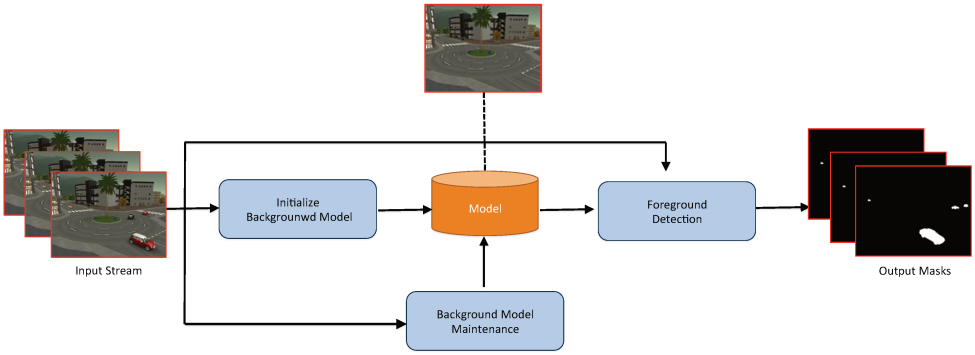


Figure 1.2: An overview of the background subtraction process.

foreground FG). A traditional background subtraction technique consists of four elements: a background model, a background initialization process, an updating mechanism, and a foreground detection operation. BS is often the first step in many computer vision applications as shown in Figure 1.3.

The background model (or representation) is the core of any BS algorithm. The key idea behind such step is to create a representation of the static scene which is robust against environmental changes in the background and also sensitive to identify all moving objects of interests. In the last decades, researchers have proposed a number of methodologies for modeling and subtracting the background, e.g. statistical methods [39, 88, 187, 188], multilayer codebook based methods [74], compressive methods for streaming videos [55], etc. Another important step is the background initialization process that consists to its generation, extraction or construction. In contrast to background modeling, the initialization of the background model was only slightly investigated (e.g. [49, 77, 130]). The main reason is that often the assumption made is that initialization can be achieved by exploiting some clean frames at the beginning of the sequence. Naturally, this assumption is rarely met in real scenarios, because of continuous clutter presence. Generally, the model is initialized using the first frame or a preliminary background model estimated over a set of training frames, which contains (or not) foreground objects. The third step consists to the background model updating (or maintenance) that relies on the mechanism used for adapting the background model when a scene changes over time. It is important that the background maintenance be incremental (an online algorithm), since new data is streamed and so dynamically provided. Robust updating mechanisms used in flexible models aim to overcome different challenges, such noisy, camera automatic adjustments and background illumination changes. Furthermore, it is in this stage where the updating mechanism that defines whether inserted objects are incorporated to the model, and whether ghosts are updated or removed. To solve these issues, various approaches have been developed [7, 115, 123, 134]. Finally, the last step is the foreground detection operation, which compares the background model with the current image to label pixels (or regions) as background or foreground. This task is a classification one, that can be achieved by crisp [115, 156], statistical [2, 180] or fuzzy [38] methods.

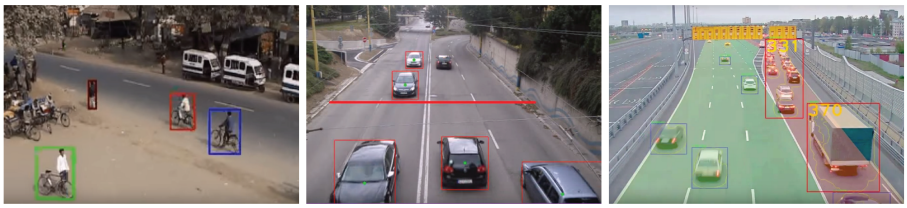


Figure 1.3: Moving object tracking results.

1.3 Why features are important in the BS context!

Researchers have been working for decades to develop BS methods to handle the different type of challenges in BS (described in Section 1.1). However, at the present time, no algorithm seems to be able to simultaneously address all the key challenges found in real environments. This limitation may occurs due to the majority of BS methods are focused on sophisticated learning models, while visual features have been received relatively little attention. The suitable choice of features in background modeling can improve the segmentation of moving objects, however, certain factors must be taken into consideration. For instance, color and intensity features are very discriminative, but they have several limitations in the presence of illumination changes, camouflage and shadows. Nevertheless, the texture and edge features are less sensitive to illumination variations that might occur in outdoor scenes due to the sun, clouds or light changes, while stereo features can differentiate moving objects from shadows avoiding some problems such as: object shape distortion, ghost objects and camouflage. Whereas, the motion features might be useful to handle dynamic scenes, containing common elements such as fountains, swaying trees or ocean ripples [24, 149]. Other type of feature that has become accessible is the multispectral-*based* images [16]. Its main advantage is the possibility to take into account the spatial (or spatio-temporal) relationships among the different spectra in a neighborhood, allowing more elaborate spectral-spatial (and -temporal) models for a more accurate segmentation. However, its primary drawback are the computational cost and complexity due to its massive and multidimensional characteristics. In this thesis we will focus on the importance of features in the background subtraction taking into account two main factors: study of new features and selection of the best features for background modeling. The development of new features and the selection of the best features can improve the foreground segmentation, mainly if the features are complementary and uncorrelated [78].

Recent advances in deep convolutional neural networks (ConvNet) have enabled a new way to extract features from images and videos. The ConvNet have a great performance in many computer vision applications including background subtraction [29, 221]. In addition, it is commonly easy to set up using modern libraries (Caffe [101], Theano [14], Torch [1], etc.) with built-in architectures. On the other hand, the ConvNets are not suitable for applications whose few images are available – training a deep ConvNet usually require a large amount of images for a better model generalization. Furthermore, the computational cost for training of ConvNets is high in term of time and memory requirements. For these reasons, the study of new features computationally simple is crucial in many real-life applications. Color

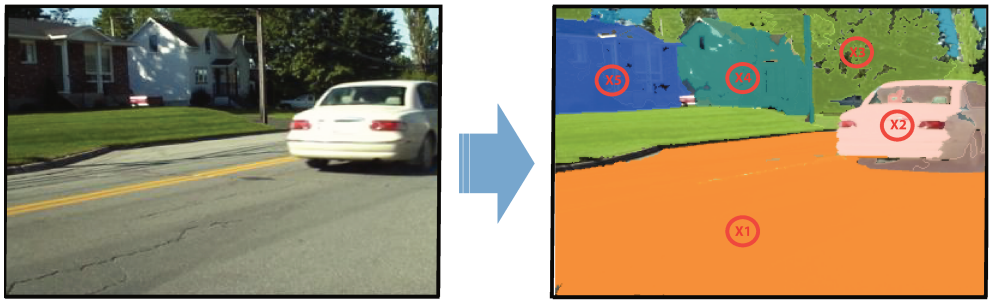


Figure 1.4: Given a complex scene with different regions X1, X2, X3, X4 and X5, these can be characterized by different features such as: texture, color, texture-color, motion and edge.

and intensity are probably the most popular features for background modeling. However, a variety of local texture descriptors recently proposed have attracted great attention for background modeling, especially the Local Binary Pattern (LBP) [146] because of its simplicity and speedy to compute. In addition, it does not require a large amount of data for feature extraction. Although LBP features are tolerable against illumination variations, they cannot deal with the presence of moving shadows. The other major problem of LBP in background modeling is that it cannot cope with local image noises when neighboring pixels are similar. Moreover, the ordinary LBP usually does not work well in dynamic scenes since it does not take into consideration the temporal information. Some researchers have put extensive effort to solve these problems by proposing new variants of local binary patterns. Nevertheless, there are still limitations to be addressed giving opportunities for further investigations.

In addition to the difficulties faced by researchers in the conception of robust features for background subtraction, a second major challenge is the definition of the best combination of features that would improve the accuracy and robustness in foreground detection. Commonly, the BS methods do not take into account the properties of each features, and they use the same feature for the whole scene [181]. Figure 1.4 shows a complex scene comprising of several elements such as waving trees, sky, soil and cars. The most discriminant features for these elements are probably different, and therefore a single-feature BS algorithm may not be appropriate. Some authors have proposed to combine two (or more) features to take advantage from both [47, 97, 122, 235]. However, the fusion of features can be helpful to a limited range, because the features chosen may not be mutually complementary, and frequently they have conflict. Despite the choice of the best features for each region is not an easy task as it requires a deep knowledge of the scene. However, it is possible to automatically select the most relevant features, and this process is commonly defined as *feature selection*. It can improve the detection of foreground objects thanks to its capability to select a subset of highly discriminant features removing irrelevant and redundant ones. Traditionally, feature selection methods can be categorized into three main groups: filter, wrapper and embedded *-based* methods. Over recent years, a new kind of feature selection that use ensemble learning to select features, called *ensemble for feature selection* have been proposed [23]. Ensemble learning is a powerful tool to combining a set of models, where each of them solves the same task in order to obtain a better global model with more robustness and the generalization

ability than a single model. The hot wave of research on ensemble learning began in 1990, however its efficiency has been proven until the current days. In the contest held in last year by ImageNet Large Scale Visual Recognition Challenge (ILSVRC), software programs compete to correctly classify and detect objects and scenes. The best performance was achieved by algorithms that used an ensemble of deep neural networks (see the results in [2].) Ensemble for feature selection extends the traditional feature selection methods by looking for a set of feature subsets that will favour disagreement among the ensemble members [186]. Surprisingly, little BS works have been done to date based on feature selection approaches, becoming this subject an interesting research topic in the BS context.

1.4 Contributions of the thesis

Given the above importance of the features in background subtraction, we present below the contributions of the thesis. The list of publications concerning the thesis can be found in Appendix C.

1. A novel texture-based descriptor, namely eXtended Center-Symmetric Local Binary Pattern (XCS-LBP). The descriptor is less sensitive to noisy pixels and produces a short histogram, while preserving robustness to illumination changes.
2. A new pixel-based ensemble for feature selection in background subtraction to deal with the challenges enumerated in the Section 1.1. The proposed approach selects automatically the best features for different pixels of the image, and the most relevant features are used for foreground segmentation. In our framework, the background is modeled by different features including our proposed XCS-LBP descriptor.
3. Our pixel-based ensemble for feature selection only reaches the highest accuracy when the number of features is huge. Furthermore, each base classifier learns a feature set instead of individual features. To overcome these limitations, we extend our previous approach by proposing a novel methodology for selecting features based on wagging. This approach is more efficient in terms of time and memory consumption. We also added an ensemble pruning technique to eliminate the importances with very low values over time.
4. A robust 3D joint color-texture descriptor, called OCLBP-TOP developed in conjunction with the Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB). This descriptor allows to extract not only color information, but also a more detailed information from video sequences.

[2] http://image-net.org/challenges/LSVRC/2016/results

1.5 Thesis outline

The rest of the thesis is organized as follows.

- ***Chapter 2:*** conducts a literature review of the main features used in the context of background subtraction. In addition, the traditional approaches for feature selection including the recent works in this domain are also discussed.
- ***Chapter 3:*** presents a novel eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) descriptor for background modeling and subtraction in videos. The experiments conducted on both synthetic and real videos (from the Background Models Challenge) show that the proposed XCS-LBP outperforms its direct competitors for the background subtraction task.
- ***Chapter 4:*** describes an online weighted pixel-based ensemble learning method able to select suitable features for each pixel to distinguish the foreground objects from the background. In addition, our proposal uses a mechanism to update the importance of each feature over time. Moreover, a heuristic approach is used to reduce the complexity of the background model maintenance while maintaining the robustness of this one. Experimental results on two datasets have shown the pertinence of the proposed approach.
- ***Chapter 5:*** extends our approach proposed in Chapter 4 by a novel methodology for selecting features based on wagging. Furthermore, we also adopted a superpixel-based approach instead of a pixel-level approach. This does not only increased the efficiency in terms of time and memory consumption, but also improved the segmentation performance.
- ***Chapter 6:*** presents a particular work realized in conjunction with Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB). This chapter describes a novel Opponent Color Local Binary Pattern from Three Orthogonal Planes (OCLBP-TOP) descriptor for applications in the field of dynamic texture recognition. The OCLBP-TOP fuses both, the texture and color information. As such, it allows to extract not only color information, but also a more detailed information from video sequences. The experiments conducted on real videos have shown that the proposed OCLBP-TOP outperforms other state-of-the-art descriptors.
- ***Chapter 7:*** summarises the thesis with remarks, advantages, and limitations of the proposed approaches. It also discuss the open issues and future works.

Chapter 2

Literature review

Features play an essential role for various computer vision applications and it is not different for background subtraction. In the long history of BS, various features have been used, improved or even proposed to address BS challenges in background modeling. Another way to deal with the BS challenges is to select a subset of highly discriminant features for each pixel, region or cluster in a image sequence. This can be done automatically by using feature selection approaches. This chapter begins with a review of the main features used in the context of BS, then we discuss the traditional and recent approaches for feature selection including the important BS works in this domain. This chapter corresponds to a concise version of our recent survey submitted to Computer Science Review, 2016 [26]. Furthermore, an open source library, called LBPLibrary[1], was developed to provide a collection of local binary patterns variants. The library was designed for the problem of background-foreground separation in videos.

2.1 Features for background modeling

Background modeling is an important step in detecting moving objects in video sequence. A very important factor in background modeling is the choice of the transformation that is applied to the original data in order to obtain the features that are used. Features (descriptors or attributes) is a set of measurements describing an object such as points, edges or corners. In background subtraction, the features characterize a picture element captured in the current frame of a video sequence and are compared against a known background model to classify it as either foreground or background. Feature representations can take multiple forms and can be computed for and from: a pixel, a region or a cluster. Practically, there are several types of features which can be computed either in the spatial, temporal, spatio-temporal or depth transform domain. Some of the features commonly used within the background modeling literature includes: color features, edge features, stereo features, motion features and texture

[1] https://github.com/carolinepacheco/lbplibrary

features. These features can be classified from different view points such as: by level, by type in a specific domain and by intrinsic properties. In the following sections, these view points are discussed in more details.

2.1.1 Classification by level

The size of the picture element chosen for interpreting necessary features that faithfully represent its characteristics plays a crucial role in background modeling. The size of the picture element that is used to model the background and hence for comparing the current image frame to the background model, can either be a pixel [70], a region [70] or a cluster [17] with a feature value.

- ***Pixel-level:*** Most approaches for background subtraction are based on pixel-level modeling which assumes adjacent pixels are independent. These approaches build a separate model for each pixel, such as Gaussian Mixture Model (GMM) [187, 244, 246], Kernel Density Estimation (KDE) [58], and non-parametric approaches based on sample consensus (Pixel-based Adaptive Segmenter (PBAS) [88] and ViBe [12]). The pixel-level approaches are usually effective, but they cannot discriminate well the variations of the pixel's value caused by the presence of foreground objects and natural illumination changes, since each model knows only history of the corresponding pixel. In fact, such illumination changes is learnt in the background model over a period of time, it is practically impossible to adapt it for sudden illumination changes [139].
- ***Region-level:*** Many studies have adopted a region-level background modeling by splitting an image into blocks and calculating the block-specific features. In this approach, instead of dealing with one pixel at time, the relationship among neighboring pixels is modeled [236]. Compared with pixel-level modeling, the region-level one gives richer features, and it is more robust in the case of illumination changes. Another important advantages is their robustness to noise and the movement in the background. However, the disadvantage is that the detection is less precise because only foreground regions are segmented, making them unsuitable for applications that require a detailed shape information of the foreground object.
- ***Cluster-level:*** A recent trend in background modeling is to consider region sizes that are non-uniform across the image sequence. First, pixels in an image frame are grouped using an application-specific homogeneity criteria, typically exploiting clustering mechanisms as discussed in [17–19]. For example in Bhaskar et al. [17], each cluster contains pixels that have similar features in the color space. Then, the background model is applied on these clusters to obtain cluster of pixels classified as background or foreground. This cluster-level approach gives less false alarms than block-level approaches. Just like the region-level modeling, the cluster-level ones boost efficiency in terms of both required memory and computation time, since fewer models have to be kept in memory and updated at every frame.

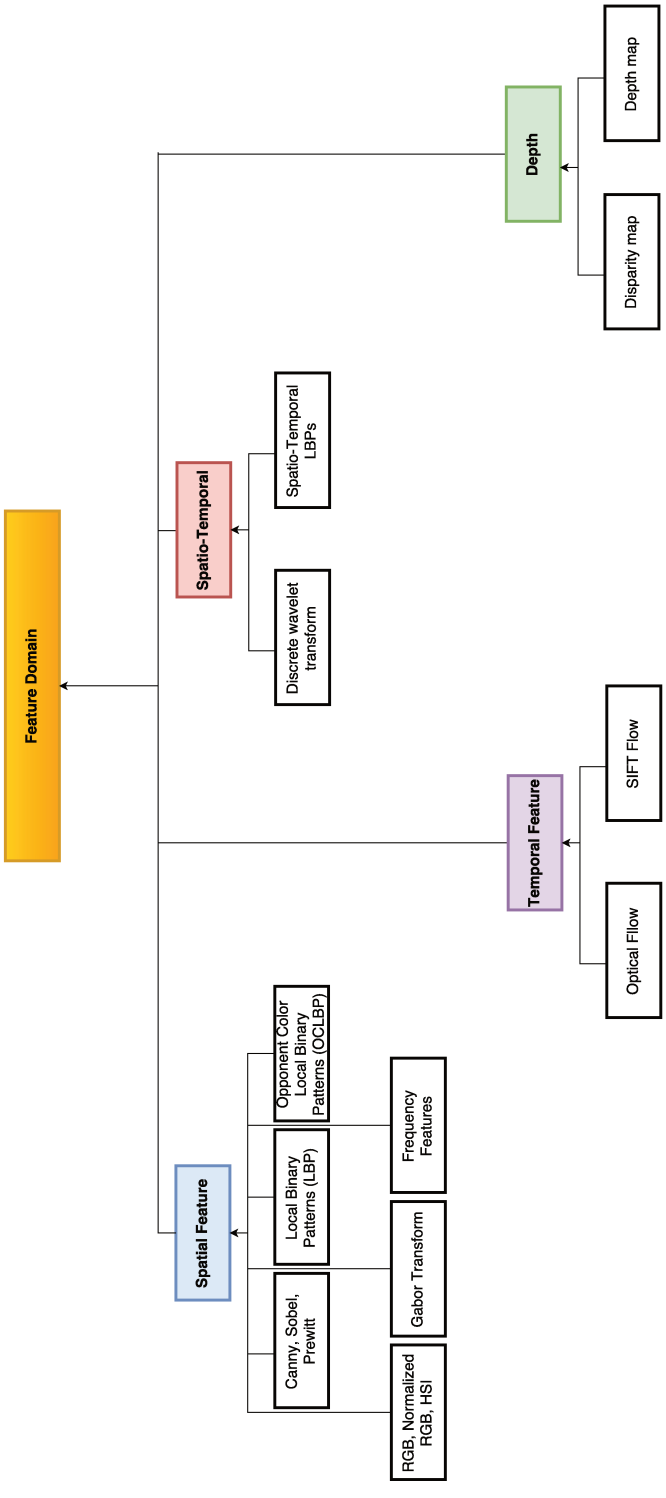


Figure 2.1: A brief overview of the features classified by its intrinsic properties.

2.1.2 Classification by intrinsic properties

In this thesis, we define the features can be classified by their intrinsic properties into the following categories:

- ***Spatial domain:*** At the beginning of the research in BS, most of the features were represented in the spatial domain. Basically, the spatial domain refers to the image plane itself, and they assume that the scenes to be modeled are often static structures with a little perturbation. Spatial features can cope well with the slight illumination changes, but cannot usually handle either large (or sudden) changes or multiple moving objects in the scene [95].

- ***Temporal domain:*** The temporal information can be an alternative choice to improve the detection of non-stationary objects. The motion information can be exploited to improve the discriminative power of the objects by including its temporal evolution. The key idea of the temporal domain methods lie in the fact that different object's motion produces a defined motion pattern. The temporal domain can be used to discriminate one object from another by analyzing its temporal motion patterns, thereby playing a crucial role in moving object detection.

- ***Spatio-temporal domain:*** The spatio-temporal domain is usually suitable to deal with dynamic background by modeling the spatial and temporal relationship and variations simultaneously. For example, in an outdoor scene containing waving trees caused by the presence of a strong wind, the regions occupied by the trees will be unstable (random motion patterns) for a some period of time. The temporal domain methods consider only the temporal variations and ignore the spatial variances which is useful for a correct modeling of the background.

- ***Depth domain:*** The recent advances on distance image sensors enabled the use of the depth information for many computer vision applications, specially in the BS field. The depth information is an attractive element for the segmentation of moving objects due to its robustness to deal with common BS problems such as shadows and camouflage (e.g. the similarity of the color and texture features of the moving object and the scene background).

Various features for the above domains have been proposed and improved for background subtraction. Figure 2.1 shows a brief overview of the features classified by its intrinsic properties.

2.1.3 Classification by type

Now, we present and analyze the different features mostly used in background modeling and foreground detection in terms of robustness against the challenges in videos taken by fixed cameras.

Figure 2.2: RGB channels of the image showed separately.

Color features

Color features have been widely used for background modeling. They provide a ***spatial information*** from captured by sensors or cameras. In spite of the color features have been largely used for an easy discrimination between the background and the foreground, the color features are generally not robust against illumination changes and shadow cast caused by moving objects. Furthermore, similar colors between background and foreground lead to the well-known problem of camouflage. In the literature, several color features in different color spaces have been proposed, and they are described as follows:

- ***RGB color space:*** The RGB color space is the most popularly used due to their direct availability from sensors or cameras. Red, Green, and Blue channels (see Fig. 2.2) of each pixel are usually measured with 8-bits resolution, where 0 is no intesity (black color) and 255 is the maximum intensity (white color), therefore, a total of 24-bit true color definition. But the RGB color space has several limitations: ***1)*** it is well known that the RGB color space does not reflect the true similarities among colors, ***2)*** depending on the scene, one color component could be more informative than others, so it should be given more importance than others, ***3)*** the three components are dependent on each other which increase its sensitivity to illumination changes. For example, global illumination changes shift the mean level of the entire RGB image, possibly with shifts of different magnitude for each color component, and ***4)*** as the three channel components are correlated, there is a need to compute inter-correlation terms in the covariance matrix which shall be incorporated into existing background models such as in the Mixtures of Gaussian (MOG) model [187]. Stauffer and Grimson [187] demonstrated that by not computing these inter-correlations terms, computational speed improves, however with increased false detections.
- ***Normalized RGB color space:*** The normalized RGB space is derived from the traditional RGB color space to be illumination invariant. Xu and Ellis [220] used the normalized RGB to allow the MOG to be robust to fast illumination changes in an outdoor environment lit by sunlight and shadowed by clouds.
- ***YUV color space:*** The YUV space separates luminance and chroma and so it is more suitable for improving the robustness of the model against illumination changes. For example, Wren et al. [216] used the normalised components, U/Y and V/Y to remove

shadows in a relatively static indoor scene. Using the MOG model, Harville et al. [82] defined a chroma validity test based on the luminance Y as the chroma (U and V) components become unstable when the luminance is low. When the test is not verified, the chroma components of the current observation are not used and so are its current Gaussian distributions. Furthermore, the detection in luminance was combined with the detection in depth, improving its robustness to color camouflage.

- ***HSV color space:*** The HSV color space is used to improve the discrimination between shadows and objects, classifying shadows as those pixels having the approximately the same hue and saturation values compared to the background, but lower luminosity. For example, Sun et al. [190] used the Hue-Saturation-Value (HSV) color space, because the likelihood term in the MOG model shows stronger contrast in HSV space rather than the RGB space, especially for objects that share similar appearance to the background (camouflage in color).
- ***HSI color space:*** HSI color space is closer to human interpretation of colors in the sense that brightness, or intensity, is separated from the base color. HSI uses polar coordinates. In the original MOG model, shadows are extracted as part of the object mask when using the RGB color space. To address this problem, Wang and Wu [210] used the HSI color space which tends to be shadow-removable. However, the obtained results are not satisfactory due to the fragmented segmentation results by using hue and saturation. In order to achieve both "shadow-rejection" and "segmentation stability over time", Wang and Wu [210] employed the MOG on chroma (hue and saturation) and luma (intensity) separately. The fused results obtained by combining chroma and luma is prepared using two criteria. This scheme reserves the advantage of using chroma (i.e. avoiding shadow) and that of luma (i.e. stability of segmentation).
- ***Luv color space:*** Yang and Hsu [228] used the Luv components assuming independence in the computation of covariance matrix required in the MOG model. Then, Yang and Hsu [228] built an hybrid feature space with spatial and color features to obtain a 6-dimensional hybrid feature vector for each pixel. A mean-shift procedure classified each hybrid feature vector to its corresponding local maximum along the gradient direction. Thus, a set of neighboring pixels associated with the same local maximum (i.e. mode) is highly similar in this hybrid feature space. Yang and Hsu [228] then assign pixel-level background likelihood for each pixel using the MOG likelihood, and further obtain a smoothed version of MOG in terms of spatial and color coherency.
- ***Improved HLS color space:*** Setiawan et al. [171] proposed to use the IHLS color space which has the following advantage against the RGB color space. That is to identify shadows region from an object by using luminance and saturation-weighted hue information directly, without any calculation of chrominance and luminance. By exploiting this color space in the MOG model, Setiawan et al. [171] obtained good sensitivity to color changes and shadow.
- ***Ohta color space:*** The axes of the Ohta space are the three largest eigenvectors of the RGB space, found from the principal components analysis of a large selection of natural images. This color space is a linear transformation of RGB. Using the mean model, Zhang and Xu [235] applied the Ohta color space. The three orthogonal

color features of the Ohta color space are important components for representing color information. Good results in the case of illumination changes and shadows in outdoor scenes are achieved by using only the first two components which are combined with a texture feature.

- ***YCrCb color space:*** YCbCr uses Cartesian coordinates. El Baf et al. [8] used the YCrCb color space combined with the texture feature to be robust to illumination changes and shadows. Experimental results in [8] showed that YCrCb color space is more robust in these cases than the Ohta and HSV color spaces.

- ***Lab/Lab2000HL color space:*** Lab color space is a color space which indicates proper changes in the direction of human color perception. Its components are the lightness of the color and two color opponent dimensions. Lab2000HL color space, which is an improved version of Lab color space, was introduced and is thought to perform a better modeling of the human perception. Particularly, Lab2000HL color space have linear hue band. So, Balcilar et al. [9] investigated the performance of the Lab2000HL color space. The average precision value of Lab2000HL is the greatest in all videos in comparison to all other color spaces. The Lab2000HL globally gives the best performance on all the video sequences, but not mandatory on each sequence. In terms of the computational costs for each color space (YCrCb, Luv, Lab,Lab2000HL), RGB color space leads to the lowest. The reason is that it does not require any transformation since the information gathered from the camera sensors is directly in RGB. Lab2000HL color space, on the other hand, has the most computational cost, since a computationally intensive procedure is required to apply first the Lab transformation, and then the computation of transformation value with respect to the transition map using interpolation.

Edge features

Edge features are based usually on intensity features given from ***spatial information***, and they are computed using a gradient approach such as Canny [37], Sobel [108] or Prewitt [154]. The gradients can be calculated from the gray level image or in each component of the color space. Edge detectors operate on the difference between neighboring pixels, hence an edge detector should be reasonably insensitive to global shifts in the mean level, i.e. global illumination changes. Therefore it would be interesting to run background-foreground separation algorithms on the output from edge detectors, hopefully reducing the effects of rapid illumination changes. So, the edge could handle the local illumination changes, but also the ghost leaved when waking foreground objects begin to move. The edge features are generally used alone or jointly with other features as follows:

- ***Edge alone:*** First, Kim and Hwang [104] proposed to use only edges to model the background. This approach used a binarized information for the existence of an edge for a given pixel. But, regions in consecutive frames may not have exactly the same edge position, and have shape and length changes due to presence of noise. This strategy may generate many false alarms in the foreground mask due to edge distortion

Features	*Acronym of papers*	*Authors - Dates*
Color features	RGB	Stauffer and Grimson (1999) [187]
	Normalized RGB	Xu et Ellis (2001) [220]
	Normalized RGB	Xu et Ellis (2001) [220]
	YUV	Wren et al. (1997) [216], Harville et al. (2001) [82]
	HSV	Sun et al. (2006) [190]
	HSI	Wang and Wu (2006) [210]
	Luv	Yang and Hsu (2006) [228]
	Improved HLS	Setiawan et al. (2006) [171]
	Ohta	Zhang and Xu (2006) [235]
	YCrCb	Baf et al. (2008) [8]
	Lab/Lab2000HL	Balcilar et al. (2013) [9]
Edge features	***Edge alone***	
		Jabri et al. (2000) [95], Kim and Hwang (2002) [104]
		Li et al. (2004) [118], Lindström et al. (2006) [123]
		Kim and Hwang (2002) [104], Murshed and Chae (2010) [141]
		Ramirez-Rivera et al. (2011) [157], Kim et al. (2013) [105]
		Mousse et al. (2014) [140], Lopez-Rubio and Lopez-Rubio (2014) [131]
		Wang and Wan (2014) [211]
	Jointly with other features	
		Jabri et al. (2000) [95], Lindström et al. (2006) [123]
		Kim et al. (2015) [106]
Depth features	***Depth from Stereo-Cameras***	
		Eveland et al. (1998) [59], Gordon et al. (1999) [68]
		Ivanov et al. (2000) [94] , Harville et al. (2001) [82]
		Braham et al. (2014) [27], Harville (2002) [81]
	Depth from Time-of-Flight Cameras	
		Tombari et al. (2008) [197], Leens et al. (2009) [117]
		Stormer et al. (2010) [189], Hu et al. (2014) [91]
		Braham et al. (2014) [27]
	Depth from RGB-D Cameras	
		Greff et al. (2012) [73], Gallego and Pardas (2013) [66]
		Camplani et al. (2013) [35], Fernandez-Sanchez et al. (2013) [62]
		Spampinato et al (2014) [183], Fernandez-Sanchez et al. (2013) [62]
		Liang et al (2016) [119]
Motion features	Optical Flow	Huang et al. (2006) [92], Zhong et al. (2008) [241]
		Huang et al. (2009) [93],Chen et al.(2014) [45]

Table 2.1: Features: An Overview (Part 1).

from consecutive frames. To solve the edge-distortion problem, edge-segment-based methods have emerged to take advantage of the edge existence and its shape information [89]. An edge-segment approach consists of the concatenation of adjacent edges, and it inherits the problems of edges: shape and position changes. Thus, basic comparison of edge-segments produces similar results as edge-pixel-based approaches. To solve this problem, statistical edge-segment-based methods extract movement of edge-segments including edge distortion [105, 141, 157]. Thus, these methods solve the edge-variation problem by accumulating edge existence from a training set [106]. Practically, each accumulated region represents an edge-segment distribution. Each region refines their statistical properties after each frame to provide a stable background model. Since edge-based and edge-segment-based methods detect foreground as edges, these methods depend of a post-processing step to extract the regions defined by the detected edges. Moreover, these methods have problems updating their background model to adapt the background.

- ***Jointly with other features:*** Jabri et al. [95] used in addition of the intensity features the intensity gradient obtained by the Sobel edge detector. Large changes in either intensity or in edges are fused. However, the involvement of the intensity model retains the sensitivity to sudden changes in illumination. Lindström et al. [123] proposed to use a Prewitt edge detector without the thresholding independently to each color component followed by a log-transformation gives a color edge image with pixel values that can be modeled using Gaussian mixtures. Experimental results [123] showed better performance against illumination changes for the log-transformed detection using the Prewitt edge detector. In another work, Kim et al. [106] used edge and texture features in a hybrid scheme to generate the background model. Thus, theses features are encoded into a coding scheme called Local Hybrid Pattern (LHP). LHP selectively models edges and texture features of each pixel. Then, each pixel is modeled with an adaptive code dictionary to take into account the background dynamism. In the background maintenance, stable codes are added in the model while unstable ones are discarded. The incoming codes that deviate from the dictionary are classified as edge or inner region. Experimental results [106] on the ChangeDetection (CDnet 2012) dataset [69] showed that this Adaptive Dictionary Model (ADM) with LHP features outperforms the original MOG [220], the ordinary LBP [84] and SALBP [144].

Texture features

Texture features are extracted from ***spatial information*** or on ***spatio-temporal information***. The texture features have been very investigated in the BS field as can be seen in Table 2.2. Generally speaking, texture can be defined to surface characteristics and appearance of an object given by the shape, size, density, arrangement, proportion of its elementary parts. By contrast with the color features, the texture features are more appropriate to cope with illumination changes and shadows. In the following, different texture descriptors are discussed following the same categorization given in [198].

- ***Statistical Texture:*** Statistical texture descriptors are useful qualities for the spatial distribution of the intensity values. This technique is one of the first methods sug-

gested in the literature of texture descriptors. In BS, some statistical texture descriptors have been proposed mainly to deal with the problem of illumination variations. For instance, Satoh et al. [165] proposed Peripheral Increment Sign Correlation (PISC) feature that encodes a value of 1 or 0 according to whether the increment near the considered pixel is positive or negative. The resulting logical code representing the trend of brightness change. However, this leads to increase false positives because the code is reversed easily with slight intensity changes in regions with small intensity differences, for example in plain regions. Plain regions often occupy large spatial region within images, which makes stabilizing on them very important. Yokoi [230] proposed a Probabilistic Bi-polar Radial Reach Correlation (PrBPRRC). It encodes the intensity difference by $-1/0/1$ ternary codes to enhance the robustness against illumination changes and background movements. In Satoh et al. [166] a novel statistical measure for robust event detection, called Radial Reach filter (RRF) is proposed. It evaluates a local texture to handle with brightness distributions of the events and the influence of shadows, etc. RRF searches for a point with the brightness difference more than a threshold from the interest pixel. This procedure is repeated about eight directions in the shape of radiation resulting in 8 sets of the "RRF pairs". At the end a binary code is given by the sign of brightness difference of each pairs.

- ***Structural texture:*** These type of descriptors are constituted by the texture elements named as texels or texton. Texels are the smallest element that creates the impression of a texture surface. Usually, structural descriptors are invariant to illuminations, however heavily depend upon the definition of texels. To the best of our knowledge, the structural texture descriptor has been less explored for moving object detection. Recently, Spampinato [184] presented a kernel density estimation method which models background and foreground by exploiting textons to describe textures within small and low contrasted regions. According to the authors, the proposed method is robust to illumination changes, but it can not be applied for real-time purposes due to computational cost.

- ***Model based texture:*** Model based texture is commonly learned for a specific texture analysis task and used as features. The most popular technique from this category for background modeling is Markov Random Fields (MRFs) [107]. They are based on the contextual information of the image. In Schick et al. [169], a novel post-processing framework to improve foreground segmentation with the use of Probabilistic Superpixel Markov Random Fields is proposed. First, they converted a given pixel-based segmentation into a probabilistic superpixel representation. Based on these probabilistic superpixels, a Markov random field exploits structural information and similarities to improve the segmentation. Xu et al. [222] also introduced a new background modeling algorithm based on MRFs. The pyramid structure is introduced and the background modeling/labeling are processed at different resolution levels. The experiments showed this algorithm segment the foreground objects accurately from scene with sharp lighting changes and background movements. Other works using MRF technique can be found in [33, 137].

- ***Filtering based texture:*** Filtering based descriptors represent an image in a space whose co-ordinate system has an interpretation that is closely related to the characteristics of a texture. For instance, the frequency masks are more common and ef-

fective in texture description. Usually, the frequency features are obtained by converting the image into the frequency space normally using Fast Fourier Transform (FFT) [41]. Fourier transform features encapsulate spatial information which are suitable for scenes that contain periodic motions. That is, scene having a significant correlation between structures and observations across time (e.g. a tree swaying in the wind or a wave lapping on a beach). In this context, Wren and Porikli [217] estimated the background model that captures spectral signatures of multi-modal backgrounds using FFT features through a method called Waviz. Here, FFT features are then used to detect changes in the scene that are inconsistent over time. Results [217] showed robustness to low-contrast foreground objects in dynamic scenes. Some others works based on frequency methods are found in the state-of-the-art: *Discrete Cosinus Transform Features* ([160, 209, 245]) and *Hadamard Transform* (also known as the *Walsh–Hadamard Transform* ([10]). Latterly, wavelet transformation [135] is one of the most famous of the time-frequency-transformations. Considering that static backgrounds correspond to the low-frequency components, Han et al. [79] removed the static backgrounds indirectly in the 3D wavelet domain. Additionally, they made use of wavelet shrinkage to remove disturbance and introduce an adaptive threshold based on the entropy of the histogram to obtainur optimal detection results. See other works using the wavelet transformation at: ([6, 50, 90, 138]). Another popular descriptor based on filtering is the Gabor Transform [65]. Some Gabor Transform works in BS can be found in [214, 227].

