Lorena de los Angeles Guachi Guachi

Background Subtraction for moving object detection

UNIVERSITÀ DELLA CALABRIA

SENESCYT
Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación

UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Scuola di Dottorato

"Archimede" in Scienze, Comunicazione e Tecnologie

Indirizzo

Scienze e Tecnologie dei Sistemi Complessi

Con il contributo della

Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT - ECUADOR)

CICLO XXVIII

BACKGROUND SUBTRACTION FOR MOVING OBJECT DETECTION

Settore Scientifico Disciplinare: ING-INF/01

Direttore: Ch.mo Prof. Pietro Pantano

Firma _____

Supervisori: Ch.mo Prof. Giuseppe Cocorullo

Firma _____

Ch.mo Prof. Stefania Perri

Firma _____

Ch.mo Prof. Pasquale Corsonello

Firma _____

Dottoranda: Dott.ssa Lorena de los Angeles Guachi Guachi

Firma _____

Background Subtraction for moving object detection

Ph.D canditate: Lorena de los Angeles Guachi Guachi

University of Calabria, 2016

Supervisors: Prof. Giuseppe Cocorullo

Prof. Stefania Perri

Prof. Pasquale Corsonello

Abstract

Background Subtraction is a technique that deals with separating input frame into meaningful moving objects (foreground) with their respective borderlines from the static (background) objects that remain quiescent for a long period of time for further analysis. It works mainly with fixed cameras focused on increase the quality of data gathering in order to "understand the images".

This technique for moving object detection has widespread applications in computer vision system with the modern high-speed technologies along with the progressively increasing computer capacity, which provides wide range of real and efficient solutions for information gathered through image/video as input sequence. An accurate background subtraction algorithm has to handle challenges such as camera jitter, camera automatic adjustments, illumination changes, bootstrapping, camouflage, foreground aperture, objects that come to

stop and move again, dynamic backgrounds, shadows, scene with several object and noisy night.

This dissertation is focused on study of the Background Subtraction technique through an overview of its applications, challenges, steps and several algorithms which have been found in literature in order to propose efficient approaches for Background Subtraction for high performance on real-time applications. The proposed approaches have allowed investigation of several representations used to model the background and the technique considered for adjusting the environmental changes, it has provided capability of several color invariant combinations for segmenting foreground as well as to perform a comparative evaluation of the optimized versions of the Gaussian Mixture Model and the multimodal Background Subtraction that are approaches with high-performance for real-time segmentation. Deep Learning has been also studied through the use of auto-encoder architecture for Background Subtraction.

Experimental test in terms of accuracy over proposed algorithms which are based on the analysis at pixel-level and exploit the use of two channels based on the color invariants and the Gray scale level information have demonstrated that Gaussian Mixture Model with two channels achieves a higher robustness, is less sensitive to noise and increases the number of pixel correctly classified as foreground for both indoor and outdoor video sequences.

Embedded algorithm on Raspberry PI provides an inexpensive implementation for low-cost embedded video surveillance systems with combination of few historical frames by the use of two channels that obtains high performance and good quality also within the Raspberry-Pi platform.

Meanwhile multimodal Background Subtraction algorithm is focused on achieve low computational cost and high accuracy on real-time applications by using a limited number of historical frames and a percentage analysis for updating background model in order to be robust in presence of dynamic background and the absence of frames free from foreground objects without undermining the accuracy achieved. For this approach, different hardware designs have been implemented for several image resolutions within an Avnet ZedBoard containing an xc7z020 Zynq FPGA device where Post-place and route characterization shows that the proposed multimodal approach is suitable for the integration in low-cost high-definition embedded video systems and smart cameras.

Sommario

Background Subtraction è una tecnica che si occupa di separare dei cornice di ingresso in significativi oggetti in movimento (foreground) con i rispettivi confini dei (background) oggetti statici che rimangono quiescente per un lungo periodo di tempo per ulteriori analisi. Questo lavora principalmente con telecamere fisse focalizzati sul migliorare la qualità della raccolta di dati al fine di "comprendere le immagini".

Questa tecnica per il rilevamento di oggetti in movimento ha diffuse applicazioni nel sistema di visione artificiale con le moderne tecnologie ad alta velocità, insieme con la progressivamente crescente capacità del computer, che fornisce un'ampia gamma di soluzioni reali ed efficienti per la raccolta di informazioni attraverso l'immagine/video come sequenza di ingresso. Un accurato algoritmo per Background Subtraction deve gestire sfide come jitter fotocamera, automatiche regolazioni della fotocamera, i cambiamenti di illuminazione, il bootstrapping, camuffamento, apertura foreground, gli oggetti che vengono a fermarsi e muoversi di nuovo, background dinamici, ombre, scena con diversi oggetti e notte rumorosa.

Questa tesi è focalizzata sullo studio della tecnica di Background Subtraction attraverso una panoramica delle sue applicazioni, le sfide, passi e diversi algoritmi che sono stati trovati in letteratura, al fine di proporre approcci efficaci per Background Subtraction per alto performance su applicazioni in tempo reale. Gli approcci proposti hanno consentito indagini di varie rappresentazioni utilizzati per modellare il background e le tecniche considerate per la regolazione dei cambiamenti ambientali, questo ha fornito capacità di vari combinazioni di colori invarianti per segmentare il foreground e anche per eseguire una valutazione comparativa delle versioni ottimizzate del Gaussian Mixture Model e il multimodale Background Subtraction che sono approcci con alte prestazioni per la segmentazione in tempo reale. Deep Learning è stato anche studiato attraverso l'uso di architettura auto-encoder per Background Subtraction.

Test sperimentale in termini di accuratezza oltre algoritmi proposti che si basano sull'analisi a livello di pixel e sfruttano l'uso di due canali in base alle invarianti colore e le informazioni di livello di Gray Scale hanno dimostrato che il Gaussian Mixture Model con due canali raggiunge una robustezza maggiore, è meno sensibile al rumore e aumenta il numero di pixel correttamente classificato come foreground sia per le sequenze video interni ed esterni.

Il algoritmo incorporato sul Raspberry PI fornisce un'implementazione economica per sistemi di videosorveglianza integrati a basso costo con combinazione di alcuni fotogrammi storiche mediante l'uso di due canali che ottiene alto performance e buona qualità anche all'interno della piattaforma Raspberry-Pi.

Intanto che il multimodale algoritmo di Background Subtraction è focalizzato sul raggiungimento del basso costo computazionale ed elevata precisione di applicazioni in tempo reale utilizzando un numero limitato di fotogrammi storici e un'analisi percentuale per aggiornare il modello di background per essere robusto in presenza di sfondo dinamico e l'assenza di frame liberi da oggetti in foreground senza compromettere la precisione raggiunta. Per questo approccio, diversi progetti hardware sono stati implementati da diversi risoluzioni di immagine all'interno di Avnet ZedBoard contenenti un dispositivo xc7z020 Zynq FPGA dove post-luogo e la caratterizzazione percorso mostra che l'approccio multimodale proposto è adatto per l'integrazione in basso-costo e alta definizione dei sistemi di video integrato e telecamere intelligenti.

Resumen

Background Subtraction es una técnica que consiste en separar los objetos en movimiento (foreground) de aquellos objetos que permanecen estáticos (background) por un largo período de tiempo, los cuales son de utilidad en análisis posteriores. Trabaja principalmente con cámaras fijas y tiene como finalidad incrementar la calidad de información recolectada para mejorar la "interpretación de las imágenes".

Esta técnica es ampliamente utilizada para detección de objetos en movimiento dentro del área de visión por computador, beneficiándose así de las modernas tecnologías con continuo incremento de capacidad en los computadores, lo cual provee un amplio rango de soluciones reales y eficientes para adquisición de información a través de secuencias de entradas de imágenes/videos. Un eficiente algoritmo para background subtraction debe solucionar problemas como camera jitter, ajustes automáticos de cámara, cambios de iluminación, bootstrapping, camouflage, foreground apertura, objetos que se detienen e inician a moverse luego de cortos períodos de tiempo, background dinámico, sombras, escenas con varios objetos en movimiento y ruido introducido por el dispositivo de captura o por los cambios de iluminación.

Esta tesis está enfocada en el estudio de la técnica Background Subtraction mediante una revisión de las aplicaciones, problemas, pasos y varios algoritmos encontrados en la literatura con la finalidad de proponer eficientes algoritmos de Background Subtraction con alto rendimiento en aplicaciones en tiempo real. Los algoritmos propuestos han permitido investigar varias representaciones utilizadas en el modelado del background, así como también las técnicas consideradas para introducir y ajustar los cambios ambientales dentro del modelo de background, lo cual ha generado el estudio de varias combinaciones de colores invariables para separación del foreground así como también el desarrollo de una evaluación comparativa de versiones optimizadas del algoritmo de Modelo de Mezcla de Gaussianas y del algoritmo multimodal de Background Subtraction, ya que los dos algoritmos son enfoques

con alto rendimiento para segmentación en tiempo real. Deep Learning ha sido también estudiado mediante el uso de la arquitectura de auto-encoder para Background Subtraction.

Pruebas experimentales de los algoritmos propuestos en términos de eficiencia y enfocadas en un análisis a nivel de pixel con el uso de dos canales basados en colores invariantes y escala de grises han demostrado que el modelo de Mezcla de Guassianas con dos canales de colores alcanza una alta robustez, es menos sensible al ruido e incrementa el número de pixeles correctamente clasificados como foreground en secuencias de video internas y externas.

El algoritmo embebido en Raspberry Pi proporciona una implementación embebida de bajo costo para sistemas de seguridad de video vigilancia con la combinación de pocos frames históricos y el uso de dos canales de colores, generando así un alto rendimiento y buena calidad dentro de la plataforma Raspberry-Pi.

Mientras que el algoritmo multimodal de Background Subtraction está enfocado en alcanzar un bajo costo computacional y alta eficiencia en aplicaciones en tiempo real con el uso de un número limitado de frames históricos y un análisis porcentual en la actualización del modelo de background para ser robusto en presencia de background dinámicos y en la ausencia de frames que no contengan objetos en movimiento, todo ello sin reducir la eficiencia alcanzada. Para éste enfoque, diferentes diseños de hardware han sido implementados para varias resoluciones dentro de un Avnet ZedBoard que contiene un dispositivo xc7z020 Zynq FPGA donde el Post-place y la caracterización de la ruta muestra que el enfoque multimodal propuesto es adaptable para la integración en bajo-costo alto-definición embebida en sistemas de video y cámaras inteligentes.

Acknowledgements

I cordially thanks to all who in diverse way contributed in the completion of this dissertation. First of all I am really thankful to God for giving me strength and ability to do my work. I am so grateful to the Ecuador's Secretary of Higher Education, Science, Technology and Innovation (SENESCYT) scholarship scheme and University of the Calabria as well for making it possible for me to study here.

I would like to forward my sincere thanks to my advisors, Prof. Giuseppe Cocorullo, Prof. Stefania Perri and Prof. Pasquale Corsonello at University of Calabria (Italy) who encouraged and directed me during my Ph.D studies. Their support, immense knowledge and guidance helped me to finish my dissertation successfully. I truly appreciate Prof. Stefania Perri's supervision and guidance who has been a wonderful and kind co-supervisor, I have received the most warm and well-intentioned advice. Prof. Stefania has been there for every trouble, struggle and progress in my PhD experience.

Warm thanks goes to Prof. Theo Gevers at Informatics Institute of the University of Amsterdam (Netherlands) for allowing me to do my internship and providing me work placement in the groups they lead "Computer Vision: object recognition and deep learning".

I am also grateful to all the staff, students and interns in the groups to whom I belonged during my Ph.D. pursuit, particularly to Dr. Fabio Frustaci, Ing. Giovanni Staino, Leticia Vaca, Hanan Elnaghy, Anil Baslamisli, Han, Javeria, Erica Calderón and Damián Andrade. Many thanks for that precious time you shared with me during work and also for supporting me in personal matters. Also my sincere appreciation and thanks to the rest of my friends and wonderful people whom I met throughout my life for their immense contribution in my personal and professional life. Many thanks to all of you for the life experience you have allowed.

Thank you from heart to my beloved family, my parents, my brothers and sisters and Cristina Robalino for all their love and encouragement and I am highly grateful for their support during my challenges and pursuits of my life. Many sincere thanks to my family in Amsterdam, Waddendijk family. I will never forget you.

Infinite thanks to all the above mentioned persons and also to those I may have forgotten, and of course to you, warm thanks dear reader.

¡Thank you very much!, ¡Grazie mille!, ¡Muchas gracias!.

Lorena de los Angeles Guachi Guachi
2016, Rende, Italia

“Try not to become a man of success, but rather try to become a man of value.”

Albert Einstein

Dedication

I dedicate this dissertation to my beloved parents, César Guachi and Vilma Guachi, to my dear brothers and sisters (Róbinson, Alexander, Emérita and Jazmin), and to my nephews and nieces (Ronny Josué, José David, Emily and Danna).