- ***Local Binary Patterns:*** Local binary patterns (LBP) proposed in [85] is the simple yet powerful gray scale invariant texture descriptor. The computation of the ordinary LBP for a neighborhood of size P = 8 is illustrated in Figure 2.3. It combines the characteristic of statistical and structural texture analysis, describing the texture with micro-primitives and their statistical placement rules. To the authors' best knowledge, the first work using LBP histograms for background modeling was proposed by Heikkilä et al. [85]. The authors showed that LBP features are tolerant against illumination variations. Therefore, they found that moving shadows could not be handled very well. The other major LBP problem in background modeling is that it cannot cope with local image noise when neighboring pixels are similar. In addition, the ordinary LBP cannot usually work well in dynamic scenes since it does not taken into account the temporal information. Consequently, several LBP variants have been proposed in the recent literature to tackle these problems. In this thesis, we grouped these variants into five categories. We describe below the main LBP variants for each category. The interested reader will find a full list of the main LBP variants in Table 2.2 and its relative equations in the Appendix B.

  - *Ordinary LBP-based:* The first category consists of the variants with small mathematical changes from ordinary LBP. Few years after using ordinary LBP in background modeling, Heikkilä et al. [84] proposed a small change in its thresholding scheme. They improved the ordinary LBP in image areas where the gray values of the neighboring pixels are very close to the center pixel, *e.g.* sky, grass, etc. The LBP-based algorithms are often invariant to local illumination changes, but they are unable to detect uniform foreground objects in large uniform background except at the objects' edges. To solve this problem, Chua et

Textures	*Acronym of papers*	*Authors - Dates*
Statistical texture	Radial Reach filter (RRF)	Satoh et al. (2002) [166]
	Peripheral Increment Sign Correlation (PISC)	Satoh et al. (2004) [165]
	Probabilistic Bi-Polar Radial Reach Correlation (PrBP-RCC)	Yokoi (2009) [230]
Structural texture	Texton	Spampinato et al. (2014) [184]
Model based texture	Markov Random Fields (MRFs)	Xu et al. (2005) [222], Bugeau and Pérez (2007) [33]
		McHugh et al. (2009) [137], Schick et al. (2012) [169]
Filtering based texture	***Frequency features***	
		Wren and Porikli (2005) [217], Zhu et al. (2005) [245]
		Wren and Porikli (2005) [217], Wang et al. (2005) [209]
		Reddy et al. (2010) [160]
	Wavelet transformation	
		Antic et al. (2009) [6], Crnojevic, et al. (2009) [50]
		Mendizabal and Salgado (2011) [138], Hsia and Guo (2014) [90]
		Han et al. (2016) [79]
	Gabor transform	
		Wei et a. (2008) [214], Xue et al. (2012) [227]
Local Binary Patterns	***(1) Ordinary LBP-based***	
	Local Binary Pattern (LBP)	Heikkilä et al. (2004) [85]
	Opponent Color Local Binary Patterns (OCLBP)	Maenpaa and Pietikainen (2004) [133]
	Modified LBP	Heikkilä et al. (2006) [84]
	εLBP	Wang and Pan (2010) [206]
	Adaptive εLBP	Wang et al. (2010) [207]
	Uniform Local Binary Patterns (ULBP)	Yuan et al. (2012) [231]
	Local Color Pattern (LCP)	Chua et al. (2012) [47]
	Local Binary Similarity Patterns (LBSP)	Bilodeau et al. (2013) [22]
	Local SVD Binary Pattern (LSBP)	Guo et al. (2016) [76]
	(2) Center-Symmetric LBP-based	
	Center-Symmetric Local Binary Patterns (CS-LBP)	Heikkilä et al. (2009) [86]
	Center-Symmetric Local Derivative Pattern (CS-LDP)	Xue et al. (2011) [225]
	eXtended Center-Symmetric Local Binary Pattern (XCS-LBP)	Silva et al. (2015) [176]
	BackGround Local Binary Patterns (BG-LBP)	Davarpanah et al. (2016) [51]
	(3) Ternary LBP-based	
	Local Ternary Pattern (LTP)	Tan and Triggs (2010) [191]
	Scale Invariant Local Ternary Pattern (SILTP)	Liao et al. (2010) [120]
	Scale Invariant Local States (SILS)	Yuk and Wong (2011) [232]
	Scene Adaptive Local Binary Pattern (SALBP)	Yin et al. (2013) [229]
	Multi-Channel Scale Invariant Local Ternary Pattern (MC-SILTP)	Ma and Sang (2013) [132]
	(4) Spatio-Temporal LBP-based	
	Spatio-temporal Local Binary Patterns (STLBP)	Shengping et al. (2008) [174]
	Spatial-Temporal Local Binary Pattern (STLBP)	Shimada and Taniguchi (2009) [175]
	Stereo Local Binary Pattern based on Appearance and Motion (SLBP-AM)	Yin et al. (2013) [229]
	(5) Hybrid LBP-based	
	Spatial Extended Center-Symmetric Local Binary Pattern (SCS-LBP)	Xue et al. (2010) [226]
	Center Symmetric Spatio-temporal Local Ternary Pattern (CS-STLTP)	Xu (2012) [223]
	Center Symmetric Spatio-temporal Local Ternary Pattern (CS-STLTP)	Wu (2013) [218]
	Spatiotemporal Scale Invariant Ternary Pattern (ST-SILTP)	Ji et al. (2014) [100]

Table 2.2: Texture Features: An Overview (Part 2).

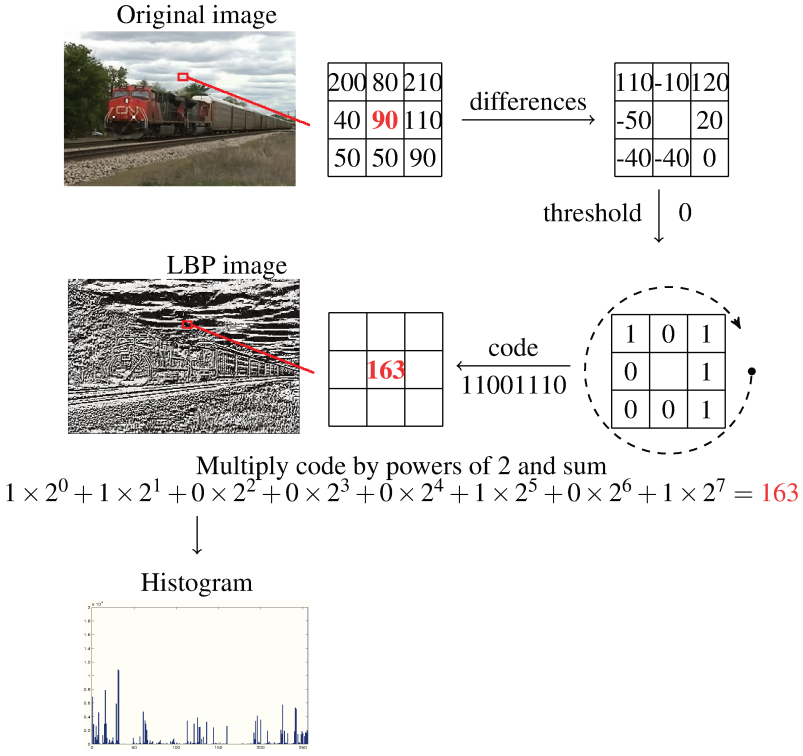


Figure 2.3: The LBP descriptor. From the original image to the histogram of its LBP image.

al. [47] proposed a robust texture-color based background modeling. Two variants of LBP, called εLBP and Adaptive εLBP, are developed in [206, 207]. They are fast to compute and less sensitive to the illumination variation or some color similarity between foreground and background.

– *Center-Symmetric LBP-based:* The computational complexity is very useful normally in background modeling which requires real-time processing. The center-symmetric LBP are based on descriptions which generates more compact binary patterns by working only with the center-symmetric pairs of the pixels. The variants of this category are extensions of the Center Symmetric Local Binary Pattern (CS-LBP) descriptor proposed by Heikkilä et al. [86]. The CS-LBP descriptor is less sensitive to noisy pixels and produces a short histogram while preserving robustness to illumination changes and slightly gaining in time consumption. Due to its performance, many other works based on CS-LBP have been proposed. For instance, in Xue et al [226], a Spatial Extended Center-Symmetric (SCS-LBP) is presented. It improves the CS-LBP by better capturing the gradient information and hence, making it more discriminative. The authors explained that their SCS-LBP produces a relatively short feature histogram with low computationally complexity. The Center-Symmetric Local

Derivative Pattern descriptor (CS-LDP) is described in [225]. It extracts more detailed local information while preserving the same feature lengths than the CS-LBP, but with a slightly lower precision than the ordinary LBP.

– *Ternary LBP-based:* This category represent the descriptors which inherit the characteristics from Local Ternary Pattern (LTP) introduced by Tan and Triggs [191]. This descriptor is more robust for local noises by introducing a small tolerative range. The intensity scale invariant property of a local comparison descriptor is very useful, because illumination variations, either global or local, usually cause sudden changes of gray scale intensities of neighboring pixels simultaneously. Nevertheless, Liao et al. [120] demonstrated that the LTP descriptor can not keep its invariance against scale transform when all local pixel values are multiplied by a constant. Therefore, to deal with these problems Liao et al. [120] presented a Scale Invariant Local Ternary Pattern (SILTP) descriptor. More recently, Ma and Sang [132] proposed to extend the SILTP to feature space and to operate on the three channels of RGB images rather than only one channel present in gray images to get the texture patterns. This texture descriptor is called Multi-Channel Scale Invariant Local Ternary Pattern (MC-SILTP). The MC-SILTP demonstrated all the properties that SILTP owns, and it can deal especially in flat areas.

– *Spatio-Temporal LBP-based:* The spatio-temporal category include the variants that extend the ordinary LBP from spatial domain to spatio-temporal domain. However, these variants can deal with dynamic scenes. In Shengping et al., [174], a novel spatio-temporal local binary patterns (STLBP) is presented. The experimental results indicate that the proposed method can adapt quickly to changes in the dynamic background. Yin et al. [229] proposed a Stereo Local Binary Pattern based on Appearance and Motion (SLBP-AM) descriptor. The motion of pixels is represented as dynamic texture in ellipsoidal domain. Then, Yin et al. [229] combined texture histograms in the XY, XT and YT planes in the ellipsoid. SLBP-AM is more robust to slight disturbance, but also adapts quickly to the large-scale and sudden changes. Shimada and Taniguchi [175] proposed an invariant feature using both spatial invariance and temporal invariance also called Spatio-Temporal LBP (STLBP) suitable for outdoor scene in which the illumination condition can change gradually.

– *Hybrid LBP-based:* These variants combine two or more characteristics of the above categories, which usually results in a descriptor even more powerful. Xue et al. [226] proposed to use a Spatial Center-Symmetric Local Binary Pattern (SCS-LBP) which not only has the property of illumination invariance, but also produces short histograms and be more robust to noise. So, Xue et al. [226] extended the CS-LBP operator from spatial domain to spatial-temporal domain and proposed a texture operator named SCS-LBP which extracts spatial and temporal information simultaneously. Then, combining the SCS-LBP operator with an improved temporal information estimation scheme, Xue et al. [226] obtained a background modeling approach which reach high accurate detection in dynamic scenes while reducing the computational complexity compared to the ordinary LBP. Wu et al. [218] extended the SILTP descriptor for handling some challeng-

ing scenes by introducing the Center-Symmetric Scale Invariant Local Ternary Pattern (CS-SILTP) descriptor. This texture descriptor explores the spatial and temporal relationships of neighborhood pixels.

Depth features

Depth features encapsulate the ***depth information*** and they have become very attractive for BS, especially, in indoor environments. The main advantage of the depth features is that it does not suffer the limitations of color features (e.g. camouflage). Depth-based detection results in a more compact silhouettes. However, using exclusive depth features still present some issues such as: depth sensors frequently raise noises at object boundaries; measurements of depth are not always available for all image pixels. Therefore, usually many BS works propose to combine both color and depth features to improve the detection results. Depth information can be obtained in real-time by different technologies. We describe below three technologies to acquire depth information.

- ***Depth from Stereo-Cameras:*** Traditional stereo cameras consist of a single device integrating two or more monocular cameras with small baseline (i.e., the distance between focal center of the cameras). The disparity map obtained that correlates the two views of a stereo camera can be used as input for a disparity-based BS algorithm. To accurately perform the background modeling, it is necessary that a dense disparity map be calculated. However, to obtain an accurated dense map of correlations between two stereo images, usually time-consuming stereo vision algorithms are employed [31, 121]. Moreover, the correlation between left and right images may not be reliable, and the disparity map can present holes due to "invalid" pixels (i.e., points with invalid depth values). Ivanov et al. [94] were among the first authors who proposed a BS method based on disparity maps to address some of these issues. By cross-verifying each pixel across three camera views, the authors were able to distinguish the foreground objects from occlusion/shadows. Practically, this method required the offline construction of disparity fields mapping the background images that contained no foreground objects. At runtime, foreground detection was made by checking background image to each of the additional auxiliary color intensity values at corresponding pixels. This algorithm could be implemented in real-time on conventional hardware. In Gordon et al. [**?**], the background model was modeled using a multidimensional mixture of Gaussians model with the (R,G,B,D) features. A significant advantage of incorporating both color and depth features within the background model is that, Gordon et al. [**?**] could correctly estimate depth and color of the background when the background is available in a fewer number of initialization frames. The authors used a disjunction of the results coming from each feature to obtain the final foreground detection. A pixel is classified as foreground based on either color or depth is taken to be foreground in the final foreground detection. Other related BS works can be found in [59, 81, 82].
- ***Depth from Time-of-Flight (ToF) Cameras:*** The ToF cameras produce a depth image, each pixel encodes the distance to the corresponding point in the scene. Apart from their advantages of high frame rates and ability to capture the scene all at once, ToF

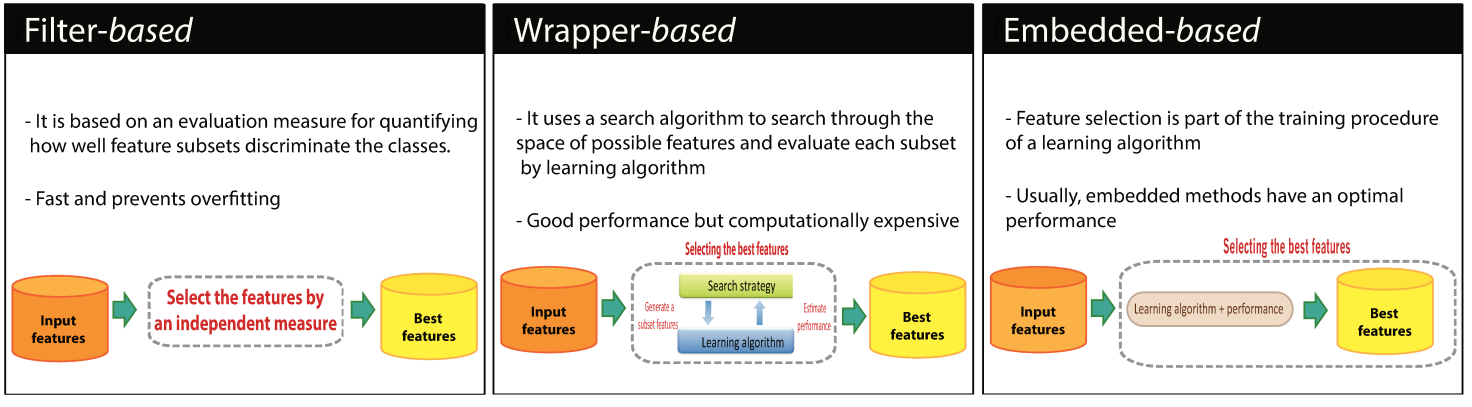
based cameras have generally the disadvantage of low resolution. In Leens et al. [117], color and depth features were obtained with a low resolution from ToF camera. The ViBe algorithm [12] is applied independently to the color and the depth features. Then, the obtained foreground masks are then combined with logical operations and then post processed with morphological operations. Stormer et al. [189] used a MoG model [187], where depth and infrared features are combined to detect foreground objects in the case of close or overlapping objects. Two independent background models are built. Each pixel is classified as background or foreground only if the two models matching conditions agree. But a failure of one of the models affects the final pixel classification. In Tombari et al. [197], an algorithm for automatic graffiti detection is presented. The algorithm compares the current intensity information with a model of the background to detect the scene changes. Next, the depth information was used for distinguishing between changes occurring in the space between the background and the ToF camera (e.g. intrusion). It presented low rate of false positives, and it can operate in a real-time manner. As the authors used a basic BS for the intensity data, the proposed method may fail by the presence of both slow and sudden changes in the scene's illumination. Hu et al. [91] realized the foreground detection by using a weighted average on the probabilities obtained from the MOG model [187]. The different weights are updated adaptively for each output of the classifier by considering foreground detections in the previous frames and the depth feature. Experimental results [91] showed that the proposed approach can effectively solve the limitations of color-based or depth-based detection.

- ***Depth from RGB-D Cameras:*** Recently, low cost RGB-D cameras such as the Microsoft's Kinect or the Asus's Xtion Pro are widely used to improve background modeling. However, the RGB-D cameras based on structured light scanner (i.e., Microsoft Kinect) are not usually suitable for outdoor environments, due to the range limitation and errors introduced by interference with the sunlight. Several BS work using Microsoft Kinect are found in the literature. For example, Camplani et al. [35] used a multiple region-based classifiers in a mixture of experts fashion to improve the final foreground detection. It is based on multiple background models that provide a description at region and pixel level by considering the color and depth features. In Camplani et Salgado [36], the combination of the four models (pixel-color, region-color, pixel-depth, region-depth) was based on a weighted average to efficiently adapt the contribution of each classifier to the final classification. Another BS algorithm based on RGB-D camera to make the background and foreground models more robust to effects such as camouflage and illumination changes was proposed by Spampinato et al. [62] and Fernandez-Sanchez [183]. The authors modeled the background and foreground scenes with a Kernel Density Estimation (KDE) [58] in a quantized x-y-hue-saturation-depth space after a preprocessing stage for aligning color and depth data and for filtering/filling noisy depth measurements. Experimental results in three different indoor environments, with different lighting conditions, showed that this approach achieved an accuracy in foreground segmentation over 90% that the combination of depth data and illumination-independent color space proved to be very robust against noise and illumination changes. More works can be seen in: [66, 73, 119].

Motion features

The motion features provide ***temporal information*** and they are useful to handle dynamic scenes, containing natural elements such as fountains, swaying trees or ocean ripples [24, 149]. The motion features are usually obtained via optical flow to deal with irrelevant motions in the background. The majority of the optical flow algorithms are computationally slow. Three alternative approaches are then used to introduce temporal attributes: **1)** the ones based only on the difference between consecutive frames. Then, the background model is only computed on stationary regions of the scene, **2)** optical flow (computed on all pixels) which is used to detect moving areas. The background model is only computed in stationary areas, and **3)** optical flow is only computed on moving areas after foreground detection. In this case, optical flow allows the algorithm to distinguish the unimportant moving areas from the moving objects. Different approaches have been proposed to extract motion features. We review in the following paragraphs the main existing ones.

Huang et al. [92] presented a dense optical flow for describing motion vectors. Regions with coherent motion are then extracted as initial motion markers. Pixels not assigned to any region are labeled uncertain ones. Finally, a watershed algorithm based on motion and color is used to associate uncertain pixels to the nearest similar mark. Further, Markov Random Fields (MRFs) [107] are used to formulate the foreground detection as a labeling problem. The optimization over the MRF model is then performed. The posterior probabilities initialized with the ones computed with the MOG model [208] are maximized to obtain the final classification result. Finally, regions which have the same classification label and similar colors are merged to derive a more consistent foreground mask. Experimental results [92] on gradual illumination changes and shadows demonstrated the robustness of this method, but the computational complexity of this technique has not been mentioned. In similar studies, Huang et al. [93] used motion information captured through the difference of consecutive frames to model the background in stationary areas. Using the EPPM [11], Chen et al. [45] ensured temporally-consistent background subtraction with optical flow estimation by tracking the foreground pixels. Here, motion information is integrated with a temporal M-smoother. A similarity measurement is obtained directly from optical flow estimation with the assumption that the background estimate for the same object appearing in the difference video frames should be identical. As the direct implementation of EPPM [11] is extremely slow as optical flow estimation is required between any two video frames, Chen et al. [242] developed a recursive implementation so that optical flow estimation is required only between every two successive frames. As described in previous approaches, the background model is initially obtained using the MOG model [187]. Then, a spatial and a temporal M-smoother are employed to obtain a spatially-temporally-consistent foreground mask. Experimental results [45] on the ChangeDetection.net dataset [69] and SABS dataset [32] showed this algorithm outperforms most of state-of-the-art algorithms. Using multiple features, Zhong et al. [241] proposed to fuse texture (εLBP [206]) and motion patterns. For each pixel, its probability to be either a background or foreground is computed from the histogram of each feature. Then, the results are combined using a weighted average mechanism. Experimental results [241] showed that the combination of εLBP and motion pattern outperforms the ordinary LBP in presence of dynamic backgrounds.



(a) Filter-*based*. (b) Wrapper-*based*. (c) Embedded-*based*.

Figure 2.4: Three traditional approaches for feature selection.

2.2 Feature selection in background modeling

Most of background subtraction methods use a uniform feature map for all pixels of the scene, disregarding the non-uniformity of the distribution of the background properties [181]. Moreover, the importance of different features on particular regions of the image is still ignored. In practice, however, for a complex scene comprising of several elements such as waving trees, sky, soil and car, the most discriminant features for these elements are probably different, and therefore a single-feature background subtraction algorithm may not be appropriate. Despite the choice of the best features for each region is not an easy task as it requires a deep knowledge of the scene, it is possible to automatically select the most relevant features, and this process is commonly defined as *feature selection* [28]. Feature selection (known as subset selection, attribute selection or variable selection) is the task of selecting a small subset of features that is enough to predict the target labels well. Three key benefits of performing feature selection on the data are [167]:

1. ***Reduced computational complexity:*** Feature selection helps to reduce the computational complexity of learning algorithms improving its prediction performance. Some learning algorithms can becoming computationally intractable when there a large number of features either in the training step as in the prediction step. When we find a small set of features that allows a good prediction of the labels, we can exclude the rest of irrelevant features. Therefore, in the prediction step we only have to measure a small set of features for each instance.
2. ***Improved accuracy:*** It is possible to improve the prediction accuracy by applying initially a feature selection method. Many of the state-of-the-art learning algorithms can given predictions greatly skewed by the presence of a big number of irrelevant or weakly relevant features. In contrast, even the simple learning algorithms may yield good performance if a a small set of good features has been previously selected.
3. ***Problem understanding:*** Normally, the key of solving an specific problem is by understanding it better. Feature selection methods can contribute to better understanding the problem at hand by selected the most useful information from a feature set.

In the background subtraction field, the use of feature selection methods have been less studied so far. Nevertheless, the feature selection can be used to improve the detection of the foreground objects [149]. This is possible due to its capability to select a subset of highly discriminant features removing the irrelevant and redundant ones. Traditionally, feature selection methods can be categorized into three main groups: filter, wrapper and embedded *-based* methods. Recent works have also proposed the use of ensemble-*based* approaches for feature selection [23, 163]. Following this, we discuss later each of these approaches and their main BS works.

2.2.1 Traditional approaches for feature selection

There are three general state-of-the-art approaches for feature selection: filter-*based*, wrapper-*based* and embedded-*based* [128, 186]. Figure 2.4 shows a brief overview these approaches.

Filter-based The filter-*based* methods were the early approaches for feature selection. The filter-*based* methods evaluate the relevance of the features based on a statistical measure estimated directly from the data to assign a score to each feature without involving any classification algorithm [48, 127, 199]. The filter methods are generally much computationally efficient and practical than wrapper methods (discussed later), especially for using it on high dimensional data. Nonetheless, it tends to select subsets with a high number of features (even all the features) and so a threshold is required for the choosing of a subset. The representation of the filter-*based* is shown in Figure 2.4a. A general filter-*based* algorithm is presented Algorithm 1 [128].

Given a training set $X = \{x_1, x_2, ..., x_N\}$ where each x_j $(j = 1, ..., N) \in \mathbb{R}^p$, the algorithm can start with one of the subsequent subsets of S_0 such as $S_0 = \{\phi\}$ or $S_0 = \{NULL\}$ or $S_0 \subset X$. An independent measure ϑ evaluates each created subset S and compares it to the previous best subset. The search iterates until a predefined stopping criterion Υ is reached. Some commonly used stopping criteria are described by Liu and Yu [128]. Lastly, the algorithm outputs the last current best subset S_{best} as the final result. Note that by changing the search strategies and evaluation measures used in Steps 5 and 6 in the Algorithm 1, we can design diversified filter-*based* algorithms.

Algorithm 1 A generalized filter-*based* approach

1: **Require:** A training set X, a feature subset S_0, a stopping criterion Υ, an independent measure ϑ
2: $S_{best} = S_0$
3: $\varphi_{best} = eval(S_0, \vartheta)$ {evaluate S_0 by using an independent measure ϑ}
4: **repeat**
5: $S = generate(X)$ {generate a subset for evaluation}
6: $\varphi = eval(S, \vartheta)$ {evaluate the current subset S by ϑ}
7: **if** $(\varphi > \varphi_{best})$ **then**
8: $\varphi_{best} = \varphi$
9: $S_{best} = S$
10: **end if**
11: **until** (Υ *is reached*)
12: **Output:** An optimal subset S_{best}

Wrapper-based The wrapper-*based* methods employ a learning algorithm as a "black box" for selecting a set of relevant features. Commonly, in this approach a learning algorithm is run over the entire training set and then measured against the testing set, or a cross-validation method can be used. This approach tends to give superior performance than the filter ones, but it is also more computationally expensive since we have to re-train the learning algorithm in each step. A representation of the wrapper-*based* method is shown in Figure 2.4b. The general wrapper approach (see Algorithm 2 [128]) is very similar to the general filter one except that it uses a predefined learning algorithm A instead of an independent measure ϑ for the subset evaluation. In a wrapper-*based* algorithm, for each created subset S, it evaluates its kindness by using the learning algorithm to the data with feature subset S and evaluating the quality of mined results. Nonetheless, different learning algorithms will provide different feature selection results. Note that it is possible to propose different wrapper-*based* algorithms by changing the function *generate()* and learning algorithms A.

Algorithm 2 A generalized wrapper-*based* approach

Require: A training set X, a feature subset S_0, a stopping criterion Υ, a learning algorithm A
$S_{best} = S_0$
$\varphi_{best} = eval(S_0, A)$ {evaluate S_0 by using a learning algorithm A}
repeat
  $S = generate(X)$ {generate a subset for evaluation}
  $\varphi = eval(S, A)$ {evaluate the current subset S by A}
  if $(\varphi > \varphi_{best})$ **then**
    $\varphi_{best} = \varphi$
    $S_{best} = S$
  end if
until (Υ *is reached*)
Output: An optimal subset S_{best}

Embedded-based The embedded methods is normally used to describe selection which is done automatically by the learning algorithm. Decision trees [155], the artificial neural networks with pruning of input neurons [114] and L1-SVM [143] are examples of methods in this category. The embedded-*based* approach interact to the learning algorithm with a lower computational cost than the wrapper-*based*. An illustration of the embedded-*based* approach is shown in Figure 2.4c. This approach employs the independent criteria to determine the best subsets for a known cardinality, and then uses the learning algorithm to choose the final best subset among the best subsets across distinct cardinality (number of elements of the set). An embedded algorithm usually initiates with an empty set S_0 by using sequential forward selection (start with an empty set of features and add features one at a time). For the best subset of cardinality c, it is searching all suitable subsets of cardinality $c+1$ adding a feature from the leftover subsets. A subset created at cardinality $c+1$ is evaluated by independent criterion ϑ and compared with the previous best subset. Next, the learning algorithm A is used to the current best subset, and performance Π is compared with the performance of the best subset at cardinality c. The algorithm continue looking for the best subset until S'_{best} is better; otherwise, it stops and return the current best subset as the final best subset. A generalized embedded procedure is shown in Algorithm 3 [128].

2.2.2 Ensemble learning for feature selection

Ensemble learning is a powerful tool in the field of machine learning and its efficiency has been demonstrated in several studies [125, 126, 159]. The main idea of ensemble learning is to combine a set of models, where each of them solves the same task in order to obtain a better global model with more robustness and the generalization ability than a single model. In the same way as in the classification tasks, ensemble learning might be employed to improve the robustness of feature selection approaches. Traditional feature selection approaches has concentrated on finding the suitable subset of significant features to be used for learning an inference model through classification or regression. In recent decades, a new kind of feature selection that uses ensemble learning to select features, called *ensemble for feature selection* has been introduced [3, 163, 173]. This approach extends the traditional feature selection methods by looking for a set of feature subsets that will favour disagreement among

Algorithm 3 A generalized embedded-*based* approach

Require: A training set X, a feature subset S_0, a learning algorithm A, an independent measure ϑ
$S_{best} = S_0$
$\varphi_{best} = eval(S_0, \vartheta)$ {evaluate S_0 by using an independent measure ϑ}
$\Pi_{best} = eval(S_0, A)$ {evaluate S_0 by using a learning algorithm A}
$C_0 =$card(S_0) {cardinality calculation of S_0}
for $c = C_0 + 1 : N$ **do**
  for $t = 0 : N - c$ **do**
    $S = S_{best} \cup \{x_t\}$ {subset generation for evaluation with cardinality t, where $x_t \in X$}
    $\varphi = eval(S, \vartheta)$ {evaluation the current subset S by ϑ}
    if $(\varphi > \varphi_{best})$ **then**
      $\varphi_{best} = \varphi$
      $S'_{best} = S$
    end if
    $\Pi = eval(S'_{best}, A)$ {evaluating subset S'_{best} by A}
    if $(\Pi > \Pi_{best})$ **then**
      $S_{best} = S'_{best}$
      $\Pi_{best} = \Pi$
    else
      break and **return** S_{best}
    end if
  end for
end for
Output: An optimal subset S_{best}

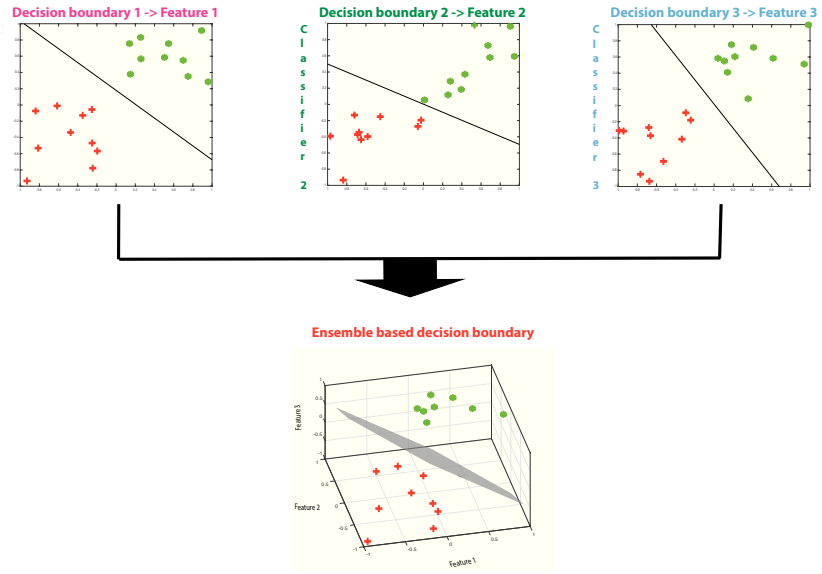


Figure 2.5: Combining an ensemble of classifiers with different features for reducing classification error.

the ensemble members. The ensemble for feature selection may increase the probability of choosing a stable feature subset, i.e once the goal of feature selection process is fixed, the meaningful features should not change for different samples of the data. For instance, in real applications are usually required that the algorithms to select features are always consistent even if new samples are added to the data. The ensemble for feature selection can also provide a better approximation to the appropriate subset or ranking of features avoiding feature subsets which can be regarded local appropriate in the space of feature subsets. Lastly, the ensemble for feature selection can expands the search space by aggregating the outputs of many classifiers allowing that optimal subsets can be achieved [163]. Note that not all ensemble methods can be considered as a feature selector. We regard that an ensemble is a feature selector when different subsets of features are used for different base classifiers in an ensemble. In this context, each ensemble member is associated with its own feature or feature subset, which can, for example, be selected by a particular feature selection algorithm or randomly sampled from the original pool of features [87]. Figure 2.5 graphically shows this concept, where each classifier is trained with different feature(s) to differentiate two classes. The combination of the (three) classifiers provide the best decision boundary. We will discuss in more detail each of ensemble for feature selection steps below.

Building an ensemble for feature selection algorithm

An ensemble for feature selection should be composed of mutually complementary individual classifiers which are characterized by the high diversity and accuracy. Otherwise, there may be a risk of the increasing the computational complexity, in addition, combining similar classifiers must not contribute much to the combined classifier under construction [215, 233]. There are usually three strategies necessary for building a successful ensemble feature selection algorithm: ***1)*** creating a set of diversified base/weak classifiers; ***2)*** ensemble pruning; and ***3)*** combining classifiers.

***1)* Creating a set of diversified base/weak classifiers** The diversity of the classifier outputs is considered a key requirement for the great success of an ensemble algorithm [20, 234]. Whenever all base classifiers produce the same output, there is nothing to be acquired by their combination. Therefore, it is necessary that the decisions of ensemble members are diversified, especially when they are making error. Random subspace methods [87] and the Boosting for feature selection [203] are two very popular strategies to generate a diverse set of classifiers in an ensemble feature selection. Each of these approaches are described below.

- ***Random Subspace:*** The random subspace approach is be able to handle issues with a huge number of features. It employs different feature subsets to train the ensemble members. Random subspace method generate each classifier in the ensemble from a randomly chosen subset of predefined features [87]. Therefore, the diversity is ensured by providing the base classifier different views (or projections) of the data. Like such views are generated randomly from a big feature set, it is very possible that every base classifier gets a different prospective of the data, which takes to the discovery of diverse and complementary structures in the data. Finally, the M classifiers are usually

combined by simple majority voting in the final decision rule [102]. The random subspace procedure is presented in the Algorithm 4 [243].

Algorithm 4 The random subspace algorithm

1: **Require:** Classifier training procedure, training set X, subspace dimension p^*, number of iterations M
2: $k \leftarrow 1$
3: **repeat**
4: $S_k \leftarrow$ SelectRandomSubspace(X,p^*)
5: Train k-th classifiers on S_k
6: $k \leftarrow k+1$
7: **until** $k > M$
8: **Output:** Combine outputs of M trained base classifiers usually according the Eq. (2.2).

- ***Boosting for feature selection:*** Boosting refers to a set of algorithms that allow to convert weak learners to strong ones. The AdaBoost (Adaptive Boosting) is a popular implementation of boosting proposed for the first time by Freund and Schapire (1996) [63]. It works by repeatedly running a weak learner on various distributed training set, then, the weak learning are combined into a single strong classifier. The aim is to find a final classifier with a low prediction error rate. A few years later, the AdaBoost version to select a number of relevant features from a high number of potential features was proposed in [129, 203, 204]. The AdaBoost for feature selection is a simple modification of the standard AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a single feature. When the classifiers are combined, a much better performance can be achieved than what can be achieved by a single classifier. The key idea behind this algorithm is concentrate on the samples which are harder to classify, increasing their representation in successive training sets. In the AdaBoost for feature selection, M features and weak classifiers are chosen to compose the final strong classifier over a number of M rounds. In each of the iterations, the space of all possible features is searched extensively to find the optimal weak classifier with the smaller weighted classification error. The error is then employed to update the weights such that the wrongly classified samples get weights increased. The final strong classifier is a weighted linear combination of all M selected weak classifiers. Details of the AdaBoost for feature selection is presented in Algorithm 5 [203, 204]. In addition to AdaBoost, some others boosting variants, such as RealBoost [149] and XGBoost [46] have also been proposed for feature selection.

2) **Ensemble pruning** An important issue in an ensemble method is to decide how many base classifiers should be used. Ensemble pruning, also known as ensemble selection (or selective ensemble) aims to select a subset of individual base classifiers to form the whole ensemble. Many ensemble algorithms do not include this additional intermediate phase into prior to combination of the base classifiers. Nonetheless, some authors have demonstrated both theoretical and empirical that ensemble selection can improve the generalization performance of ensemble, therefore, the ensemble selection phase may reach better performance than the original ensemble [13, 136]. Furthermore, a great number of base classifiers in an

Algorithm 5 The AdaBoost for feature selection

1: **Require:** Training set $(x_1, y_1), ..., (x_N, y_N)$ where $x_i \in X$, $y_i \in Y = 0, 1$ for negative and positive examples respectively, number of iterations M
2: $k \leftarrow 1$
3: Initialize weights $w_{1,i} = \frac{1}{2b}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where b and l are the number of negative ans positive examples respectively.
4: **repeat**
5: Normalize the weights $w_{k,i} \leftarrow \frac{w_{k,i}}{\sum_{j=1}^{N} w_{k,j}}$ so that w_k is a probability distribution.
6: For each, ρ_j, train a classifier Ψ_j which is restricted to using a single feature. The error is evaluated with respect to w_k, $error_j = \sum_i w_i \mid \Psi_j(x_i) - y_i \mid$.
7: Choose the classifier Ψ_k, with the lowest $error_k$
8: Update the weights: $w_{k+1,i} = w_{k,i} \nu_k^{1-e_i}$
9: where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\nu_k = \frac{e_k}{1-e_k}$
10: $k \leftarrow k+1$
11: **until** $k > M$
12: **Output:** The strong classifier is:

$$H(x) = \begin{cases} 1 & \text{if } \sum_{k=1}^{M} \beta_k \Psi_k(x) \geq \frac{1}{2} \sum_{k=1}^{M} \beta_k \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

13: where $\beta_k = \log \frac{1}{\nu_k}$

ensemble demand large memory and computational overhead. Consequently, this will result in an increase of the training cost, storage demands, and prediction time. According to Rokach and Maimon [162], there are four factors that may determine the size of an ensemble: ***1)*** suitable number of base classifier should be chosen to achieve the desired accuracy in an ensemble. A study conducted by Hansen and Salamon [80] showed that ten classifiers are usually sufficient to reduce the error rate; ***2)*** the size limit of ensemble should be predefined to preventing from increasing computational cost and the decreasing comprehensibility between the base classifiers; ***3)*** the nature of the classification problem can be responsible by the number of base classifiers in an ensemble; and ***4)*** the quantity of processors available for parallel learning can also be used as parameter to define the number of base classifiers in an ensemble. There are three approaches for determining the ensemble size [161, 162]:

- ***Pre-selection of the ensemble size:*** In this category, the user can define the ensemble size by "number of iterations", (such as in the Random Subspace, Bagging, etc.) or by the nature of the classification problem (such as in the Error-Correcting Output Coding (ECOC) [110]).
- ***Selection of the ensemble size while training:*** The algorithms belonging to this category attempt to define the best ensemble size during the training. Normally, while new classifiers are introduced to the ensemble, these algorithms verify if the contribution of the last classifier to the ensemble performance is still meaningful. Otherwise, the ensemble algorithm stops. These algorithms often also have a controlling parameter that limits the size of ensemble as in the previous category.
- ***Pruning-post selection of the ensemble size:*** This category allows the ensemble grow

freely and thereafter prune the ensemble to obtain small and efficient ones. Post selection of the ensemble typically uses performance metrics, such as accuracy, cross entropy, mean, precision, etc. This approach can be separated into two categories: pre-combining and post-combining approaches. Pre-combining pruning is realized before combining the base classifiers whereas in post-combining, the base classifiers are eliminated based on their contribution with others.

***3)* Combining classifiers** The last stage for any ensemble feature selection algorithm is the combination of the outputs of several base classifiers. There different methods to combine classifiers, however the scheme for combining which is going to be utilized, in part, depends on the type of classifiers used as ensemble member. For instance, the majority voting is typically used for classifiers that give discrete-valued label outputs. Nonetheless, there is a variety of scheme for combining classifiers that give continuous outputs, such as arithmetic (sum, product, mean, etc.), voting-based methods, etc. A detailed review of the different kinds of combiners can be found in [112, 161, 243]. In this thesis, only some of the most common methods for combining classifiers will be explained. Given the output of each classifier k is a i-long vector $q_{k,1},...,q_{k,i}$. The value $q_{k,j}$ corresponds to the support that the sample x belongs to the class j according to the classifier k. For simplicity, it is also defined that $\sum_{j=1}^{i} q_{k,j} = 1$. If we are dealing with a crisp classifier k, which attributes the sample x to a determined class l, therefore it can still be transformed to i-long vector $q_{k,1},...,q_{k,i}$ such that $q_{k,l} = 1$ and $q_{k,j} = 0, \forall j \neq l$ [161].

- ***Majority voting:*** Majority voting is a simple and most intuitive method for combining classifier outputs. A comprehensive analysis of the majority voting approach can be found in [112]. Basically, the combining scheme classify an unlabeled sample by counts the votes for each class over the input classifiers and choose the majority class. Mathematically, majority voting can be expressed as follows:

$$H(x) = \arg\max_{\omega_k \in Y} \sum_{i=1}^{M} \mathbb{I}(h_i(x), \omega_k) \tag{2.2}$$

  where $h_k(x)$ is the classification of the k-th classifier and $\mathbb{I}(h, \omega)$ is an indicator function defined as:

$$\mathbb{I}(h, \omega) = \begin{cases} 1 & \text{if } h = \omega \\ 0 & \text{if } h \neq \omega \end{cases}$$

- ***Weighted majority voting:*** This approach consists in combining the base classifiers assigning weights for each of them. The more competent classifiers will have greatest power in the final decision. Normally, the classifiers' weight can be determined either upon preliminary information or based on their performance for a certain validation set. More details on weighted majority voting can also be found in [124]. In mathematical terms, the weighted voting can be given as:

$$H(x) = \text{sign}(\sum_{i=1}^{M} \beta_i (h_i(x), \omega_k)) \tag{2.3}$$

where β_i is the weight of each classifiers.

- ***Bayesian combination:*** In the Bayesian combination approach the classifiers' weight is a posterior probability of the classifier given the training set [34].

$$H(x) = \arg\max_{\omega_k \in Y} \sum_{i=1}^{M} P(\Psi_i|X)\hat{P}_{\Psi_i}(Y = \omega_k|x) \tag{2.4}$$

  where $P(\Psi_i|X)$ indicates the probability that the classifier Ψ_i is correct given the training set X. The estimation of $P(\Psi_i|X)$ depends on the classifier's representation.

- ***Näive bayes:*** Considering that the classifiers are mutually independent given a class label (conditional independence), the Bayes' rule can be used for combining various classifiers.

$$H(x) = \arg\max_{\substack{\omega_j \in Y \\ \hat{P}(Y=\omega_j)>0}} \hat{P}(Y = \omega_j) \prod_{i=1} \frac{\hat{P}_{\Psi_i}(Y = \omega_j|x)}{\hat{P}(Y = \omega_j)} \tag{2.5}$$

2.2.3 Feature selection in background subtraction

Surprisingly, a little BS works based on feature selection have been done to date. Some works based on the traditional feature selection methods are presented below. For instance, Li et al. [118] presented one of the first works based on this category. The authors introduced a novel method to detect changes based into static and dynamic pixels in accordance with inter frame changes. The Bayes decision theory is used for classification of a certain pixel in static or dynamic class. The static pixels belong to stationary objects, and they are described by color and gradient statistics whereas dynamic pixels belong to non-stationary, and the are represented by color co-occurrence statistics. According to Li et al. [118], the proposed method can be affected by the problem of intermittent object motion, since the statistics are associated to each individual pixel without considering its neighborhood. Furthermore, the method can mistakenly learn the features of non-stationary objects as stationary if crowded foreground objects are frequently showed in the scenes. In Javed et al. [98], a simple dynamic feature selection scheme for background scenes is proposed. An Online Robust Principal Component Analysis (OR-PCA) with dynamic feature selection provides a framework to select multiple features frame by frame. The means and variances are used as a criterion for selecting the best features. The authors mentioned that the potential problem of the proposed approach is the time computation, since features are extracted from every incoming video block. Most recently, Braham and Van Droogenbroeck [28] presented a generic feature selection method for background subtraction. The authors proposed a strategy for selecting the best features by comparing the current input feature values with local background ones. Initially, local feature background models are created from a set of features. Then it checks, if the each model predicts the correct class of input samples. Finally, the best feature/threshold combination is selected by a performance metric computed from a confusion matrix. Experiments conducted on the ViBe algorithm [12] showed that the proposed feature selection method improves the segmentation results.

Authors/Date	*Strategy*	*Level*	*Features*
	Traditional methods		
Li et al. (2004) [118]	Bayes decision rule	Pixel	RGB, gradient, and color co-occurrence
Javed et al. (2015) [98]	Means and variances criterion	Region	RGB, gray, LBP, gradients, and HOG
Braham and Van Droogenbroeck (2015) [28]	Performance metric	Pixel	RGB, HSV, and YCbCr
	Ensemble-based		
Grabner and Bischof (2006) [70, 72]	AdaBoost	Region	Haar-like features, orientation histograms and LBP
Parag et al. (2006) [149]	RealBoost	Pixel	RGB, gray, and gradients
Grabner et al. (2008) [71]	AdaBoost	Region	Haar-like features
Klare and Sarkar (2009) [109]	Ensemble of Mixture of Gaussians	Pixel	RGB, gradients, and Haar-like features.

Table 2.3: The main BS works based on features selection approaches.

In the last decades, some papers have been published addressing the ensemble for feature selection for the BS context. Most ensemble for feature selection algorithms for BS use widely the boosting and its variants. In Grabner and Bischof [70, 72], a feature selection framework using the online AdaBoost [64] is introduced for the BS task. In the learning step a weak classifier is created for all image patches supposing that all input images are positive samples. For this purpose, the gray value of each pixel is given as uniformly distributed, the Haar features are computed by standard statistics the parameters of the negative distribution and, the orientation histogram features consists of equally distributed orientations. Afterwards, the new input images are analyzed, and the background model is updated. According to the authors, this method is robust to illumination changes and dynamic backgrounds since the classifiers are consistently updated. However, this approach has many restrictions concerning robust adaptiveness. To overcome this limitation, Grabner et al. [71] introduced a controllable time dependency into online boosting. The algorithm used an exponential forgetting of the samples over time and a simple sum-rule is used in the method to adjusting its temporal behavior to the underlying scene by using a control system that regulates the model parameters (e.g. errors, and importance). Parag et al. [149] proposed a generic model that is capable of automatically selecting the features that obtain the best invariance to the background changes while maintaining a high detection rate for the foreground detection. In this study, the authors proposed the use of a RealBoost algorithm [168]. Unlike AdaBoost algorithm which combines weak hypotheses having outputs in $\{-1, +1\}$, RealBoost algorithm computes real-valued weak classifiers given real numbered feature values, and generates a linear combination of these weak classifiers that minimizes the training error. To generate the background model, Parag et al. [149] used the Kernel Density estimation (KDE) [58] into RealBoost algorithm to select the most appropriate features for each pixel. The authors used 9 types of features, such as three color values R, G, B and spatial derivatives for each of these color channels in both x and y directions for each pixel of a color image. According to authors, once trained, the algorithm is able to adequately detect the moving objects unless there are some structural changes in the scene. In Klare and Sarkar [109], an ensemble of 13 Mixture of Gaussians (MoG) classifiers is presented. Each classifier uses exclusively one of the 13 (e.g. RGB, gradients, and Haar-like) features from the feature set, then they are fused using equally weighted hypotheses, resulting in a single hypothesis. The experimental results showed an evident improvement compared to the original MoG algorithm that uses only color intensities. The main BS works based on feature selection reported here as well as its principal differences are shown in Table 2.3.

2.3 Conclusion

As discussed in this chapter, numerous approaches for background subtraction have been proposed until the present date. However, there still exist open research questions to be investigated, as for example no traditional algorithm today still seem to be able to simultaneously address all the key challenges of illumination variation, dynamic camera motion, cluttered background and occlusion. We believe that an way of solving this issue is by the systematic investigation concerning the role and importance of features within background modeling and foreground detection. In the next chapters of this thesis, we tackle the problem by starting proposing a new descriptor that produces a short histogram while preserving robustness to illumination changes. Moreover, this novel descriptor is less sensitive to noisy pixels too. Furthermore, we present a feature selection approach to select automatically the best features for different pixels/regions of the image, and the more relevant ones are used for foreground segmentation.

Chapter 3

A novel texture descriptor for background subtraction in videos

In this chapter, we propose an eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) descriptor for background modeling and subtraction in videos. By combining the strengths of the ordinary LBP and the similar Center-Symmetric (CS) ones, it is robust to illumination changes and noise, and produces short histograms, too. The experiments conducted on both synthetic and real videos (from the Background Models Challenge) of outdoor urban scenes under various conditions showed that the proposed XCS-LBP outperforms its direct competitors for the background subtraction task. The work presented here was published at the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Berlin, Germany (oral presentation) [176]. The reader can find the related source code on Matlab at[1].

3.1 Motivation

Recently, a variety of local texture descriptors have been attracted great attention for background modeling, especially the Local Binary Pattern (LBP) because it is simple and fast to compute. Figure 3.1 (*top*) shows how a (center) pixel is encoded by a series of bits, accordingly to the relative gray levels of its circular neighboring pixels. It shows great invariance to monotonic illumination changes, do not require many parameters to be set, and have a high discriminative power. However, the ordinary LBP descriptor in [146] is not efficient for background modeling because of its sensitivity to noise, see Figure 3.1 (*bottom*) where a little change of the central value greatly affects the resulting code.

The LBP feature of an image consists in building a histogram based on the codes of all the pixels within the image. As it only adopts first-order gradient information between the

[1] https://fr.mathworks.com/matlabcentral/fileexchange/49815-xcs-lbp-descriptor-for-background-modeling-and-subtraction-in-videos

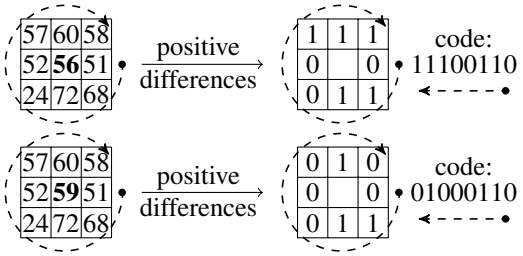


Figure 3.1: Examples of LBP encoding

center pixel and its neighbors, see [225], the produced histogram can be rather long. We have mentioned in the literature review (Chapter 2) that the Center-Symmetric LBP variants have been proposed to address this problem. It generates more compact binary patterns by working only with the center-symmetric pairs of pixels. In this chapter, we propose a Center-Symmetric LBP variant by introducing a new neighboring pixels comparison strategy that allows the descriptor to be less sensitive to noisy pixels and to produce a short histogram, while preserving robustness to illumination changes and slightly gaining in time consumption when compared to its direct competitors.

The rest of this chapter is organized as follows. The new descriptor that we propose is described in Section 3.2. Comparative results obtained on both synthetic and real videos are given in Section 3.3. Finally, the conclusion drawn at the last section closed the Chapter 3.

3.2 Proposed XCS-LBP descriptor

The ordinary LBP descriptor introduced by [146] has proved to be a powerful local image descriptor. It labels the pixels of an image block by thresholding the neighborhood of each pixel with the center value and considering the result as a binary number. The LBP encodes local primitives such as curved edges, spots, flat areas, etc. In the context of BS, both the current image and the image representing the background model are encoded such that they become a texture-based representation of the scene.
Let a pixel at a certain location, considered as the center pixel $c = (x_c, y_c)$ of a local neighborhood composed of P equally spaced pixels on a circle of radius R. The LBP descriptor applied to c can be expressed as:

$$LBP_{P,R}(c) = \sum_{i=0}^{P-1} s\,(g_i - g_c)\; 2^i \tag{3.1}$$

where g_c is the gray value of the center pixel c and g_i is the gray value of each neighboring pixel, and s is a thresholding function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

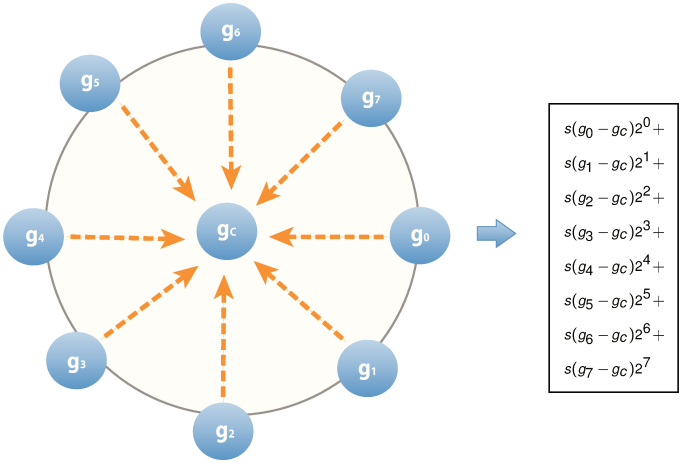


Figure 3.2: The LBP descriptor.

From (3.1), it is easy to show that the number of binary terms to be summed is $\sum_{i=0}^{P-1} 2^i = 2^P - 1$, so that the length of the resulting histogram (including the bin-0 location) is 2^P. An illustration of the LBP descriptor t is shown in Figure 3.2. The underlying idea of CS-LBP in [86] is to compare the gray levels of pairs of pixels in centered symmetric directions instead of comparing the central pixel to its neighbors. Assuming an even number P of neighboring pixels, the CS-LBP descriptor is given by:

$$CS-LBP_{P,R}(c) = \sum_{i=0}^{(P/2)-1} s(g_i - g_{i+(P/2)})\, 2^i \tag{3.3}$$

where g_i and $g_{i+(P/2)}$ are the gray values of center-symmetric pairs of pixels, and s is the thresholding function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x > T \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where T is a user-defined threshold. Since the gray levels are normalized in [0,1], the authors recommend to use of a small value. We will set it to 0.01 in the experiments presented in Section 3.3. By construction, the length of the histogram resulting from the CS-LBP descriptor falls down to $1 + \sum_{i=0}^{P/2-1} 2^i = 2^{P/2}$. For BS, the CS-LBP encodes the two images to be compared as texture-based images with a lower quantization that slightly favors robustness.

We propose to extend the CS-LBP descriptor by comparing the gray values of pairs of center-symmetric pixels so that the produced histogram are short as well, but considering the central pixel also. This combination makes the resulting descriptor less sensitive to noise for the BS application. The new LBP variant, called XCS-LBP (eXtended CS-LBP), expresses as:

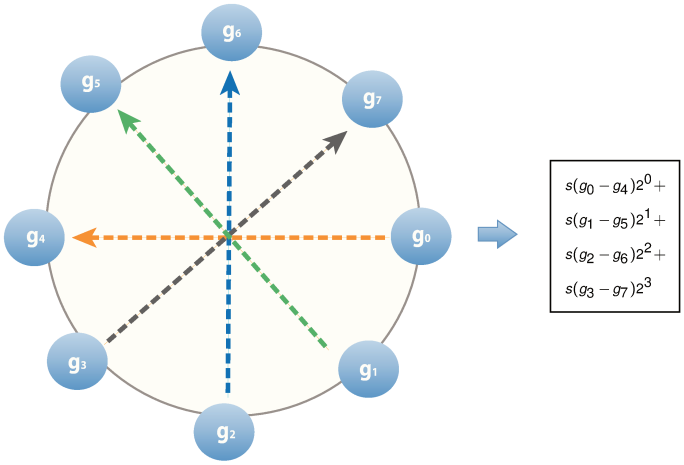


Figure 3.3: The CS-LBP descriptor.

$$XCS-LBP_{P,R}(c) = \sum_{i=0}^{(P/2)-1} s\left(g_1(i,c)+g_2(i,c)\right)2^i \tag{3.5}$$

where the threshold function s, which is used to determine the types of local pattern transition, is defined as a characteristic function:

$$s(x_1+x_2) = \begin{cases} 1 & \text{if } (x_1+x_2) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

and where $g_1(i,c)$ and $g_2(i,c)$ are defined by:

$$\begin{cases} g_1(i,c) = (g_i - g_{i+(P/2)}) + g_c \\ g_2(i,c) = (g_i - g_c)\,(g_{i+(P/2)} - g_c) \end{cases} \tag{3.7}$$

with the same notation conventions than in equations (3.1) and (3.3). It is worth noting that the threshold function does not need a user-defined threshold value, contrary to CS-LBP.

The computation of the ordinary LBP for a neighborhood of size $P = 8$ is illustrated in Figure 3.2 and the computation of the proposed XCS-LBP is shown in Figure 3.4 in order to make the comparison more understandable for the reader. Note the respective code lengths of 8 and 4 that lead to respective image compressions.

The proposed XCS-LBP produces a shorter histogram than LBP, as short as CS-LBP, but it extracts more image details than CS-LBP because (i) it takes into account the gray value of the central pixel, and (ii) it relies on a new strategy for neighboring pixels comparison. Since it is also more robust to noisy images than both LBP and CS-LBP, the proposed descriptor

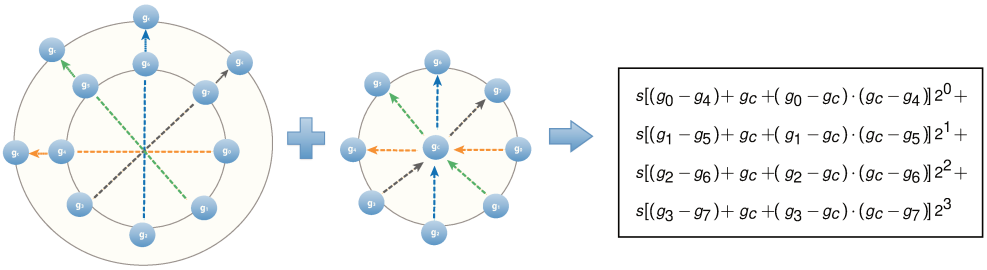


Figure 3.4: The XCS-LBP descriptor.

appears to more efficient for background modeling and subtraction. The main characteristics of different LBP variants, including those we will compare our new descriptor to, are summarized in Table 3.1.

3.3 Experimental results and discussions

Several experiments were conducted to illustrate both the qualitative and quantitative performances of the proposed descriptor XCS-LBP. We use datasets from the BMC (Background Models Challenge) which comprises synthetic and real videos of outdoor situations (urban scenes) acquired with a static camera, under different weather variations such as: wind, sun or rain [201].

3.3.1 Comparing direct competitor descriptors

We compare XCS-LBP with three other texture descriptors among the reviewed ones, namely: ordinary LBP, CS-LBP, and CS-LDP.

- A description of the **ordinary LBP** as well as **CS-LBP** [86] are presented in the Section 3.2.
- **CS-LDP [225]**: The Center-Symmetric Local Derivative Pattern descriptor (CS-LDP) proposed by Xue et al. [225] is an effective variant to CS-LBP. Like our XCS-LBP, it extracts more detailed local information while preserving the same feature lengths than the CS-LBP. This descriptor is given by:

$$CS-LDP_{P,R}(x_c,y_c) = \sum_{p=0}^{(P/2)-1} s\left[\left[(g_p - g_c)\left(g_c - g_{c+(P/2)}\right)\right]2^p \right. \tag{3.8}$$

$$s(x_1,x_2) = \begin{cases} 1 & \text{if } x\cdot y \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

The CS-LDP descriptor is illustrated in Figure 3.5.

Table 3.1: Comparison of LBP and variants.

Descriptor	Robust to noise	Robust to illumination changes	Uses color information	Uses temporal information	Histogram size with 8 neighbors
Ordinary LBP [146]		●			256
Modified LBP [84]	●	●			256
CS-LBP [86]		●			16
STLBP [175]		●		●	256
εLBP [206]		●			256
Adaptive εLBP [207]		●			256
SCS-LBP [226]	●			●	16
SILTP [120]	●				256
CS-LDP [225]	●				16
SCBP [225]			●		64
OCLBP [116]			●		1536
Uniform LBP [231]	●				59
SALBP [144]	●				128
SLBP-AM [229]	●			●	256
LBSP [22]	●	●			256
CS-SILTP [218]	●			●	16
XCS-LBP [176] (in this thesis)	●	●			16

We choose these the CS-LBP and CS-LDP descriptors for fair comparison purpose. Indeed, among those who rely on the same construction principle, *i.e. Center Symmetric* (CS), they are the only ones that use neither color nor temporal information, see Table 3.1. For all descriptors, the neighborhood size is empirically selected so that $P = 8$ and $R = 1$.

3.3.2 The BS methods used in this work

We evaluate the performance with two popular background subtraction methods: Adaptive Background Learning (ABL) and Gaussian Mixture Models (GMM). A summary of these approaches are presented below:

- **Adaptive Background Learning (ABL)**: This method consists to compute the absolute difference between the current frame and the static representation of the background model. Initially, the background is modeled using an average, a median or an histogram analysis over time then it is updated via running average. Once the model is computed, pixels of the current image are classified as foreground by thresholding the difference between the background image and the current frame [24].

- **Gaussian Mixture Models (GMM)**: In this algorithm, each pixel is represented by a sum of weighted Gaussian distributions defined for a given color space. These distributions are generally updated using an online expectation-minimization algorithm.

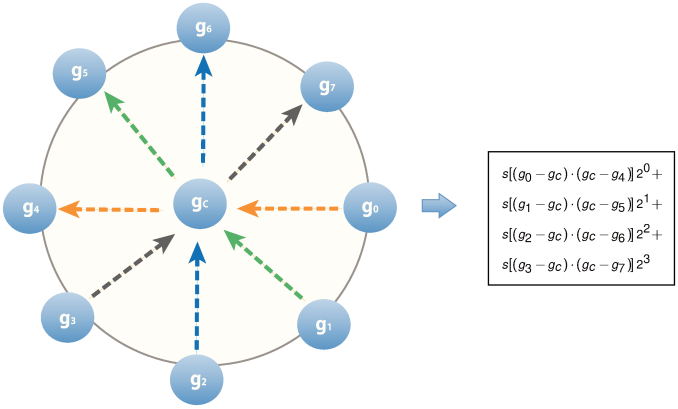


Figure 3.5: The CS-LBP descriptor.

Rotary (frame #1140) – scenes 122, 222, 322, 422 and 522

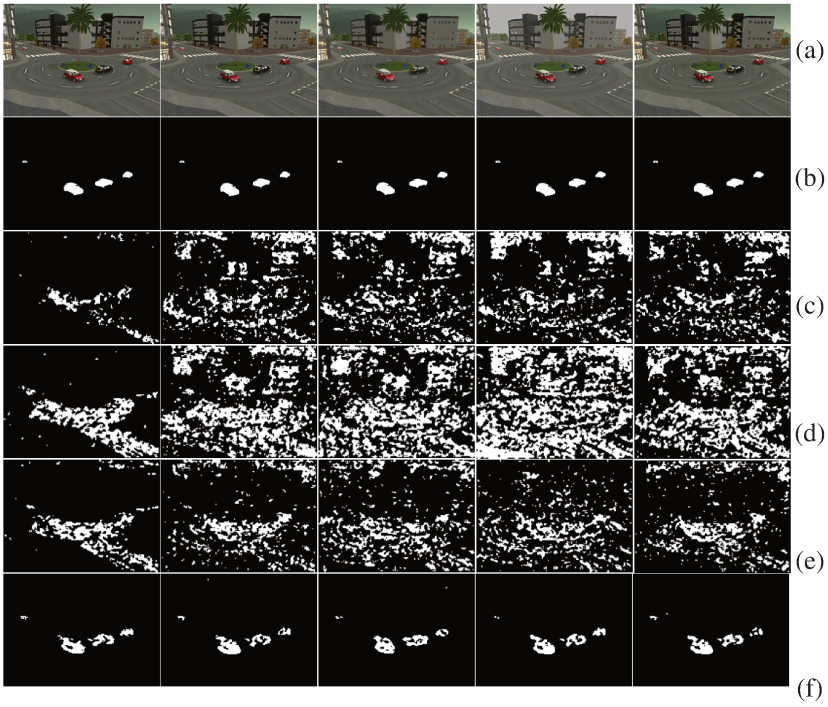


Figure 3.6: Background subtraction results using the ABL method on synthetic scenes – (a) original frame, (b) ground truth, (c) LBP, (d) CS-LBP, (e) CS-LDP and (f) proposed XCS-LBP.

More precisely, as a new image is processed, the GMM parameters for each pixel are updated to explain the colors variations over time [24].

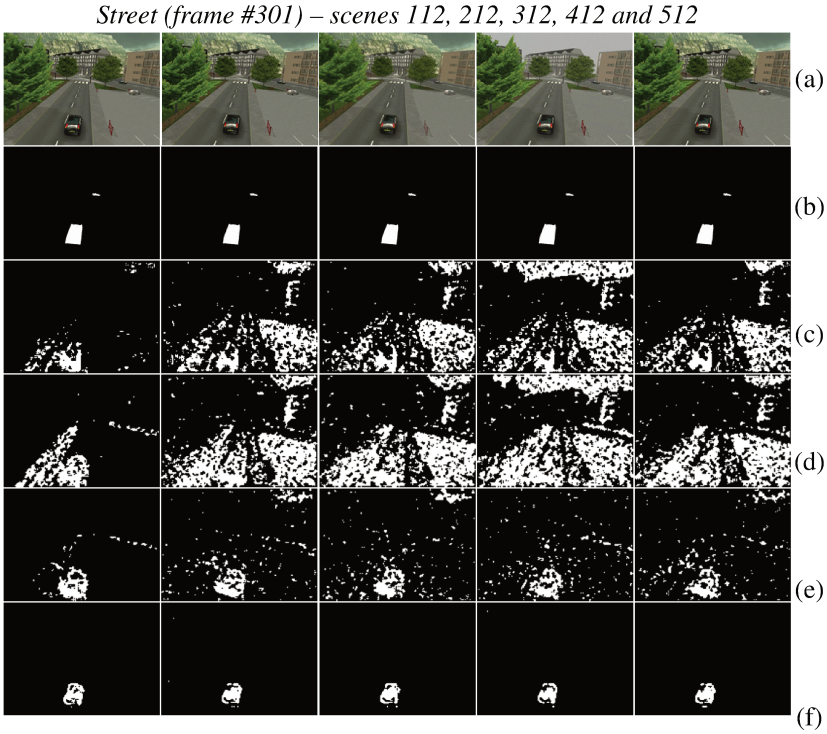


Figure 3.7: Background subtraction results using the ABL method on synthetic scenes – (a) original frame, (b) ground truth, (c) LBP, (d) CS-LBP, (e) CS-LDP and (f) proposed XCS-LBP.

First, we present results of background subtraction on individual frames of five different scenes from two video sequences: *Rotary* (frame #1140) and *Street* (frame #301). Figures 3.6, 3.7, 3.8 and 3.9 show the foreground detection results using the ABL and the GMM methods, respectively. Our descriptor clearly appears to be less sensitive to the background subtraction method, whereas the three others are very useless in detecting the moving objects when using the ABL method, unless a strong post-processing procedure.

Next, we give quantitative results on the same data. We use three classical measures based on the numbers of true positive TP pixels (correctly detected foreground pixels), false positive FP pixels (background pixels detected as foreground ones), false negative pixels FN (foreground pixels detected as background ones), and true negative pixels (correctly detected background pixels):

- $Recall = \dfrac{TP}{TP+FN}$,
- $Precision = \dfrac{TP}{TP+FP}$, and
- $F-score = 2 \times \dfrac{Recall \times Precision}{Recall + Precision}$.

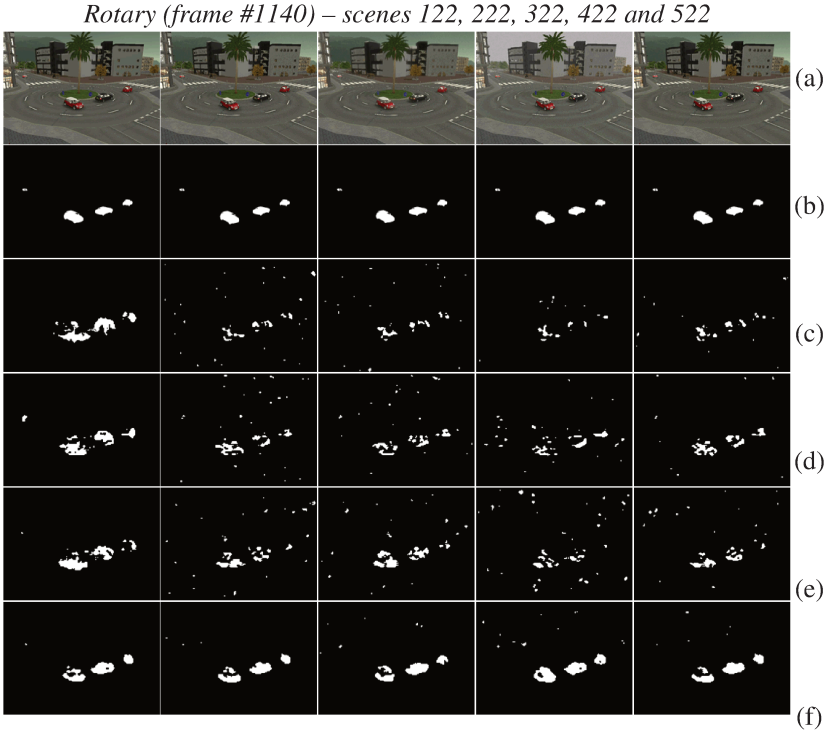


Figure 3.8: Background subtraction results using the GMM method on synthetic scenes – (a) original frame, (b) ground truth, (c) LBP, (d) CS-LBP, (e) CS-LDP and (f) proposed XCS-LBP.

Tables 3.3 and 3.4 shows the scores of the different descriptors obtained on the *Rotary* and *Street* entire scenes when using the ABL and the GMM method, respectively. Best scores are in bold. The proposed XCS-LBP gives the highest value for each score on almost all scenes, except for scene *Street*-[112, 312,412], for which CS-LBP and CS-LDP has achieved the best Recall using ABL, and scene *Street*-112 for which LBP gives the best Recall using GMM.

Note that both CS-LBP and CS-LDP gives lower scores (Precision and F-score) than LBP for some scenes, while our XCS-LBP descriptor takes always the advantage on the others, as shown by the average scores reported at the bottom of each Table.

Finally, we evaluate the proposed descriptor on nine long duration (about one hour) real outdoor video scenes from BMC. Each video sequence shows different challenging situations of real world: moving trees, casted shadows, the presence of a continuous car flow near to the surveillance zone, general climatic conditions (sunny, rainy and snowy conditions), fast light changes and the presence of big objects. The scores obtained using the ABL and the GMM methods are given in Table 3.5 and 3.6, respectively. Once again, our descriptor achieved the best scores on almost all the scenes, even when using the simple ABL method whereas it dramatically affect the other descriptors. The average scores reported at the bottom of each Table show that our XCS-LBP outperforms the ordinary LBP and both the similar

Table 3.2: Elapsed CPU times (averaged on the nine real-world videos of the BMC) over LBP times

Descriptor	CS-LBP	CS-LDP	XCS-LBP
ABL	1.10	1.12	1.09
GMM	1.06	1.07	1.05

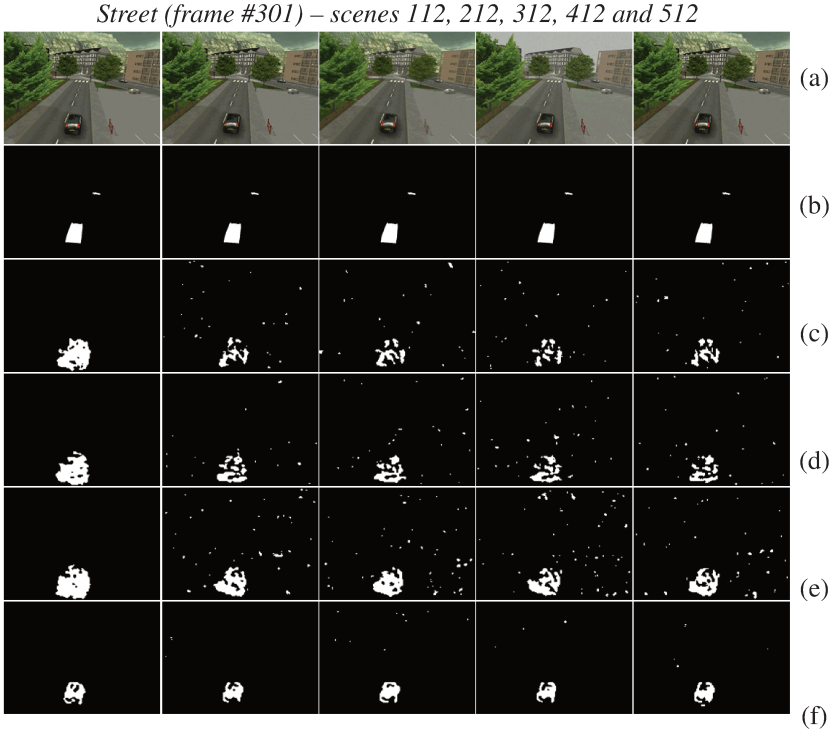


Figure 3.9: Background subtraction results using the GMM method on synthetic scenes – (a) original frame, (b) ground truth, (c) LBP, (d) CS-LBP, (e) CS-LDP and (f) proposed XCS-LBP.

construction-based CS-LBP and CS-LDP descriptors, the latter one being less performant than the LBP using GMM method. We use Matlab R2013a on a MacBook Pro (OS X 10.9.4) equipped with 2.2 GHz Intel Core i7 and 8 GB - 1333 MHz DDR3.