Thank you all for helping to give me the life I love today.

Table of contents

List of figures

List of tables

Abbreviations

List of abbreviations

AMBA	Advanced High-performance Bus
BS	Background subtraction
CB	Codebook
CI	Color invariant
CIHW	Color invariant H and W
CNN	Convolutional Neural Networks
F1	F1-score
FBU	Fuzzy Background Update
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FPS	Frames per second
FRA	Fuzzy Running Average
GMM	Gaussian mixture model
GMMHG	Gaussian Mixture model with color invariant H and gray scale
HD	High definition
KDE	Kernel density estimation
LBP	Local binary pattern
MBSCIG	Multimodal Background Subtraction with color invariant and gray scale
MBSCIGA	Multimodal Background Subtraction with color invariant and gray scale approximated
PCB	Percentage of Correct Background Classification
PCC	Percentage of Correct Classification
PCF	Percentage of Correct Foreground Classification
Pr.	Precision

PWC	Percentage of Wrong Classification
QQVGA	Quarter Quarter Video Graphics Array
QVGA	Quarter Video Graphics Array
RAM	Random Access Memory
Rec.	Recall
RPCA	Robust principal component analysis
RT	Real Time
SDM	Sigma Delta Multimodal
SG	Single Gaussian
Sm.	Similarity
SOBS	Self-Organizing Background Subtraction
SOC	ystem-On-Chip
Sp.	Specificity
TN	True Negative
TP	True Positive
VHDL	VHSIC (Very High Speed Integrated Circuit) Hardware Description Language

Chapter 1

Introduction

The advancement in high-speed technologies and the progressive increase in computer capacity over recent years have made it possible that computers process, analyze and interpret both video and image sequences, which gave rise to image/video processing as a specialized discipline in computer vision. This property comprehend and decode features in image/video sequences for increasing the quality of data collection in order to understand image/video sequence.

Spotlight of image and video processing are as follows:

Pattern recognition [83], [130]: It consists of classifying input data into objects or classes using key pre-established features.

Object tracking [115], [70]: This process is used to trace an object (or many other objects of interest) with the aim of identifying the location of objects and their movements precisely.

Reconstruction [18], [74]: It is used to reconstruct objects in two and three dimensions. It is suitable for medicine, biology, earth sciences, archaeology, and material sciences.

Feature extraction [96], [56]: It reduces the image dimensionality by transforming the input data into a set of features that represent the essential characteristics of the input data.

Segmentation [75], [27]: It identifies and isolate areas of interest in the input data in order to analyze the content. It splits up the input data into non-overlapping regions using features such as colors and edges among others. Segmentation is an important step in computer vision to provide detailed information about the objects present in the input data and particularly, to segment an input data into moving and static objects that humans can easily identify and separate, which is called Background Subtraction (BS). An accurately BS extracts the shapes with their respective borderlines of the moving objects present in an input sequence.

This chapter provides an introduction to Background Subtraction, our motivation, the outline of the contribution and a brief overview of the organization of the thesis.

1.1 Motivation

The wide range of technological advances allows us in obtaining enormous amount of information over time [57]. This huge amount of data can be easily acquired through computer and scanner's images, digital cameras or a mobile phones, but their analysis requires complex operations to obtain useful information for subsequent computer vision applications that are constantly expanding in fields such as filtering, human interaction, optical motion capture, medicine, remote sensing, security (surveillance systems), and so on [10].

Identification of moving objects is very important in fast recognition among objects, crowd detection, action recognition [57], [68], [138]. Detection of moving objects is a pre-processing task in many computer vision systems, which must be performed efficiently and accurately to reduce misclassification, false alarms, and missed positives, as well as it must provide fast execution and flexibility in diverse scenarios. In fact, an efficient moving object detection gives absolute identification and is more reliable in getting the same object through input sequence, if its shape, borderlines and position are accurately detected. At the same time this detection should be so fast to identify several objects, normal patterns and also to detect unusual events.

BS is an effective technique to detect and to extract objects of interest (moving objects) as people, cars, animals, abandoned objects between others, which consists in classify a pixel as static or dynamic. The classification/segmentation process separates the static object that remain unchanged for a period of time (background), of moving object (foreground) for further analysis [22]. The background can be composed of motionless objects as doors, walls, rugs, furniture, office supplies, or motion objects as escalators, swaying trees, moving water, waves, rain and others. Most of the time, these objects change from day to night, dark to light, indoor to outdoor and can experience climatic conditions as sunny, rainy and snowy. At the same time, the moving objects can become motionless objects along with the time, as when an escalator stops working, and vice-versa, as when a monitor screen is turned on. As a consequence of these dynamic backgrounds, the pixels can be misclassified as foreground.

Motivated by the challenges to extract automatically the moving objects, several solutions have been envisioning, some of them are classified in terms of the mathematical models used [10]. Peculiarly, the segmentation in environments with color similarity and environmental

illumination change is held with the color invariants [137]. The distortion of the form of the moving objects by shadows is faced handling brightness and color discrimination [67].

In the last decades, a large number of real, powerful, suitable and efficient solutions for moving object detection has been developed [8], their goal is to identify the objects in the scene as do the human vision. However, the ability to understand the images of the video sequences to automatically identify and extract objects of interest, remains a challenging problem in computer vision as a consequence of the computational requirements [102], specializations for certain environments and conditions to reduce the misclassification [23], majority of them are oriented to indoor environments, where ambiental changes as sun shine, rain, wind, etc. have minor effect than in an outdoor environment.

The best solution, therefore, should achieve high speed to incorporate changes from the environment with the ability to run in real-time. In other words, the objective is reaching towards accuracy to classify correctly a pixel as background or foreground without demanding high computational capabilities.

1.2 Scope of the dissertation

The scope of this dissertation is to investigate the models used for the BS to make sense of the environment, using data gathering from images until accomplish a desirable foreground detection even in dynamic environments. For this reason, a comprehensive review of BS algorithms is first presented.

With the help of the gathered information in this work through software tests, some innovative approaches have been introduced, going through the convergence between accuracy and suitability for hardware implementations where computational and memory resource are typically limited. In fact, this thesis presents: (i) a study of the effects induced by combining color invariants H, N, C, W [42], and Gray scale pixels to build a robust color descriptor; (ii) novel algorithms for BS. First one models the Background model using the Gaussian mixture model (GMM), and the second one uses historical frames in conjunction with one modeled frame and global percentage threshold; (iii) hardware implementations and designs of the novel algorithms within different supports, such as the Raspberry Pi for embedded solution and Xilinx FPGAs devices.

The study of color combinations [46], provides a viewpoint to choose the best merge of Gray scale with four candidates of color invariants provided by the Kubelka-Munk theory [42], taking into account the quantitative results of several accuracy metrics, and the channel numbers which can be used for image segmentation.

A novel background subtraction method based on color invariants [45], exploits the Gaussian mixtures for each pixel through two channels: the color invariants [42], which are derived from a physical model, and the gray colors obtained as a descriptor of the image.

Particularly, the novel algorithm proposed in [25], its goal is to achieve low computational cost and high accuracy in real-time applications. It computes the background model using a limited number of historical frames. Thus it is suitable for a real-time embedded implementation. To compute the background model, grayscale information and color invariant H [42], are jointly exploited. Differently from state-of-the-art competitors, the background model is updated by analyzing the percentage changes of current pixels with respect to corresponding pixels within the modeled background and historical frames. Several performed tests have demonstrated that the proposed approach is able to manage several challenges, such as the presence of dynamic background and the absence of frames free from foreground objects, without undermining the accuracy achieved.

Different hardware designs have been implemented for the novel BS algorithm [25], for several images resolutions, within an Avnet ZedBoard containing an xc7z020 Zynq FPGA device. Post-place and route characterization results demonstrate that the proposed approach is suitable for the integration in low-cost high-definition embedded video systems and smart cameras. In fact, the presented system uses 32MB of external memory, 6 internal Block RAM, less than 16000 Slices FFs, a little more than 20000 Slices LUTs and it processes Full HD RGB video sequences with a frame rate of about 74fps.

As an alternative to reduce the portability limitations for computer solutions due its weight, size and power consumption, the Raspberry Pi board is used in the novel implementation proposed in [22], in order to provide an inexpensive and efficient low cost embedded BS solution that does not demands external processing units.

Based on the studies and results obtained in this work, the author demonstrates that a good accuracy is achieved with the combination of only two channels, characterized by Gray scale and color invariant H [42]. Moreover, the experimental software tests of the novel algorithm (based on limited historical frames), reflects its overall performance closer to most suitable BS algorithms for real-time solutions in both indoor and outdoor situations, with a low computational complexity. As well its portable embedded solution shows to be efficient in the presence of noises, opening new trends for portable solutions in low cost embedded platforms onboard.

1.3 Dissertation Overview

The rest of the dissertation is organized as follows: Chapter 2 presents the essential theory of the BS technique, starting with a description of the several applications, followed by a brief explanation of the challenges of a good Background Subtraction algorithm, and the types of descriptors (Picture element and features) used to model the background. Then overview of its steps are presented and the characteristics considered to be an efficient BS solution are discussed at the end of this chapter. Then, in chapter 3, a review of the several BS algorithms will be discussed. A comprehensive and thorough analysis of the results obtained by proposed approaches will be depicted in chapter 4. Finally, chapter 5 synthetizes several keypoints and issues presented in the previous described sections. Therein, the proofs gathered to be deal with the conclusions. Moreover, the areas for future research will be given in this chapter.

Chapter 2

Background Subtraction

2.1 Introduction

Efficient identification of moving objects to extract the essential information from images as do the human vision is a well-known challenging task in computer vision. In recent decades, great interest has been shown for BS technique for this purpose [45], trying to achieve a precise pixel classification as background and foreground and then to identify the objects of interest.

The basic approach for BS consists of a reference image (background model), in which there is no movement and then in every next frame subtracts the reference image to extract objects in movement from the scene; objects in movement are classified as foreground after that they can be used for further analysis. Since the background is not constant following environmental conditions on indoor and outdoor situations as light, noise, reflect colors, wind, movement trees, climatic changes, etc. The development of an approach that includes strategies for updating the dynamic background is required, so these changes in the background are part of the background subtraction and the algorithm allows us to facilitate adaptive background subtraction.

In this chapter, the general approach is presented, starting with several important applications that require BS algorithms. Then, different challenges in background identification environments will be covered. Elements and features used in the background modeling to be robust in critical and dynamic situations are presented. After that, the BS steps will also be described. Finally, the major concern is to be a much more efficient BS algorithm is discussed.

2.2 Applications

BS algorithms are no longer restricted to security applications. It is used in private office, government institutions, private organizations, industrial applications and marketing purposes [30]. BS is an essential task that is often used in the following computer vision applications.

- **Visual surveillance**: Where objects of interest might be moving or abandoned objects, which are identified to assure the security of the concerned area or to provide statistics on road, airport, office, buildings, stores [10], [113].
- **Entomological applications**: Where objects of interest might include beetles, fruit flies, soil insects, parasitic wasps, predatory mites, ticks, and spiders [92]. Moving objects are also responsible for animal activities in protected areas or zoos [10].
- **Optical motion capture**: The goal is to extract a precise silhouette to identify human activities [80], [17].
- **Tracking for video teleconferencing**: Object tracking is the process being used to track the object of interest over the time by locating their positions in every frame of the video sequence [57], [93].
- **Human-machine interaction**: BS technique is useful in interactive applications to provide a human with control over the interaction [50], [126].
- **Video edditing**: Editing functions can be included in video programs or movies in which object of interest can look with different appearance. For instance, people could appear as actors or actresses [93].

2.3 Challenges

Fixed camera, constant illumination, and static background are principal conditions to achieve high accuracy as possible in BS task. However, it is not possible in real-life. Therefore, a good BS algorithm should handle the following challenges under real-life environments. Therein, the list was extended to 16 challenging situations, the first 13 situations were presented in [10].

- **Noise**: It might be introduced in the image due to a poor quality image source, during transmission from the source to the further processing, or caused by environmental factors such as wind, fog, sun-rays, and clouds.

- **Camera jitter**: It is an unexpected and undesirable camera movement. This kind of motion produces false detection without a robust maintenance mechanism.
- **Camera automatic adjustments**: Different cameras may have different adjustments, such as automatic exposure adjustment resulting in global brightness fluctuations in time. It might generate different color levels during frame sequence.
- **Illumination changes**: Throughout the day, outdoor environments often can experience gradual changes. While, sudden changes such as light on/off happen commonly in indoor environments.
- **Bootstrapping**: Bootstrapping is presence of moving objects during model initialization period.
- **Camouflage**: Foreground pixels are included in the background model due to the background and moving objects have very similar color/texture.
- **Foreground aperture**: It is caused by uniform color regions in moving objects. Thus the entire object might not appear as foreground.
- **Moved background objects**: A background object can be moved. These objects always should be considered part of the background.
- **Inserted background objects**: A background object can be inserted. These objects should be considered part of the background since it is introduced.
- **Dynamic background**: It is due to small movements of background objects such as tree branches and bushes blowing in the wind. It requires model which can represent disjoint sets of pixel values.
- **Beginning moving objects**: When a background object initially moves, both it and the newly moved parts of the background appear to change. It produces the ghost regions.
- **Sleeping foreground object**: When a foreground object becomes motionless, it cannot be distinguished from background. Thus, it is quickly incorporated erroneously in the background.
- **Shadows**: Shadows can be detected erroneously as foreground and are projected by background objects or moving objects.