We collected the elapsed CPU times needed to segment the foregrounds using the ABL and the GMM methods, averaged over the nine real videos of BMC. Since the reference is the (fastest) LBP descriptor, the times are divided by LBP ones. Table 3.2 reports the resulting ratios for the compared CS descriptors. Our XCS-LBP shows slightly better time performance than both CS-LBP and CS-LDP.

Table 3.3: Performance of the different descriptors on synthetic videos of the BMC using the ABL method.

Scenes	Descriptor	Recall	Precision	F-score
Rotary 122	LBP	0.682	0.564	0.618
	CS-LBP	0.832	0.520	0.640
	CS-LDP	0.809	0.523	0.635
	XCS-LBP	**0.850**	**0.784**	**0.816**
Rotary 222	LBP	0.611	0.505	0.553
	CS-LBP	0.673	0.504	0.577
	CS-LDP	0.753	0.510	0.608
	XCS-LBP	**0.852**	**0.782**	**0.815**
Rotary 322	LBP	0.603	0.505	0.550
	CS-LBP	0.647	0.504	0.566
	CS-LDP	0.733	0.507	0.600
	XCS-LBP	**0.829**	**0.793**	**0.810**
Rotary 422	LBP	0.573	0.502	0.535
	CS-LBP	0.609	0.503	0.550
	CS-LDP	0.733	0.508	0.600
	XCS-LBP	**0.751**	**0.780**	**0.765**
Rotary 522	LBP	0.610	0.505	0.553
	CS-LBP	0.663	0.504	0.573
	CS-LDP	0.745	0.509	0.605
	XCS-LBP	**0.852**	**0.732**	**0.787**
Street 112	LBP	0.702	0.530	0.604
	CS-LBP	**0.839**	0.512	0.636
	CS-LDP	0.826	0.525	0.642
	XCS-LBP	0.803	**0.793**	**0.798**
Street 212	LBP	0.636	0.504	0.562
	CS-LBP	0.716	0.503	0.591
	CS-LDP	0.798	0.513	0.624
	XCS-LBP	**0.808**	**0.790**	**0.799**
Street 312	LBP	0.627	0.504	0.558
	CS-LBP	0.699	0.503	0.585
	CS-LDP	**0.801**	0.511	0.624
	XCS-LBP	0.800	**0.796**	**0.798**
Street 412	LBP	0.580	0.501	0.558
	CS-LBP	0.599	0.501	0.546
	CS-LDP	**0.754**	0.507	0.607
	XCS-LBP	0.748	**0.781**	**0.764**
Street 512	LBP	0.628	0.503	0.559
	CS-LBP	0.677	0.503	0.577
	CS-LDP	0.771	0.508	0.612
	XCS-LBP	**0.800**	**0.575**	**0.669**
Average scores	LBP	0.625	0.512	0.565
	CS-LBP	0.695	0.506	0.584
	CS-LDP	0.772	0.512	0.616
	XCS-LBP	**0.809**	**0.761**	**0.782**

Table 3.4: Performance of the different descriptors on synthetic videos of the BMC using the GMM method.

Scenes	Descriptor	Recall	Precision	F-score
Rotary 122	LBP	0.817	0.701	0.755
	CS-LBP	0.830	0.705	0.763
	CS-LDP	0.819	0.677	0.741
	XCS-LBP	**0.831**	**0.800**	**0.815**
Rotary 222	LBP	0.636	0.653	0.644
	CS-LBP	0.741	0.687	0.713
	CS-LDP	0.651	0.616	0.633
	XCS-LBP	**0.825**	**0.794**	**0.809**
Rotary 322	LBP	0.661	0.646	0.653
	CS-LBP	0.741	0.656	0.696
	CS-LDP	0.674	0.613	0.642
	XCS-LBP	**0.821**	**0.767**	**0.793**
Rotary 422	LBP	0.611	0.585	0.598
	CS-LBP	0.673	0.575	0.620
	CS-LDP	0.611	0.548	0.578
	XCS-LBP	**0.748**	**0.702**	**0.724**
Rotary 522	LBP	0.636	0.627	0.631
	CS-LBP	0.743	0.672	0.706
	CS-LDP	0.605	0.650	0.627
	XCS-LBP	**0.825**	**0.760**	**0.791**
Street 112	LBP	**0.940**	0.674	0.785
	CS-LBP	0.924	0.675	0.780
	CS-LDP	0.938	0.656	0.772
	XCS-LBP	0.844	**0.755**	**0.808**
Street 212	LBP	0.676	0.642	0.659
	CS-LBP	0.752	0.658	0.702
	CS-LDP	0.694	0.577	0.630
	XCS-LBP	**0.833**	**0.760**	**0.795**
Street 312	LBP	0.684	0.633	0.657
	CS-LBP	0.742	0.627	0.680
	CS-LDP	0.729	0.581	0.647
	XCS-LBP	**0.821**	**0.713**	**0.763**
Street 412	LBP	0.619	0.566	0.591
	CS-LBP	0.705	0.567	0.628
	CS-LDP	0.659	0.539	0.593
	XCS-LBP	**0.751**	**0.619**	**0.679**
Street 512	LBP	0.662	0.566	0.610
	CS-LBP	0.727	0.568	0.638
	CS-LDP	0.689	0.551	0.612
	XCS-LBP	**0.828**	**0.629**	**0.715**
Average scores	LBP	0.694	0.629	0.658
	CS-LBP	0.758	0.639	0.693
	CS-LDP	0.707	0.601	0.648
	XCS-LBP	**0.813**	**0.730**	**0.769**

Table 3.5: Performance of the different descriptors on real-world videos of the BMC using the ABL method

Videos	Descriptor	Recall	Precision	F-score
Boring parking, active bkbg	LBP	0.555	0.512	0.533
	CS-LBP	0.663	0.539	0.595
	CS-LDP	**0.712**	0.556	0.624
	XCS-LBP	0.673	**0.628**	**0.650**
Big trucks	LBP	0.456	0.490	0.473
	CS-LBP	0.664	0.583	0.621
	CS-LDP	**0.675**	0.673	0.674
	XCS-LBP	0.623	**0.788**	**0.696**
Wandering students	LBP	0.500	0.500	0.500
	CS-LBP	0.632	0.525	0.573
	CS-LDP	0.691	0.566	0.622
	XCS-LBP	**0.854**	**0.714**	**0.778**
Rabbit in the night	LBP	0.562	0.515	0.537
	CS-LBP	0.657	0.515	0.577
	CS-LDP	0.742	0.561	0.639
	XCS-LBP	**0.818**	**0.706**	**0.758**
Snowy christmas	LBP	0.568	0.516	0.541
	CS-LBP	0.640	0.508	0.567
	CS-LDP	0.684	0.513	0.586
	XCS-LBP	**0.719**	**0.557**	**0.628**
Beware of the trains	LBP	0.542	0.511	0.526
	CS-LBP	0.608	0.556	0.581
	CS-LDP	0.711	0.618	0.662
	XCS-LBP	**0.780**	**0.674**	**0.723**
Train in the tunnel	LBP	0.524	0.505	0.514
	CS-LBP	0.636	0.640	0.638
	CS-LDP	**0.668**	0.659	0.663
	XCS-LBP	0.655	**0.688**	**0.672**
Traffic during windy day	LBP	0.491	0.497	0.494
	CS-LBP	0.597	0.528	**0.560**
	CS-LDP	**0.589**	0.515	0.550
	XCS-LBP	0.572	**0.529**	0.550
One rainy hour	LBP	0.536	0.508	0.521
	CS-LBP	0.563	0.504	0.532
	CS-LDP	0.658	0.520	0.581
	XCS-LBP	**0.694**	**0.649**	**0.671**
Average scores	LBP	0.526	0.506	0.515
	CS-LBP	0.629	0.544	0.583
	CS-LDP	0.681	0.576	0.558
	XCS-LBP	**0.710**	**0.659**	**0.681**

Table 3.6: Performance of the different descriptors on real-world videos of the BMC using the GMM method

Videos	Descriptor	Recall	Precision	F-score
Boring parking, active bkbg	LBP	0.684	0.587	0.632
	CS-LBP	**0.716**	0.593	**0.649**
	CS-LDP	0.674	0.579	0/623
	XCS-LBP	0.680	**0.607**	0.641
Big trucks	LBP	0.695	0.778	**0.734**
	CS-LBP	**0.698**	0.773	0.733
	CS-LDP	0.649	0.758	0.699
	XCS-LBP	0.630	**0.792**	0.702
Wandering students	LBP	0.704	0.667	0.685
	CS-LBP	0.700	0.640	0.668
	CS-LDP	0.654	0.634	0.643
	XCS-LBP	**0.826**	**0.742**	**0.782**
Rabbit in the night	LBP	0.767	0.659	0.709
	CS-LBP	**0.826**	0.626	0.712
	CS-LDP	0.706	0.619	0.659
	XCS-LBP	0.805	**0.684**	**0.740**
Snowy christmas	LBP	**0.750**	0.519	0.614
	CS-LBP	0.734	0.516	0.606
	CS-LDP	0.625	0.510	0.562
	XCS-LBP	0.726	**0.538**	**0.618**
Beware of the trains	LBP	0.657	0.685	0.671
	CS-LBP	0.699	0.664	0.681
	CS-LDP	0.641	0.642	0.642
	XCS-LBP	**0.759**	**0.731**	**0.744**
Train in the tunnel	LBP	**0.724**	**0.711**	**0.717**
	CS-LBP	0.710	0.675	0.692
	CS-LDP	0.679	0.697	0.688
	XCS-LBP	0.695	0.680	0.687
Traffic during windy day	LBP	0.523	0.509	0.516
	CS-LBP	**0.553**	**0.520**	**0.536**
	CS-LDP	0.527	0.510	0.518
	XCS-LBP	0.532	0.518	0.525
One rainy hour	LBP	**0.867**	0.574	**0.691**
	CS-LBP	0.774	0.589	0.669
	CS-LDP	0.797	0.556	0.655
	XCS-LBP	0.761	**0.628**	0.688
Average scores	LBP	0.708	0.632	0.663
	CS-LBP	0.712	0.622	0.661
	CS-LDP	0.661	0.612	0.632
	XCS-LBP	**0.713**	**0.658**	**0.681**

3.4 Conclusion

In summary, a new texture descriptor for background modeling is proposed. It combines the strengths of the ordinary Local Binary Pattern (LBP) and the Center-Symmetric (CS) ones. Thus, the new variant XCS-LBP (eXtended CS-LBP) produces a shorter histogram than LBP, by its CS-construction. It is also tolerant to illumination changes as LBP and CS-LBP are whereas CS-LDP is not, and robust to noise as CS-LDP is whereas LBP and CS-LBP are not. We compared the XCS-LBP to the ordinary LBP and to its two direct competitors on both synthetic and real videos of the Background Modeling Challenge (BMC) using two popular background subtraction methods. The experimental results have shown that the proposed descriptor qualitatively and quantitatively outperforms the mentioned descriptors, making it a serious candidate for the background subtraction task in computer vision applications.

In the next chapter, we present an ensemble pixel-based for feature selection in BS to deal with the challenges enumerated in the Section 1.1. The proposed approach selects automatically the best features for different pixels of the image, and the more relevant ones are used for the foreground segmentation task. In this framework, the background model is modeled by different features including our XCS-LBP descriptor presented in this chapter.

Chapter 4

A pixel-based ensemble for feature selection in background subtraction

This chapter presents an Online Weighted Ensemble of One-Class SVMs (Support Vector Machines) able to select suitable features for each pixel to distinguish the foreground objects from the background. In addition, our proposal uses a mechanism to update the relative importance of each feature over time. Moreover, a heuristic approach is used to reduce the complexity of the background model maintenance while maintaining the robustness of the background model. Results on two datasets show the pertinence of the approach. This chapter is based on our recent publication presented at the International Conference on Pattern Recognition (ICPR), Cancun, Mexico (oral presentation) [177].

4.1 Motivation

A single-feature background subtraction algorithm may not be appropriate in a complex scene because the most discriminant features for each element are probably different. A complex scene comprising of several elements such as waving trees, sky, soil and cars is shown in Figure. 4.1. We have argued in the Chapter 2 that the ensemble feature selection technique as a great way to able select automatically the most relevant features in a scene. Relatively little approach based on *ensemble for feature selection* has been proposed for BS task. Most of these approaches use a multi-class boosting approach and its variants to select the best features (see Table 4.1). However, the BS can be considered an one-class classification (OCC) problem, therefore usually only exemplars of one-class elements are available (i.e. the background component is always present), whereas the other classes are unknown (i.e. foreground objects can appear/disappear several times in the scene). To overcome this problem, most of BS approaches have been used statistical distributions to generate the unrealistic foreground samples. In this chapter, we propose an online weighted ensemble of one-class SVMs (Support Vector Machines) for feature weighting and selection for foreground-background separation. The main BS works based on ensemble for feature selection as well as its principal

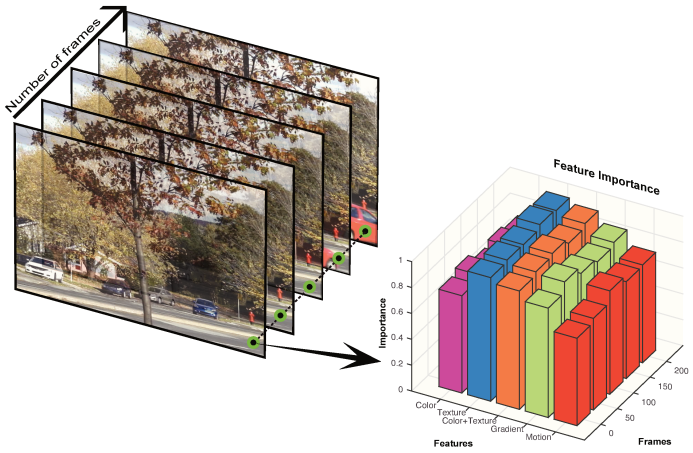


Figure 4.1: A conceptual illustration of a complex scene (left) and its features importance over time. The bar-graph (right) shows the feature importance variations for a certain region of the scene along time.

differences, including our proposal are shown in Table 4.1 and Table 4.2. A brief overview of the proposed framework given is illustrated in Figure 4.2. Firstly, a set of multispectral features jointly with well-known features (ie. color, texture, etc.) are extracted from the training image sequence. Next, a weighted version of random subspace creates a diversity of classifiers pool, each classifier represented by a incremental weighted version of one-class SVM. A heuristic approach called Small Votes Instance Selection (SVIS) is used in the IWOC-SVM model updating step. Only the best week classifiers are selected and combined to form a final classifier. Finally, we use a mechanism called Adaptive Importance (AI) computation to update the importance of the classifiers pool over time. The whole framework described here works as online manner. The main contributions of this work are:

1. An incremental version of the WOC-SVM algorithm, called Incremental Weighted One-Class Support Vector Machine (IWOC-SVM).

2. An online weighted version of random subspace (OW-RS) to increase the diversity of the classifiers pool.

3. A mechanism called Adaptive Importance Calculation (AIC) to suitably update the relative importance of each feature over time.

4. A heuristic approach for IWOC-SVM model updating to improve speed.

The rest of this chapter is as follows. In Section 4.2, we remind the offline WOC-SVM and show how we extend it for incremental learning. Then, we present an overview of the proposed method in Section 4.3. Experimental results are presented in Section 4.4, and concluding remarks are given in Section 4.5.

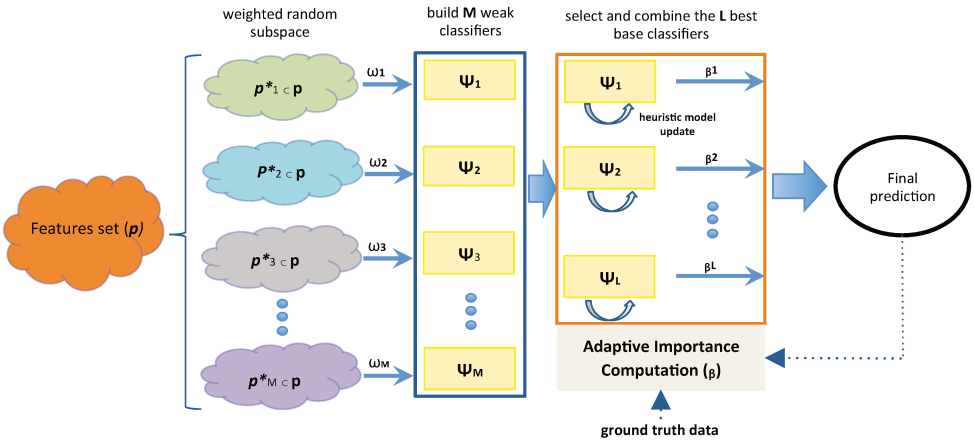


Figure 4.2: A brief overview of the proposed framework.

Authors/Date	*Strategy*	*Level*	*Type*
	Boosting-based		
Grabner and Bischof (2006) [70, 72]	AdaBoost	Region	multi-class
Parag et al. (2006) [149]	RealBoost	Pixel	multi-class
Grabner et al. (2008) [71]	AdaBoost	Region	multi-class
	Other approaches		
Klare and Sarkar (2009) [109]	Ensemble of Mixture of Gaussians	Pixel	one-class
OWOC-RS [in this chapter [177]	Weighted Random Subspace	Pixel	one-class

Table 4.1: The main BS works based on ensemble for features selection approaches.

Authors/Date	Intensity	Color	Edge	Texture	Depth	Motion	Multispectral
Grabner and Bischof (2006) [70, 72]		●	●				
Parag et al. (2006) [149]	●	●	●				
Grabner et al. (2008) [71]				●			
Klare and Sarkar (2009) [109]		●	●	●			
OWOC-RS [in this chapter] [177]	●	●	●	●		●	●

Table 4.2: Comparison of the main BS works based on ensemble for features selection approaches and its features.

4.2 Incremental weighted one-class SVM

The One-Class Support Vector Machine (OC-SVM) [192] is considered as one of the most efficient one-class based non linear classifier. Given a labeled training data set $X = \{x_1, ..., x_N\}$ in $\mathbb{R}^p$, it consists in learning for each target class ω_T the minimum volume contour that enclose all the data in X whose label is ω_T, using a *one class against all* scheme. It is well adapted to BS for which there is only one target class: the background pixels class ω_T. The

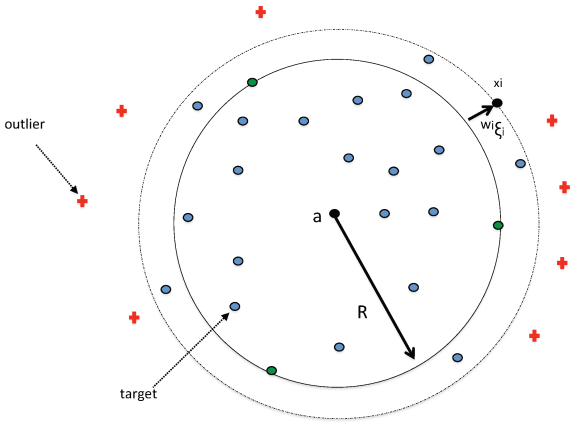


Figure 4.3: Data of a single class is covered by the hypersphere with center a and radius R.

volume can be defined as an hypersphere of radius R centered on a, both to be learned from X. Minimizing the hypersphere volume implies the minimization of R^2. To prevent the classifier from over-fitting with noisy data, slack variables ξ_i are introduced to allow some target points (respectively outliers) outside (respectively inside) the hypersphere. Therefore the problem is to minimize the objective function $\Theta(a,R)$ [192]:

$$\Theta(a,R) = R^2 + C\sum_{i=1}^{N}\xi_i, \tag{4.1}$$

where a and R are the center and the radius of the hypersphere, subject to: $\forall 1 \leq i \leq N$,

$$\xi_i \geq 0 \tag{4.2}$$

$$||x_i - a||^2 \leq R^2 + \xi_i \tag{4.3}$$

In Eq. (4.1), C is a user-defined parameter that controls the trade-off between the volume and the number of target points rejected. The larger C, the less outliers in the hypersphere. Bicego and Figueiredo [21] proposed a Weighted version (WOC-SVM) that allows to use weights $W = \{w_1, ..., w_N\}$ comprised in [0,1] for the data. The objective function $\Theta(a,R)$ becomes:

$$\Theta(a,R) = R^2 + C\sum_{i=1}^{N} w_i\xi_i, \tag{4.4}$$

subject to (4.2-4.3). The smaller w_i, the smaller penalty, the smaller the influence points far from center of the hypersphere on a and R. Figure 4.3 illustrates the data of a single class is covered by the hypersphere with center a and radius R. The hypersphere defines a boundary separating the target and outlier samples. Incorporating the constraints in (4.4) allows to construct the Lagrangian and the dual problem is to minimize:

$$L_\Theta(a,R) = \sum_{i=1}^{N}\alpha_i\langle x_i, x_i\rangle - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\langle x_i, x_j\rangle \tag{4.5}$$

where $\langle .,. \rangle$ stands for the inner product that can be replaced by any kernel function $K(.,.)$, and α_i are the associated Lagrangian multipliers subject to:

$$0 \leq \alpha_i \leq w_i C, \tag{4.6}$$

$$\sum_{i=1}^{N} \alpha_i = 1. \tag{4.7}$$

The solution of this quadratic programming problem is twofold. On one hand, the center a is a linear combination of the data points:

$$a = \sum_{i=1}^{N} \alpha_i x_i. \tag{4.8}$$

On the other hand, the radius R is subject to the following Karush-Kuhn-Tucker (KKT) conditions that correspond to inliers, the so-called support vector (SV) points and outliers, respectively:

$$inliers \quad : \quad \alpha_i = 0 \Rightarrow ||x_i - a||^2 < R^2 \tag{4.9}$$

$$SV \quad : \quad 0 < \alpha_i < C \Rightarrow ||x_i - a||^2 = R^2 \tag{4.10}$$

$$outliers \quad : \quad \alpha_i = C \Rightarrow ||x_i - a||^2 > R^2 \tag{4.11}$$

and can be computed from SV points given by (4.10). The classification of an incoming point x is straightforward: it is assigned to ω if it falls inside the class boundary (positive case), otherwise it is associated to an outlier class ω_o (negative case).

Traditional WOC-SVM is an offline or batch process, so that classification boundaries are not updated. This can limit its use for many machine learning applications. For the BS task, it is required to adjust the learned model to the scene variations over time. We propose an Incremental Weighted One-Class Support Vector Machine (IWOC-SVM) to handle this issue which is closely related to the procedure proposed by Tax and Laskov [194]. In the IWOC-SVM algorithm, SV set and non-SV set in previous training set Z_0 may be converted into SV. Samples which violate KKT conditions in new samples are chosen as training set and the other useless samples are eliminated in the training process. Given new samples $Z_1 = \{z_1, z_2, ..., z_s\}$ and its respective weights not learned by the IWOC-SVM, first we defined the corresponding $\alpha_i = 0$, and then we calculated the distance to center of the hypersphere. There are no new SVs in the new samples Z_1 when the distance is smaller than the radius. In addition, some non-SVs in the old samples may be transformed into SVs along with incremental learning of the new samples. Note that non-SVs can be transformed into new SVs if they always exist nearby the hypersphere. The mathematical model can be defined as:

$$R - \theta \leq ||x - a|| \leq R \tag{4.12}$$

where $\theta \in [0, R]$ is relative to the distribution of previous training set, and the loose distribution will make the value of θ be high. In addition, with the incremental learning, the value of θ will be low for more and more samples located near the previous SV set. The resulting IWOC-SVM is summarized in Algorithm 6.

Algorithm 6 Incremental Weighted One-Class SVM

1: **Require:** Previous training set Z_0, newly added training set Z_1 and its respective weights
2: Train IWOC-SVM classifier on Z_0, then split $Z_0 = SV_0 \cup NSV_0$
3: Input new samples Z_1. Put samples that violate KKT conditions in Z_1^V. If $Z_1^V = \emptyset$, then goto 2.
4: Put samples from NSV_0 that satisfy Eq. (4.12) into NSV_0^S.
5: Set $Z_0 = SV_0 \cup NSV_0^S \cup Z_1^V$ and train IWOC-SVM classifier on Z_0.
6: **Output:** IWOC-SVM classifier Ω and the new training set Z_0.

4.3 Online weighted one-class random subspace ensemble for feature selection (OWOC-RS)

For the background subtraction task, diversity models are initially learned for each pixel contained in the first N images, say training set $X = \{x_1, x_2, ..., x_N\}$ where each x_j $(j = 1, ..., N)$ $\in \mathbb{R}^p$ is a certain pixel over time N described by p original features.

4.3.1 Generating multiple base models

For each classifier, $p^* < p$ features are randomly selected so that x reduces to S_k $(k = 1, ..., M)$, where M is the user-defined number of base classifiers. Then, for each reduced object x_j^* $(j = 1, ..., N)$ of S_k, weights are assigned to the features in accordance to an exponential distribution. We opted for a Poisson distribution because it is usually employed in re-sampling ensemble methods such as bagging and wagging [147]. In this work, we used the version of the Poisson distribution that describes the process in which events occur continuously and independently at a constant average rate. The weights drawn from the Poisson distribution are used to generate the IWOC-SVM base classifier [111]. Thus, a hybridization between random subspace and incremental one-class learning is done. The above approach increases the diversity of base classifiers since different weights of each random subspace are taken to distinguish the decision boundaries computed by the classifiers. Indeed, these base classifiers represent a set of diverse base background models $\Psi = \{\Psi_1, \Psi_2, ..., \Psi_M\}$. The pseudo-code of the proposed approach for multiple base background models generation is given in Algorithm 7.

Algorithm 7 Generate multiple base background models

1: **Require:** IWOC-SVM training procedure, training set X, subspace dimension p^*, number of base classifiers M, weight distribution $\delta(x)$
2: $k \leftarrow 1$
3: **repeat**
4: $S_k \leftarrow$ SelectRandomSubspace(X,p^*)
5: Train k-th IWOC-SVM on S_k with respect to weights $w \sim \delta(x)$
6: $k \leftarrow k+1$
7: **until** $k > M$
8: **Output:** Trained IWOC-SVM base classifiers $\Psi = \{\Psi_1, \Psi_2, ..., \Psi_M\}$

4.3.2 Adaptive Importance (AI)

Along time, the selected feature set may become inadequate if any major change in the scene occurs. Since the objective is to use the more useful models, namely the best features from the pool of p features, an adaptive importance taking values in [0,1] can be introduced as proposed in [215] for each base model to weight the class labeling (see Eq. 4.15) of the incoming pixels. The higher the importance which lies in [0,1], the more the classifier influences the decision. Let $\lambda_k^{correct}$ (respectively λ_k^{wrong}) be the number of times a pixel was correctly (respectively incorrectly) classified by the k-th ($k = 1, ..., M$) base classifier from given ground truth data. Then, the corresponding error is given by:

$$error_k = \frac{\lambda_k^{wrong}}{\lambda_k^{correct} + \lambda_k^{wrong}} \tag{4.13}$$

Note that only the base classifiers that have the smallest errors are combined and used to differentiate the moving objects from the background model in the scene. The computation of the adaptive importance of each best base classifier is given in Algorithm 8.

Algorithm 8 Adaptive Importance (AI) computation

1: **Require:** Final classifier H, validation set $(t_1, y_1), ..., (t_N, y_N)$ where $t_i \in T$, $y_i \in Y = 0, 1$ for background and foreground examples respectively, set of L best base classifiers $\Psi = \{\Psi_1, \Psi_2, ..., \Psi_L\}$, learning rate parameter γ
2: Initialize all L best classifiers with importance: $\beta_l = 1$
3: **repeat**
4: Classify t_i using the final classifier H according to Eq. (4.16)
5: **for** $l = 1 : L$ **do**
6: Checks response of Ψ_l and calculates their $error_l$ according to Eq. (4.13)
7: For each best classifier, Ψ_l, update the importance $\beta_l = \beta_l(i-1) + \frac{P_a(\Psi_l) - P_a(H(i-1))}{(N+\gamma)}$
8: where $P_a(\Psi_l) = 1 - error_l$ according to Eq.(4.13).
9: **end for**
10: **until** $i < N$
11: Normalize the importance β of each L best classifier.
12: **Output:** New importance assigned to the best classifiers $\beta = \{\beta_1, \beta_2, ..., \beta_L\}$

4.3.3 Background detection

Given an incoming pixel x to be classified, one can define a support function associated to the class ω for each of the L best base classifiers: $\forall l = 1, ..., L$

$$F_l(x, \omega) = \frac{1}{s_1} exp(-d(x, a)/s_2) \tag{4.14}$$

where $d(x, a)$ is a distance metric from x to the center a of the target class ω, s_1 is a normalization factor and s_2 is a scale parameter. Each $F_l(x, \omega)$ is then compared to a threshold t_1 to obtain the positive or negative class labels: $\forall l = 1, ..., L$

$$c_l(x,\omega) = \begin{cases} 1 & \text{if } F_l(x,\omega) \geqslant t_1 \\ -1 & otherwise \end{cases} \tag{4.15}$$

Comparing the weighted sum of theses L class labels as in [193] to another threshold t_2 allows to define the final classifier for x as follows:

$$H(x) = \begin{cases} 1 & \text{if } \frac{1}{L}\sum_{l=1}^{L} \beta_l \, c_l(x,\omega) \geqslant t_2 \\ 0 & otherwise \end{cases} \tag{4.16}$$

A pixel x is classified as a background pixel if $H(x) = 0$.

4.3.4 Heuristic approach for background model maintenance

The background maintenance relies on the mechanism used for adapting the learned model to the scene over time. For this step, we propose to suitably update the learned model by our IWOC-SVM using a new ensemble margin-based data selection approach called Small Votes Instance Selection (SVIS) introduced by Guo and Boukir [75]. The SVIS relies on a simple and efficient heuristic approach to provide SV candidates: selecting lowest margin samples. This heuristic significantly reduces the IWOC-SVM training task complexity while maintaining the accuracy of the IWOC-SVM classification. Once only support vector candidate samples are used to update the IWOC-SVM's models. The SVIS consists of an unsupervised ensemble margin that combines the first $c_{(1)}$ and second most voted class $c_{(2)}$ labels under the learned model. Let $v_{c_{(1)}}$ and $v_{c_{(2)}}$ denote the relative number of votes. Then the margin, taking value in [0,1] is:

$$m(x) = \frac{v_{c_{(1)}} - v_{c_{(2)}}}{L} \tag{4.17}$$

where L represents the number of best base classifiers in the ensemble. The first smallest margin samples are selected as support vector candidates. The final model is updated by the first smallest margin samples. This procedure is presented in the Algorithm 9.

4.4 Experimental results

The experiments were conducted to show both the qualitative and quantitative performances of the proposed method. We used the MSVS dataset [1] [16] which consists of a set of 5 video sequences containing 7 multispectral bands and color video sequence (RGB). We also present the results on the ChangeDetection (CDnet 2014) dataset [2] [212]. Three video sequences categorized into baseline scenes, intermittent object motion and dynamic scenes are used.

[1] http://www.fluxdata.com/articles/universit%C3%A9-de-bourgogne-uses-fluxdata-fd-1665-create-dataset-background-subtraction

[2] http://changedetection.net/

Algorithm 9 Heuristic approach for model maintenance

Require: Final classifier H, test set $Z = \{z_1, z_2, ..., z_t\}$, weight distribution $\delta(z)$, user defined parameter *time*, user defined parameter η.
$i \leftarrow 1$
repeat
  if $H(z_i) = 1$ (background) **then**
    Compute the margin $m(z_i)$ by Eq. (4.17).
  end if
  if *time* is reached **then**
    Order all the test samples according to their margin values, in ascending order.
    The η smallest margin samples are selected as support vectors.
    $H(x)$ is updated using Z_1 and its weight $w \sim \delta(x)$.
  end if
  $i \leftarrow i + 1$
until $i > t$

Figure 4.4: Results using the MSVS dataset [16] – (a) original frame, (b) ground truth and (c) proposed method.

The baseline scenes include pets2006 while dynamic scenes include canoe and intermittent object motion scenes include sofa.

In the training step, we used kernalized IWOC-SVM as a base classifier with C = 1, with the same RBF (Radial Basis Function) kernel $K(.,.)$ [192]. The main advantage of RBF kernel is its good performance on non-linearly separable data. The pool of classifiers was homogeneous and consisted of 10 base classifiers of the same type. The classification threshold t_1 was set to 0.9 and t_2 to 0.5 for combining the best one-class classifiers. The video sequences was resized to 160×120 pixels in our experiments due computational cost. We set p* = 5 for the random subspace dimension from the original p = 26-dimensional features space on the MSVS dataset while p = 19-dimensional features space on the CDnet 2014 dataset. These features were chosen to have at least one feature in the five type of features commonly used in BS: color feature (R,G,B, H,S,V and gray-scale), texture feature (XCS-LBP [176]),

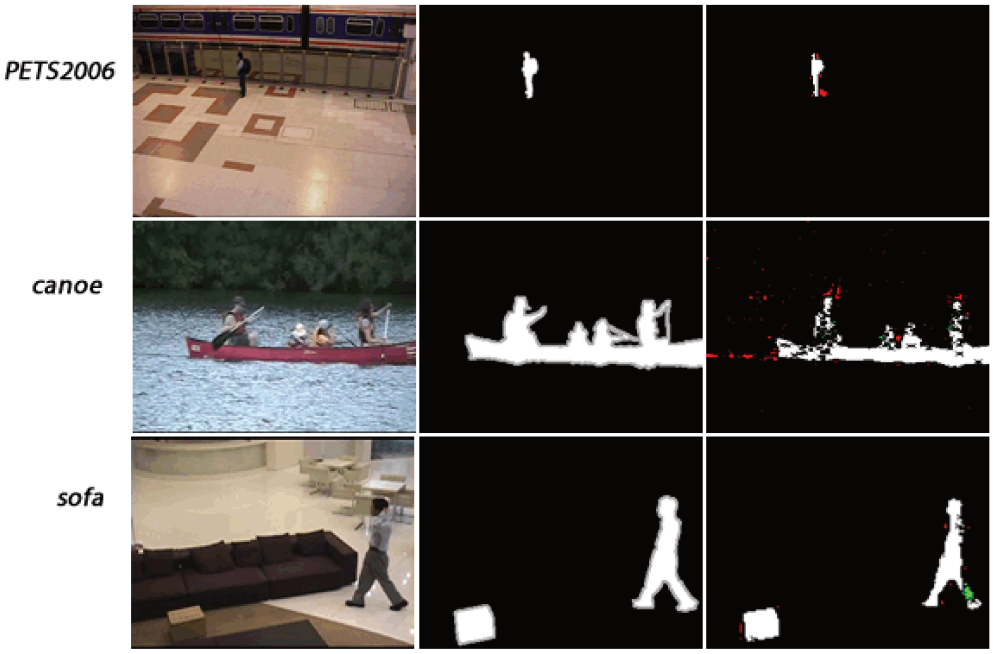


Figure 4.5: Results using the CDnet 2014 dataset [16]– (a) original frame, (b) ground truth and (c) proposed method.

color-texture (OC-LBP [133]), edge feature (gradient orientation and magnitude), and motion feature (optical flow). In addition, we used also the multispectral bands included in the MSVS dataset (a total of 7 spectral narrow bands).

We present the visual results on individual frames of two different scenes: *Scene* 02 (frame #51) and *Scene* 03 (frame #75) from MSVS dataset and three scenes: *PETS2006* (frame #345), *canoe* (frame #980) and *sofa* (frame #630) from CDnet 2014 dataset. Figure 4.4 and 4.5 show the foreground detection results using our approach were displayed without any post-processing technique. The true positives (TP) pixels are in white, true negatives (TN) pixels in black, false positives (FP) pixels in red and false negatives (FN) pixels in green. Our method is able to detect the moving objects with fewer number of false detection. Next, the performance of the BS is evaluated at pixel-level. Given the ground truth data, the correctness of foreground segmentation is measured using three classical measures: recall, precision and F-score. We divided the training set into three parts - first, we generate the base BS, next we select the best base BS models and finally, the adaptive importance are calculated for each best BS model. In addition, we used a set of images to test our framework (detection step without ground truth for testing). All tests were done by a 10-fold cross validation.

We compare the results obtained by our method from MSVS dataset with two other approaches proposed by Benezeth et. al [16]: 1) BS with the Mahalanobis distance using color video sequence (RGB) and multispectral video sequence (7B), and 2) Pooling using multispectral video sequence (7B). We use these two approaches because they use multispectral

features to perform BS. Table 4.5 shows the score of the Mahalanobis distance and Pooling methods evaluated on five scenes. The best scores are in bold. The proposed approach presented the best scores for *Scene 01*, *Scene 03*, *Scene 04* and *Scene 05*. In these scenes, the most frequent challenges for BS are color saturation, dynamic background, illumination changes, camouflage effects and intermittent object motion. In the *Scene 02*, the framework's performance was impacted due to gradual illumination changes. The best score for *Scene 02* was presented for Mahalanobis distance. Table 4.6 also shows the score of our method from CDnet 2014 dataset. Note that for this dataset the best score was presented for the *sofa* scene. It contains abandoned objects and objects stopping for a short while and then moving away. A suggestion to further improve our method score for both datasets is adding new feature descriptors and/or its variants can be added to deal with specific background subtraction challenges. In general, we can see that the ensemble feature selection is a suitable and efficient approach for BS.

Figure 4.6 and 4.7 illustrate the importance of each feature through video scenes from MSVS and CDnet 2014 datasets. For each pixel, certain features are ignored or receive relatively low importance in favor of other more informative features. Then, a global histogram was then normalized to obtain scores from 0-1, where higher scores meant highly informative features. Unlike traditional methods that the same feature (or set of features) is used globally for the whole video scene (and usually with the same level of importance), we present the potential of the proposed approach and its effectiveness to select the best features for background subtraction task. As can be seen on the MSVS dataset, the most important features for overall scenes were OCLBP and gradients with high or medium contribution of some features such as multispectral. It is important to note several BS algorithms uses color as main feature, whereas in our experiments the color feature is the one with lowest importance except for Scene 02. Notice that on CDnet 2014 dataset, all features are important for *PETS2006* and *canoe* scenes while in the *sofa* scene only OCLBP-GG is less important. Table 4.3 and 4.4 show the most and less significant features for both datasets used in this work. The experimental were made in Matlab R2013a a MacBook Pro with 2.2 GHz Intel Core i7. We collected the elapsed CPU time for training/validation and foreground detection. For training/validation the elapsed time is 5.44 sec/frame, while in foreground detection the elapsed time is 1.05 sec/frame.

Table 4.3: The most (+) and less (-) significant features from MSVS scenes [16].

Videos	Importance	
	most (+)	**less (-)**
Scene 01	Gradient Direction with medium contrib. multispectral features	OCLBP-GB
Scene 02	MS1,MS2 and MS6 with Color, Gradient X features	XCS-LBP and MS4
Scene 03	OCLBP-GG,RR with medium contrib. of other OCLBP channels and gradient features	Hue, Optical flow and multispectral features
Scene 04	OCLBP-BB,RR,RG and GG with medium contrib. of gradient features	Multispectral and color features
Scene 05	OCLBP-RR with high contrib. of other OCLBP channels and multispectral features	Gradient Magnitude

Table 4.4: The most (+) and less (-) significant features from CDnet 2014 dataset [212].

Videos	Importance	
	most (+)	less (-)
PETS2006	Relatively a high contribution of most of the features except for a high contribution of the saturation, OCLBP-RB and Y gradient features, respectively	none
canoe	High contribution of all features	none
sofa	High contribution of most of the features except for a medium contribution of the saturation, OCLBP-RG, and Y gradient features, respectively	OCLBP-GG

Table 4.5: Performance of the different methods using the MSVS dataset [16].

Videos	Method	Precision	Recall	F-score
Scene 01	MD (RGB) [16]	0.6536	0.6376	0.6536
	MD (MSB) [16]	0.7850	0.8377	0.8105
	Pooling (MSB) [16]	0.7475	0.8568	0.7984
	OWOC-RS [in this chapter]	**0.8500**	**0.9580**	**0.9008**
Scene 02	MD (RGB) [16]	0.8346	0.9100	0.8707
	MD (MSB) [16]	0.8549	0.9281	**0.8900**
	Pooling (MSB) [16]	**0.8639**	**0.8997**	0.8815
	OWOC-RS [in this chapter]	0.8277	0.8245	0.8727
Scene 03	MD (RGB) [16]	0.7494	0.5967	0.6644
	MD (MSB) [16]	0.7533	0.6332	0.6889
	Pooling (MSB) [16]	0.8809	0.5134	0.6487
	OWOC-RS [in this chapter]	**0.9326**	**0.9965**	**0.9635**
Scene 04	MD(RGB) [16]	0.8402	0.7929	0.8158
	MD (MSB) [16]	0.8430	0.8226	0.8327
	Pooling (MSB) [16]	0.8146	**0.8654**	0.8392
	OWOC-RS [in this chapter]	**0.9534**	0.8374	**0.8997**
Scene 05	MD (RGB) [16]	0.7359	0.7626	0.7490
	MD (MSB) [16]	0.7341	0.8149	0.7724
	Pooling (MSB) [16]	**0.7373**	0.8066	0.8066
	OWOC-RS [in this chapter]	0.7316	**0.8392**	**0.8400**

*MD = *Mahalanobis distance*

Table 4.6: Performance of our method using the CDnet 2014 dataset [212].

Videos	Precision	Recall	F-score
PETS2006	0.8555	0.9395	0.8955
canoe	0.9034	0.9216	0.9124
sofa	0.9682	0.9160	0.9414

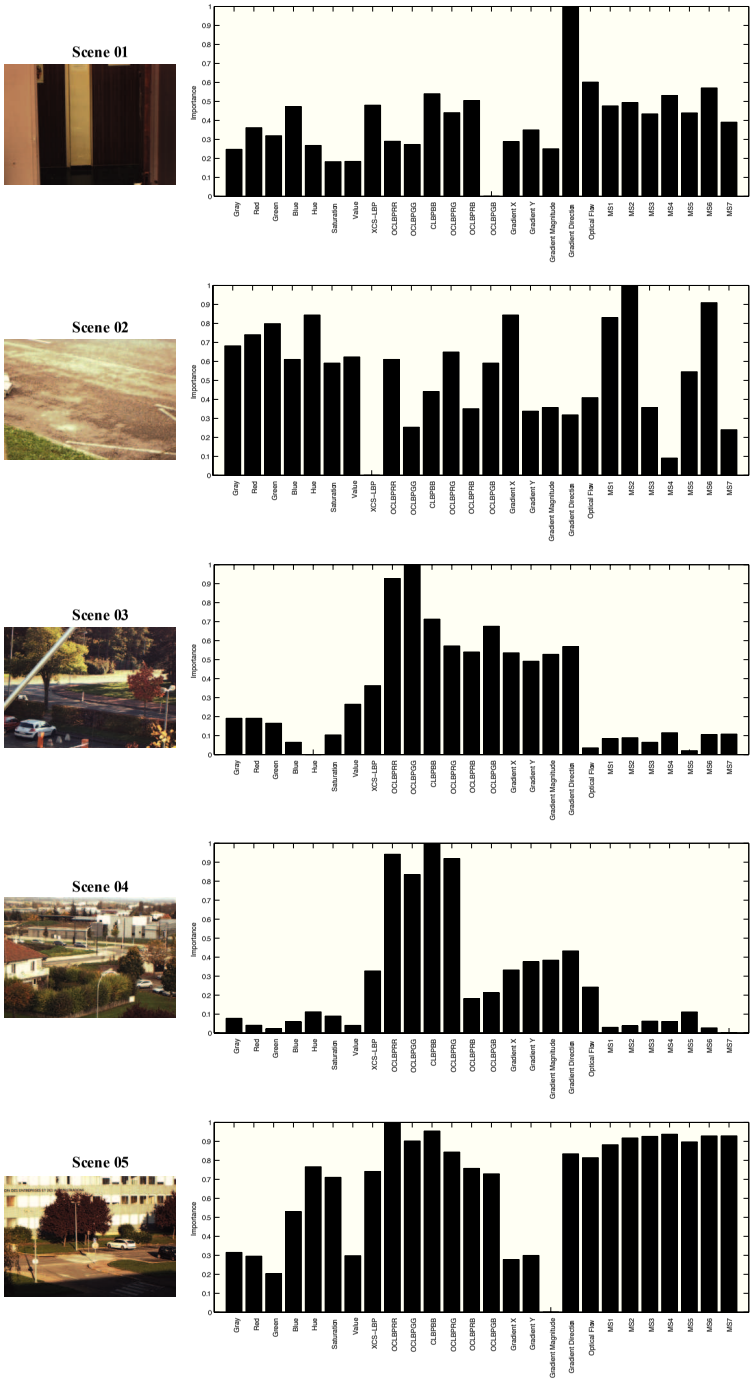


Figure 4.6: The visual features importance through video scenes from the MSVS dataset [16].

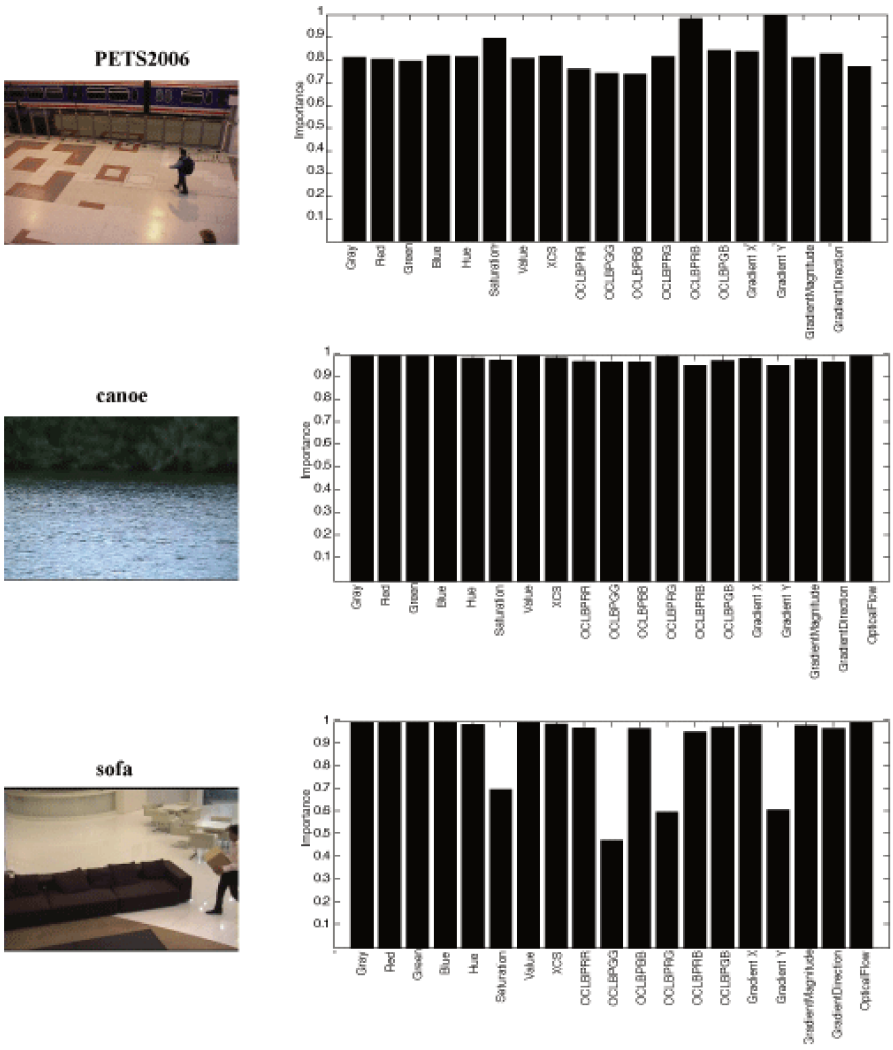


Figure 4.7: The visual features importance through video scenes from the CDnet 2014 dataset [212].

4.5 Conclusion

Online Weighted Ensemble of One-Class SVMs is able to select suitable features for each pixel to distinguish the foreground objects from the background. In addition, an Online and Weighted version of the Random Subspace (OW-RS) is used to assign a degree of importance to each feature set, and these weights are used directly in the training step of our IWOC-SVM. Moreover, a heuristic approach is used to reduce the complexity of the background model maintenance while maintaining the robustness of the background model. Experimental results on different video sequences show the potential of the proposed approach and its effectiveness to select the best features for distinct regions in a video sequence. However, the ensemble pixel-based for feature selection described in this chapter only reaches the highest accuracy when the number of features is huge. In summary, each base classified learns a feature set instead of individual features. To overcome these limitations, in the next chapter we extend the approach proposed here by developing a novel methodology for selecting features based on wagging.

Chapter 5

A superpixel-based ensemble for feature selection in background subtraction

In this chapter, we present a novel superpixel based one-class ensemble to select the best features based on wagging. Our proposal is able to select suitable features to each region of a certain scene to distinguish the foreground objects from the background. In addition, we propose a mechanism to update the importance of each feature discarding insignificant features over time. Results on two challenging datasets show the pertinence of the proposed approach. The work presented here was recently submitted to Pattern Recognition Letters Journal [178].

5.1 Motivation

In Chapter 4, we presented an online weighted one-class random subspace ensemble pixel-based able to select automatically the best features for different pixels of the image, and the most relevant features are used for foreground segmentation. The main drawback is that this method only reaches the highest accuracy when the number of features is huge. Furthermore, each base classifier learns a feature set instead of individual features. To overcome these limitations, in this chapter we extend our previous approach by proposing a novel methodology for selecting features based on wagging. It is important to note that the ensemble learning methods usually require high computation time and memory consumption. In order to circumvent this issue, an alternative way is to use efficient strategies that not further increase the computational cost of the ensemble. So, In this chapter, we adopted a superpixel-based approach instead of pixel-level approach used in our previous work (Chapter 4). This does not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance. We propose further a mechanism called Adaptive Importance Computation and Ensemble Pruning (AIC-EP). Chapter 4 also propose a mechanism to select the features over time, however, in this chapter we have added an ensemble pruning to eliminate the features that will not have impact on the ensemble's final decision.

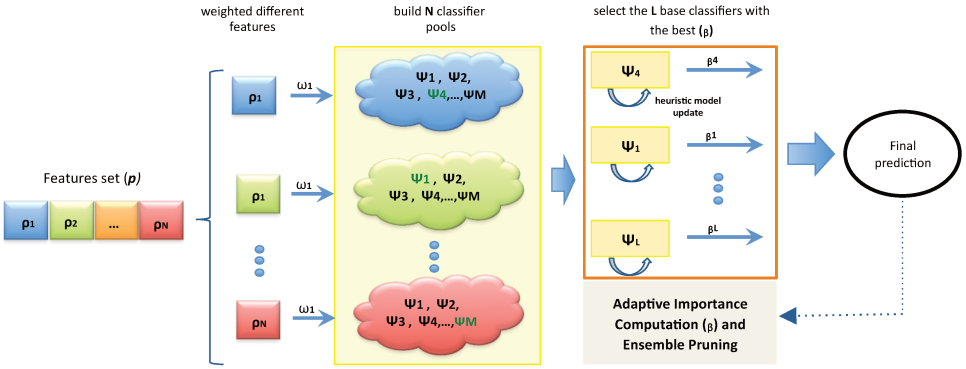


Figure 5.1: A brief overview of the proposed framework

This can improve the generalization performance of the ensemble, furthermore, it prevents the increase of the training cost, storage demands, and prediction time. Table 5.1 and Table 5.2 show the main differences of the present work compared to the work presented in the previous chapter including other state of the art works. The main contributions of this chapter can be summarized as follows:

1. A novel methodology to select the best features based on wagging.

2. A superpixel segmentation strategy to improve the segmentation performance, increasing the computational efficiency of our ensemble.

3. A mechanism called Adaptive Importance Computation and Ensemble Pruning (AIC-EP) to suitably update the importance of each feature discarding insignificant features over time.

A brief overview of the proposed framework in Figure 5.1. Firstly, a set of features are extracted from the training image sequence. Next, our wagging version creates different pools of IWOC-SVM classifiers from a certain feature. A heuristic approach called Small Votes Instance Selection (SVIS) is used in the IWOC-SVM model updating step. Finally, we use a mechanism called Adaptive Importance and Ensemble Pruning (AIC-EP) to update the importance of the classifiers discarding insignificant classifiers over time. Only the classifiers with high importance are selected and combined to form a strong classifier. The whole framework described here works as incremental manner.

The rest of this chapter is as follows. We present an overview of the proposed method in Section 5.2. Experimental results are presented in Section 5.3, and concluding remarks are given in Section 5.4.

Authors/Date	*Strategy*	*Level*	*Type*
Boosting-based			
Grabner and Bischof (2006) [70, 72]	AdaBoost	Region	multi-class
Parag et al. (2006) [149]	RealBoost	Pixel	multi-class
Grabner et al. (2008) [71]	AdaBoost	Region	multi-class
Other approaches			
Klare and Sarkar (2009) [109]	Ensemble of Mixture of Gaussians	Pixel	one-class
OWOC-RS [177]	Weighted Random Subspace	Pixel	one-class
Superpixel-OWAOC [in this chapter] [178]	Wagging for feature selection	Cluster	one-class

Table 5.1: The main BS works based on ensemble for features selection approaches.

Authors/Date	Intensity	Color	Edge	Texture	Depth	Motion	Multispectral
Grabner and Bischof (2006) [70, 72]		•	•				
Parag et al. (2006) [149]	•	•	•				
Grabner et al. (2008) [71]				•			
Klare and Sarkar (2009) [109]		•	•	•			
OWOC-RS [177]	•	•	•	•		•	•
Superpixel-OWAOC [in this chapter] [178]	•			•	•		•

Table 5.2: Comparison of the main BS works based on ensemble for features selection approaches and its features.

5.2 Superpixel-based Online WAgging One-Class Ensemble for Feature Selection (Superpixel-OWAOC)

Wagging is a variant of Bagging algorithm [15]. It trains each base classifier on the entire training, since for each sample is assigned a weight. Therefore, each sample has a level of influence on the classifier's training process. The standard wagging is a powerful strategy to generate a diverse set of base classifiers, but it is not designed for feature selection. We propose to extend the standard wagging for feature selection restricting the base learner so that each base classifier can focus only on a single feature. An overview of our wagging for feature selection is presented in Alg. 10.

For the background subtraction task, we initially computed the superpixel by SLIC (Simple Linear Iterative Clustering) [16], which is an adaptation of k-means in the *labxy* image space for robust superpixel creation. Next, diversity models are learned from a training set $X = \{x_1, x_2, ..., x_N\}$ where each x_j $(j = 1, ..., N) \in \mathbb{R}^p$ is a certain superpixel (maximum value) over time N described by p features.

5.2.1 Generate multiple base models

Our wagging for feature selection assign weights for each sample of a given features ρ according to an exponential distribution. We opted to use the version of the Poisson distribution

Algorithm 10 The wagging for feature selection

1: **Require:** IWOC-SVM training procedure, training set X, weight distribution $\delta(x)$, number of base classifier M, user defined parameter ε
2: $j \leftarrow 1$
3: **repeat**
4: **for** $k = 1 : M$ **do**
5: $\Psi_k \leftarrow$ train an IWOC-SVM classifier for each, ρ_j feature according to random weights drawn from $\delta(x)$.
6: Calculate the error of Ψ_k according to Eq. (5.1)
7: **if** $error_k \geq \varepsilon$ **then**
8: Choose the classifier Ψ_k
9: **break**
10: **else**
11: **continue**
12: **end if**
13: **end for**
14: $j \leftarrow j+1$
15: **until** $j > N$
16: // choose base classifiers with the best importances to according the Algorithm (11)
17: **Output:** Combine outputs the best base classifiers to according the Eq. (5.4).

that describes the process in which events occur continuously and independently at a constant average rate [111]. Therefore, these weights together with the samples are used as input to generate the Incremental Weighted One-Class Support Vector Machine (IWOC-SVM) base classifiers. The reader can find details of the IWOC-SVM in Chapter 4. The search iterates until an IWOC-SVM with the smallest error (defined by the user) is found or M rounds is reached. Let $\lambda_k^{correct}$ (respectively λ_k^{wrong}) be the number of times a region was correctly (respectively incorrectly) classified by the k-th $(k = 1, ..., M)$ base classifier from given ground truth data. Then, the corresponding error is given by:

$$error_k = \frac{\lambda_k^{wrong}}{\lambda_k^{correct} + \lambda_k^{wrong}} \tag{5.1}$$

The Algorithm 10 (lines 1-16) is responsible by created many base classifiers with small error representing a set of diverse base background models $\Psi = \{\Psi_1, \Psi_2, , ..\Psi_M\}$.

5.2.2 Adaptive Importance Computation and Ensemble Pruning (AIC-EP)

Along time, the selected feature set may become inadequate if any major change in the scene occurs. Since the objective is to use the more useful models, namely the best features from the p features set, an adaptive importance taking values in [0,1] can be introduced as proposed in [215] for each base model to weight the class labeling (see Eq. 5.4) of the incoming regions. The higher the importance which lies in [0,1], the more the classifier influences the decision. Note that the difference of Algorithm 11 for the Algorithm 8, proposed in

the Chapter 4, is just that we have added an ensemble pruning to eliminate the importances with very low values over time. This can can improve the generalization performance of the ensemble. Furthermore, it can prevent the increase of the training cost, storage demands, and prediction time since it allows to eliminate classifiers with very low importance that will not have impact on the ensemble's final decision. Note that only the base classifiers that have the highest importance are combined and used to differentiate the moving objects from the background model in the scene.

Algorithm 11 Adaptive Importance Computation and Ensemble Pruning (AIC-EP)

Require: Final classifier H, validation set $(t_1, y_1), ..., (t_N, y_N)$ where $t_i \in T$, $y_i \in Y = 0, 1$ for background and foreground examples respectively, set of L base classifiers $\Psi = \{\Psi_1, \Psi_2, ..., \Psi_L\}$, learning rate parameter γ, user defined parameter ς
Initialize all L classifiers with importance: $\beta_l = 1/L$ and estimate their $P_a(\Psi_l)$
where $P_a(\Psi_l) = 1 - error_l$ according to Eq.(5.1).
$i \leftarrow 1$
repeat
    Classify t_i using the final classifier H according to Eq. (5.4)
    for $l = 1 : L$ **do**
        Checks response of Ψ_l and calculates their $error_l$ according to Eq. (5.1)
        For each best classifier, Ψ_l, update the importance $\beta_l = \beta_l(i-1) + \frac{P_a(\Psi_l) - P_a(H(i-1))}{(N+\gamma)}$
    end for
    $i \leftarrow i+1$
until $i < N$
Normalize the importance β_l of each l classifier
for $l = 1 : L$ **do**
    if $B_l \leq \varsigma$ **then**
        discard the l-th classifier
    end if
end for
Output: The best classifier(s) and its/their β which could be used in Eq. (5.4)

5.2.3 Background detection

The procedure for background detection is the same as used in Chapter 4. However, we recover some of the principal definitions as follows. Given an incoming regions x to be classified, one can define a support function associated to the class ω for each of the L best base classifiers: $\forall l = 1, ..., L$

$$F_l(x, \omega) = \frac{1}{s_1} exp(-d(x, a)/s_2) \tag{5.2}$$

where $d(x, a)$ is a distance metric from x to the center a of the target class ω, s_1 is a normalization factor and s_2 is a scale parameter. Each $F_l(x, \omega)$ is then compared to a threshold t_1 to obtain the positive or negative class labels: $\forall l = 1, ..., L$

$$c_l(x,\omega) = \begin{cases} 1 & \text{if } F_l(x,\omega) \geqslant t_1 \\ -1 & otherwise \end{cases} \tag{5.3}$$

Comparing the weighted sum of theses L class labels as in [193] to another threshold t_2 allows to define the strong classifier for x as follows:

$$H(x) = \begin{cases} 1 & \text{if } \frac{1}{L}\sum_{l=1}^{L} \beta_l \, c_l(x,\omega) \geqslant t_2 \\ 0 & otherwise \end{cases} \tag{5.4}$$

A region x is classified as a background region if $H(x) = 0$.

5.2.4 Heuristic approach for background model maintenance

The procedure for background model maintenance is the same as that used in Chapter 4. In order to facilitate the reading, we recover some of the principal definitions as follows. The background maintenance relies on the mechanism used for adapting the learned model to the scene over time. For this step, we propose to suitably update the learned model by our IWOC-SVM using a new ensemble margin-based data selection approach called Small Votes Instance Selection (SVIS) introduced by Guo and Boukir [75]. The SVIS relies on a simple and efficient heuristic approach to provide SV candidates: selecting lowest margin samples. This heuristic significantly reduces the IWOC-SVM training task complexity while maintaining the accuracy of the IWOC-SVM classification. Once only support vector candidate samples are used to update the IWOC-SVM's models. The SVIS consists of an unsupervised ensemble margin that combines the first $c_{(1)}$ and second most voted class $c_{(2)}$ labels under the learned model. Let $v_{c_{(1)}}$ and $v_{c_{(2)}}$ denote the relative number of votes. Then the margin, taking value in [0,1] is:

$$m(x) = \frac{v_{c_{(1)}} - v_{c_{(2)}}}{L} \tag{5.5}$$

where L represents the number of best base classifiers in the ensemble. The first smallest margin samples are selected as support vector candidates. The final model is updated by the first smallest margin samples. This procedure is presented in the Algorithm 12.

5.3 Experimental results

The experiments were conducted in two recent public datasets: MSVS dataset [16] and RGB-D object detection dataset [36]. These datasets were chosen because they provide two types of informations so far been little explored in BS: multispectral and depth, respectively. The MSVS dataset consists of a set of 5 video sequences containing 7 multispectral bands and color video sequence (RGB) with different challenges such as gradual illumination changes,

Algorithm 12 Heuristic approach for model maintenance

Require: Final classifier H, test set $Z = \{z_1, z_2, ..., z_t\}$, weight distribution $\delta(z)$, user defined parameter *time*, user defined parameter η.
$i \leftarrow 1$
repeat
  if $H(z_i) = 1$ (background) **then**
    Compute the margin $m(z_i)$ by Eq. (5.5).
  end if
  if *time* is reached **then**
    Order all the test samples according to their margin values, in ascending order.
    The η smallest margin samples are selected as support vectors.
    $H(x)$ is updated using Z_1 and its weight $w \sim \delta(x)$.
  end if
  $i \leftarrow i + 1$
until $i > t$

shadows, camouflage effects (color similarity of object and background) and intermittent object motion. While the RGB-D dataset includes four different sequences of indoor environments, acquired with the Microsoft Kinect RGB-D camera, that contain different situations such as cast shadows, color and depth camouflage.

In the training step, we used kernalized IWOC-SVM as a base classifier with C = 1, with the same RBF (Radial Basis Function) kernel $K(.,.)$ [192]. The main advantage of RBF kernel is its good performance on non-linearly separable data. The pool of classifiers was homogeneous and consisted of 10 base classifiers of the same type. The pool of classifiers consisting of a maximum of 10 base classifiers. The classification threshold t_1 was set to 0.9 and t_2 to 0.5 for combining the best one-class classifiers. We divided the training set into three parts - first, we generate the base BS, next we calculate the adaptive importance for each BS model and finally, the base BS models with high importance are selected. In addition, we used a set of images to test our framework (detection step without ground truth for testing). All tests were done by a 10-fold cross validation. The video sequences was resized to 160×120 pixels in our experiments due computational cost. We used 9-dimensional features space for the MSVS dataset and 4-dimensional features space for the RGB-D dataset. In both datasets were used the grayscale and XCS-LBP [176] features. However, 7 multispectral bands and 1 depth information were adding for MSVS and RGB-D datasets, respectively.

5.3.1 Background detection on the MSVS and RGB-D datasets

We present the visual results on individual frame for *Scene* 05 (frame #413) from MSVS dataset and *GenSeq* (frame #996) from RGB-D dataset. Figure 5.2 shows the foreground detection results using our approach were displayed without any post-processing technique. The true positives (TP) regions are in white, true negatives (TN) regions in black, false positives (FP) regions in red and false negatives (FN) regions in green. Our method is able to detect the moving objects with fewer number of false detection for both datasets. Next, the performance of the BS is evaluated at region-level. Given the ground truth data, the correct-

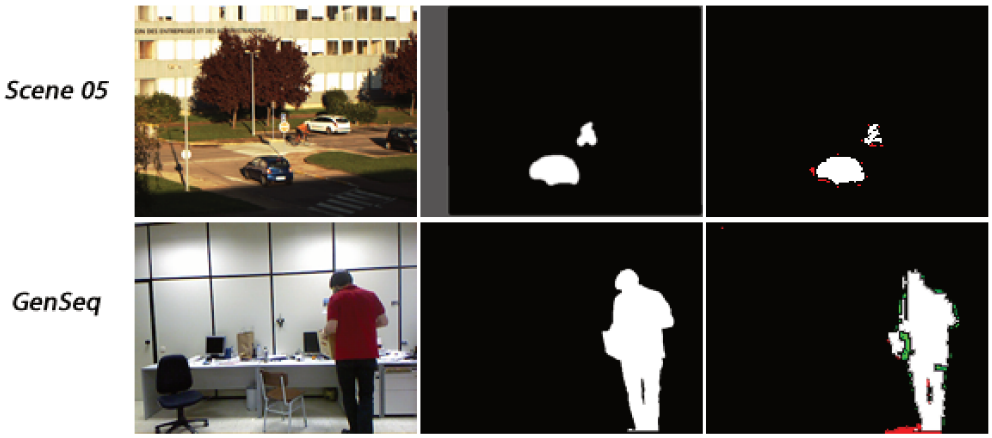


Figure 5.2: Results using the MSVS [16] (top row) and RGB-D [36] (bottom row) datasets – (a) original frame, (b) ground truth and (c) proposed method.

ness of foreground segmentation is measured using three classical measures: recall, precision and F-score. We compare the proposed method with OWOC RS [177](Chapter 4) and with the traditional classification approach for both datasets. This last one has the same setting of our ensemble, however, it uses only one IWOC-SVM classifier and grayscale feature. Table 5.3 shows the score of these methods evaluated on five scenes from MSVS dataset. The best scores are in bold. The proposed approach presented the best scores for *Scene 01*, *Scene 02* and *Scene 05*. In these scenes, the most frequent challenges for BS are color saturation, dynamic background, illumination changes, camouflage effects and intermittent object motion. In the *Scene 03* and *Scene 04*, the best score was presented for OWOC-RS approach. For the RGB-D dataset, as can be seen from Table 5.4 our approach presented the best score for all scenes, except for scene *DCamSeq*. Note that both the OWOC-RS and the proposed approach presented better performance than traditional classifications using only one classifier for both datasets. This prove the efficiency of the ensemble for feature selection in the BS task. An alternative that may further improve the score of our approach is the use of other features, however, as we can see in Tables 5.3 and 5.4, using only three types of resources were sufficient to achieve good results. More robust methods of superpixel as proposed in [213] can also be used to further improve the results of our approach.

Figures 5.3 and 5.4 illustrate the importance of each feature through its respective map features showing most important feature for each region and histogram for five video scenes from MSVS dataset and four scenes from RGB-D dataset. For each region, certain features are ignored or receive relatively low importance in favor of other more informative features. Then, a global histogram was normalized to obtain scores from 0-1, where higher scores meant highly informative features. Unlike traditional methods that the same feature (or set of features) is used globally for the whole video scene (and usually with the same level of importance), we present the potential of the proposed approach and its effectiveness to select the best features for BS task. As can be seen, the most important features for overall scenes were grayscale with high contribution, then XCS-LBP with medium contribution and multispectral features that presented low contribution from MSVS dataset. The grayscale

Table 5.3: Performance using the MSVS dataset [16].

Videos	Method	Precision	Recall	F-score
Scene 01	IWOC-SVM	**0.9814**	0.3378	0.5027
	OWOC-RS [177] (Chapter 4)	0.8500	**0.9580**	0.9008
	Superpixel-OWAOC [in this chapter]	0.9498	0.8799	**0.9135**
Scene 02	IWOC-SVM	0.7671	0.9410	0.8452
	OWOC-RS [177] (Chapter 4)	0.8277	0.8245	0.8727
	Superpixel-OWAOC [in this chapter]	**0.9627**	**0.9555**	**0.9591**
Scene 03	IWOC-SVM	0.8945	0.6123	0.7270
	OWOC-RS [177] (Chapter 4)	0.9326	**0.9965**	**0.9635**
	Superpixel-OWAOC [in this chapter]	**0.9787**	0.8999	0.9376
Scene 04	IWOC-SVM	0.9279	0.4287	0.5865
	OWOC-RS [177] (Chapter 4)	**0.9534**	0.8374	**0.8997**
	Superpixel-OWAOC [in this chapter]	0.8236	**0.9509**	0.8827
Scene 05	IWOC-SVM	0.0331	0.5430	0.0624
	OWOC-RS [177] (Chapter 4)	0.7316	**0.8392**	0.8400
	Superpixel-OWAOC [in this chapter]	**0.8691**	0.8695	**0.8693**

feature presented also the highest contribution from RGB-D dataset. Nonetheless, note that for *ColCamSeq* scene the XCS-LBP was the most important. It is important to note several state-of-the-art BS algorithms use grayscale feature for the whole image sequence, however, it is possible to observe from the feature map in the Figures 5.3 and 5.4 that different features were used for different regions of the image.

5.3.2 Computational costs

The key of success of the BS is due to its simplicity and also the low cost computational usually required by most of its methods. Ensemble for feature selection has proven to be an effective tool for BS, but usually it demands an high availability of computational resources. Therefore strategies to improve the computational time could prove interesting, for instance in our previous framework we proposed a weighted random subspace ensemble that require a large quantity of features to guarantee a good performance. Yet there is very little BS datasets that provide a lot of features, in addition, a huge feature set required also a high computational power. In our previous work, we used 26-dimensional features space while in this work only 9 (MSVS dataset) and 3 (RGB-D dataset) dimensional feature space were enough to achieve a good result. In this chapter, to further improve the computational costs we propose to use the superpixel approach instead pixel approach. The superpixel approach allow us to measure the feature statistics on a semantically meaningful atomic regions instead of individual pixels which can be provide redundant information. The experiments were made in Matlab R2013 a MacBook Pro with 2.2 GHz Intel Core i7. We collected the elapsed CPU time for training/validation and foreground detection. OWOC-RS has presented for training/validation the elapsed time is 5.44 sec/frame, while in foreground detection the elapsed time is 1.05 sec/frame. In this chapter, we define approximately 4000 superpixels for each scene instead of 19200 pixels from OWOC-RS. Note that the proposed approach can be up

Table 5.4: Performance using the RGB-D dataset [36].

Videos	Method	Precision	Recall	F-score
ColCamSeq	IWOC-SVM	**0.9898**	0.6706	0.7995
	OWOC-RS [177]	0.8887	0.7555	0.8167
	Superpixel-OWAOC [in this chapter]	0.9859	**0.8041**	**0.8858**
DCamSeq	IWOC-SVM	0.9255	0.8172	0.8680
	OWOC-RS [177]	**0.9774**	**1.0000**	**0.9885**
	Superpixel-OWAOC [in this chapter]	0.9245	0.9488	0.9365
GenSeq	IWOC-SVM	0.7427	0.7513	0.7470
	OWOC-RS [177]	0.7029	0.9239	0.7984
	Superpixel-OWAOC [in this chapter]	**0.8427**	**0.9513**	**0.8937**
ShSeq	IWOC-SVM	0.6024	0.6385	0.6199
	OWOC-RS [177]	0.7316	0.7392	0.7354
	Superpixel-OWAOC [in this chapter]	**0.7325**	**0.8389**	**0.7821**

to 4 times faster than OWOC-RS. The computational cost can be reduced by increasing the number of superpixels. However, this may lead to less accurate segmentations.