- **Scene with several moving objects**: Multiple objects moving in the scene both for long and short periods of time.
- **Stopping moving objects**: A foreground object might stop for long period of time and requires be introduced as part of the background.
- **Noisy Night**: The most challenging task includes a typical scenario at night, where foreground and background contrast is low, which might result in camouflage of foreground objects.

2.4 Picture Element

The picture element used in the BS steps can be a pixel-level, a block-level, a cluster-level or a frame-level.

Some approaches which models the background with pixel-level [68], [118], [81], use statistics as median, mean or complex multimodal distributions. A block-level analysis [23], [123], [51], is used in order to capture spatial relationship among pixels (the blocks can be of different sizes). Cluster-level [26], further subdivides each frame into constituent clusters, which can be characterized by a weight and a centroid (K-means [68]), or intensity, frequency and number of accesses (Codebook [112]). Whereas, methods that use frame-level [26], [106], [94], have taken into account the entire frame. Particularly, frame-level is performed by methods focused on handling the shade issue by computing ratio of intensity between background model and current frame.

It is relevant that the selection of the picture element establishes the robustness to noise, the accuracy, and the performance. For instance, a pixel-level is less sensible to noise than others but provides high accuracy. Better results can be achieved by combining them.

2.5 Feature

Feature or descriptor, describes essential element in order to distinguish them and particularly to detect them. The goal is to facilitate meaningful matches, through a design of a distinctive feature for each interest point. The main properties of a good feature should be highly distinctive and robust, in order to capture the peculiar information relating to areas of interest and discard changes due to noise and other issues such as time of the day, illumination changes, camouflage, camera jitter, camera automatic adjustments, bootstrapping, among others critical situations (presented in 2.3 Challenges).

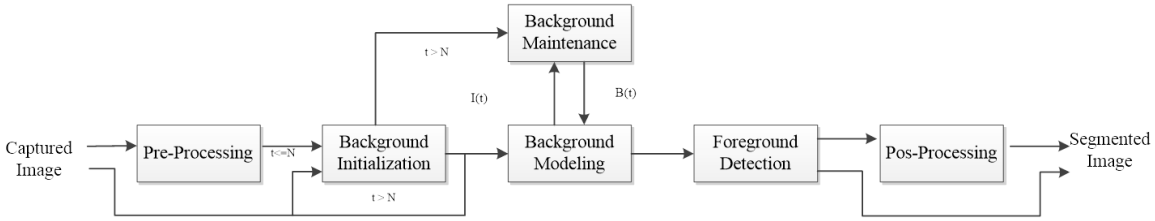


Fig. 2.1 Work flow of the Background Subtraction process.

The highly used features are: color, edges, texture, stereo and motion [10]. Each one is particularly robust to handle critical issues in a different way. For instance, color feature is highly discriminative but depends on the way of representing colors in the image. Therefore, different color representations obtain different accuracies which are limited in presence of shadows, illumination changes, and camouflage [46]. On the other hand, edge feature is very discriminative in presence of ghost and illumination variations. Texture feature works well in presence of shadows and illumination variations. Stereo is robust in order to handle the camouflage issue. While motion feature is useful for detecting articulated objects at cost of increasing the computational cost [96].

2.6 Background Subtraction Steps

An overview of the BS steps is illustrated in Figure 2.1. Generally, the BS process starts with a period of training to obtain the first background model. Followed by the classification process, it is necessary to segment the image by comparing the current frame with respect to the background model. Hence, to detect the foreground objects. Background maintenance is executed over time to adapt changes on the background model. In order to obtain reliable images, pre and post processing operations could be included. In essence, the pre-processing step consists of applying operations over captured image to generate a compressed and reliable image. Post-processing steps as morphological operations often are applied at the end in order to remove noise and enhance the recognized foreground regions. The defined basic steps are described in the following sub-sections.

2.6.1 Pre-Processing

Pre-processing step provides a reliable image by passing through different smoothing filters for blurring and for removing image noise. Geometric and radiometric/intensity adjustments are usually used for this purpose. For geometric operations, frame registration is used to align several frames into the same coordinated frame. Similarity or projective transformations are

used in situations with small camera motions. Intensity adjustment is focused on compensating the illumination variations between consecutive frames. In the same way, color, spatial and temporal derivatives are used to include motion information and edges [110].

To improve the processing speed for the posterior steps and to be able to process heavy data images, the pre-processing phase often compress the image using scaling through bilinear interpolation [115], [110]. As a consequence of this, the frame-size and frame-rate is reduced.

An efficient survey about several BS algorithms with pre-processing operations is presented in [36].

2.6.2 Background Initialization

This step is performed to obtain the first background model. It can be done with different approaches which do not require any training relating to initialize the first frame. Several traditional based training emphasizes on learning focused on statistical properties of background, and have initialized the background model over a set of captured frames (N) during training time, as have done in [127], [37], [80], [55], [70]. The initialization step is critical, especially when looking for an immediate response in sudden interruptions over the moving object detection. Thus, convenient algorithms as [72], [78], [75], [95], use the first frame in order to provide an instant initialization and reduction in the amount of memory required for storage purpose at the cost of misclassification rate by the ghost effect in presence of foreground objects.

Initializations based on first or short number of N frames have a strong assumption that no moving objects are present in the training time. However, in real situations, it is difficult to get a clear background (without presence of moving objects), and the long time duration is required to eliminate the foreground objects in order to obtain the first background model.

In fact, the presence of foreground object is a major challenge for the initialization process. It is also known as bootstrapping issue (presented in 2.3 Challenges), which cannot be controlled and occlude part of the background in presence of moving objects [54], [[70], [129]. It is often observed in crowded environment applications such as schools, banks, transport stations (bus, train), shopping malls, airports, lobby, etc.

The background model can be initialized and reinitialized using non-supervised [80], [129], [21], [79], [30], or supervised [37], [16], [139], [88], [118] procedures. In non-supervised procedure, the background model can be built depending upon the static or moving patterns, the most of its parameters can be learn online automatically, and not requires any human intervention even in complex and dynamic environments. On the other

hand, supervised procedures are dependent of the human intervention to the parameters updating whether the scene is dynamic and is computationally more efficient.

2.6.3 Background Modeling

The aim is to build the representation used to model the background. When the background is particularly static and the camera is fixed, the background is often represented through a single static frame (uni-modal). On the other hand, robust multi-modal backgrounds are required to face some typical variations in the background. It defines the robustness to handle with complex dynamic backgrounds, bootstrapping scenes, illumination variation, and others [10], [36]. The accuracy and performance of BS mostly depends on the background modeling representation that exploits (several models which are presented in chapter 3).

2.6.4 Background Maintenance

Background maintenance defines the technique used for adjusting the environmental changes to the background model. Depending on the environments and its application, it is required to regularly update the background model in order to avoid an obsolete background model that increases the rate of detection errors.

For maintenance, Bouwmans [10], considered following key points: the maintenance scheme, learning rate, update mechanism and frequency. Maintenance scheme establishes which pixels of the background model are updated and rules should satisfy. Learning rate determines how fast new information is introduced. Meanwhile, the update mechanism determines the taken time by a static foreground object before being included in the background model. Finally, the frequency is attempted to update just when it is necessary.

2.6.5 Foreground Detection

It consists of extracting the object of interest in the video sequences through a classification process, which identify a pixel as background or foreground. The classification can be performed through difference, statistical and clustering techniques according to Elhabian et al. [36].

- **Difference technique**: The most traditional technique to segment an image consists of thresholding the computed difference between current frame and background model. The difference can be absolute, relative, normalized or predictive. Absolute difference is often used when the value or modulation is limited to ranges or signals [81], [27]. Relative difference is used to emphasize the contrast in dark areas such as shadow

[110] while, normalized difference technique is applied to balance the color in the bright distortion and chromaticity distortion [50], [103]. At the end, the difference value is thresholded to identify foreground and background pixels.

- **Statistical technique**: This technique uses knowledge of a set of significant earlier frames to learn the background model, and afterwards thresholding the statistical representation of the background model to identify the variations between the model and the currently captured frames. Here, standing out methods like Single Gaussian distribution, Gaussian mixture model (GMM), kernel density estimation, local binary pattern (LBP) and autoregressive estimation.

  One of the most common statistical techniques is based on modeling each pixel with a single Gaussian distribution [126], [49], [82], whereas any recent pixel is classified as a foreground pixel whether it belongs to the distribution or not. In GMM [45], [111], [19], each pixel value of the background model is represented with a few Gaussian distributions. Whereas that KDE avoid any distribution assumption and estimates the pixel intensity value from most recent samples of data [72], [38], [33].

  LBP classifies each pixel by thresholding the eight surrounding neighborhood with the model pixel value [133], [69], [117]. Consequently, autoregressive estimation technique as Kalman filter [70], Hidden Markov models [26], classify the pixel over time considers the previous pixel values.

- **Clustering technique**: Individual features like brightness, intensity, weights are grouped into clusters of this kind of technique [55], [139], [112]. The current pixel is classified as background if it satisfies the cluster conditions, otherwise it belongs to foreground

2.6.6 Post-Processing

Post-processing step consists of further processing of the segmented image to minimize the effect of noise and the pixels that are not part of the foreground. Many BS algorithms includes this step, while others one relegate quietly its use to add some form of correction or consistency to their results [8]. According to Parks et al. [97], techniques like noise removal, morphological closing, area thresholding, saliency test, optical flow test and object-level feedback can be performed to enhance the final detected foreground results.

Noise removal technique consists of applying noise filtering algorithms in order to remove the misclassified blobs, which are oftenly produced by camera noise and background model

limitations. Morphological closing is performed in order to fill internal holes and small gaps, while area thresholding is applied to remove blobs that are so small to be a foreground object.

Saliency test examines if a blob contains a sufficient percentage of most noticeable pixels to represents a valid foreground objects. Optical flow test checks and removes the presence of ghosts blobs while object-level feedback checks whether foreground objects that remain in static for a long period of time are properly incorporated into the background model or not.

Hsiao et al. [51] remove the noise and shadow in the final segmented results through two morphological operations. Mohamed et al. [86], includes a data validation stages function to inspect and remove the pixels that does not belong to the foreground objects. Therefore, it is important to note that each technique can be used individually, repeatedly, or in a combined form, and the enhancement in the final result depends on the selection of adequate parameters by each technique.

2.7 Background Subtraction Considerations

The major concerns of BS applications to identify the moving objects on time are as follows:

2.7.1 Speed

The frequency at which an imaging device displays consecutive images called frames is known as frame rate. This term applies to film and video cameras, computer graphics, and motion capture systems. Frame rate is usually expressed in frames per second (FPS) [125].

Considering the previous definition, the total speed includes the acquisition frame rate and the processing time. The acquisition frame speed often are adjusted on the capture device, which in some cases is associated with the available bandwidth [8]. The processing time is dependent on the size of the image (which can be compressed to improve the speed) [115], the amount of operations per pixel (computational complexity), the texture type of the background such as smoothness and regularity, and the analysis level, which could be at pixel or region (subset of pixels).

The processing rate of operations per pixel in hardware or software relies on the processor or the compiler [8], taking into account that the computing of fixed values are faster that float point values. However, it is difficult to give a precise analysis of this processing rate. Instead, as an alternative, the authors in [22] and [45] evaluated the number of operations included in the algorithm steps. The diminishing of the number per-pixel of operations can significantly increase the speed but at the cost of detection accuracy.

In presence of shadows, moving trees and sudden illumination changes, the texture information has allowed to build robust algorithms as follows [131], [121], [53], [127], [114], [107]. However, the high changes in the texture increases the number of operations. Therefore, in the survey and comparative evaluation [107], some algorithms for shadow detection excludes the small region texture-based in order to diminish the number of operations per pixel, knowing that the large region texture-based algorithms achieve good performance than others.

Recently, the interest in region level has augmented, which associates high accuracy with heavy computation with respect to pixel level analysis. For this reason, the algorithm presented in [115], introduces an implementation in parallel form and compression process with the aim to speed up the detection process. On the other hand, in order to reduce the misclassification rate, the iterative algorithm proposed in [123], uses the Gaussian Mixture background and segments from larger to smaller rectangular region based on color histograms and texture information.

The royal challenge of all these computations is to be quiet short to run efficiently in real-time with process heavy data flow.