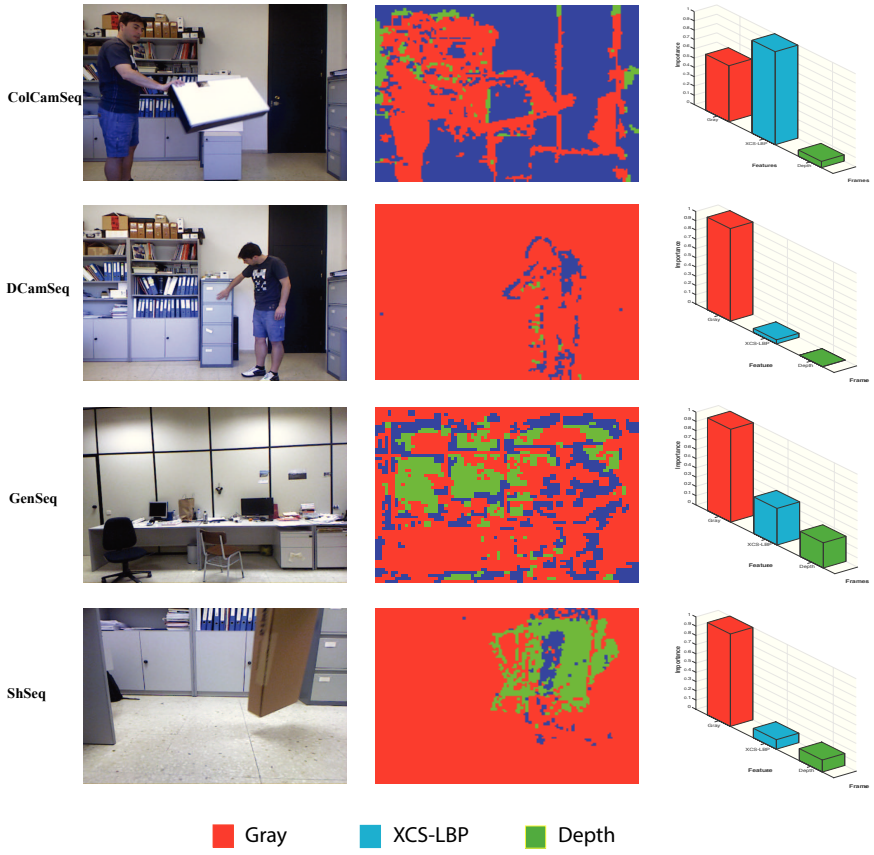


Figure 5.3: Results on RGB-D dataset [36] – (a) original frame, (b) features map and (c) its respective histogram of features importance.

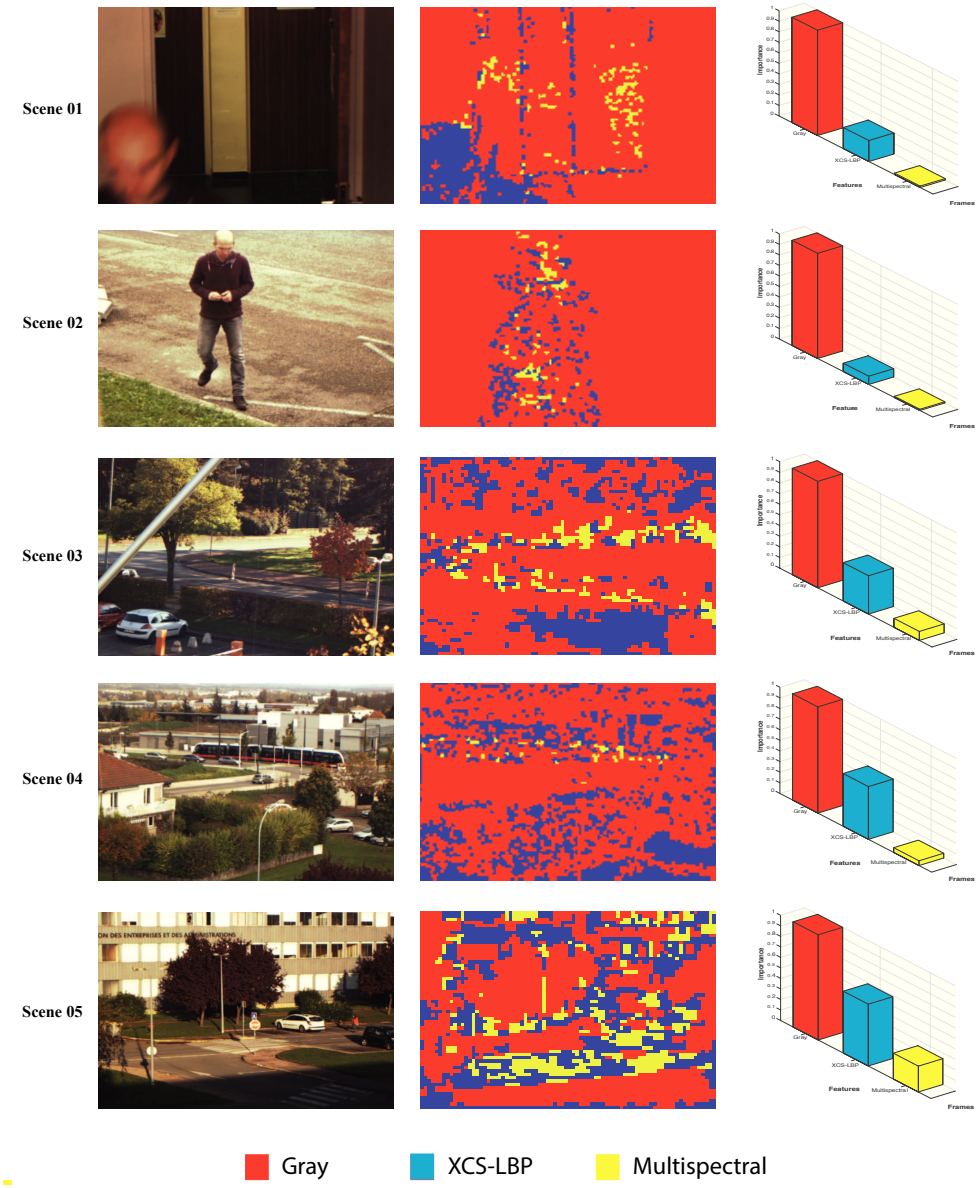


Figure 5.4: Results on MSVS dataset [16] – (a) original frame, (b) map feature and (c) its respective histogram of features importance.

5.4 Conclusion

In summary, we proposed a novel methodology to select the best features based on wagging. Our proposal is able to select suitable features for each region to distinguish the foreground objects from the background ones. In addition, it uses a superpixel approach that not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance. Our framework also uses a mechanism to update the importance of each feature discarding insignificant features over time. Experimental results on two challenging datasets have shown the potential of the proposed approach and its effectiveness to select the best features for distinct regions in a video sequence. A future work may address how to update the importance of each feature, discarding insignificantly features over time without ground-truth data.

In the next chapter we present a novel Opponent Color Local Binary Pattern from Three Orthogonal Planes (OCLBP-TOP) descriptor for applications in the field of dynamic texture recognition.

Chapter 6

A novel joint color-texture descriptor for dynamic texture recognition

In this chapter, we propose a novel Opponent Color Local Binary Pattern from Three Orthogonal Planes (OCLBP-TOP) descriptor for applications in the field of dynamic texture recognition. The OCLBP-TOP fuses the texture and color information, combining the Opponent Color Local Binary Patterns (OCLBP) with LBP on Three Orthogonal Planes (LBP-TOP). As such, it allows to extract not only color information, but also a more detailed information from video sequences. The experiments conducted on real videos from the Dyntex++ and YUPENN Dynamic Scenes show that the proposed OCLBP-TOP outperforms not only LBP-TOP and OCLBP as expected, but also three state-of-the-art descriptors, in particular its direct recent competitor, called Local Gabor Binary Patterns from Three Orthogonal Planes (LGBP-TOP). These descriptors were especially designed for the dynamic texture recognition. This chapter presents a particular work realized in conjunction with the Computer Vision Center (CVC) at Autonomous University of Barcelona (UAB). The work presented here is currently under revision for publication in the IET Computer Vision Journal [179].

6.1 Motivation

Dynamic (or temporal) texture analysis attracts growing attention in the computer vision community for applications such as automatic environment surveillance, synthesis, segmentation and recognition. Unlike static textures which are patterns describing pixel intensity variations that repeat spatially in an image, dynamic textures are motion patterns, *i.e.* image sequences of moving scenes that present certain stationarity properties not only in space but also in their dynamics over time [56,224]. Dynamic textures are then of prime importance when the video sequence at hand continuously changes in shape and appearance. Some examples of dynamic textures in the real world are shown in Figure 6.1. From left to right and top to bottom: forest fire, waterfall, flock of birds in flight, vegetation in the wind, water, vehicle traffic, crowd of people running and insect swarms. Given such a video sequence, the recognition of dy-

Figure 6.1: Examples of dynamic textures in the real world.

namic textures consists in identifying to which class (*e.g.* water, vehicle traffic, fire, etc.) it belongs to. Many approaches have been proposed for that purpose, *e.g.* Linear Dynamic System (LDS) [4, 42, 56, 158], GIST [148, 182] Wavelets-based methods [53, 61, 99, 224], Spatiotemporal Oriented Energies (SOE) [52], Slow Feature Analysis (SFA) [195, 196, 237], Local Space-Time (HOG/HOF) [113] and Complementary Spacetime Orientation descriptor (CSO) [60] descriptor. However, the descriptors based on local binary patterns have also attracted the attention of the image processing and pattern recognition community for other tasks. The Local Binary Pattern histograms from Three Orthogonal Planes (LBP-TOP) is the best known descriptor based on local binary pattern that combines motion and appearance for describing dynamic textures. It is based on the differences between neighboring pixels on three orthogonal planes: the space plane XY, the space-time transitions planes XT and YT. The LBP-TOP has been successfully used in various applications such as dynamics facial expression recognition [240], action recognition [103], segmentation [43] and analysis of facial paralysis [83]. Note that very recent variants and modifications have been proposed to further increase its robustness and its discriminative power [30, 153]. They have been successfully used in various applications, such as: background subtraction (see Chapter 2), human activity recognition [191], speaker identification [40, 239], facial actions [5, 152, 219, 238] and texture segmentation [44].

Until recently, to our best knowledge, there have been no previous descriptors based on local binary patterns which processed altogether color-texture information for the dynamic texture problem. State-of-the-art dynamic texture descriptors operate on gray-scale level scene, ignoring color information. The color and texture are two of the most significant low level visual cues for visual recognition. In the past decades, the combination of color and texture concerning the static texture problem in joint descriptors has been debated [133]. The research indicates that joint color-texture descriptors and combined color and texture features are outperformed by either color or gray-scale texture. In the last few years, Mäenpää and Pietikäinen [133] introduced an Opponent Color Local Binary Pattern (OCLBP) descriptor to describe color-texture joint. It extracts more detailed information and it has a state-of-art performance for the static texture problem. However it is not suitable for dynamic texture as it does not capture the motion information. We believe the joint color-texture information may provide useful scene and motion information for dynamic texture recognition. In this chap-

ter, we propose to extend the spatial color-texture OCLBP descriptor to the spatio-temporal domain by combining it with the LBP-TOP one. By fusing color and dynamics textures, the derived OCLBP-TOP extracts more detailed information from the video sequence to be analyzed. Our contributions can be summarized as follows:

- A robust combination of the descriptor OCLBP with the descriptor LBP-TOP, that allow us to be more robust on the dynamic texture recognition in presence of the main challenges such as illumination changes.
- A detailed comparative evaluation of our descriptor OCLBP-TOP against other five state-of-the art descriptors on two large scale dataset that are Dyntex++ and YUPENN.

The rest of this chapter is organized as follows. The construction of the new 3D joint color-based texture descriptor is presented in Section 6.2. In Section 6.3, we give experimental results obtained on real videos that compare the proposed OCLBP-TOP descriptor to its direct competitors. Finally, the conclusion is shown in Section 6.4.

6.2 3D joint color-texture descriptor

It is challenging to find joint color-texture descriptors based on local binary patterns for dynamics texture tasks. To address this issue, we have developed an Opponent Color Local Binary Pattern from Three Orthogonal Planes (OCLBP-TOP). Given a finite color video sequence of a texture in motion and considering the cooccurrences statistics on the three planes (XY_k plane, XT_k plane and YT_k plane), we extract six-opponent-color video on these three orthogonal planes, where k is the opponent color space. The opponent color space can be computed as [96, 202]:

$$\begin{aligned} red-green &: \mathrm{O}_1 = (r-g)/\sqrt{2}, \\ yellow-blue &: \mathrm{O}_2 = ((r+g)-2b)/\sqrt{6} \\ luminance &: \mathrm{O}_3 = (r+g+b)/\sqrt{3}. \end{aligned}$$

The intensity is represented in channel O_3 and the color information is in the channels O_1 and O_2. In addition to the perception correlation properties of the opponent color space, one important advantage of this space is that the O_3 axis, can be more closely sampled than O_1 and O_2, thereby decreasing the sensitivity of color matching to a difference in the global brightness of the video. Then the LBP is computed on three orthogonal planes XY_k, XT_k and YT_k on the six new opponent color video. Note that in the following, we will remind the LBP equation already defined in the Chapter 3. Given a pixel at a certain location, considered as the center pixel $c = (x_c, y_c)$ of a local neighborhood composed of P equally spaced pixels on a circle of radius R, the LBP descriptor applied to can be expressed as:

$$LBP_{P,R}(c) = \sum_{i=0}^{P-1} s\left(g_i - g_c\right) 2^i \tag{6.1}$$

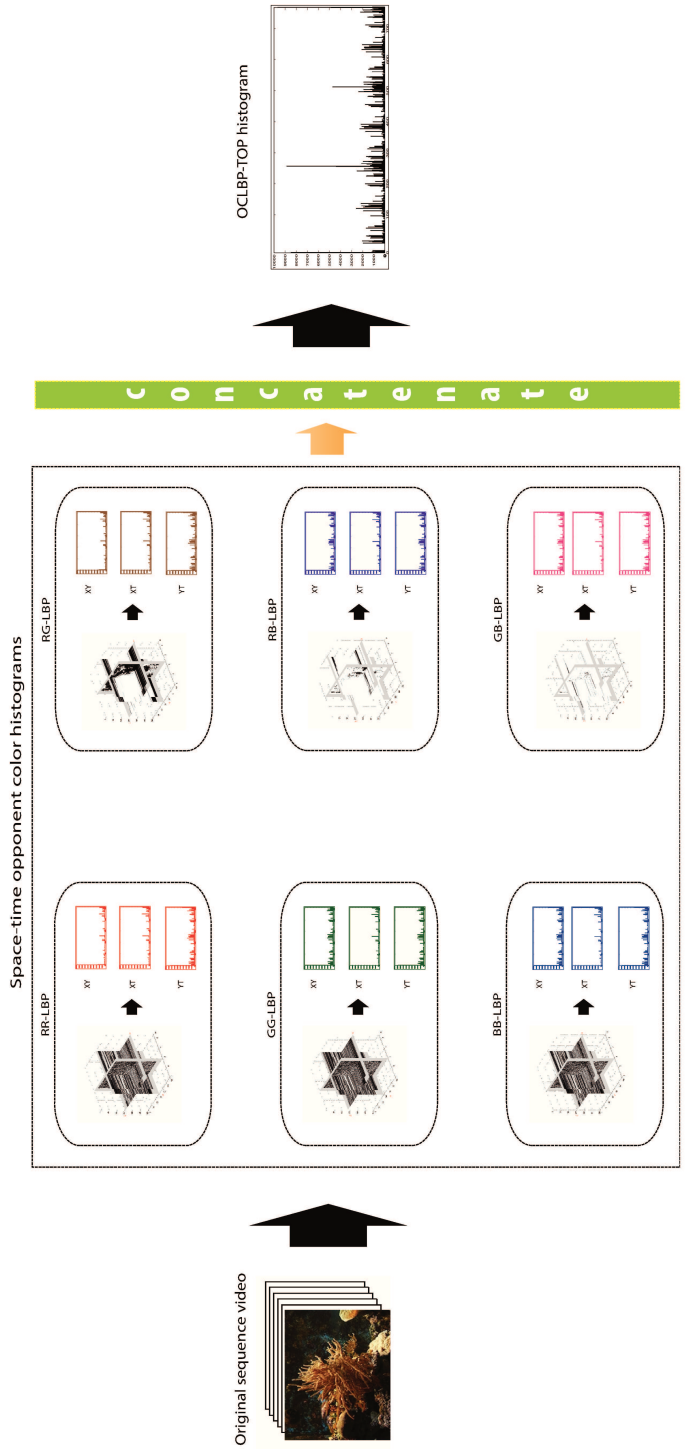


Figure 6.2: Our OCLBP-TOP descriptor.

in our case g_c is an opponent color value of the center pixel c, g_i is an opponent color of each neighboring pixel, and s is a thresholding function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

The resulting binary number is of length P, and there are 2^P possible different labels to be obtained from an LBP-image which histogram can be used as a texture descriptor. The computation of the ordinary LBP for a neighborhood of size $P = 8$ on a circle of radius $R = 1$, resulting in an histogram of size $2^8 = 256$, is illustrated in Figure 2.3 (see Chapter 2).

The opponent color local patterns are extracted from the XY_k, XT_k and YT_k. The XY_k plane contain information about the appearance, while the co-occurrence statistics of motion in horizontal and vertical directions are included in the labels from the XT_k and YT_k planes. In the OCLBP-TOP descriptor, the three planes intersect in the center pixel and six distinct patterns are extracted in function of that central pixel for each XY_k, XT_k and YT_k. For each pixel in opponent-color images from XY_k, XT_k and YT_k planes, a six binary code is built by thresholding its neighborhood in a circle from these three planes separately with the value of the center pixel. Three inter-channel (*RG*, *RB*, *GB*) and three intra-channel (*RR*, *GG*, *BB*) histograms for each individual XY_k, XT_k and YT_k are created to collect the occurrences of different binary patterns, which are denoted as *RG*-LBP, *RB*-LBP, *GB*-LBP, *RR*-LBP, *GG*-LBP and *BB*-LBP. This results in $3 \times 6 \times 2^P$ dimensional histograms, which are then concatenated into a single histogram to create a global description of the dynamics texture with the spatial-temporal and joint color-texture features. The final histogram can be expressed as:

$$H_i = \sum_{x,y,t} I\left(f_{j_k}(x,y,t) = i\right) \qquad i = 0,1,...,n_j; \qquad j = 1,2,3; \qquad k = 1,...,6 \tag{6.3}$$

where n_j is the number of different labels produced by the OCLBP-TOP descriptor in the jth plane, k is the number of opponent colors, f_j is the central pixel at coordinates (x,y,t) in the jth plane and $I(A)$ is 1 if A is true and 0 otherwise.

In the OCLBP-TOP, the dynamic texture is encoded by the LBP, while the appearance and the motion in two directions of the joint dynamic color-texture are taken, incorporating spatial-domain information and two spatio-temporal co-occurrence statistics together. In the OCLBP-TOP descriptor, the R_k is applied in the axes X_k, Y_k and T_k and the P_k number in the XY_k, XT_k, and YT_k. The planes can be also different, which can be indicated as R_{X_k}, R_{Y_k}, R_{T_k}, P_{XY_k}, P_{XT_k} and P_{YT_k}. The corresponding OCLBP-TOP is called as OCLBP-TOP$_{P_{XY_k},P_{XT_k},P_{YT_k},R_{X_k},R_{Y_k},R_{T_k}}$ planes, that is, P_k= P_{XY_k}=P_{XT_k}=P_{YT_k} and R = R_{X_k}= R_{Y_k}=R_{T_k}. At times, the R_k in three planes are the same and the P_k in XY_k, XT_k and YT_k axis. In that case, we denote OCLBP-TOP$_{P_k,R_k}$.

The OCLBP-TOP descriptor may be useful for dynamic-texture analysis, mainly because of the large quantity of richer information that it can extract from the video. It is because our descriptor describes joint color texture in spatio-temporal domain. The OCLBP-TOP extracts six times greater than LBP-TOP. The LBP-TOP considers only grayscale information in the

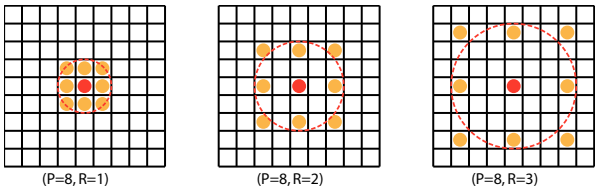


Figure 6.3: Circularly symmetric neighbor sets for different R and $P = 8$ in the LBP space.

spatio-temporal domain. We show in the next section that the proposed approach allows to improve the performance of a dynamic texture classification method, as compared to other local binary pattern based approaches and two popular methods in the filed of video sequence recognition.

6.3 Experiments

6.3.1 Datasets

The performance of our proposed descriptor was evaluated on two public large and diverse datasets dedicated to the color dynamic-texture recognition. We give a brief introduction of these datasets as follows.

- the Dyntex++ [67] which is a selected version of the Dyntex dataset [151], composed of 3 600 video sequences grouped in 36 classes, each of which containing 100 sequences of a fixed size $50 \times 50 \times 50$ (width$\times$height$\times$# of frames). Various kinds of dynamic texture are present, ranging from struggling flames to whelming waves, from sparse curling smoke to dense swaying branches.

- the YUPENN dataset [52] that contains 420 videos of dynamic scene categories grouped in 14 classes, each class containing 30 videos. The sequences in YUPENN have important variations such as frame rate, scene appearance, scale, illumination, and camera viewpoint.

There is a limited number of dynamic-texture datasets in the literature because of the difficulties in collecting DT sequences. Results of many existing approaches have been reported based on the UCLA dynamic texture dataset [164]. But this dataset presents only gray-scale images and our descriptor needs color features making its application impossible on this dataset. Figures 6.4 and 6.5 show examples frames of some scenes of Dyntex++ and YUPENN datasets used in this chapter, respectively.

Table 6.1: Overall classification results (%) for evaluation different values of P, R in the OCLBP-TOP space.

R	Our Descriptor	YUPENN Dynamic Scenes (%)
1	OCLBP-TOP$_{2,1}$	45.00
	OCLBP-TOP$_{4,1}$	76.90
	OCLBP-TOP$_{8,1}$	**86.90**
2	OCLBP-TOP$_{2,2}$	26.19
	OCLBP-TOP$_{4,2}$	72.85
	OCLBP-TOP$_{8,2}$	82.85
3	OCLBP-TOP$_{2,3}$	24.04
	OCLBP-TOP$_{4,3}$	73.57
	OCLBP-TOP$_{8,3}$	85.47

6.3.2 Parameter settings

The selection of appropriate parameters is always a key issue. The OCLBP-TOP has only few parameters to optimize, making this task much easier. The P and R parameters of our OCLBP must be carefully selected not to affect the descriptor performance. In addition, small changes in P may cause big differences in the length of the feature vector. According to previous studies on LBP [146, 240], the best R are normally smaller than 3 and P is 2^i $(i = 1, 2, 3...)$. In our proposed descriptor, when the number of neighboring points increases, the number of patterns OCLBP-TOP will become large: $3 \times 6 \times 2^P$. Thus only the results for $P = 2, 4$ and 8 are given in Table 6.1. In all our experiments, we used a leave-one-out-cross-validation strategy [54] with linear SVM (Support Vector Machine) to evaluate our descriptor. The Dyntex++ dataset was used to evaluate different values of P and R in the OCLBP-TOP. Table 6.1 presents the overall recognition rate. It can be seen that the OCLBP-TOP performs very well for $P = 8$ and $R = 1$. For the influence of P, we can obtain a shorter feature vector, however a small P loses more information. Nonetheless, the large P value improves the recognition accuracy, but it generates a long histogram and therefore a high memory consumption. For the influence of R, we can see that for a fixed P the best performance is obtained for $R = 1$. Figure 6.3 shows the case among different values of R and $P = 1$. We note there is a loss of information as the R value is higher because neighboring pixels are not considered in the calculation of the LBP. Therefore, we opted to use of the $P = 8$ and $R = 1$ for all the experiments in this chapter. Table 6.1 shows that an accuracy of 86.90% is obtained for OCLBP-TOP using P = 8 with a feature vector length of 4608 bits.

6.3.3 Comparison with state-of-the-art

A brief summary of all the descriptors we compared can be found in Table 6.2. First, we compare our descriptor to some LBP-based descriptors with $P = 8$ neighbors on a circle of radius $R = 1$:

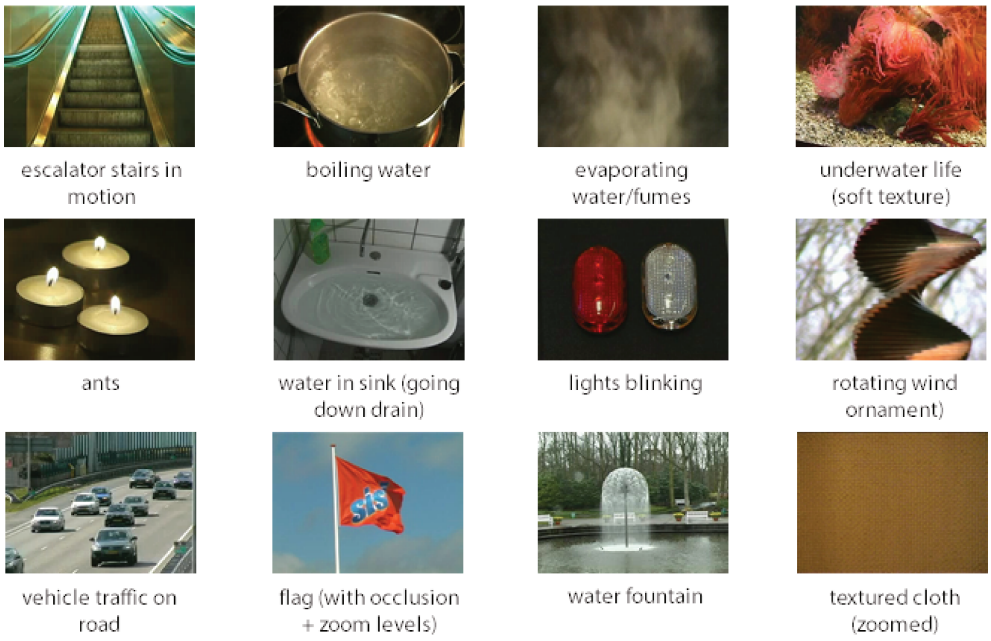


Figure 6.4: Sample frames of classes from the Dyntex++ dataset.

- OCLBP [133],
- LBP-TOP [240] as the reference local binary pattern spatio-temporal extension,
- LGBP-TOP [5], which extends the Local Gabor Binary Pattern [170] to the temporal domain. LGBP consists of applying a set of g Gabor filters at different scales and orientations to a number of s non-overlapping sub-images, then describing each resulting filtered sub-image using Uniform LBP [145, 146], and concatenating the corresponding histograms. A local binary pattern is called uniform if the number of bitwise transitions from 0 to 1 or vice versa, considering the pattern circular, is at most 2. For a P-dimensional pattern, there are only $P \times (P-1)+3$ different labels, so that the LGBP-based feature size is $g \times s \times (P \times (P-1)+3)$, and its TOP extension is obviously 3 times larger. In the experiments, we used $g = 18$ and $s = 16$ (4×4 grid). Combining spatial and dynamic texture analysis with Gabor filtering allows to achieve unprecedented levels of recognition accuracy in real-time. While LBP-TOP features risk being sensitive to misalignment of consecutive images, a rigorous analysis of the descriptor shows the relative robustness of LGBP-TOP to image registration errors caused by errors in rotational alignment.
- our proposed OCLBP-TOP.

Since the two last descriptors produce high dimensional features, respectively $3 \times 18 \times 16 \times (8(8-1)+3) = 50976$ and $6 \times 3 \times 2^8 = 4608$, a post-processing step was also tested. It consisted of applying Principal Component Analysis (PCA) to reduce the dimensionality,

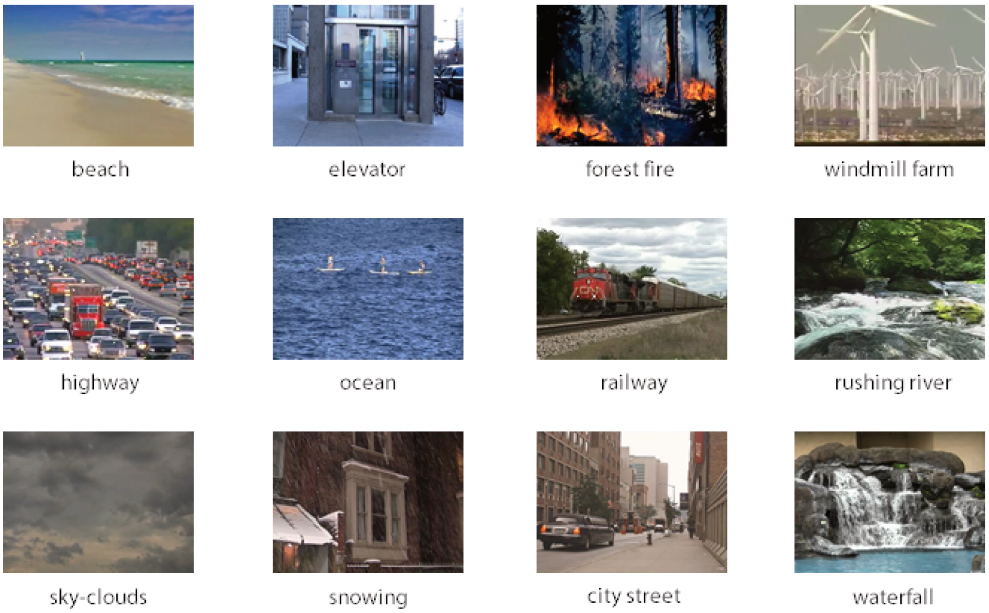


Figure 6.5: Sample frames of scenes from the YUPENN Dynamic Scenes dataset.

keeping the first $3 \times 2^P = 768$ principal components which equals the smallest feature size of the local binary pattern competitors (LBP-TOP).
We also used two other methods among those most popular in recognizing video sequences, namely:

- fast and dense HOG/HOF [200] is a local space-time descriptor derived from the original one in [113] which consists in using a bag of spatio-temporal features that identify 3D interest points with high variations in appearance captured by the Histogram of Oriented Gradients (HOG) and motion captured by the Histogram of Optical Flow (HOF). Gradients magnitude responses and flow displacements vectors are computed on a number $n_x \times n_y \times n_t$ of blocks in space and time, each orientation is quantized on a single scale sampling into b_g and b_f bins respectively for HOG and HOF. The descriptor is then of size $n_x \times n_y \times n_t \times (b_g + b_f)$. We used $n_x = 3$, $n_y = 3$, $n_t = 2$ and $b_g = b_f = 8$ so that the feature space was $3 \times 3 \times 2 \times (8+8) = 288$ dimensional, as in [200]. The dense HOG/HOF is a local descriptor, thus it is less sensitive to noise or occlusion, but it requires the detection of sufficient and relevant interest points. In addition, the HOG/HOF requires the quantization of large amount of data, because of the bag-of-features model.
- GIST3D [182] is a global video descriptor which is computed by applying a bank of 3-D spatio-temporal Gabor filters on the frequency spectrum of a video sequence, so it integrates information about both the motion and the scene structure. The GIST3D feature space is $g \times s \times k$ dimensional, where g is the number of Gabor filters at dif-

ferent scales and orientations in the spatio-temporal frequency domain, s is a number of non-overlaping sub-volumes of user-defined size ($width \times height \times \#frames$) and k is a number of key-clips (by default, the authors define a key-clip as a video block of 64 frames). We used $g = 68$, $s = 512$ and $k = 1$, resulting in 34 816 features. The local video descriptors may require the background subtraction or tracking, whereas the GIST3D does not need these steps and represents each video with a single feature vector. The global descriptors are in general not invariant to viewpoint changes and camera motion.

6.3.4 Results and discussions

The classification results (correct classification rate in %) of the tested methods are reported in Table 6.2 for both datasets. Highest scores are shown in bold. As it can be seen, the proposed descriptor gives the highest accuracy for both datasets. Without PCA dimensionality reduction, it outperforms all the others (including the two video sequences recognizers), in particular on the larger and more diverse Dyntex++ dataset for which it appears to be approximatively 10% better. The same increase of performance (10%) can be noticed as compared to the ordinary OCLBP it extends to the spatio-temporal domain, for both datasets. The performance of OCLBP-TOP is not necessarily related to the feature size when compared to LGBP-TOP and HOG/HOF whose produced histograms are respectively 11 and 7 times bigger. Even if the dimensionality reduction by PCA affects the performance of the proposed OCLBP-TOP as one could expect, especially for the Dyntex++ dataset, it is worthy of note that it gives quite similar results to LBP-TOP, but significantly better results than LGBP-TOP.

Table 6.2: Overall classification results (%)

Descriptors	Dyntex++ (%)	YUPENN Dynamic Scenes (%)	Feature Size
OCLBP (2004) [133]	70.14	77.85	1 536
LBP-TOP (2007) [240]	71.88	85.37	768
OCLBP-TOP [this thesis]	**80.58**	**86.90**	4 608
LGBP-TOP (2013) [5]	68.69	84.47	50 976
LGBP-TOP + PCA	52.08	63.57	768
OCLBP-TOP + PCA [this thesis]	73.04	84.76	768
HOG/HOF (2008) [113]	72.75	78.80	288
GIST3D (2012) [182]	70.43	63.33	34 816

To go a little bit further in the analysis, Tables 6.4 and 6.5 show some popular class performance measures (Precision, Recall and F-score in %) obtained on the Dyntex++ and YUPENN dataset using the tested local binary pattern on Three Orthogonal Planes (TOP) descriptors. We analyze the descriptors studied in this work in various cases:

Case 1. Performance of descriptors close to 100%: For the former dataset, both the color and the texture are very important in some scenes such as: *blossoming tree in the*

wind, *escalator stairs in motion*, *waves on beach*, *underwater life (soft texture)*, *underwater life (pulsating jellyfish)*, *underwater life (flowers swaying with current)*, *waterfall*, *branches swaying in wind* and *smoke*. This explains the much better precision, recall and F-score measures obtained with our OCLBP-TOP on these particular classes. However, in the scenes *boiling water* and *wash cycle* that are known to present more structured texture, the texture information alone is sufficient, so that LBP-TOP reaches better scores while LGPB-TOP fails and OCLBP-TOP is in the middle. On the opposite, for scenes where the color information is crucial such as: *river water* and *rain on water*, LGBP-TOP is much more efficient thanks to the Gabor filtering and the huge feature size, but our OCLBP-TOP is most of times between the two. Same remarks hold for the latter dataset, where the color and texture appeared to be very relevant in scenes like: *beach*, *lightning storm* and *rushing river*, whereas the texture information is sufficient in some other scenes such as: *forest fire*, *highway*, *ocean* and *snowing*. Although the proposed descriptor gives the highest average class performance measures for both datasets (see Table 6.6), the errors in classification may occur if the color and texture are similar as situations shown in the Figure 6.6.

In the case in which only one descriptor performs better than others: There are some classes from Dyntex++ dataset that only LBP-TOP and OCLBP-TOP are able to classify correctly, such as *grass swaying in wind*, *evaporating water/fumes*, *underwater life (soft texture)* and *water in sink*. This is due to the fact that color and texture components are more discriminative in these classes. However, the same descriptors cannot classify properly some scenes such as *artificial hair*, *ants* and *birds flying in sky*. This implies that all measures (precision, recall F-score) had a performance of 0%. For example, the *artificial hair* class was misclassified as *underw. life (more structured)* class. This can be explained by color and texture similarities in these classes. On the other hand, the LBP-TOP classified *artificial hair* scene as *water fountain* due to its high texture similarities. It's important to note that only LGBP-TOP was able to classify the *artificial hair*, *ants* and *birds flying in sky*, possibly due to color similarities in these scenes.

Case 3: In the case in which each descriptors reach 100%: In some cases precision, recall F-score measures had 100% of success. The OCLBP-TOP was also as successful for textures as: *underw. life (pulsating jellyfish)*, *underwat. life (flowers swaying with current)* and *lamp globes swaying.*. In these scenes the texture and color are very significant. The LGBP-TOP also had 100% of success in some scenes in which color is very predominant such as: such as: *artificial hair*, *rain on water* and *water fountain*. Meanwhile in scenes as the *evaporating water/fumes* the OCLBP-TOP had 100% of accuracy. In these scenes only the texture feature is more significant.

Case 4. In which descriptors have bad performance (near 0%): We noted also that any descriptor evaluated in this study was able to classify correctly the scene *Faucet water*, please see Table 6.4. This may be explained by the fact that only one sequence is available for this class in the Dyntex++ dataset, combined to a leave-one-out strategy. Note however that the OCLBP-TOP classified *textured cloth* scene as *blossoming tree in the wind*,

and LBP-TOP classified the same scene as *water fountain*. In addition, the LBPG-TOP classified *textured cloth* scene as *Faucet water* scene.

Figure 6.6: Similar images of different classes. From left to right: flag, and water fountain classes from Dyntex++ dataset, fountain, and waterfall classes from YU-PENN dataset.

6.3.5 Computational costs

The final result we give is about the computational time which may be important for some application. Table 6.3 shows the average computational time (in seconds) to process a video block of $256 \times 256 \times 64$ (width$\times$height$\times$# of frames). Not surprisingly, the proposed OCLBP-TOP needs much more time than the others local binary pattern based descriptors, because of both the TOP extension (as compared to OCLBP), and the six separate channels computation (as compared to LBP-TOP). This is the price to be paid for combining color information together with the texture so that the classification performance of dynamic textures increase. Note that the times obtained using HOG/HOF and GIST3D are not achievable using local binary patterns.

Table 6.3: Average computational time results

Descriptors	Computational Time (s)
OCLBP (2004) [133]	39.94
LBP-TOP (2007) [240]	47.88
OCLBP-TOP [this thesis]	357.87
LGBP-TOP (2013) [5]	19.03
HOG/HOF (2008) [113]	5.41
GIST3D (2012) [182]	**4.93**

Table 6.4: Class performance measures (%) of the local binary patterns on Three Orthogonal Planes (TOP) for the Dyntex++ dataset

	LBP-TOP			LGBP-TOP			OCLBP-TOP		
Class	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
Textured cloth	0	0	0	0	0	0	0	0	0
Artificial hair	0	0	0	**100**	**100**	**100**	0	0	0
Blossoming tree in the wind	80.0	**100**	88.9	83.3	62.5	71.4	**88.9**	**100**	**94.1**
Escalator stairs in motion	80.0	**57.1**	66.7	71.4	45.5	55.6	**100**	**57.1**	**72.7**
Waves on beach	85.7	82.8	84.2	66.7	50.0	57.1	**93.3**	**96.6**	**94.9**
Grass swaying in wind	70.8	68.0	69.4	0	0	0	**87.5**	**84.0**	**85.7**
Boiling water	**80.0**	**100**	**88.9**	66.7	**100**	80.0	50.0	**100**	66.7
Evaporating water/fumes	**100**	**100**	**100**	0	0	0	80.0	80.0	80.0
River water	66.7	64.0	65.3	**87.5**	**87.5**	**87.5**	83.3	80.0	81.6
Faucet water	**100**	33.3	50.0	70.0	77.8	73.7	75.0	**100**	**85.7**
Fish swimming	66.7	**100**	**80.0**	**100**	25.0	40.0	60.0	75.0	66.7
Underwater life (soft texture)	63.6	63.6	63.6	0	0	0	**80.0**	**72.7**	**76.2**
Underw. life (more structured)	40.0	**50.0**	44.4	**66.7**	40.0	**50.0**	28.6	**50.0**	36.4
Underw. life (pulsating jellyfish)	66.7	66.7	66.7	83.3	**100**	90.9	**100**	**100**	**100**
Underwat. life (flowers swaying with current)	66.7	**100**	80.0	**100**	66.7	80.0	**100**	**100**	**100**
Ants	0	0	0	**66.7**	**57.1**	**61.5**	0	0	0
Waterfall	57.1	50.0	53.3	50.0	28.6	36.4	**88.9**	**100**	**94.1**
Candles	85.7	66.7	75.0	**100**	66.7	**80.0**	70.0	**77.8**	73.7
Rain on water	75.0	75.0	75.0	**100**	**100**	**100**	75.0	75.0	75.0
Flushing water	75.0	60.0	66.7	56.5	**81.3**	66.7	**100**	60.0	**75.0**
Water in sink	**83.3**	**100**	**90.9**	0	0	0	**83.3**	**100**	**90.9**
CD in CD player	**66.7**	**66.7**	**66.7**	35.7	29.4	32.3	**66.7**	**66.7**	**66.7**
Wash cycle	**87.5**	**100**	**93.3**	83.3	62.5	71.4	85.7	85.7	85.7
Water pouring into sink	71.4	71.4	71.4	72.7	**88.9**	**80.0**	**83.3**	71.4	76.9
Lamp globes swaying	**100**	66.7	80.0	**100**	44.4	61.5	**100**	**100**	**100**
Lights blinking	**100**	33.3	50.0	69.7	**76.7**	**73.0**	50.0	66.7	57.1
Leaves on branches swaying with wind	76.5	81.3	78.8	54.5	60.0	57.1	**93.3**	**87.5**	**90.3**
Birds flying in sky	0	0	0	**53.2**	**67.6**	**59.5**	0	0	0
Pond water	64.3	52.9	58.1	**77.8**	**77.8**	**77.8**	75.0	70.6	72.7
Rotating wind ornament	69.2	**100**	**81.8**	75.0	75.0	75.0	**77.8**	77.8	77.8
Vehicle traffic on road	72.7	88.9	80.0	**100**	28.6	44.4	81.8	**100**	**90.0**
Flag	75.9	73.3	74.6	72.2	**89.7**	**80.0**	**81.5**	73.3	77.2
Branches swaying in wind	61.5	80.0	69.6	77.8	84.0	80.8	**90.0**	**90.0**	**90.0**
Water fountain	58.1	67.6	62.5	**100**	**100**	**100**	73.2	81.1	76.9
Clouds	**100**	66.7	80.0	71.4	**100**	**83.3**	85.7	66.7	75.0
Smoke	90.9	62.5	74.1	70.4	76.0	73.1	**92.3**	**75.0**	**82.8**

Table 6.5: Class performance measures (%) of the local binary patterns on Three Orthogonal Planes (TOP) for the YUPENN dataset

	LBP-TOP			LGBP-TOP			OCLBP-TOP		
Class	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
Beach	89.3	83.3	86.2	81.3	86.7	83.9	**93.3**	**93.3**	**93.3**
Elevator	90.6	96.7	93.5	**96.8**	**100**	**98.4**	93.5	96.7	95.1
Forest Fire	**90.0**	**90.0**	**90.0**	72.2	86.7	78.8	86.7	86.7	86.7
Fountain	65.4	56.7	60.7	**83.3**	**66.7**	74.1	74.1	**66.7**	70.2
Highway	**80.6**	**83.3**	**82.0**	79.3	76.7	78.0	80.0	80.0	80.0
Lightning Storm	89.3	83.3	86.2	86.7	86.7	86.7	**93.5**	**96.7**	**95.1**
Ocean	**100**	**100**	**100**	96.7	96.7	96.7	96.8	**100**	98.4
Railway	84.4	**90.0**	87.1	**96.0**	80.0	**87.3**	86.2	83.3	84.7
Rushing River	84.8	93.3	88.9	80.6	83.3	82.0	**93.5**	**96.7**	**95.1**
Sky-Clouds	**96.3**	86.7	**91.2**	84.8	**93.3**	88.9	92.6	83.3	87.7
Snowing	**96.3**	**86.7**	**91.2**	78.8	**86.7**	82.5	86.7	**86.7**	86.7
Street	**92.9**	86.7	89.7	87.1	90.0	88.5	85.3	**96.7**	**90.6**
Waterfall	62.2	**76.7**	68.7	**88.0**	73.3	80.0	**75.0**	70.0	72.4
Windmill Farm	80.6	83.3	82.0	**90.0**	**90.0**	**90.0**	77.4	80.0	78.7

Table 6.6: Average measures (%) of the local binary patterns on Three Orthogonal Planes (TOP) for the Dyntex++ and YUPENN datasets

	LBP-TOP			LGBP-TOP			OCLBP-TOP		
Data set	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F
Dyntex++	67.7	65.2	64.7	66.2	59.7	60.6	**71.7**	**72.8**	**71.3**
YUPENN	85.9	85.5	85.5	85.8	85.5	85.4	**86.8**	**86.9**	**86.8**

6.4 Conclusion

In summary, a new 3-dimensional joint color-texture descriptor for dynamic texture analysis is proposed. It combines the strengths of local binary patterns and it describes joint color-texture in a spatio-temporal domain. Then, we compared the OCLBP-TOP with its direct competitors LBP-TOP and LGBP-TOP on real videos of Dyntex++ and YUPENN Dynamic Scenes datasets. The experimental results have shown that OCLBP-TOP outperforms the LBP-TOP, LGBP-TOP descriptors, and other three traditional methods. In addition, our descriptor can be applied in various type of applications including facial expression analysis, human activity recognition, among others.

Chapter 7

Conclusions

In this thesis we set out to improve background subtraction by focusing on visual features. Background subtraction is a crucial task in many computer vision applications including surveillance devices in public spaces, traffic monitoring and industrial machine vision. We focused on developing robust texture descriptor to deal with illumination changes, noise, and produces short histograms. In addition, we present two efficient approaches able to select suitable features for each pixel/region to distinguish the foreground objects from the background. The key contributions of the thesis are as follows.

- **An eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) Descriptor**. The XCS-LBP descriptor is introduced in this thesis. It combines the strengths of the ordinary Local Binary Pattern (LBP) and the Center-Symmetric (CS) LBPs. Thus, the new variant XCS-LBP produces a shorter histogram than LBP, by its CS-construction. It is also tolerant to illumination changes as LBP and CS-LBP are whereas CS-LDP is not, and robust to noise as CS-LDP is whereas LBP and CS-LBP are not. Despite our descriptor have been proposed recently, it has been widely improved and used in different applications by some authors. For instance, Du and Qin (2016) [57] presented a uniform pattern version of our descriptor (called UXCS-LBP). The authors combined the histograms extracted by UXCS-LBP and CS-LDP. The experimental results show that this combination is robust under scenes ranging from dynamic background to changing illuminations. Nagananthini and Yogameena (2017) [142] used the XCS-LBP for crowd count application. Firstly, the authors extracted XCS-LBP features of the images under sudden illumination changes. Then, these features are trained using deep Convolutional Neural Network (CNN). The proposed approach display a warning message if the people count overcome a threshold by avoiding crowd disaster.

- **An Ensemble Pixel-based for Feature Selection in Background Subtraction**. We proposed an online weighted one-class random subspace ensemble for feature selection (OWOC-RS). The proposed method is designed to automatically select the best features for different pixels of the image, and the more relevant features are used for foreground segmentation. In addition, a mechanism to update these importances fea-

tures over time is presented.

- **An Ensemble Superpixel-based for Feature Selection in Background Subtraction**. We extended our OWOC-RS approach by proposing a novel methodology for selecting features based on wagging. Our proposal is able to select suitable features for each region to distinguish the foreground objects from the background. In addition, it uses superpixel approach that not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance. The experiments conducted on challenging videos have show that this approach is more efficient in terms of time and memory consumption than our previous approach.
- **An 3D Joint Color-Texture Descriptor for Dynamic Texture Recognition**. The last contribution of this thesis is the proposed 3-dimensional joint color-texture descriptor for dynamic texture analysis. We extended the spatial color-texture OCLBP descriptor to the spatio-temporal domain by combining it with the LBP-TOP one. By fusing color and dynamics textures, the derived OCLBP-TOP extracts more detailed information from the video sequence to be analyzed.

7.1 Limitations

The benefits of the contributions introduced in this thesis have been demonstrated in the several evaluative experiments. Nonetheless, there are limitations which could open opportunities for further investigations or new lines of thought.

- As the proposed XCS-LBP descriptor does not include temporal relationships between neighboring pixels, it is not very suitable to deal with dynamic scenes. However, the temporal domain can be used to discriminate one object from another by analyzing its temporal motion patterns, thereby playing a crucial role in moving object detection.
- Our proposed online weighted ensemble of one-class SVMs (Support Vector Machines) pixel-based for feature selection is designed to automatically select the best features for different regions of the image. The main drawback is that this method only reaches the highest accuracy when the number of features is huge. Furthermore, each base classifier learns a feature set instead of individual features. To overcome these limitations, in this thesis we extended our approach by proposing a novel methodology for selecting features based on wagging. In addition, we also adopted a superpixel-based approach instead of pixel-level approach. This does not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance. Both approaches proposed to select the best feature use a mechanism to update the relative importance of each feature, discarding insignificant features over time. This mechanism requires ground-truth data, but usually ground truth data is not available for BS in real environments.
- Not surprisingly, the proposed OCLBP-TOP needs much more time than the others local binary pattern based descriptors, because of both the TOP extension (as compared to OCLBP), and the six separate channels computation (as compared to LBP-TOP).

This is the price that must be paid for combining color information together with the texture, so that the classification performance of dynamic textures increase. In order to solve this problem, feature selection methods can be used for selecting the best channels before of the dynamic texture classification.

7.2 Future works

- **Developing local binary patterns features**. Local binary pattern features are important to describe different scenes in many computer vision applications. In this thesis, we proposed a robust local binary patterns descriptor for background subtraction called XCS-LBP as well as a second descriptor named OCLBP-TOP for dynamic texture recognition. A future work will be the extension of XCS-LBP to include temporal properties. We also intend to reduce the computation time of our OCLBP-TOP by proposing to use only the best channels instead of all the channels to recognize dynamic textures.
- **Feature selection in background subtraction**. In the BS field, the use of feature selection methods is less studied so far. Nevertheless, the feature selection can be used to improve the detection of foreground objects [149] in complex scenes thanks to their capability to select a subset of highly discriminant features removing irrelevant and redundant ones, *e.g.* in [149]. Therefore, the feature selection approaches provide opportunity for future research. A possible future work is hte extension of our proposed approaches in this thesis by developing a mechanism to suitably update the importance of each feature discarding insignificantly features over time without ground-truth data.

Appendix A

Notations and Symbols

P	number of neighboring pixels
R	radius of neighboring pixels; radius of a hypersphere
g_i	gray value of a pixel in a neighborhood
g_c	gray value of the center of a neighborhood
x	universal variable used with many functions
x_c	x coordinate of the center of a neighborhood
y_c	y coordinate of the center of a neighborhood
$T, t_1, t_2, time, \eta$	a user-defined threshold; number of iterations
X	training set
ξ	slack variables
a	center of a hypersphere
C	user parameter that controls the trade-off of a hypersphere
w	weight samples
Z_0	previous training set
Z_1	newly added training set
θ	distribution of a previous training set
p	original features
N	number images/samples
M	user-defined number of base classifier
Ψ	set of diverse base classifier/background models
δ	weight distribution
$\lambda^{correct}$	number of times a pixel was correctly classified
λ^{wrong}	number of times a pixel was incorrectly classified
γ	learning rate parameter
L	number of best base classifiers
P_a	accuracy of a base classifier
d	distance metric
s_1	normalization factor
s_2	scale parameter

F	support function
$H(x)$	final/strong classifier
$m(x)$	margin samples
$v_{c(1)}$	number the first voted class
$v_{c(2)}$	number the second voted class
TP	true positive
FP	false positive
FN	false negative
FN	false negative
k	number of base classifiers
ω	class
E	normalization factor
r	rank of a class
Υ	stopping criterion
φ_{best}	variable of evaluation
ϑ	independent measure
S	subset of features
A	learning algorithm
ν, β	importance feature/classifier
$b1$	number of negative examples
$b2$	number of positive examples

Appendix B

Local Binary Patterns Descriptors

The standardized formulas of the main LBPs are presented in the Tables below.

Table B.1: Local Binary Patterns and its variants

Ordinary Local Binary Pattern (LBP) [85]	$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p$	$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$ The T is a threshold value.
Modified LBP [84]	$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c + a)\, 2^p$	$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
Uniform Local Binary Patterns (ULBP) [231]	$LBP^{riu}_{P,R}(x_c, y_c) = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p & \text{if } U_{P,R} \leq 2 \\ P+1 & \text{otherwise.} \end{cases}$ where $U_{P,R} = \sum_{p=0}^{P-2} (s(g_p - g_c) \bigoplus s(g_{p+1} - g_c)) + s(g_{P-1} - g_c) \bigoplus s(g_0 - g_c)$	$s(x) = \begin{cases} 1 & \text{if } x \geq T \\ 0 & \text{if } x < T \end{cases}$ A relatively small value for T should be used, for example, $2 \leq T \leq 5$.

Opponent Color Local Binary Patterns (OCLBP) [116, 133]	$$LBP_{P_{RR},R_{RR}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{R,p}-g_{R,c})$$ $$LBP_{P_{GG},R_{GG}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{G,p}-g_{G,c})$$ $$LBP_{P_{BB},R_{BB}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{B,p}-g_{B,c})$$ $$LBP_{P_{RG},R_{RG}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{R,p}-g_{G,c})$$ $$LBP_{P_{RB},R_{RB}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{R,p}-g_{B,c})$$ $$LBP_{P_{GB},R_{GB}}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_{G,p}-g_{B,c})$$ $$OC-LBP_{P_{oc},R_{oc}}(x_c,y_c)=LBP_{P_{RR},R_{RR}}(x_c,y_c)\bigoplus$$ $$LBP_{P_{GG},R_{GG}}(x_c,y_c)\bigoplus LBP_{P_{BB},R_{BB}}(x_c,y_c)\bigoplus$$ $$LBP_{P_{RG},R_{RG}}(x_c,y_c)\bigoplus LBP_{P_{RB},R_{RB}}(x_c,y_c)\bigoplus$$ $$LBP_{P_{GB},R_{GB}}(x_c,y_c)$$ where $g_{R,c},g_{G,c},g_{B,c}$ correspond to the opponent color values of the center pixel, respectively; $g_{R,p},g_{G,p},g_{B,p}$ correspond to the opponent color values of the neighborhoods on the circles of radius R_{oc} in the opponent color channels and $\bigoplus$ denotes concatenation descriptor.	$$s(x)=\begin{cases}1 & \text{if } x\geq 0\\ 0 & \text{otherwise.}\end{cases}$$

εLBP [206]	$$\varepsilon LBP_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1}s\left(\frac{\hat{g_p}-\check{g_p}}{g_c}-\varepsilon\right)2^p$$ where $\hat{g_p}$ and $\check{g_p}$ denote the gray value of the clockwise and counter-clockwise neighborhood of g_p. The ε is a noise parameter.	$$s(x)=\begin{cases}1 & \text{if } x\geq 0\\ 0 & \text{otherwise.}\end{cases}$$
Adaptive εLBP [207]	$$\varepsilon LBP_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1}s\left(\frac{\hat{g_p}-\check{g_p}}{g_c}-\varepsilon_c^p\right)2^p$$ when $\lvert\mu^B\rvert>\alpha\cdot\sigma_{p(\mu^F)}$, the threshold ε_c^p is calculated: $$\varepsilon_c=\begin{cases}\max\left(\eta,\frac{\mu^B-\gamma\cdot\sigma^B}{g_c}\right) & \text{if } \mu^B>\alpha\cdot\sigma_{p(\mu^F)}\\ \min\left(-\eta,\frac{\mu^B-\gamma\cdot\sigma^B}{g_c}\right) & \text{if } \mu^B<-\alpha\cdot\sigma_{p(\mu^F)}\end{cases}$$ when $\lvert\mu^B\rvert\leq\alpha\cdot\sigma_{p(\mu^F)}$, the threshold ε_c^p is calculated: $$\varepsilon_c=\begin{cases}-\eta & \text{if } \mu^B\leq\alpha\cdot\sigma_{p(\mu^F)}\ \&\ \mu^B\geq 0\\ \eta & \text{if } \mu^B\geq-\alpha\cdot\sigma_{p(\mu^F)}\ \&\ \mu^B<0\end{cases}$$ where μ^B is the first obtained from the start N frames, the $\sigma_{p(\mu^F)}$ is the mean distribution of the N other frames. The γ, α and η are the constants, g_c corresponds to the gray value of the center pixel, and the $\max(\cdot)$ and $\min(\cdot)$ operators are used to restrict the threshold.	$$s(x)=\begin{cases}1 & \text{if } x\geq 0\\ 0 & \text{otherwise.}\end{cases}$$

Local Color Pattern (LCP) [47]	$$LBP_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_p-g_c+a)\,2^p$$ The mapping from LBP to uniform LBP is as follows: $$LBP_{P,R}^{riu}(x_c,y_c)=\begin{cases}\sum_{p=0}^{P-1}B(LBP_{P,R}) & \text{if } U_{P,R}\leq 2\\ P+1 & \text{otherwise.}\end{cases}$$ Finally, the uniform LBP histogram is obtained as follows: $$H_{LBP,i}=\sum_{(x_c,y_c)\in R}I\{LBP(x_c,y_c)=i\}\ \ |i=0,1,...,2^{P-1}$$ Finally, local color pattern (LCP) histogram is formed by concatenating the quantized hue, luminance, and saturation histograms, summed over the structuring element as follows: $$H_{LCP}=[H_{hue}\ H_{lum}\ H_{sat}]$$	$$I(A)=\begin{cases}1 & \text{if } A \text{ is true,}\\ 0 & \text{otherwise.}\end{cases}$$

Local Binary Similarity Patterns (LBSP) [22]	$$LBSP_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_p-g_c)2^p$$ where g_c corresponds to the central pixel (whether from the current image for intra-LBSP or from a reference frame for inter-LBSP), and g_p corresponds to the neighbor pixel (always in the current image).	$$s(x)=\begin{cases}1 & \text{if } \lvert x\rvert \leq T\\ 0 & \text{otherwise.}\end{cases}$$ The T is a similarity threshold.
Local SVD Binary Pattern (LSBP) [76]	$$LBSP_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1}s(g_p,g_c)2^p$$ where g_c and g_p are obtained as follow: $$g(x_c,y_c)=\sum_{q=2}^{M}\tilde{\lambda}_q, \quad and \quad \tilde{\lambda}_q,=\lambda_q/\lambda_1$$ where λ_q indicates the *jth* singular value.	$$s(x)=\begin{cases}0 & \text{if } \lvert x-y\rvert \leq T\\ 1 & \text{otherwise.}\end{cases}$$

Table B.2: Center-Symmetric Local Binary Patterns and its variants

Center-Symmetric Local Binary Patterns (CS-LBP) [86]	$CS-LBP_{P,R}(x_c,y_c)=\sum_{p=0}^{(P/2)-1} s(g_p-g_{p+(P/2)})\,2^p$ where g_i and $g_{i+(P/2)}$ are the gray values of center-symmetric pairs of pixels.	$s(x)=\begin{cases}1 & \text{if } x>T\\ 0 & \text{otherwise}\end{cases}$
Center-Symmetric Local Derivative Pattern (CS-LDP) [225]	$CS-LDP_{P,R}(x_c,y_c)=\sum_{p=0}^{(P/2)-1} s\left[\left[(g_p-g_c)\left(g_c-g_{c+(P/2)}\right)\right]\right]2^p$	$s(x,y)=\begin{cases}1 & \text{if } x\cdot y\leq 0\\ 0 & \text{otherwise.}\end{cases}$

eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) [176]	$$XCS-LBP_{P,R}(x_c,y_c)=\sum_{p=0}^{(P/2)-1} s\left(g_1(p,c)+g_2(p,c)\right)2^p$$ where g_1 and g_2 are defined by: $$g_1(p,c)=\left(g_p-g_{p+(P/2)}\right)+g_c$$ $$g_2(p,c)=\left(g_p-g_c\right)\left(g_{p+(P/2)}-g_c\right)$$	$$s(x,y)=\begin{cases}1 & \text{if } (x+y)\geq 0\\ 0 & \text{otherwise.}\end{cases}$$
BackGround Local Binary Patterns (BG-LBP) [51]	$$BGLBP_{P,R}(x_c,y_c)=\begin{cases}\sum_{p=0}^{(P/2)-1} s(g_p,m,g_{p+(P/2)})\,2^p & U\left(LB^p_{\frac{P}{2},P,R}\right)\leq 2\\ P_2^P & \text{otherwise.}\end{cases}$$ $$m=\frac{1}{P}\left(g_c+\sum_{p=0}^{P-1} g_p\right)$$	$$s(x)=\begin{cases}1 & if(((g_p\geq m\geq g_{p+(P/2)})\,||\,(g_p<m<g_{p+(P/2)}))\&\&\\ & ((abs(g_p-m)+abs(g_{p+(P/2)}-m))\geq T))\\ 0 & otherwise.\end{cases}$$

Table B.3: Local Ternary Pattern and its variants

Local Ternary Pattern (LTP) [191]	$LTP^{\tau}_{P,R}(x_c,y_c)=\sum_{p=0}^{P-1} s_\tau(g_p-g_c)2^p$ where τ is scale factor indicating the comparing range.	$s_\tau(x)=\begin{cases}1 & \text{if } x\geq\tau,\\ 0 & \text{if } -\tau<x<\tau,\\ -1 & \text{if } x<-\tau.\end{cases}$
Scale Invariant Local Ternary Pattern (SILTP) [120]	$SILTP^{\tau}_{P,R}(x_c,y_c)=\bigoplus_{p=0}^{P-1} s_\tau(g_c,g_p)$ where $\bigoplus$ denotes concatenation operator of binary strings.	$s_\tau(x,y)=\begin{cases}01 & \text{if } x>(1+\tau)y,\\ 10 & \text{if } x<(1-\tau)y,\\ 00 & \text{otherwise.}\end{cases}$
Scale Invariant Local States (SILS) [232]	$SILS^{\tau}_{P,R}(x_c,y_c)=s_\tau(g_c,g_p)$.	$s_\tau(x,y)=\begin{cases}01 & \text{if } x>(1+\tau)y,\\ 10 & \text{if } x<(1-\tau)y,\\ 00 & \text{otherwise.}\end{cases}$

Scene Adaptive Local Binary Pattern (SALBP) [229]	$$SALBP^{\tau}_{P,R}(x_c,y_c)=\sum_{p=1}^{6}s_{\tau}\left(diff,CB\right)2^{p}$$ where $diff$ is defined as subtraction of an gray value of a center pixel from that of $p-$neighborhood pixel, $CB(x_c,y_c)=\{c_1\|1\leq l\leq L(x_c,y_c)\}$ implies the corresponding codebook composed of $L(x_c,y_c)$number of codewords.	$$s_{\tau}(x,y)=\begin{cases}1 & \text{if } x \text{ is matched to } CB\\ 0 & \text{otherwise.}\end{cases}$$
Multi-Channel Scale Invariant Local Ternary Pattern (MC-SILTP) [132]	$$SILTP^{\tau}_{P,R}(x_c,y_c)=\bigoplus_{p=0}^{P-1}s_{\tau}\left(g_{R,c},g_{B,p}\right)$$ $$SILTP^{\tau}_{P_G,R_G}\left(x_c,y_c\right)=\bigoplus_{p=0}^{P-1}s_{\tau}\left(g_{G,c},g_{R,p}\right)$$ $$SILTP^{\tau}_{P_B,R_B}\left(x_c,y_c\right)=\bigoplus_{p=0}^{P-1}s_{\tau}\left(g_{B,c},g_{G,p}\right)$$ $$MC-SILTP^{\tau}_{P_{RGB},R_{RGB}}\left(x_c,y_c\right)=SILTP^{\tau}_{P_R,R}\left(x_c,y_c\right)\bigoplus SILTP^{\tau}_{P_G,R}\left(x_c,y_c\right)\bigoplus SILTP^{\tau}_{P_B,R}\left(x_c,y_c\right)$$ where $g_{R,c},g_{G,c},g_{B,c}$ correspond to the RGB values of the center pixel, respectively; $g_{R,p},g_{G,p}$, $g_{B,p}$ to the RGB values of the neighborhoods on the circles of radius R_{RGB} in the RGB channels and $\bigoplus$ indicates concatenation operator of binary strings.	$$s_{\tau}(x,y)=\begin{cases}01 & \text{if } x>(1+\tau)y,\\ 10 & \text{if } x<(1-\tau)y,\\ 00 & \text{otherwise.}\end{cases}$$

Table B.4: Spatial-Temporal Pattern and its variants

Spatio-temporal Local Binary Patterns (STLBP) [174]	$$LBP^{t}_{P,R}(x_{t,c}, y_{t,c}) = \sum_{p=0}^{P-1} s(g_{t,p} - g_{t,c})\, 2^{p},$$ $$LBP^{t-1}_{P,R}(x_{t,c}, y_{t,c}) = \sum_{p=0}^{P-1} s(g_{t-1,p} - g_{t-1,c})\, 2^{p},$$ $$H_{t,i} = \sum_{(x_c,y_c)\in R} I\left\{LBP^{t}_{P,R}(x_{t,c}, y_{t,c}) = i\right\} \; | i = 0, 1, ..., 2^{P-1}$$ $$H_{t-1,i} = \sum_{(x_c,y_c)\in R} I\left\{LBP^{t-1}_{P,R}(x_{t,c}, y_{t,c}) = i\right\} \; | i = 0, 1, ..., 2^{P-1}$$ where t corresponds to the time, $H_{t,i}$ and $H_{t-1,i}$ are the histogram values at i^{th} bin of H_t and H_{t-1}, respectively. $$STLBP_t = \omega H_{t-1,i} + (1-\omega) H_{t,i} \; | i = 0, 1, ..., 2^{P-1}$$	$$I(A) = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

Spatial-Temporal Local Binary Pattern (STLBP) [175]	$STLBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)\,2^p + \sum_{p=0}^{P-1} u(g_p - g_z)\,2^{P+p}$ where g_z corresponds to the predictive values of the P.	$u(x) = \begin{cases} 1 & \lvert x\rvert \geq T \\ 0 & \text{otherwise.} \end{cases}$
Stereo Local Binary Pattern based on Appearance and Motion (SLBP-AM) [229]	$LBP_j = \sum_{p=0}^{P-1} s(g_p - g_c)\,2^p,$ where j denotes the corresponding plane: 0 for the XY plane, 1 for the XT plane and 2 for the YT plane. $H_{i,j} = \sum_{(x_c,y_c)\in R} I\{LBP_j(g_p - g_c) = p\} \mid i = 0,1,...,2^{P-1}$ where $H_{i,j}$ is the histogram value. $SLBP - AM = \sum_{j=0,1,2} \omega_j H_{i,j} \mid i = 0,1,...,2^{P-1}$	$I(A) = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$

Table B.5: Hybrid Local Binary Pattern and its variants

Spatial Extended Center-Symmetric Local Binary Pattern (SCS-LBP) [226]	$SCS-LBP_{P,R}(x_c,y_c,t) = \sum_{p=0}^{(P/2)-1} s\left(g_{(p,t)} - g_{(p+(P/2),t)}\right) 2^p + f\left(g_{(x_c,y_c,t)} - \bar{\mu}_{(x_c,y_c,t-1)}\right) 2^{P/2}$ where $\bar{\mu}_{(x_c,y_c,t-1)}$ and $\bar{\sigma}_{(x_c,y_c,t-1)}$ are estimated mean value and standard deviation respectively corresponding to pixel $g_(x_c,y_c)$.	$f(t) = \begin{cases} 0 & \text{if } \lvert g_{(x_c,y_c,t-1)} - \bar{\mu}_{(x_c,y_c,t-1)} \rvert < 2.5\bar{\sigma}_{(x_c,y_c,t-1)}, \\ 1 & \text{otherwise.} \end{cases}$
Center Symmetric Spatio-temporal Local Ternary Pattern (CS-STLTP) [223]	$CS-STLTP^j(x_c,y_c,z_c) = \biguplus_{p=0}^{(P/2)-1} s_\tau\left(g_{(p)}, g_{(p+(P/2))}\right)$ where sign $\biguplus$ indicates stretching elements into a vector and j denotes the planes: XY, XT, and YT.	$s_\tau(x,y) = \begin{cases} 1 & \text{if } x > (1+\tau)y, \\ 0 & \text{if } x < (1-\tau)y, \\ -1 & \text{otherwise.} \end{cases}$

Center Symmetric Scale Invariant Local Ternary Patterns (CS-SILTP) [218]	$CS-SILTP^{\tau}_{P,R}(x_c,y_c) = \bigoplus_{r=-R}^{R} \bigoplus_{p=0}^{P/2-1} s_\tau\left(g_p^{t+r}, g_{p+P/2}^{t-r}\right)$ where g^t denotes the scene image captured at the time instant t, g_p^{t+r} and $g_{p+P/2}^{t-r}$ are the center-symmetric pixel locations lying on the cubic surface.	$s_\tau(x,y) = \begin{cases} 01 & \text{if } x > (1+\tau)y, \\ 10 & \text{if } x < (1-\tau)y, \\ 00 & \text{otherwise.} \end{cases}$
Spatiotemporal Scale Invariant Ternary Pattern (ST-SILTP) [100]	$ST-SILTP^{\tau}_{P,R}(x_c,y_c) = \bigoplus_{p=0}^{P-1} s_\tau(g_c,g_z)$ where g_z denote the gray values of neighboring pixels in the spatiotemporal neighborhood.	$s_\tau(x,y) = \begin{cases} 01 & \text{if } x > (1+\tau)y, \\ 10 & \text{if } x < (1-\tau)y, \\ 00 & \text{otherwise.} \end{cases}$

Appendix C

List of Publications

This dissertation has led to the following communications:

Journal Papers

- Bouwmans, T. and Silva, C. and Marghes, C. and Zitouni, S. and Bhaskar, H. and Frélicot, C. "On the Role and the Importance of Features for Background Modeling and Foreground Detection". Computer Science Review, 2016 (submitted).

- Silva, C. and Gonzàlez, J. and Bouwmans, T. and Frélicot, C. "3D joint color-texture descriptor for dynamic texture recognition". IET Computer Vision, 2017 (in revision).

- Silva, C. and Bouwmans, T. and Frélicot, C. "Superpixel-based incremental wagging one-class ensemble for feature selection in foreground/background separation". Pattern Recognition Letters (PRL), 2017 (submitted).

Book chapters

- Silva, C. and Bouwmans, T. and Frélicot, C. "Features and Strategies Issues". Chapter on the handbook "Background Subtraction for Moving Object Detection: Theory and Practices", 2017 (in progress)

Conferences

- Silva, C. and Bouwmans, T. and Frélicot, C. "An eXtened Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos". In the Proceedings of the 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Berlin, Germany (oral presentation), March, 2015.

- Silva, C. and Bouwmans, T. and Frélicot, C. "Online Weighted One-Class Ensemble for Feature Selection in Background/Foreground Separation". In the Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico (oral presentation), December, 2016.