2.7.2 Accuracy

An accurate classification furnishes detailed information about the moving objects present in an image and their respective boundaries. The ability to classify correctly a pixel as background and foreground can be measured on video sequences through several quantitative metrics such as: Percentage of Correct Classification (PCC), Percentage of Correct Background Classification (PCB), Percentage of Correct Foreground Classification (PCF), Recall (Rec), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Precision (Pr), F1-score (F1), Percentage of Wrong Classification (PWC), Similarity (Sm) [45], [124], [66]. Rec measures the accuracy of the approach at the pixel level with a low False Negative Rate; Sp stimulates combinations with a low False Positive Rate; Pr favors combinations with a low False Positive Rate, and PCC measures the percentage of correct classifications for background and foreground pixels [44]. Each metric is computed in terms of number of true and false negatives (TN, FN) and true and false positive (TP, FP) as it is presented in the following equations.

$$PCC=(TP+TN)\ /(TP+TN+FP+FN)\times 100 \tag{2.1}$$

$$PCB=TN /(TN+FP) \times 100 \tag{2.2}$$

$$PCF=TP /(TP+FN) \times 100 \tag{2.3}$$

$$Rec= TP /(TP+FN) \times 100 \tag{2.4}$$

$$Sp= TN /(TN+FP) \times 100 \tag{2.5}$$

$$FPR= FP /(FP+TN) \times 100 \tag{2.6}$$

$$FNR= FN /(TP+FN) \times 100 \tag{2.7}$$

$$Pr= TP /(TP+FP) \times 100 \tag{2.8}$$

$$F1= 2 \text{ x } (Pr \text{ x } Rec)/(Pr + Rec) \times 100 \tag{2.9}$$

$$PWC=(FN+FP)/(TP+TN+FP+FN) \times 100 \tag{2.10}$$

$$Sm= TP /(TP+FP+FN) \times 100 \tag{2.11}$$

The computational complexity, processing time and computational requirements have been increased as a result of enhanced robust algorithms to improve the accuracy. Thus, to cope with these issues, it is advised to handle the sudden global illumination changes and make it attractive for real-time applications, automatic parameter estimation and a robust principal component analysis (RPCA) with Markov random field which are introduced by authors in [53]. The adaptive algorithm for object detection in presence of noise and fast-varying environment [23], increases the accuracy in unpredictable backgrounds with the use of the temporal persistence through the simultaneous modeling of background and foreground. The accuracy is also improved in the neural approach with unsupervised learning [31], where each pixel is mapped to the 3x3 neural map and proposes the use of inference

system based on illumination and saturation of the current frame in order to obtain the threshold value that allows to classify the pixel as foreground or background.

2.7.3 Computational Capacity

Good results of accuracy often are achieved at the cost of additional computational requirements such as memory, processors between others. In order to cope with this issue, some algorithms tend to combine different techniques. For instance, the algorithm proposed by Manadhi Santhosh Kumar [68], uses K-means algorithm and analyzes color, gradient and Hear-Like features to handle the pixel variation to be robust in presence of random noise and sudden illumination changes. On the other hand, it also tends to reduce the detection latency, computational complexity and memory consumption through the introduction of temporal differencing technique, which analyses two consecutive frames to extract the variations.

Particularly, temporal differencing technique is computationally less complex and is adaptive to dynamic backgrounds [57]. Its major issues are the presence of holes in detected foreground and the sensibility to the threshold value for segmenting process.

Moreover, the approximated use of integer values instead of floating point values [111], and the limitation of the number of bits for the representation of the integer and fractional part [66], are considered as an alternative to reduce the computational load and memory use. However, despite its benefits for the computational capacity, it is one of the most important design decisions for the hardware implementations, because it can diminish accuracy with respect to the floating-point operations used by the software implementations [16], [41].

Chapter 3

Background Subtraction Algorithms

3.1 Introduction

Conceptually, the extraction of the background consists of recognizing pixels which remain static for a certain period of time that belongs to stationary objects in contrast to pixels with significant variations instead of moving objects.

In the last years, many different BS algorithms have been introduced, and nearly each of them can provide improvements over the basic algorithms and among each other. They can range from very simple algorithms, usually providing poor performance to more robust algorithms that demand a high computational cost which commonly are unsuitable for real-time applications (applications that function within a time frame that the user senses as immediate or current [90]).

Several approaches have been found in literature that can be classified according to the representation which is used to model the background (statistical representations, intensity values, among others), technique considered for adaptation (recursive, predictive) or the picture element used (pixel-level, region-level, frame-level) [16].

In this chapter, a fundamental classification of several BS algorithms is elaborated taking into account the taxonomy proposed in [116], mainly focused on the representation used to model the background and the technique considered for adjusting the environmental changes to the background model.

3.2 Basic Models

Its goal is to maximize speed and reduce the memory requirements. They model the background (B), computing average, mode or median image of the input sequence, which

often do not contain moving objects. Then, at each time (t), new frame (I_t) is subtracted from background model image, after that this result is thresholded using a threshold value (Th), to classify a pixel as background or foreground [25], [100].

$$|(I_t\ )\text{-B}| > \text{Th} \tag{3.1}$$

The threshold value is set for every application, while other algorithms use dynamic thresholds per pixel knowing that the background pixels change dynamically where a poor static threshold can result in poor segmentation [48].

This kind of model can adapt to slow illumination changes in the scene by recursively updating the model using adaptive filters. However, in real outdoor applications, the background of the scene contains many non-static objects (dynamic background) such as tree branches and bushes whose movement depends on the wind in the scene. Dynamic background causes the pixel intensity values to vary significantly over time. This intensity distribution is multi-modal so that the basic models for the pixel intensity/color would not hold and would reach low overall accuracy [34].

3.3 Statistical Models

They use a statistical analysis on individual pixels to build the background model. The pixel information of each processed frame is used to dynamically update the statistics of pixels that belong to the background model [25]. Examples of statistical models are:

- **Running Gaussian average**: It considers that the background pixels are static for most of time and the main source of variation in a pixel value is due to camera noise [126]. Therefore, it is common to model each pixel in the background as a Gaussian distribution considering that the camera noise is commonly modeled with Gaussian model:

$$P(x,\mu,\sigma) = \frac{1}{\sqrt[2]{2\sigma^2\Pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.2}$$

  In [137], three color channel are used for background modeling, then, each pixel of the color channel is modeled with single Gaussian distribution.

- **Gaussian Mixture Model (GMM)**: It offers more robustness against frequent and small illumination changes thus, it usually achieves good accuracy in outdoor environ-

ments [119], [118], [111], [119], [14]. In such models, the history of each pixel is modeled over the time by the mean and variance values of a fix number of Gaussian distribution. The probability that current pixel has value (x_t) at time (t) is estimated in terms of the mean (μ), the weight (w) and co variance (Σ) as:

$$P(x_t) = \Sigma_{j=1}^{k} \frac{w_j}{(2\Pi)^{\frac{d}{2}} \left|\Sigma_j\right|} e^{-\frac{1}{2}(x_t-\mu_j)^T \Sigma_{j=1}^{-1}(x_t-\mu_j)} \tag{3.3}$$

The K distributions are ordered based on (w) and the first B distributions are used to model the background of the input scene where B is estimated as:

$$B = \arg_b^{min} \frac{\Sigma_{j=1}^{b} w_j}{\Sigma_{j=1}^{K} w_j} > T \tag{3.4}$$

T is the threshold and it represents the fraction of the total weight given to the background model.

In this kind of model, input pixels are analyzed by calculating the difference between the pixel and the mean of each Gaussian mixture. If a match is found, the parameters of the matching Gaussian are updated accordingly. Otherwise, if no match is found with any Gaussian mixture, the least probable distribution is replaced with a new one having the mean value equal to the current pixel value, a low weight and a high variance. The weights of the GMM are arranged in descending order. Gaussians that are more frequently matched are more likely to model background pixels and so input pixels are appropriately classified.

Similar analysis is based on Gaussian distributions that can be performed at different levels such as block-level, region-level. For instance, multivariate Gaussian model approach [105], splits each input image into blocks and particularly, probability measurement descriptor uses for each block location is a two component Gaussian mixture model. In the same way, author in [23], divides the observed scene into homogeneous regions and each region is modeled by a Gaussian distribution in the joint spatio-colorimetric feature space.

In presence of abrupt variations, a relatively high number of Gaussians must be considered correctly to model the background. Therefore, an interesting approach is applied in [67], [23], [139] where recursive equations are used to constantly

update the parameters and also to simultaneously select the appropriate number of components for each pixel in order to reduce the algorithm's memory requirements, increase its accuracy, and improve overall performance when the background is highly multi-modal.

- **Kernel Density Estimation (KDE)**: It was initially presented by Elgammal [33], like a non-parametric approach to cope with the drawbacks of manually tuning. After that some enhancements have been proposed to decrease the computational cost using techniques such as histogram approximation and recursive density estimation [72].

  These algorithms are popular due to their robustness in critical situations, such as the presence of noise, shadows and illumination changes [36]. Author in [35], considered that there are at least two sources of variations in a pixel value. Firstly, there are large jumps due different objects such as sky, branch, leaf, which are projected to the same pixel value at different times. Secondly, for those very short variations when the pixel is the projection of the same object where, there are local intensity variations due to blurring in the input. Therefore, the kernel's main aim reflects the local variance in the pixel intensity due to the local variations from input blur but not the intensity jumps.

- **Other statistical background modeling**: This kind of models exclude any approach like to use a specifically single model such as the previous frame or a temporal average, global thresholding, GMM or adaptive GMM. It can include approaches based on obtaining the centroid of the connected pixels moving on the foreground [70], low-rank matrix [138], distance transform [129], background model from Gaussian and Laplacian images [54], among others. For instance, edge segmentation approach is proposed in [89], based on phase feature and distance transform to adapt the motion variation of the background environment. This approach stores static and moving edges into lists and it statistically models the background in terms of weight, position, motion, size, and shape variations information. A pixel-level $\Sigma - \Delta$ decision is proposed by Manzanera [81], which use $\Sigma - \Delta$ filter to provide multiple observation, computing temporal statistics for each pixel of input sequence. This algorithm estimates the background as the simulation of a digital conversion of a time-varying analog signal using modulation (Analogic/Digital conversion using only comparison and elementary increment/decrement). In [50], each pixel is modeled statistically by a 4-tuple (color value, standard deviation, brightness variation, chromacity variation). A probabilistic foreground mask generation is proposed by Reddy [105], to exploit block overlaps and integrate interim block-level decisions into final pixel-level foreground segmentation.

3.4 Cluster Models

The vast majority of algorithms analyze input sequences on a pixel-by-pixel basis, which performs an independent decision for each pixel. A habitual restriction of such processing is that rich contextual information is not taken into account [105]. Thus, cluster methods have been proposed to deal with noise, illumination variations and dynamic backgrounds.

Particularly, in cluster model the background is modeled by a group of clusters where each cluster contains compressed information based on a set of characteristics such as intensity, minimal and maximal brightness, frequency, so when one occurred the longest interval of the time during which it has not reappeared among others like cluster-based approaches which have been used by [112], [39], [29].

Detection involves analyzing the difference of the current input image from the background model with respect to the set of characteristics of compressed information of each cluster. Each incoming pixel verifies if some characteristics (color distortion, brightness range, among others) are less than or within the detection threshold in order to classify it as background or foreground.

Cluster models assume that pixels represented by clusters are able to capture structural background motion over a long period of time. For instance, the Brox-Malik algorithm [12], analyzes the point trajectories along the sequence and segment them into clusters to provide a motion clustering approach that can be used potentially for unsupervised learning. The algorithm presented in [61], quantizes each background pixel into codebooks which represents a compressed form of background model for a long image sequence and are composed of one or more codewords. This allows us to capture structural background variation due to periodic motion over a long period of time under limited memory and can handle scenes with moving background, shadows and highlights.

A variation of codebook algorithm is proposed in [62], where not all the pixels are handled with the same number of codewords. The codebook is mostly used to compress information in order to achieve a high efficient processing speed. A hierarchical proposed in [47], involves two types of codebooks (block-based and pixel-based) to filter areas with different size.

The K-mean algorithm proposed in [14], [91], [98], [15], model each pixel of the input frame by a group of clusters then, they are sorted in order of the likelihood to deal with lighting variations and dynamic background. Incoming pixels are analyzed against the corresponding cluster group and are classified according to whether or not the analysis cluster is considered as a part of the background.

Performance Analysis and Augmentation of K-means Clustering proposed by Parmar [98], who used a clustering technique in order to adjust the input data with an approximation

of a mixture of Gaussians. It provides efficient dynamic background estimation based on the previous N frames. Meanwhile in [15] each pixel was modeled by a group of clusters and sort them in order of probability so that they model the background. Then they are adapted to deal with illumination variations.

3.5 Fuzzy Models

They use Fuzzy logic which is an approach to computing based on "range of truth" rather than the usual Boolean logic ("true or false", 1 or 0) on which the modern computer is based [3]. Fuzzy rules may be performed in terms of rules of the type: if (condition) then (action), to include knowledge of the world in which the system works, such as knowledge of objects (static or moving) and their spatial relations. When the condition is satisfied, the action is performed [78].