Websites

- Behance.net project: `https://www.behance.net/carolinepacheco`
- LBPLibrary: `https://github.com/carolinepacheco/lbplibrary`
- Caroline Silva's homepage: `http://lolynepacheco.wixsite.com/carolinesilva`

Social networks

- ResearchGate: `http://https://www.researchgate.net/profile/Caroline_Silva6`
- Linkedin: `https://www.linkedin.com/in/carolinepes`
- Academia: `https://univ-larochelle.academia.edu/CarolineSilva`

Bibliography

[1] Torch. www.torch.ch. Accessed: 2015-03-11. 5

[2] T. Aach, L. Dumbgen, R. Mester, and D. Toth. Bayesian illumination-invariant motion detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 640–643, 2001. 4

[3] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, pages 392–398, 2010. 29

[4] B. Afsari, R. Chaudhry, A. Ravichandran, and R. Vidal. Group action induced distances for averaging and clustering linear dynamical systems with applications to the analysis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2208–2215, 2012. 84

[5] T. Almaev and M. Valstar. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 356–361, 2013. 84, 90, 92, 94

[6] B. Antic, J. Castaneda, D. Culibrk, A. Pizurica, and V. Crnojevic. *Robust Detection and Tracking of Moving Objects in Traffic Video Surveillance*. Advanced Concepts for Intelligent Vision Systems (ACIVS), 2009. 19, 20

[7] F. El Baf, T. Bouwmans, and B. Vachon. A fuzzy approach for background subtraction. In *IEEE International Conference on Image Processing (ICIP)*, pages 2648–2651, 2008. 4

[8] F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. *IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1729–1736, 2008. 15, 16

[9] M. Balcilar, F. Karabiber, and A. Sonmez. Performance analysis of Lab2000HL color space for background subtraction. *EEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2013. 15, 16

[10] D. Baltieri, R. Cucchiara, and R. Vezzani. Fast background initialization with recursive Hadamard transform. *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2010. 19

[11] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patch match for large displacementoptical flow. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 25

[12] O. Barnich and M.V. Droogenbroeck. Vibe: a powerful random technique to estimate the background in video sequences. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 945–948, 2009. 10, 24, 35

[13] P. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, pages 525–536, 2006. 32

[14] F. Bastien, R. Lamblin, P.and Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2012. 5

[15] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, pages 105–139, 1999. 73

[16] Y. Benezeth, D. Sidibe, and J. B. Thomas. Background subtraction with multispectral video sequences. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. ix, xii, 5, 62, 63, 64, 65, 66, 67, 73, 76, 78, 79, 81

[17] H. Bhaskar, L. Mihaylova, and A. Achim. Video foreground detection based on symmetric alpha-stable mixture models. *IEEE Transactions on Circuits, Systems and Video Technology*, 2010. 10

[18] H. Bhaskar, L. Mihaylova, and S. Maskell. Background modeling using adaptive cluster density estimation for automatic human detection. In *LNCS from the 3rd German Workshop on Sensor Data Fusion: Trends, Solutions, Applications*, pages 130–134, 2007. 10

[19] H. Bhaskar, L. Mihaylova, and S. Maskell. *Automatic object detection based on adaptive background subtraction using symmetric alpha stable distribution*, pages 197–203. The Institution of Engineering and Technology Conference on Target Tracking and Data Fusion, 2008. 10

[20] S. Bianco, G. Ciocca, and R. Schettini. How far can you get by combining change detection algorithms? *Computing Research Repository (CoRR)*, 2015. 3, 31

[21] M. Bicego and M. Figueiredo. Soft clustering using weighted one-class support vector machines. *Pattern Recognition (PR)*, pages 27–32, 2009. 58

[22] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier. Change detection in feature space using local binary similarity patterns. In *Conference on Computer and Robot Vision (CRV)*, pages 106–112, 2013. 20, 44, 108

[23] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M Benítez, and F. Herrera. A review of microarray datasets and applied feature selection methods. *Information Sciences*, pages 111–135, 2014. 6, 27

[24] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. In *Computer Science Review*, pages 31–66, 2014. 5, 25, 44, 45

[25] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant. Background modeling and foreground detection for video surveillance. In *Chapman & Hall/CRC*, 2014. 1

[26] T. Bouwmans, C. Silva, C. Marghes, S. Zitouni, H. Bhaskar, and C. Frélicot. On the role and the importance of features for background modeling and foreground detection. In *Computer Science Review*, 2016. 9

[27] M. Braham, A. Lejeune, and M. Van Droogenbroeck. A physically motivated pixel-based model for background subtraction in 3d images. In *International Conference on 3D Imaging (IC3D)*, pages 1–8, 2014. 16

[28] M. Braham and M. Van Droogenbroeck. A generic feature selection method for background subtraction using global foreground models. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 717–728, 2015. 27, 35, 36

[29] M. Braham and M. Van Droogenbroeck. Deep background subtraction with scene-specific convolutional neural networks. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, 2016. 5

[30] S. Brahnam, L. C. Jain, L. Nanni, and A. Lumini. *Local Binary Patterns: New Variants and Applications*. Studies in Computational Intelligence, 2014. 84

[31] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 993–1008, 2003. 23

[32] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1937–1944, 2011. 25

[33] A. Bugeau and P. Pérez. Detection and segmentation of moving objects in highly dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 18, 20

[34] W. Buntine. *A theory of learning classification rules*. PhD thesis, 1992. 35

[35] M. Camplani, C. Blanco, L. Salgado, F. Jaureguizar, and N. Garcia. Multi-sensor background subtraction by fusing multiple region-based probabilistic classifiers. *Pattern Recognition Letters (PRL)*, 2013. 16, 24

[36] M. Camplani and L. Salgado. Background foreground segmentation with RGB-D kinect data: an efficient combination of classifiers. *Journal on Visual Communication and Image Representation (JVCIR)*, 2013. ix, xii, 24, 76, 78, 80

[37] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1986. 15

[38] M. Chacon-Murguia and S. Gonzalez-Duarte. An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems. *IEEE Transactions on. Industrial Electronics (TIE)*, pages 3286–3298, 2012. 4

[39] A. Chan, V. Mahadevan, and N. Vasconcelos. Generalized stauffer–grimson background subtraction for dynamic scenes. *Machine Vision and Applications (MVA)*, pages 751–766, 2011. 4

[40] C. Chan, B. Goswami, J. Kittler, and W. Christmas. Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication. *IEEE Transactions on Information Forensics and Security*, pages 602–612, 2012. 84

[41] V. Charles. *Computational Frameworks for the Fast Fourier Transform*. Society for Industrial and Applied Mathematics, 1992. 19

[42] R. Chaudhry, G. Hager, and R. Vidal. Dynamic template tracking and recognition. *International Journal of Computer Vision (IJCV)*, pages 19–48, 2013. 84

[43] J. Chen, G. Zhao, and M. Pietikäinen. Unsupervised dynamic texture segmentation using local spatiotemporal descriptors. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 84

[44] J. Chen, G. Zhao, and M. Pietikainen. An improved local descriptor and threshold learning for unsupervised dynamic texture segmentation. In *International Conference on Computer Vision Workshops*, pages 460–467, 2009. 84

[45] M. Chen, Q. Yang, Q. Li, G. Wang, and M. Yang. Spatiotemporal background subtraction using minimum spanning tree and optical flow. *European Conference on Computer Vision (ECCV)*, 2014. 16, 25

[46] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016. 32

[47] T. Chua, Y. Wang, and K. Leman. Adaptive texture-color based background subtraction for video surveillance. In *IEEE International Conference on Image Processing (ICIP)*, pages 49–52, 2012. 6, 20, 21, 107

[48] M. Cord and P. Cunningham. *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval (Cognitive Technologies)*. Springer-Verlag, 2008. 28

[49] M. Cristani, M. Bicego, and V. Murino. Multi-level background initialization using hidden markov models. In *ACM SIGMM International Workshop on Video Surveillance*, pages 11–20, 2003. 4

[50] V. Crnojevic, B. Antic, and D. Culibrk. Optimal wavelet differencing method for robust motion detection. In *IEEE International Conference on Image Processing (ICIP)*, pages 645–648, 2009. 19, 20

[51] S. Davarpanah, F. Khalid, Abdullah L. N., and M. Golchin. A texture descriptor: Background local binary pattern (BGLBP). *Multimedia Tools and Applications (MTA)*, pages 6549–6568, 2016. 20, 110

[52] K. Derpanis, M. Lecce, K. Daniilidis, and R. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1306–1313, 2012. 84, 88

[53] K. Derpanis and R. Wildes. Dynamic texture recognition based on distributions of spacetime oriented structure. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 191–198, 2010. 84

[54] P. Devyver and J. Kittler. *Pattern recognition: a statistical approach*. Prentice-Hall, 1982. 89

[55] B. Dey and M. K. Kundu. Robust background subtraction for network surveillance in h.264 streaming video. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, pages 1695–1703, 2013. 4

[56] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. In *International Journal of Computer Vision (IJCV)*, pages 91–109, 2003. 83, 84

[57] X. Du and G. Qin. Foreground and detection in surveillance videos via a hybrid local texture based method. In *International Journal on Smart Sensing and Intelligent Systems*, 2016. 97

[58] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767, 2000. 3, 10, 24, 36

[59] C. Eveland, K. Konolige, and R. Bolles. Background modeling for segmentation of video-rate stereo sequences. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 266–271, 1998. 16, 23

[60] C. Feichtenhofer, A. Pinz, and R. Wildes. Spacetime forests with complementary features for dynamic scene recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2013. 84

[61] C. Feichtenhofer, A. Pinz, and R. Wildes. Bags of spacetime energies for dynamic scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2681–2688, 2014. 84

[62] E. Fernandez-Sanchez, J. Diaz, and E. Ros. Background subtraction based on color and depth using active sensors. *Sensors*, pages 8895–8915, 2013. 16, 24

[63] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning (ICML)*, pages 148–156, 1996. 32

[64] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, pages 119–139, 1997. 36

[65] D. Gabor. Theory of communication. *Institution of Electrical Engineering*, pages 429–457, 1946. 19

[66] J. Gallego and M. Pardas. Region based foreground segmentation combining color and depth sensors via logarithmic opinion pool decisions. *Journal of Visual Communication and Image Representation (JVCIR)*, 2013. 16, 24

[67] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *Proceedings of the European Conference on Computer Vision: Part II*, pages 223–236, 2010. 88

[68] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 459–464, 1999. 16

[69] N. Goyette, P. Jodoin, P. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. *IEEE Workshop on Change Detection (CDW) at CVPR*, 2012. 17, 25

[70] H. Grabner and H. Bischof. On-line boosting and vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 10, 36, 57, 73

[71] H. Grabner, C. Leistner, and H. Bischof. Time dependent on-line boosting for robust background modeling. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2008. 36, 57, 73

[72] H. Grabner, P. Roth, M. Grabner, and H. Bischof. Autonomous learning a robust background model for change detection. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2006. 36, 57, 73

[73] K. Greff, A. Brandao, S. Krauss, D. Stricker, and E. Clua. A comparison between background subtraction algorithms using a consumer depth camera. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2012. 16, 24

[74] J.-M. Guo, C.-H. Hsia, Y.-F. Liu, M.-H. Shih, C.-H. Chang, and J.-Y. Wu. Fast background subtraction based on a multilayer codebook model for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, pages 1809–1821, 2013. 4

[75] L. Guo and S. Boukir. Fast data selection for svm training using ensemble margin. *Pattern Recognition Letters (PRL)*, 2015. 62, 76

[76] L. Guo, D. Xu, and Z. Qiang. Background subtraction using local svd binary pattern. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016. 20, 108

[77] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. M. Lyons, and A. K. Jain. A background model initialization algorithm for video surveillance. In *International Conference on Computer Vision (ICCV)*, pages 733–740, 2001. 4

[78] B. Han and L. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1017–1023, 2012. 5

[79] G. Han, J. Wang, and X. Cai. Background subtraction based on three-dimensional discrete wavelet transform. *Sensors (Basel, Switzerland)*, 2016. 19, 20

[80] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 993–1001, 1990. 33

[81] M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. *European Conference on Computer Vision (ECCV)*, 2002. 16, 23

[82] M. Harville, G. Gordon, and J. Woodfill. Foreground segmentation using adaptive mixture models in color and depth. *International Workshop on Detection and Recognition of Events in Video*, 2001. 14, 16, 23

[83] S. He, J. Soraghan, B. O'Reilly, and D. Xing. Quantitative analysis of facial paralysis using local binary patterns in biomedical videos. *IEEE Transactions on Biomedical Engineering*, pages 1864–1870, 2009. 84

[84] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 657–662, 2006. 17, 19, 20, 44, 104

[85] M. Heikkilä, M. Pietikäinen, and J. Heikkilä. A texture-based method for detecting moving objects. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2004. 19, 20, 104

[86] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition (PR)*, pages 425–436, 2009. 20, 21, 41, 43, 44, 109

[87] T. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 832–844, 1998. 31

[88] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 38–43, 2012. 4, 10

[89] M. Hossain, M. Dewan, and O. Chae. Moving object detection for real time video surveillance: An edge based approach. *IEICE Transactions on Communications*, 2007. 17

[90] C.-H. Hsia and J.-M. Guo. Efficient modified directional lifting-based discrete wavelet transform for moving object detection. *Signal Processing*, pages 138–152, 2014. 19, 20

[91] L. Hu, L. Duan, X. Zhang, and J. Yang. Moving object detection based on the fusion of color and depth information. *Journal of Electronics and Information Technology*, pages 2047–2051, 2014. 16, 24

[92] S. Huang, L. Fu, and P. Hsiao. Region-level motion-based foreground detection with shadow removal using mrfs. *Asian Conference on Computer Vision (ACCV)*, 2006. 16, 25

[93] S. Huang, L. Fu, and P. Hsiao. Region-level motion-based foreground segmentation under a bayesian network. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, pages 522–532, 2009. 16, 25

[94] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. *International Journal of Computer Vision (IJCV)*, pages 199–207, 2000. 16, 23

[95] S. Jabri, Z. Duric, and H. Wechsler. Detection and location of people in video images using adaptive fusion of color and edge information. In *International Conference on Pattern Recognition (ICPR)*, pages 627–630, 2000. 12, 16, 17

[96] A. Jain and G. Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Transactions on Image Processing*, pages 124–128, 1998. 85

[97] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. *IEEE Workshop on Motion and Video Computing (WMVC)*, 2002. 6

[98] S. Javed, A. Sobral, T. Bouwmans, and S. K. Jung. OR-PCA with dynamic feature selection for robust background subtraction. In *ACM Symposium on Applied Computing*, pages 86–91, 2015. 35, 36

[99] H. Ji, X. Yang, H. Ling, and Y. Xu. Wavelet domain multifractal analysis for static and dynamic texture vlassification. *IEEE Transactions on Image Processing*, pages 286–299, 2013. 84

[100] Z. Ji and Z. Wang. Detect foreground objects via adaptive fusing model in a hybrid feature space. In *Pattern Recognition (PR)*, pages 2952–2961, 2014. 1, 3, 20, 116

[101] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Arxiv preprint hep-th*, 2014. 5

[102] P.-M. Jodoin, S. Piérard, Y. Wang, and M. Van Droogenbroeck. *Overview and Benchmarking of Motion Detection Methods*, pages 24–1–24–26. Chapman and Hall/CRC, 2014. 32

[103] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 885–894, 2008. 84

[104] C. Kim and J. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002. 15, 16

[105] J. Kim, A. Ramirez-Rivera, G. Song, B. Ryu, and O. Chae. Edge-segment-based background modeling: Non-parametric online background update. *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2013. 16, 17

[106] J. Kim, A. Rivera, B. Ryu, and O. Chae. Simultaneous foreground detection and classification with hybrid features. *International Conference on Computer Vision (ICCV)*, 2015. 16, 17

[107] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society (AMS), 1980. 18, 25

[108] J. Kittler. On the accuracy of the Sobel edge detector. *Image and Vision Computing*, pages 37–42, 1983. 15

[109] B. Klare and S. Sarkar. Background subtraction in varying illuminations using an ensemble based on an enlarged feature set. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 66–73, 2009. 36, 57, 73

[110] E. Kong and T Dietterich. Error-correcting output coding corrects bias and variance. In *International Conference on Machine Learning (ICML)*, pages 313–321, 1995. 33

[111] B. Krawczyk and M. Woźniak. Wagging for combining weighted one-class support vector machines. In *International Conference on Computational Science (ICCS)*, pages 1565–1573, 2015. 60, 74

[112] L. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms, Second Edition*. Wiley-Interscience, 2014. 34

[113] I. Laptev, M. Marszaaek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 84, 91, 92, 94

[114] Y. Le Cun, J. Denker, and S. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pages 598–605, 1990. 29

[115] D.-S. Lee. Improved adaptive mixture learning for robust video background modeling. In *Machine Vision for Application (MVA)*, pages 443–446, 2002. 4

[116] Y. Lee, J. Jiyoung, and I.-S. Kweon. Hierarchical on-line boosting based background subtraction. In *Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–5, 2011. 44, 105

[117] J. Leens, O. Barnich, S. Pierard, M. Droogenbroeck, and J. Wagner. Combining color, depth, and motion for video segmentation. *Computer Vision Systems*, pages 104–113, 2009. 16, 24

[118] L. Li, W. Huang, Gu I., and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, pages 1459–1472, 2004. 16, 35, 36

[119] Z. Liang, X. Liu, H. Liu, and W. Chen. A refinement framework for background subtraction based on color and depth data. *IEEE International Conference on Image Processing (ICIP)*, 2016. 16, 24

[120] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1301–1306, 2010. 20, 22, 44, 111

[121] S. Lim, A. Mittal, L. Davis, and N. Paragios. Fast illumination-invariant background subtraction using two views: Error analysis, sensor placement and applications. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1071–1078, 2005. 23

[122] H. Lin, T. Liu, and J. Chuang. Learning a scene background model via classification. *IEEE Transactions on Signal Processing*, pages 1641–1654, 2009. 6

[123] J. Lindström, F. Lindgren, K. Åström, J. Holst, and U. Holst. Background and foreground modeling using an online em algorithm. In *IEEE International Workshop on Visual Surveillance at ECCV*, pages 9–16, 2006. 4, 16, 17

[124] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, pages 212–261, 1994. 34

[125] B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics*, 2004. 29

[126] H. Liu, L. Liu, and H. Zhang. Ensemble gene selection by grouping for microarray data classification. *Journal of Biomedical Informatics*, pages 81–87, 2010. 29

[127] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall/CRC, 2008. 28

[128] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pages 491–502, 2005. 27, 28, 29

[129] P. Long and V. Vega. Boosting and microarray data. *Machine Learning*, pages 31–44, 2003. 32

[130] W. Long and Y.-H. Yang. Stationary background generation: An alternative to the difference of two images. *Pattern Recognition (PR)*, pages 1351–1359, 1990. 4

[131] F. Lopez-Rubio and E. Lopez-Rubio. Features for stochastic approximation based foreground detection. *Computer Vision and Image Understanding (CVIU)*, 2014. 16

[132] F. Ma and N. Sang. Background subtraction based on multi-channel siltp. In *Workshop on Asian Conference on Computer Vision (ACCV)*, pages 73–84, 2013. 20, 22, 112

[133] T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition (PR)*, pages 16291–1640, 2004. 20, 64, 84, 90, 92, 94, 105

[134] D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing (IVC)*, pages 143–155, 2004. 4

[135] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 837–842, 1996. 19

[136] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 245–259, 2009. 32

[137] J. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin. Foreground-adaptive background subtraction. In *IEEE Signal Processing Letters*, pages 390–393, 2009. 18, 20

[138] A. Mendizabal and L. Salgado. A region based approach to background modeling in a wavelet multi-resolution framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 929–932, 2011. 19, 20

[139] I. Mitsugami, H. Fukui, and M. Minoh. Extraction of potential sunny region for background subtraction under sudden illumination changes. In *International Journal of Computer Vision (IJCV)*, 2014. 10

[140] M. Mousse, E. Ezin, and C. Motamed. Foreground-background segmentation based on codebook and edge detector. *Computing Research Repository (CoRR)*, 2014. 16

[141] M. Murshed and O. Chae. Statistical background modeling: an edge segment based moving object detection approach. *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2010. 16, 17

[142] C. Nagananthini and B. Yogameena. *Crowd Disaster Avoidance System (CDAS) by Deep Learning Using eXtended Center Symmetric Local Binary Pattern (XCS-LBP) Texture Features*. International Conference on Computer Vision and Image Processing (CVIP), 2017. 97

[143] A.Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *International Conference on Machine Learning (ICML)*, 2004. 29

[144] S. Noh and M. Jeon. A new framework for background subtraction using multiple cues. In *Asian Conference on Computer Vision (ACCV)*, pages 493–506, 2012. 17, 44

[145] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition (PR)*, pages 51–59, 1996. 90

[146] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 971–987, 2002. 6, 39, 40, 44, 89, 90

[147] O. Okun. *Feature selection and ensemble methods for bioinformatics: algorithmic classification and implementations*. Information Science Reference - Imprint of: IGI Publishing, 2011. 60

[148] A. Oliva and A Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, pages 145–175, 2001. 84

[149] T. Parag, A. Elgammal, and A. Mittal. A framework for feature selection for background subtraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1916–1923, 2006. 5, 25, 27, 32, 36, 57, 73, 99

[150] J. Paruchuri, E. P. Sathiyamoorthy, S.-C.S. Cheung, and C.-H. Chen. Spatially adaptive illumination modeling for background subtraction. In *International Conference on Computer Vision (ICCV) Workshop on Visual Surveillance*, pages 1745–1752, 2011. 3

[151] R. Péteri and M. Fazekas, S.and Huiskes. DynTex: A comprehensive database of dynamic textures. *Pattern Recognition Letters (PRL)*, pages 1627–1632, 2010. 88

[152] T. Pfister, X. Li, G. Zhao, and M. Pietikainen. Differentiating spontaneous from posed facial expressions within a generic facial expression recognition framework. In *IEEE International Conference on Computer Vision Workshops (CVPRW)*, pages 868–875, 2011. 84

[153] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Studies in Computational Imaging and Vision, 2014. 84

[154] J. Prewitt. Object enhancement and extraction. *Picture Processing and Psychopictorics*, pages 75–149, 1970. 15

[155] J. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, pages 77–90, 1996. 29

[156] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, pages 294–307, 2005. 4

[157] A. Ramirez-Rivera, M. Murshed, and O. Chae. Object detection through edge behavior modeling. *International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011. 16, 17

[158] A. Ravichandran, R. Chaudhry, and R. Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 342–353, 2013. 84

[159] M. Re and G. Valentini. Integration of heterogeneous data sources for gene function prediction using decision templates and ensembles of learning machines. *Neurocomputing*, pages 1533–1537, 2010. 29

[160] V. Reddy, C. Sanderson, and B. Lovell. Robust foreground object segmentation via adaptive region-based background modelling. *International Conference on Pattern Recognition (ICPR)*, 2010. 19, 20

[161] L. Rokach. *Pattern Classification Using Ensemble Methods*. World Scientific Publishing, 2010. 33, 34

[162] L. Rokach and O. Maimon. *Data Mining with Decision Trees - Theory and Applications*. Series in Machine Perception and Artificial Intelligence. WorldScientific, 2014. 33

[163] Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 313–325, 2008. 27, 29, 31

[164] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 58–63, 2001. 88

[165] Y. Satoh, S. Kaneko, and S. Igarashi. Robust object detection and segmentation by peripheral increment sign correlation image. *Systems and Computers in Japan*, pages 70–80, 2004. 18, 20

[166] Y. Satoh, H. Tanahashi, C. Wang, S. Kaneko, Y. Niwa, and K. Yamamoto. Robust event detection by radial reach filter (RRF). *International Conference on Pattern Recognition (ICPR)*, pages 623–627, 2002. 18, 20

[167] C. Saunders. *Subspace, Latent Structure and Feature Selection*. Springer, 2006. 27

[168] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999. 36

[169] A. Schick, M. Bäuml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *IEEE Workshop on Change Detection (CDW) at CVPR*, 2012. 18, 20

[170] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multi-kernel learning. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, pages 993–1005, 2012. 90

[171] N. Setiawan, S. Hong, J. Kim, and C. Lee. Gaussian mixture model in improved ihls color space for human silhouette extraction. *International Conference on Artificial Reality and Telexistence (ICAT)*, pages 732–741, 2006. 14, 16

[172] S. Shaikh, K. Saeed, and N. Chaki. *Moving Object Detection Using Background Subtraction*. Springer International Publishing, 2014. 1

[173] Q. Shen, R. Diao, and P. Su. Feature selection ensemble. *EasyChair*, pages 289–306, 2012. 29

[174] Z. Shengping, Y. Hongxun, and L. Shaohui. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *IEEE International Conference on Image Processing (ICIP)*, pages 1556–1559, 2008. 20, 22, 113

[175] A Shimada and R.-I Taniguchi. Hybrid background model using spatial-temporal lbp. In *IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, pages 19–24, 2009. 20, 22, 44, 114

[176] C. Silva, T. Bouwmans, and C. Frélicot. An extended center-symmetric local binary pattern for background modeling and subtraction in videos. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, pages 1–8, 2015. 20, 39, 44, 63, 77, 110

[177] C. Silva, T. Bouwmans, and C. Frélicot. Online weighted one-class ensemble for feature selection in background/foreground separation. In *nternational Conference on Pattern Recognition (ICPR)*, pages 1–6, 2016. 55, 57, 73, 78, 79, 80

[178] C. Silva, T. Bouwmans, and C. Frélicot. Superpixel-based incremental wagging one-class ensemble for feature selection in foreground/background separation. In *Pattern Recognition Letters (PRL)*, pages 1–7, 2017. 71, 73

[179] C. Silva, J. Gonzàlez, T. Bouwmans, and C. Frélicot. 3d joint color-texture descriptor for dynamic texture recognition. *IET Computer Vision*, 2017. 83

[180] M. Singh, V. Parameswaran, and V. Ramesh. Order consistent change detection via fast statistical significance testing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 4

[181] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding (CVIU)*, pages 4–21, 2014. 6, 27

[182] B. Solmaz, S. A. Modiri, and M. Shah. Classifying web videos using a global video descriptor. *Machine Vision and Applications (MVA)*, 2012. 84, 91, 92, 94

[183] C. Spampinato, S. Palazzo, and D. Giordano. Kernel density estimation using joint spatial-color-depth data for background modeling. *International Conference on Pattern Recognition (ICPR)*, 2014. 16, 24

[184] C. Spampinato, S. Palazzo, and I. Kavasidis. A texton-based kernel density estimation approach for background modeling under extreme conditions. *Computer Vision and Image Understanding (CVIU)*, pages 74–83, 2014. 18, 20

[185] P. St-Charles, G. Bilodeau, and R. Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. In *IEEE Transactions on Image Processing*, pages 359–373, 2015. 3

[186] U. Stanczyk and L. C. Jain. Feature selection for data and pattern recognition. In *Springer Publishing Company, Incorporated*, 2014. 7, 27

[187] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Computer Vision and Pattern Recognition (CVPR)*, pages 246–252, 1999. 4, 10, 13, 16, 24, 25

[188] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 747–757, 2000. 4

[189] A. Stormer, M. Hofmann, and G. Rigoll. Depth gradient based segmentation of overlapping foreground objects in range images. *IEEE International Conference on Information Fusion (Fusion)*, pages 1–4, 2010. 16, 24

[190] Y. Sun, B. Li, B. Yuan, Z. Miao, and C. Wan. Better foreground segmentation for static cameras via new energy form and dynamic graph-cut. *International Conference on Pattern Recognition (ICPR)*, pages 49–52, 2006. 14, 16

[191] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, pages 1635–1650, 2010. 20, 22, 84, 111

[192] D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters (PRL)*, pages 1191–1199, 1999. 57, 58, 63, 77

[193] D. Tax and R. Duin. Combining one-class classifiers. In *Proceedings of the Second International Workshop on Multiple Classifier Systems (MCS)*, pages 299–308, 2001. 62, 76

[194] D. Tax and P. Laskov. Online SVM learning: from classification to data description and back. In *IEEE Workshop on Neural Network for Signal Processing (NNSP)*, pages 499–508, 2003. 59

[195] C. Theriault, N. Thome, and M. Cord. Dynamic scene classification: learning motion descriptors with low features analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2603–2610, 2013. 84

[196] C. Theriault, N. Thome, M. Cord, and P. Perez. Perceptual principles for video ccassification with slow feature analysis. *IEEE Journal of Selected Topics in Signal Processing*, pages 428–437, 2014. 84

[197] F. Tombari, L. Di Stefano, S. Mattoccia, and A. Zanetti. *Graffiti Detection Using a Time-Of-Flight Camera*. Advanced Concepts for Intelligent Vision Systems (ACIVS), 2008. 16, 24

[198] M. Tuceryan and A. K. Jain. *Texture Analysis*. Handbook of Pattern Recognition and Computer Vision, 1993. 17

[199] E. Tuv, A. Borisov, G. Runger, K. Torkkola, I. Guyon, and A. Saffari. Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research (JMLR)*, 2009. 28

[200] J. Uijlings, I. Duta, N. Rostamzadeh, and N. Sebe. Realtime video classification using dense HOF/HOG. In *Indian Council of Medical Research (ICMR)*, 2014. 91

[201] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for outdoor foreground/background extraction. In *Asian Conference on Computer Vision (ACCV)*, pages 291–300, 2012. 43

[202] K. Van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *Signal Processing: Image Communication*, pages 1582–1596, 2010. 85

[203] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001. 31, 32

[204] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, pages 137–154, 2004. 32

[205] L. Vosters, C. Shan, and T. Gritti. Real-time robust background subtraction under rapidly changing illumination conditions. In *Image and Vision Computing*, pages 1004–1015, 2012. 3

[206] L. Wang and C. Pan. Fast and effective background subtraction based on εLBP. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1394–1397, 2010. 20, 21, 25, 44, 106

[207] L. Wang, H-Y. Wu, and C. Pan. Adaptive εLBP for background subtraction. In *Asian Conference on Computer Vision (ACCV)*, pages 560–571, 2010. 20, 21, 44, 106

[208] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. *IEEE Workshop on Change Detection (CDW) at CVPR*, 2014. 25

[209] W. Wang, D. Chen, W. Gao, and J. Yang. Modeling background from compressed video. *International Conference on Computer Vision (ICCV)*, 2005. 19, 20

[210] W. Wang and R. Wu. Fusion of luma and chroma gmms for hmm-based object detection. *Pacific Rim Symposium on Advances in Image and Video Technology (PSIVT)*, pages 573–581, 2006. 14, 16

[211] X. Wang and W. Wan. Motion segmentation via multi-task robust principal component analysis. *Journal of Applied Sciences, Electronics and Information Engineering*, pages 473–480, 2014. 16

[212] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: An expanded change detection benchmark dataset. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 393–400, 2014. ix, xii, 62, 66, 68

[213] X. Wei, Q. Yang, Y. Gong, M. Yang, and N. Ahuja. Superpixel hierarchy. *Computing Research Repository (CoRR)*, 2016. 78

[214] Z. Wei, S. Jiang, and Q. Huang. A pixel-wise local information-based background subtraction approach. *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1501–1504, 2008. 19, 20

[215] M. Wozniak. *Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination*. Springer Publishing Company, Incorporated, 2013. 31, 61, 74

[216] C. Wren and A. Azarbayejani. Pfinder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 780–785, 1997. 13, 16

[217] C. Wren and F. Porikli. Waviz: Spectral similarity for object detection. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2005. 19, 20

[218] H. Wu, N. Liu, X. Luo, J. Su, and L. Chen. Real-time background subtraction-based video surveillance of people by integrating local texture patterns. *Signal, Image and Video Processing*, pages 665–676, 2013. 20, 22, 44, 116

[219] X.Huang, G. Zhao, W. Zheng, and M. Pietikainen. Spatiotemporal local monogenic binary patterns for facial expression recognition. *IEEE Signal Processing Letters*, pages 243–246, 2012. 84

[220] M. Xu and T. Ellis. Illumination-invariant motion detection using color mixture models. *British Machine Vision Conference (BMVC)*, pages 163–172, 2001. 13, 16, 17

[221] P. Xu, M. Ye, X. Li, Q. Liu, Y. Yang, and J. Ding. Dynamic background learning through deep auto-encoder networks. In *International Conference on Multimedia*, pages 107–116, 2014. 5

[222] W. Xu, Y. Zhou, Y. Gong, and H. Tao. Background modeling using time dependent markov random field with image pyramid. In *IEEE Workshop on Applications of Computer Vision (WACV/MOTION)*, 2005. 18, 20

[223] Y. Xu. Moving object segmentation by pursuing local spatio-temporal manifolds. *PhD Thesis*, 2012. 20, 115

[224] Y. Xu, Y. Quan, H. Ling, and H. Ji. Dynamic texture classification using dynamic fractal analysis. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1219–1226, 2011. 83, 84

[225] G. Xue, L. Song, J. Sun, and M. Wu. Hybrid center-symmetric local pattern for dynamic background subtraction. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2011. 20, 22, 40, 43, 44, 109

[226] G. Xue, J. Sun, and L. Song. Dynamic background subtraction based on spatial extended center-symmetric local binary pattern. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1050–1054, 2010. 20, 21, 22, 44, 115

[227] G. Xue, J. Sun, and L. Song. Background subtraction based on phase feature and distance transform. *Pattern Recognition Letters (PRL)*, pages 1601–1613, 2012. 19, 20

[228] S. Yang and C. Hsu. Background modeling from GMM likelihood combined with spatial and color coherency. *EEE International Conference on Image Processing (ICIP)*, pages 2801–2804, 2006. 14, 16

[229] H. Yin, H. Yang, H. Su, and C. Zhang. Dynamic background subtraction based on appearance and motion pattern. In *IEEE International Conference on Multimedia & Expo (ICME)*, pages 1–6, 2013. 20, 22, 44, 112, 114

[230] K. Yokoi. Probabilistic BPRRC: Robust change detection against illumination changes and background movements. *Conference on Machine Vision Applications (MVA)*, pages 148–151, 2009. 18, 20

[231] G-W. Yuan, Y. Gao, D. Xu, and M-R. Jiang. A new background subtraction method using texture and color information. In *Advanced Intelligent Computing Theories and Applications*, pages 541–548, 2012. 20, 44, 104

[232] J. Yuk and K. Wong. An efficient pattern-less background modeling based on scale invariant local states. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 285–290, 2011. 20, 111

[233] G. Zenobi and P. Cunningham. *Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error*. European Conference on Machine Learning (ECML), 2001. 31

[234] C. Zhang and Y. Ma. *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012. 31

[235] H. Zhang and D. Xu. Fusing color and gradient features for background model. *International Conference on Signal Processing (ICSP)*, pages 887–893, 2006. 6, 14, 16

[236] S. Zhang, H. Yao, S. Liu, X. Chen, and W. Gao. A covariance-based method for dynamic background subtraction. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 10

[237] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 436–450, 2012. 84

[238] G. Zhao, T. Ahonen, J. Matas, and M. Pietikäinen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, pages 1465–1477, 2012. 84

[239] G. Zhao, X. Huang, Y. Gizatdinova, and M. Pietikäinen. Combining dynamic texture and structural features for speaker identification. In *Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and intelligence*, pages 93–98, 2010. 84

[240] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 915–928, 2007. 84, 89, 90, 92, 94

[241] B. Zhong, X. Hong, H. Yao, S. Shan, X. Chen, and W. Gao. Texture and motion pattern fusion for background subtraction. *Joint Conference on Information Sciences (JCIS)*, pages 1–7, 2008. 16, 25

[242] D. Zhou and H. Zhang. Modified GMM background modeling and optical flow for detection of moving objects. *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2224–2229, 2005. 25

[243] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 2012. 32, 34

[244] C. Zhu, C. Bichot, and L. Chen. Multi-scale color local binary patterns for visual object classes recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 3065–3068, 2010. 10

[245] J. Zhu, S. Schwartz, and B. Liu. A transform domain approach to real-time foreground segmentation in video sequences. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005. 19, 20

[246] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition (ICPR)*, pages 28–31, 2004. 10

Extraction et sélection de caractéristiques pour la détection d'objets mobiles dans des vidéos

Résumé :

Dans ce manuscrit de thèse, nous présentons un descripteur robuste pour la soustraction d'arrière-plan qui est capable de décrire la texture à partir d'une séquence d'images est proposé. Ce descripteur est moins sensible aux bruits et produit un histogramme court, tout en préservant la robustesse aux changements d'éclairage. Un autre descripteur pour la reconnaissance dynamique des textures est également proposé. Le descripteur permet d'extraire non seulement des informations de couleur, mais aussi des informations plus détaillées provenant des séquences vidéo. Enfin, nous présentons une approche de sélection de caractéristiques basée sur le principe d ?apprentissage par ensemble qui est capable de sélectionner les caractéristiques appropriées pour chaque pixel afin de distinguer les objets de premier plan de l'arrière-plan. En outre, notre proposition utilise un mécanisme pour mettre à jour l'importance relative de chaque caractéristique au cours du temps. De plus, une approche heuristique est utilisée pour réduire la complexité de la maintenance du modèle d'arrière-plan et aussi sa robustesse. Par contre, cette méthode nécessite un grand nombre de caractéristiques pour avoir une bonne précision. De plus, chaque classificateur de base apprend un ensemble de caractéristiques au lieu de chaque caractéristique individuellement. Pour compenser ces limitations, nous avons amélioré cette approche en proposant une nouvelle méthodologie pour sélectionner des caractéristiques basées sur le principe du « wagging ». Nous avons également adopté une approche basée sur le concept de « superpixel » au lieu de traiter chaque pixel individuellement. Cela augmente non seulement l'efficacité en termes de temps de calcul et de consommation de mémoire, mais aussi la qualité de la détection des objets mobiles.

Mots clés : détection d'objets mobiles, soustraction de l'arrière-plan, apprentissage par ensemble, sélection de caractéristique, extraction de caractéristique.

Feature extraction and selection for background modeling and foreground detection

Summary:

In this thesis, we present a robust descriptor for background subtraction which is able to describe texture from an image sequence is proposed. The descriptor is less sensitive to noisy pixels and produces a short histogram, while preserving robustness to illumination changes. Moreover, a descriptor for dynamic texture recognition is also proposed. This descriptor extracts not only color information, but also a more detailed information from video sequences. Finally, we present an ensemble for feature selection approach that is able to select suitable features for each pixel to distinguish the foreground objects from the background ones. Our proposal uses a mechanism to update the relative importance of each feature over time. For this purpose, a heuristic approach is used to reduce the complexity of the background model maintenance while maintaining the robustness of the background model. However, this method only reaches the highest accuracy when the number of features is huge. In addition, each base classifier learns a feature set instead of individual features. To overcome these limitations, we extended our previous approach by proposing a novel methodology for selecting features based on wagging. We also adopted a superpixel-based approach instead of a pixel-level approach. This does not only increases the efficiency in terms of time and memory consumption, but also can improves the segmentation performance of moving objects.

Keywords: moving object detection, background/foreground separation, ensemble learning, feature selection, feature extraction.

Laboratoire MIA - Mathématiques, Image et Applications

Faculté des Sciences et Technologies, Université de La Rochelle, Avenue Michel Crépeau

17042 La Rochelle - Cedex 01 - France