Fuzzy models were recently proposed to exploit the advantages of uncertainties and imprecision of the fuzzy logic also in the background subtraction to enhance performance of some approaches and to tackle different challenges of detecting moving objects.

For instance, a fuzzy inference for thresholding is proposed in [75], [30], in order to improve the thresholding technique so to avoid the empirical selection of threshold values by trial and error. Authors in [113], improved performance of running average method using a saturating linear function instead of hard limiter in fuzzy background subtraction. An extended SOBS algorithm presented in [78], incorporates spatial coherence into background subtraction to enhance robustness against false detections and formulate a fuzzy model to cope with decision problems which are arising typically when parameter settings are involved such as the uncertainty in the establishment of suitable thresholds in the background model.

Mahapatra et al. [80], has proposed a fuzzy inference system to model a robust background where distance feature, angle feature and ratio feature are extracted from the contours of the detected objects. These features are used as inputs to a fuzzy rule for classification of the detected motion. In order to exploit the effectiveness of correlogram (inter-pixel relationships in a region) for modeling the dynamic backgrounds, a multi-channel kernel fuzzy correlogram approach is proposed in [21], reducing simultaneously the computational complexity as well as incorporating fuzzy concepts into the correlogram to be less sensitive to small intensity changes and quantization noise. In order to obtain reduced bin and handle dynamic background, illumination variation and camouflage, a novel fuzzy color difference histogram is presented in [95], by using fuzzy c-means clustering to classify the bin local histogram into clusters.

3.6 Predictive Models

They model a scene as a time series and develop a dynamic model to evaluate the current input based on the past observations. The magnitude of the variation between the predicted and incoming data can then be used as a measure of change [85]. The background model can be predicted by Kalman filter [84], [65]; Wiener filter [122], [20]; and neural networks [28], [77]; where pixels of the incoming image which vary significantly from its predicted value are classified as foreground.

- **Kalman filter**: It predicts parameters of interest from indirect, inaccurate and uncertain observations. It estimates X_t (current state) and $X_(t+1)$ (next state) recursively. This technique minimizes the mean square error of the estimated parameters when all noise is Gaussian so the Kalman filter has only the mean and standard deviation of noise. It is considered as the best linear estimator [63]. Knowing the input u_t and the output z_t of the system, the Kalman filter is characterized by the following equations [84]:

$$X_t = Ax_t - 1 + B\mu_t - 1 + w_t - 1 \tag{3.5}$$

$$Z_t = Cx_t + v_t \tag{3.6}$$

  Where A is the state transition matrix, B is the external control transition matrix, w is the process noise, C represents the transition matrix that maps the process state to the measurement and v is the measurement noise.

  In order to enhance the performance in presence of sudden changes with respect to first versions of Kalman filter [58], [9]; author in [84], proposed a background updating algorithm which enables to deal with gradual and sharp global illumination changes. This enhancement introduced a module that measures global changes and uses this information as an external input to the system considering that variations caused by global illumination changes are external events and they are different from variations caused by foreground objects. In the same field, Koler et al. [65] used Kalman filter for modeling the dynamics of the state at a pixel-level to adjust the background model to the lighting conditions change where the parameters are based on an estimation of the rate of change of the background.

  An adaptive version of Kalman filter is proposed in [64], which estimates adaptively the background model taking into account the known effects of weather and the time of

day on the intensity values. Another important variation is presented in [135], where the dynamic texture is modeled by an Autoregressive Moving Average Model in order to solve the disadvantage of previous versions that assume a static or slowly changing background.

- **Wiener filter**: Toyama et al. [122], performed a simpler version of the Kalman filter called Wiener filter which produces a single background estimation based on the past samples to make probabilistic predictions of the expected background at pixel-level. It works well for periodically changing pixels but for random variations, it produces a larger value of the threshold used in the foreground detection. Its major advantage is that it reduces the uncertainty of a pixels value by taking into account for how it varies with time.

  Wiener filter is an optimum linear filter which involves linear estimation of a desired signal sequence from another related sequence. In the statistical approach, to the solution of the linear filtering problem where it assumes the availability of certain statistical parameters (mean and correlation functions). Its goal is to design a linear filter with the noisy data as input and the requirement of minimizing the effect of the noise at the filter output according to some statistical criteria. The Wiener filter is not suitable for situations in which non-stationarity of the signal and/or noise is intrinsic to the problem. In such situations, the optimum and robust filter has to be assumed as a time-varying form. Kalmar filter is more adequate for this difficult problem [Zhehuo].

  Sankari et al. [108] used Wiener filter for dynamic background subtraction in noisy environment after extracting foreground objects for further processing using estimated background and foreground mask as input images in order to minimize the expected squared error between the estimated and perfect images.

- **Neural networks**: Neural network is a beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data [99]. The weights of a neural network are properly trained on N input frames which are used to model the background. They are often temporally updated to reflect the changes observed in the environment.

  Authors in [28], proposed a background neural network architecture to model background image for object segmentation based on an unsupervised Bayesian classifier. The approach proposed by Maddalena [77], is based on self-organizing through artificial neural networks. It can handle the bootstrapping problem, dynamic scenes containing moving backgrounds, gradual illumination variations and camouflage which can be included into the background model shadows that cast by moving objects and

achieves robust detection for different types of videos taken with stationary cameras. Although both methods can selectively update the background model by the learnt background model through a map of motion and stationary patterns its particular disadvantages are that a neural network method requires more memory to store the corresponding weights and the initialization of the weights depends on the first input image of the sequence.

Deep learning is a powerful set of techniques for learning in deep neural networks (solution through multiple layers of abstraction). These multiple layers of abstraction seem likely to give a compelling advantage to deep networks in order to solve complex pattern recognition problems. Currently deep neural networks have captured enthusiastic interest within computer vision and provides a high solution to many problems in image recognition, speech recognition and natural language processing [99].

Deep autoencoders and Convolutional Neural Networks (CNN) are types of architecture of deep neural networks. If the data is highly nonlinear, one solution could be add more hidden layers to the network to have a deep autoencoder. CNN provides translational invariance and requires modifications in the common network architecture. It has obtained a remarkable improvement in object recognition [71].

In deep learning approach, Pei Xu et al. [128] proposed a novel method based on deep autoencoder networks to learn dynamic background. This method uses two autoencoders, the first one extracts the dynamic background image from input sequence containing foreground objects. Then, the second one learns the background using the extracted dynamic background as input. This leads to good performance in environments with large varying background.

A background subtraction algorithm based on spatial features learned with CNN is proposed in [11], where it is learnt that how to subtract the background from an input image patch. The goal is to detect the classification potential of deep features learned with CNN for the background subtraction task.

3.7 Hybrid Models

In order to improve the quality and accuracy of the detection results, enhanced background estimations have been introduced with methods that fusion different models to build complementary approaches. For instance authors in [87], showed that fusion of background estimation algorithm for motion detection in non-static backgrounds in conjunction with an

enhanced background estimation method with a long-term model and a short-term model, improves the quality and reliability of the detection results.

Mahadevan et al. [79] proposed a BS algorithm considering biological vision to define locally the saliency and using center-surrounded computations that measure local feature contrast. The novel BS algorithm proposed in [129], works highly efficient under complex environments and consists of two phases: foreground detection and foreground refinement. The first one model the background pixel as a group of adaptive phase features. While the second one adopts the distance transform to aggregate the pixels surrounding the foreground so that the final result is more clear and integrated.

In the same context, the models of color, locality and temporal coherence are learned online from complex dynamic backgrounds in [32]. This algorithm used a mixture of nonparametric regional model KDE and parametric pixel-level model GMM to build the background color distribution. While that the foreground color distribution is learned from neighboring pixels of the previous frame. Then, the locality distributions of background and foreground are approximated within the nonparametric model KDE. Markov chain is used to model the temporal coherence. After it color, locality, temporal coherence and spatial consistency are fused together in the same framework.

3.8 Algorithms for Real Time applications

This kind of algorithm attempts to reduce the memory use and computational cost. Some approaches such as [25] and [41] have approximated the integers precision to overcome the lack of floating point in low-cost processors.

Recently, several hardware-oriented BS algorithms developed to support real time applications have been proposed in [75], [113], [126], [5], and specific IP modules have been designed for FPGA platforms [16]. The Single Gaussian (SG) algorithm presented in [126] that furnishes an efficient way of modeling the generic pixel through a single Gaussian distribution. Conversely, the approach demonstrated in [5] exploits a statistical model based on a $\Sigma - \Delta$ multi modal modulation that models each pixel by K distributions where each one is characterized by $\Sigma - \Delta$ mean, $\Sigma - \Delta$ variance and weight. The Fuzzy Running Average (FRA) and the Fuzzy Background Update (FBU) algorithms presented in [113] and [75] respectively, provide examples of RT algorithms in which the generic pixel within the background model is updated by means of fuzzy approaches that are taken into account either as the misclassified pixels in the past frames [113] or the neighborhood pixels [75]. In [16], the above algorithms have been compared when hardware implemented using a Xilinx Spartan-3A FPGA chip.

Chapter 4

Discussion of Results

Moving object detection has gained an important role in computer vision systems with the emerging technologies and digital devices which are the easy way to acquire and use high quality and economical video cameras and the increasing demand for understand automatically video/images trough computers.

In this chapter, new approaches have been presented for BS to extract moving objects. They incorporate color information and an analysis at pixel-level. Its main goals are to increase the overall performance by reducing the need to incorporate post-processing task after obtaining the segmented image and to reduce the computational complexity in comparison to other approaches in the state-of-the-art, as well as to run successfully in real time and to be suitable for embedded systems. In order to perform experimental test for the proposed approaches for BS C++ software routines have been implemented.

4.1 Picture element and feature chosen

They allow how to detect, describe and match key-points of areas of interest in an input sequence in order to improve the estimation and identification of moving objects. A proper selection of this will make an easy identification of the same key-points through input sequence of the scene.

The BS approaches presented in this work are interested to exploit the advantages provided by pixel-level analysis as picture element and color are main feature.

4.1.1 Pixel-level

It is chosen to perform an analysis at a very low level as [139], [118], [61], [52], where each pixel is independently processed to model the background of the input sequence and for

each pixel in the input image is to detect variation in pixel values from the model in order to classify the pixel as background or foreground. The pixel-level analysis is more precise than the block-level analysis in order to measure the probability of pixel which belongs to background but it is sensitive to spatial noise, illumination change and small movement of background [32].

4.1.2 Color feature

Color is selected as feature which provides powerful information for object detection and its ability to discriminate foreground and background objects is basically related to the way of representing colors in the processed images [45].

Color descriptor has been used in several approaches as [26], [39], [32], but in certain environments it has several limitations in the presence of camouflage, shadows and illumination changes. However, the combination among different color models allows us in achieving more robust descriptor for the BS [45], [26].

RGB color model has been widely used in BS algorithms. As alternative color models, the HSI, the YCrCbCg, the HSv and the color invariants (CI) [42] are also widely used either to cope with the color similarity or to improve the stability of the illumination change [45].

Although most of the work presented in the literature in [46] have demonstrated how the color features interfere with the achieved accuracy, typical descriptors are based on specific spectral information. On contrary, the CIs are derived from a physical model and can take into account for color spectral information and color spatial structure. Therefore, in order to build a robust descriptor, handling the issues of pixel-level analysis, an experimental study is presented in [46], which evaluated the color spaces with properties independent of illumination intensity, reflectance property, viewing direction, and object surface orientation are defined as the color invariants [43], in conjunction with Gray scale color model.

- **Color Invariant (CI)**: Any method for describing CI model relies on assumptions about the physical variables involved on photometric configuration [42]. Photometric CIs are characterized as a function of surface reflectance, illumination spectrum and the sensing device, which consider the spatial configuration of color, and also the color spectral energy distribution coding color information [137].

  Color invariant properties [43] characterize the image color configuration discounting highlights, shadows, noise and shading. As an example, the Gaussian color model with spectral and spatial parameters is exploited in [137] to define a framework for the robust measurement of colored object reflectance.

Table 4.1 Set of color invariants

CI	Definition
H	$E\lambda/E\lambda\lambda$
N	$E\lambda x.E - E\lambda.Ex/E^2$
C	$E\lambda/E$
W	Ex/E

The CIs are derived from a physical reflectance model based on the Kubelka-Munk theory for colorant layers [42], where illumination and geometrical invariant properties depend on the use of reflectance model. The invariants are useful for materials as dyed paper and textiles, paint films, opaque plastics, dental silicate cements and up to enamel. The CIs derived from Kubelka-Munk theory is listed in Table 4.1. Set of color invariants. The latter shows how computing the CIs named H, N, C, and W, with E, $E\lambda$ and $E\lambda\lambda$ being the spectral differential quotients based on the scale-space theory [40]. The CIs defined in Table 4.1 can be combined incrementally to achieve an alternative to invariant features extraction [42].

- **Gray scale**: The Gray color space model is based on the brightness information and uses the measurement of amount of light (intensity). It is applied for object tracking often on a blob or a specific region [109]. However, taking into account that the color furnishes more information on the objects in a scene, it would be expected that this model can be used in conjunction with other models to achieve more robust solutions and higher accuracy than the basic separated models. For this reason, the Gray color space computing by (4.1) is included in the proposed approaches with the additional advantage of using a color space that does not require complex color transformations.

$$GS = 0.299R + 0.587G + 0.114B \tag{4.1}$$

4.2 Color Invariant Study for Background Subtraction

Being the color widely used as descriptor to improve accuracy in several BS algorithms, a study of the effects induced by combining the CIs presented in 4.1 and Gray scale to build a robust color descriptor is presented in [46]. The experimental study provides a point-of-view to choose the best color combination considering accuracy and the channel numbers which can be further applied for image segmentation.

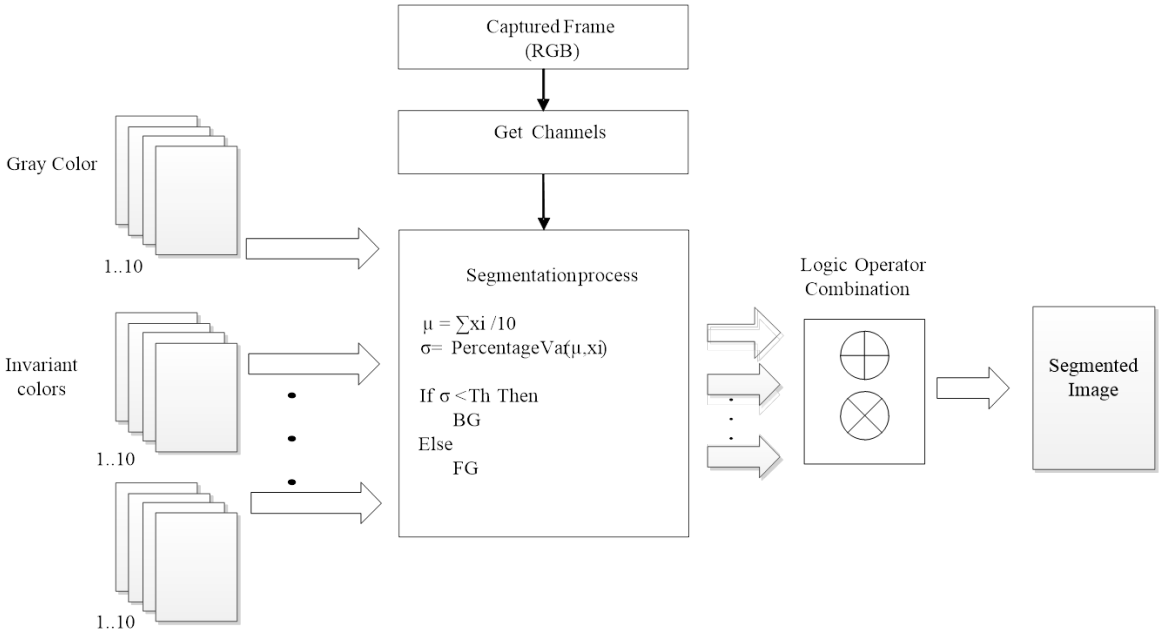


Fig. 4.1 Work flow of the Background Subtraction process

4.2.1 Background subtraction algorithm

The main computational steps required to classify the foreground pixels by using CIs can be summarized as follows: 1) RGB input frames are processed to obtain the CIs; 2) the background model is initialized by collecting, as the historical frames, the CIs obtained for the first Nf frames and the current background is computed; 3) as soon as the (Nf+1)-th frame is acquired, the foreground detection initiates and it is executed pixel-by-pixel by comparing the CIs of the current pixel the CIs of historical frames; 4) the current background model is updated taking into account the obtained classification. The algorithm schematized in Figure 4.1 is used to study the performance of CIs defined in Table 4.1. Some evaluated combinations of the selected features include a channel with Gray scale information whereas others are compounded only by CIs. Each channel is analyzed separately by computing the percentage variation between the current frame and the historical mean. To classify the pixels within the generic frame of a video sequence into the background and the foreground sets, a threshold is performed for each adopted descriptor. In our study, we refer to H, W, N, C and Gray scale components with the threshold values Th=55, Tw=90, Tn=90, Tc=90, Tg=60 that have been set experimentally to the values for which the number of wrong classified pixels is minimized for typical benchmark video sequences [44], [73], [59], [1]. Several tests have demonstrated that higher threshold values reduce the accuracy in detecting foreground pixels, whereas smaller values increase the noise sensitivity.

Each component of the generic pixel of the current frame is compared to the mean value computed from the corresponding channel of historical frames. When the difference between the current examined channel and the historical mean overcomes the relative threshold, the current component is classified as belonging to a foreground pixel.

Otherwise it is recognized as the component of a background pixel. Partial results obtained separately from the examined channels are then combined through appropriate logic operators to obtain the final segmented images. Background model is updated by introducing a new frame at a position zero and discarding the oldest frame of position nine, all frames are sorted after each analysis.

4.2.2 Experimental results

Experimental tests have been done on different video sequences, related to both indoor and outdoor environments and the achieved performances are measured in terms of recall (Rec), specificity (Sp), precision (Pr), percentage of correctly classified pixels (PCC), false negative rate (FNR), false positive rate (FPR) and percentage of wrong classification (PWC). Rec measures the accuracy of the approach at the pixel level with a low False Negative Rate; Sp stimulates combinations with a low False Positive Rate; Pr favors combinations with a low False Positive Rate, and PCC measures the percentage of correct classifications [44]. The set of metrics was classified in two groups considering that in recall, specificity, precision and PCC with a high performance value favors to the combinations by the opposite way, a low performance of FPR, FNR and PWC allows in establishing a well suited combination for segmenting image.

The overall results are summarized in Table 4.2 and Table 4.3. The first column shows the logic operation applied to classify foreground pixels. As an example, the combination (H OR W) AND GRAY detects the generic pixel as foreground only if either its component H or its component W belongs to a foreground pixel, and also its Gray scale data is associated to a foreground pixel.

The results presented in Table 4.2 and Table 4.3 show that as expected, differently combining CIs with Gray scale data vary differently and accuracy can be achieved in detecting foreground objects. It is worth pointing out that the number of channels used to achieve a given accuracy significantly affects the computational complexity.

Indoor (Pets2006, Bootstrap, Office) and outdoor (Highway, Fountain) environments of benchmark video sequences was evaluated separately. Its overall results are summarized in Table 4.4, demonstrating that combination among CIs and Gray scale data achieve higher performance in detecting foreground objects in outdoor environments.

Table 4.2 Performance results of recall, specificity, precision and PCC

Combination	Rec	Sp	Pr	PCC
H AND GRAY	11.13	99.77	81.82	93.88
H OR GRAY	52.65	89.87	27.61	87.50
H AND N	13.58	98.31	33.87	92.68
H OR N	54.60	82.18	18.08	81.74
(H OR N) AND GRAY	13.19	99.72	81.13	93.98
(H OR N) OR GRAY	59.34	82.10	19.27	80.68
H AND C	19.95	96.17	29.22	91.07
H OR C	50.09	85.77	26.24	83.40
(H OR C) AND GRAY	15.08	99.07	79.21	93.44
(H OR C) OR GRAY	56.93	89.11	27.61	87.13
H AND W	9.27	98.67	31.63	92.79
H OR W	59.79	76.14	15.44	76.41
(H OR W) AND GRAY	13.70	99.72	81.33	94.03
(H OR W) OR GRAY	64.02	76.06	16.26	75.30
H OR N OR C	64.08	75.50	15.97	74.77
(H OR N OR C) OR GRAY	68.49	70.14	14.34	70.05
(H OR N OR C) AND GRAY	15.09	99.70	81.60	94.13
H OR N OR W	65.31	70.20	13.78	69.84
(H OR N OR W) AND GRAY	14.76	99.70	81.15	94.09
(H OR N OR W) OR GRAY	66.92	75.44	16.47	74.93
H OR N OR C OR W	68.76	69.63	14.19	69.59
(H OR N OR C OR W) AND GRAY	15.74	99.68	81.83	94.16
(H OR N OR C OR W) OR GRAY	70.95	969.59	14.54	69.72

Table 4.3 Performance results of FPR, FNR and PWC

Combination	FPR	FNR	PWC
H AND GRAY	0.23	6.62	6.12
H OR GRAY	10.13	3.40	12.50
H AND N	1.69	6.53	7.40
H OR N	17.82	3.37	19.64
(H OR N) AND GRAY	0.28	6.45	6.02
(H OR N) OR GRAY	17.90	2.93	19.32
H AND C	3.83	5.98	8.93
H OR C	14.23	22.40	16.91
(H OR C) AND GRAY	0.93	6.35	6.56
(H OR C) OR GRAY	10.89	3.03	12.87
H AND W	1.33	6.92	7.41
H OR W	23.86	2.98	24.97
(H OR W) AND GRAY	0.28	6.40	5.97
(H OR W) OR GRAY	23.94	2.60	24.70
H OR N OR C	24.50	2.60	25.23
(H OR N OR C) OR GRAY	29.86	2.29	29.95
(H OR N OR C) AND GRAY	0.30	6.26	5.87
H OR N OR W	29.80	2.59	30.16
(H OR N OR W) AND GRAY	0.30	6.31	5.91
(H OR N OR W) OR GRAY	24.56	2.36	25.07
H OR N OR C OR W	30.37	2.28	30.41
(H OR N OR C OR W) AND GRAY	0.32	6.21	5.84
(H OR N OR C OR W) OR GRAY	30.41	2.09	30.28

Table 4.4 Average of recall, specificity, precision and PCC by environment type

Combination	Indoor	Outdoor
H AND GRAY	70.23	73.07
H OR GRAY	65.54	63.27
H AND N	58.90	60.32
H OR N	59.78	58.53
(H OR N) AND GRAY	70.40	73.61
(H OR N) OR GRAY	60.45	60.24
H AND C	60.00	58.21
H OR C	65.08	57.67
(H OR C) AND GRAY	69.61	73.79
(H OR C) OR GRAY	66.28	64.11
H AND W	58.00	58.19
H OR W	57.35	56.54
(H OR W) AND GRAY	70.59	73.80
(H OR W) OR GRAY	57.92	57.90
H OR N OR C	57.65	57.51
(H OR N OR C) OR GRAY	55.10	56.40
(H OR N OR C) AND GRAY	70.95	74.32
H OR N OR W	54.14	55.42
(H OR N OR W) AND GRAY	70.75	74.10
(H OR N OR W) OR GRAY	54.48	58.41
H OR N OR C OR W	54.93	56.15
(H OR N OR C OR W) AND GRAY	71.02	74.44
(H OR N OR C OR W) OR GRAY	55.59	56.81

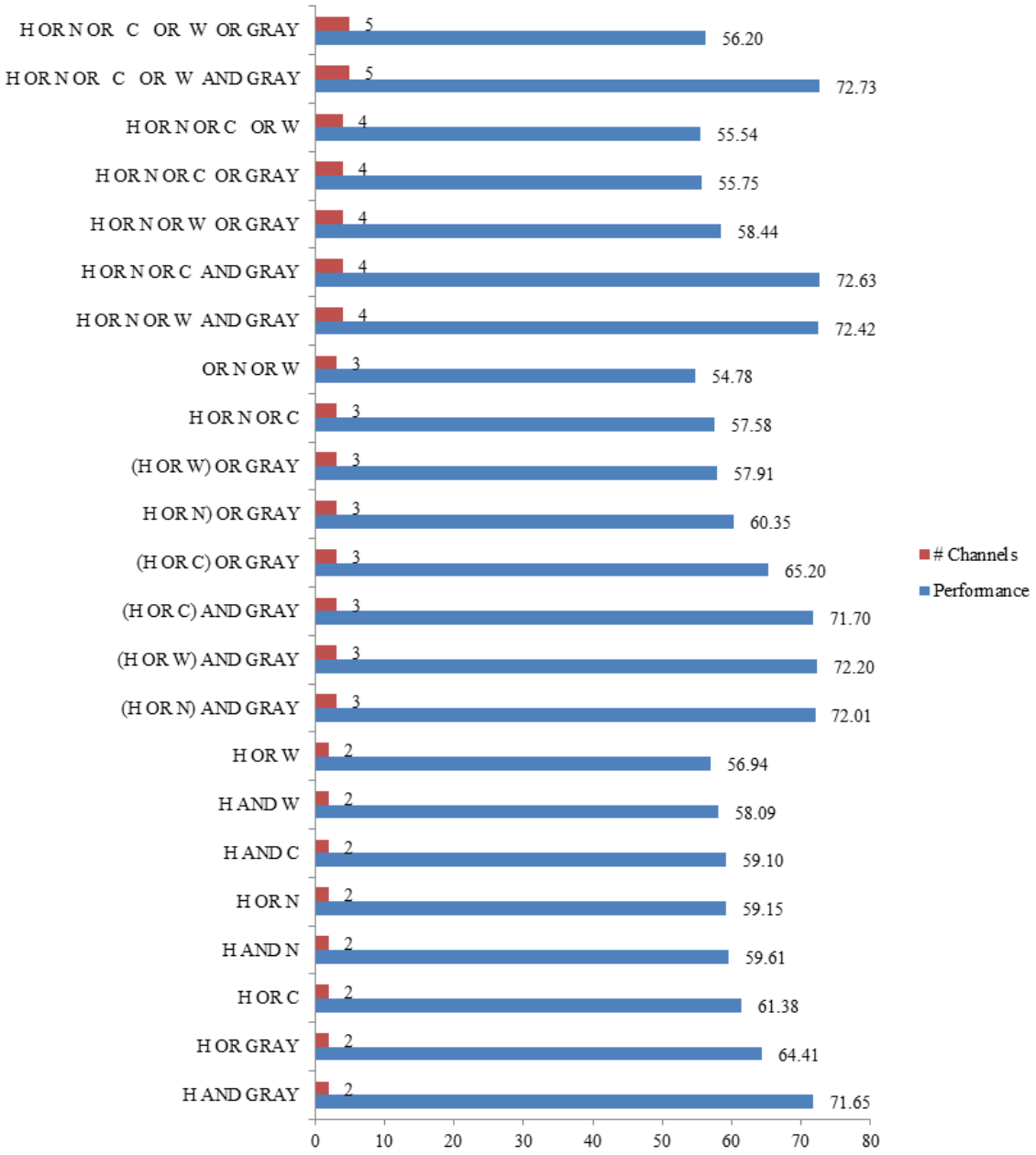


Fig. 4.2 Analysis of the adopted combinations

In Figure 4.2, the average accuracy obtained with each combination is directly related to the number of channels involved. Based on numeric analysis we can see that the combination (H OR N OR C OR W) AND GRAY achieves the best accuracy for indoor and outdoor experimental environments, and focused on the number of channels, the set of H AND GRAY reaches good performance on average with the minimum number of color channels.

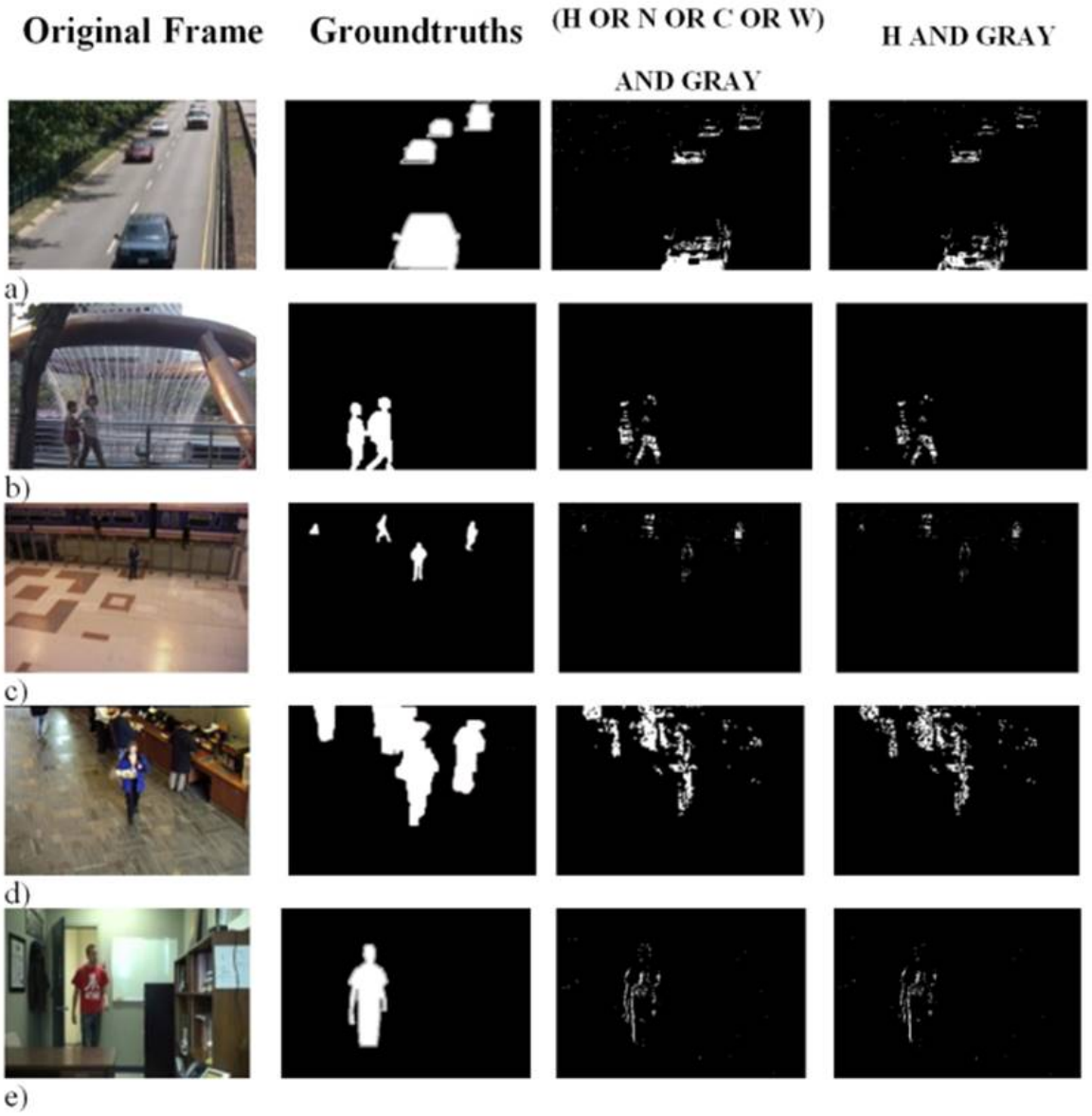


Fig. 4.3 Results related to: a) Highway; b) Fountain; c) Pets2006; d)Bootstrap; e)Office

Figure 4.3 shows some of the segmented images obtained with these two combinations. A complete set of segmented images of each combination for each benchmark video sequence is presented in APPENDIX A. The results depicted in Figure 4.4, show the benefits achieved by introducing Gray scale in the set of CI combination to reduce the noise and to improve the accuracy.

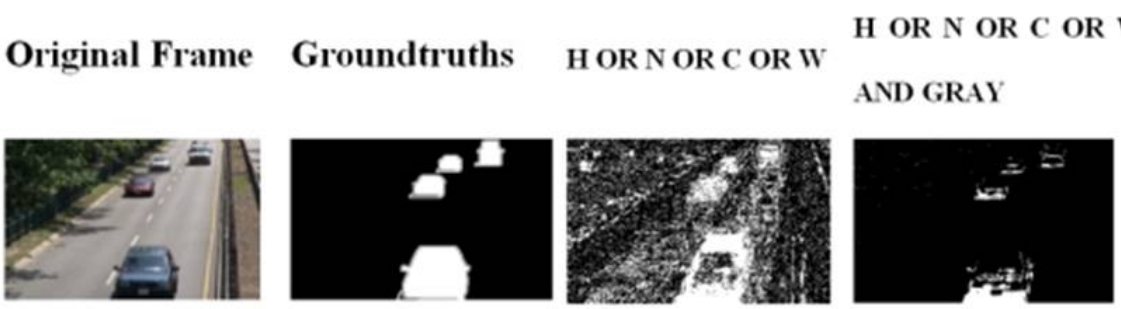


Fig. 4.4 Results obtained introducing Gray scale information

4.3 Gaussian Mixture Model with Color Invariant and Gray Scale

The author in [45] presented a novel background subtraction method based on color invariants, which takes advantages of using the color invariants combined with Gray scale. Gaussian mixtures are exploited for each pixel through two channels: color invariant Hx [42] and the Gray colors obtained as a descriptor of the input image. The update of background model is performed by using a selected random process, considering that in many practical situations it is not required to update each background pixel model for each new input frame. The novel algorithm has been compared with respect to codebook [62], GMM [139] and algorithm based on CI [137]. The novel algorithm has been compared with respect to codebook [62], GMM [139] and algorithm based on CI [137]. Experimental results demonstrate that the proposed method achieves a higher robustness, is less sensitive to noise and increases the number of pixel correctly classified as foreground for both indoor and outdoor video sequences.

4.3.1 Background subtraction algorithm

The main aim of the algorithm is to reduce the sensitivity to noise that may lead to the erroneous classification of foreground objects. To reduce the sensitive to noise ratio, each frame is characterized by two channels: the first one represents the invariant color Hx, and the second channel represents the Gray scale information. Each pixel of each input frame is then modeled by using mixture of Gaussians represented in terms of the mean (μ), the weight (w) and the variance (σ). Thresholding is then separately applied to the channels to recognize both background and foreground pixels. Background pixels are updated based on a random process. The independent results obtained in this way are finally combined to

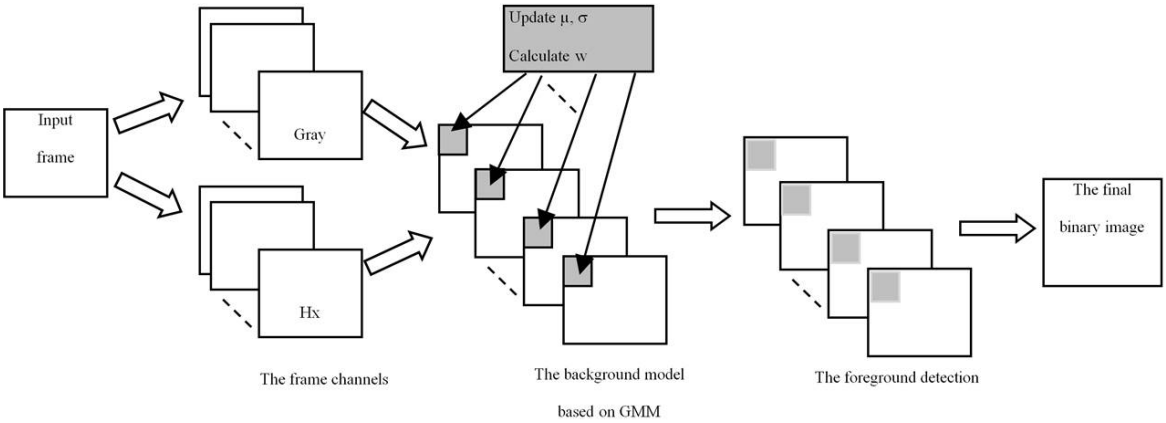


Fig. 4.5 The computational flow of the novel algorithm

generate the final binary image. The computational flow of the algorithm is schematized in Figure 4.5.

The computation flow consists of the following steps:

- **Background modeling**: Each pixel in a frame is modeled as a Gaussian mixture. The RGB frame sequences of a video are converted to gray level and to invariant color Hx. The values from the first frame are used to initialize the model. Each pixel is analyzed calculating the difference between the pixel and then by taking mean of each Gaussian mixture. If the squared difference is less than the threshold Th multiplied by σ, the pixel is classified as background. If the difference does not match with any Gaussian mixture, we replace one of the existing mixtures with a new one having the mean equal to the current pixel value and also having low weight and a high variance.

  The weights of the GMM are arranged in descending order. The sum of the weights must be less than the threshold and it is used to determine the Gaussian mixture that model the background. Different thresholds, ThH and ThG, are used for the Hx and the gray channel respectively, in order to determine whether a pixel is background or not. A match of an incoming pixel allows us to label the pixel as background; otherwise it is classified as foreground.

  When a pixel is classified as a background pixel, its model is updated by using (4.2, 4.3 and 4.4). When the gray channel is considered, ρ represents the grayscale value of the pixel and the threshold Th is equal to ThG, otherwise ρ represents the color invariant Hx and Th is set to ThH. Details on the background modeling are provided in the pseudo-code reported in Figure 4.6, where K is the number of mixtures of Gaussians exploited in the computations.

```
1. for each frame do
2. for each pixel p do
3.    kHit=−1;
4.    for k=1 to K do
5.        D = (p−μ[k])²
6.        if D < Th · σ[k]
7.        // Update the model
8.            if (rand()% 8 = = 1 ) then
9.                μ[k] = ((1-α) · μ[k]) + (α·p)
10.           End if
11.           σ[k] = α · (D − σ[k])
12.           w[k] = α · (1 − w[k]) + w[k]
13.           Sort the model following descending
              ordering of the weights
14.           kHit=k;
15.       End if
16.   End for
17.   if (kHit<0) then // Forced Update
18.       μ[k] = p;
19.       σ[k] = bigVar;
20.       w[k] = littleWeight;
21.   End if
22. Foreground detection
23. End for
24. End for
```

Fig. 4.6 The pseudo-code of the background modeling

$$\mu = (1-\alpha).\mu + .\alpha\rho \tag{4.2}$$

$$w = w + \alpha.(1.w) \tag{4.3}$$

$$\sigma = \alpha.(\mu-\rho)^2 \tag{4.4}$$

- **Foreground detection**: All the weights obtained by the described background modeling are normalized so that their sum is equal to 1. To determine whether a pixel is a foreground, the weights are sorted in descending order and they are summed. The first j weigths that satisfy equation (4.5) are considered as related to background components, whereas the (K+1)-th Gaussian mixture is associated to a foreground component. The detection step is the same in both the channels of the proposed algorithm, only different threshold values could be required. The threshold ThD establishes the fraction of the

Table 4.5 Average PCC values

Method	Lobby	Fountain	Watersurface
Codebook [62]	82.42	47.00	98.22
GMM [139]	97.77	89.74	96.18
Color invariants [137]	93.97	89.65	89.41
New: GMM with color invariants	97.85	89.90	93.08

weights that determine the model of background; this favors Gaussians with higher weights to be selected as the background. The overall foreground detection (Line 22 in Figure 4.6) is obtained by combining the results coming from the two channels. A simple logic AND is then used and a pixel is actually recognized as foreground if both the channels have identified it as foreground.

$$\Sigma_{i=1}^{j} w_i \geq thD \tag{4.5}$$

4.3.2 Experimental results

Experiments were done by running the software routines on a 2.83GHz Intel Xeon processor with 12GB of RAM memory. The benchmark video sequences Lobby, Fountain and WaterSurface [73] have been processed assuming ThD=0.75 for both the Hx and the gray channels. For the Hx channel ThH=0.0121, whereas for the gray channel ThG= 6.5. Some of the results obtained from the comparison for the referred video sequences are depicted in Figure 4.7. It may be observed that, the novel algorithm [45] leads to less noisy results, and it also achieves a higher percentage of correct classification (PCC).

The PCC has been evaluated for all the compared algorithms. Obviously, a different value of the PCC can be obtained for each frame within a video sequence. For this reason, the average PCC values have been computed for each algorithm for each processed video sequence. Results summarized in Table 4.5 demonstrate that as expected, by combining color invariants represented by Hx and gray color information, the new algorithm guarantees higher accuracies in almost all the examined environments.

The PCC has been evaluated for all the compared algorithms. Obviously, a different value of the PCC can be obtained for each frame within a video sequence. For this reason, the average PCC values have been computed for each algorithm for each processed video sequence. Results summarized in 4.5 demonstrate that as expected, by combining color

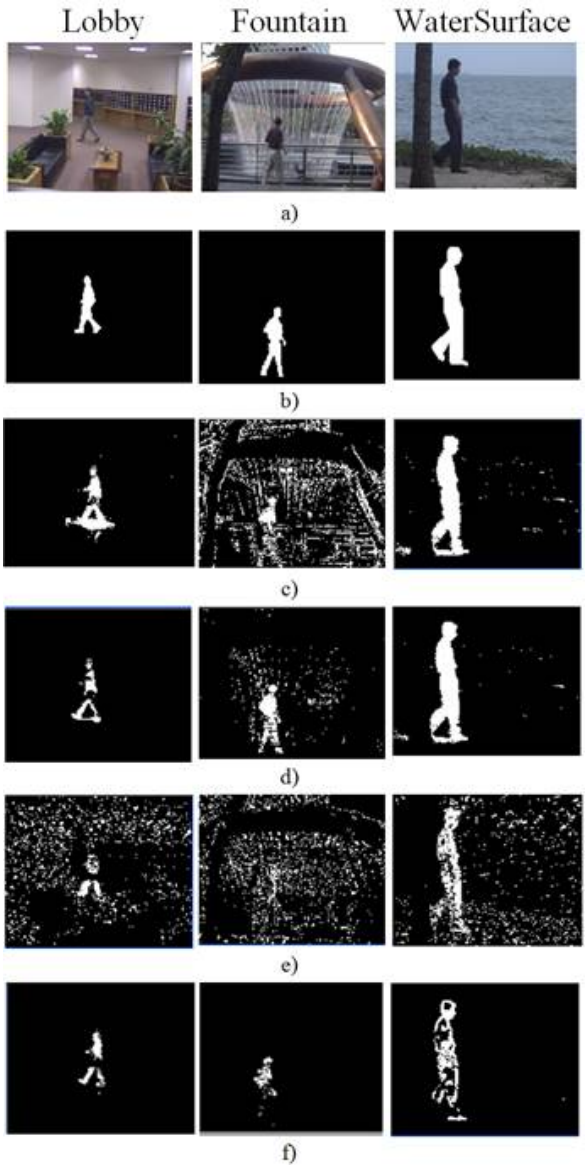


Fig. 4.7 Results related to: a) Original frames; b) ground truths; c) results obtained by [62]; d) results obtained with [139]; e) results achieved by [137]; f) results obtained with the new algorithm

invariants represented by Hx and gray color information, the new algorithm guarantees higher accuracies in almost all the examined environments.

The computational complexities of the new algorithm has also been evaluated in terms of number of multiplications/divisions (MD) and additions/subtractions (AS) required to process the Np pixels within the generic frame of a given video sequence.

Table 4.6 furnishes the number of operations required for each computational step in comparison with [137]. The counterparts codebook [62] and GMM [139] are not included

Table 4.6 Computational cost

Computation	Color invariants [137]		New: GMM with color invariants	
	MD	AS	MD	AS
$E, E\lambda, E\lambda\lambda$	3xNp	3xNp	3xNp	3xNp
Color invariants	3xNp	0	5xNp	2xNp
Background modeling	6xNp+3xnpx(N-1)	6xNpx(N-1)	$\frac{9}{8}Np$	$\frac{3}{2}Np$
Foreground detection	3xNp	6xNp	0	(k1)xNp

in the comparison since they do not exploit CIs. The parameter N appearing in Table 4.6 is the number of frames required by initializing the background model and it is usually equal to 20 [8]. Whereas k is the number of Gaussians used in the new algorithm to detect foreground objects. In the experiments done for comparison with existing counterparts k=5 has been used.

4.3.3 Hardware architecture

A possible hardware implementation of the new algorithm is finally proposed in Figure 4.8 . The top-level architecture depicted in Figure 4.8a shows how the foreground detection can be separated only in parallel design performed through the Hx and the Gray channel. It is also important to note that only two blocks of SRAM memory are required to store the mixture of Gaussians exploited in both the channels to update the background. Details related to the hardware module devoted to check and updated steps are provided in Figure 4.8b. A SRAM stores the Gaussian mixtures (i.e. μ , σ and w) of each pixel of the frame. For each pixel, its Gaussian mixtures are read from the SRAM and stored in local registers. Two control signals (en1 and en2) regulate the updating of μ, σ and w according to the conditions described in Figure 4.6. After the updating, the new Gaussian mixtures are stored in the SRAM and the weights in the w component are summed to detect whether the pixel belongs to the foreground or to the background.

4.4 Embedded surveillance system using BS and Raspberry Pi

When BS is used in embedded platforms, it must be computational efficient due to limited resources. Therefore, the authors in [22] have presented the development and the inexpensive implementation of an efficient algorithm based on the background subtraction technique

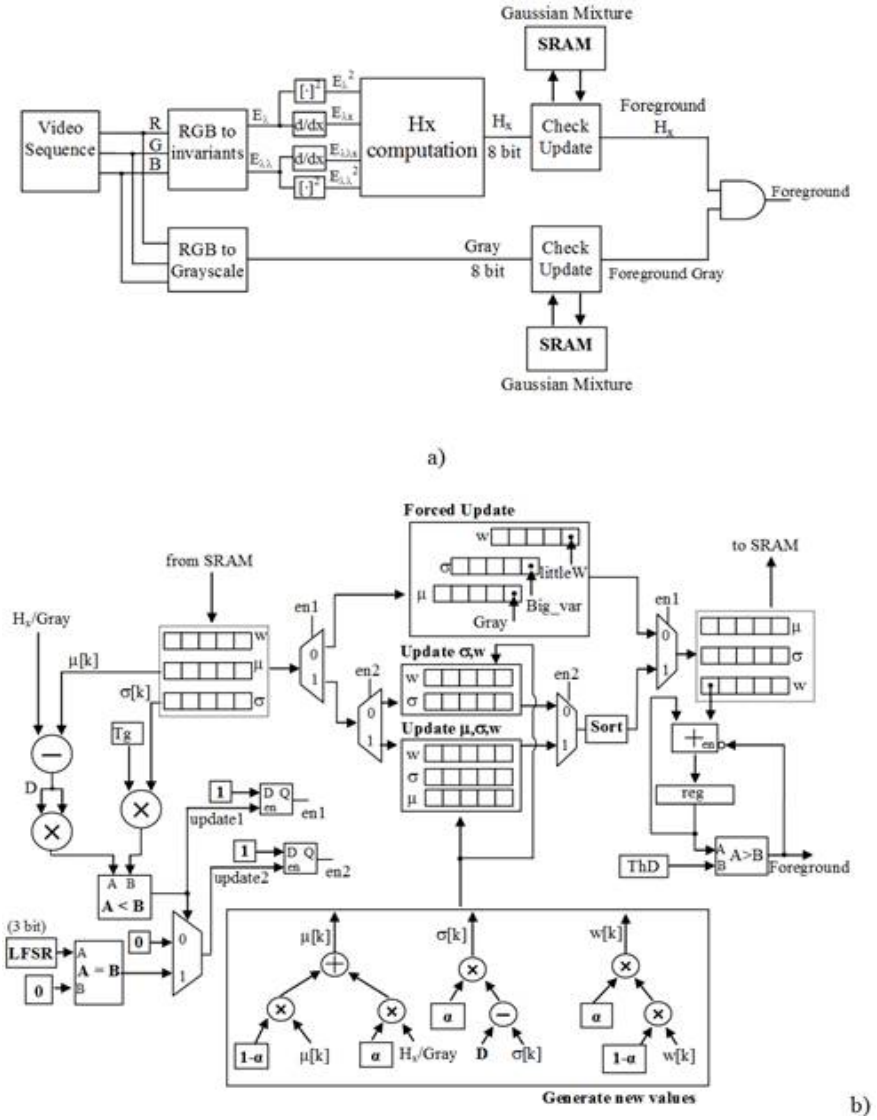


Fig. 4.8 A possible hardware structure designed for the new algorithm: a) the top-level architecture; b) the check-update module

which is adequated for low-cost embedded video surveillance systems. The algorithm exploits the combination of few historical frames with use of two channels based on the invariant color H and the grayscale level information to achieve high performance and good quality also within the Raspberry-Pi[1] platform. Experimental result show that the implemented algorithm is robust against noises typically occurring in both indoor and outdoor environments.

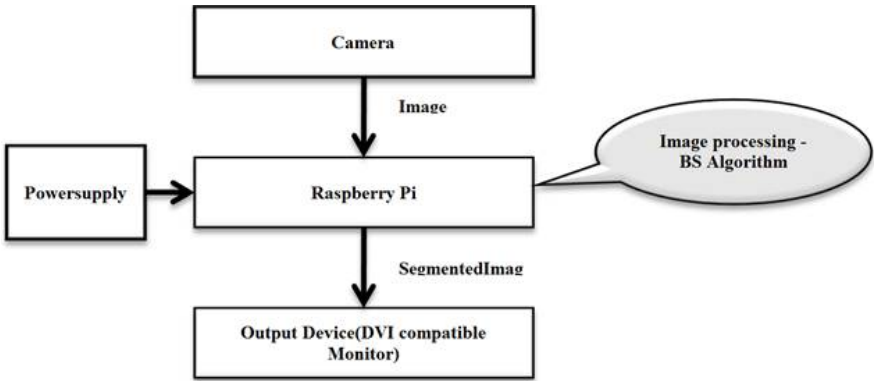


Fig. 4.9 Top-level architecture of the proposed embedded system

Fig. 4.10 Hardware design of the embedded system

4.4.1 Background subtraction algorithm

The surveillance system presented here is organized as depicted in Figure 4.9. The Raspberry Pi is the central element exploited to run the image processing software devoted to the background subtraction, and implemented by purpose designed C++ routines. As the auxiliary hardware, a camera is required to acquire video sequences, and a DVI monitor was used just the purpose of tests.

The system is composed by modules hardware and software as following:

- **Hardware**: Due to its low cost and low energy consumption, the Raspberry Pi and its camera module are used for the surveillance solution. The camera board is suitable for mobile and tiny surveillance systems where weight and size are significant, due its small size (25 mm x 20 mm x 9 mm) and weight (3g). The video sequences are captured by using the Raspberry Pi camera rev 1.3, which plugs directly on the

[1]Efficient low cost hardware platform

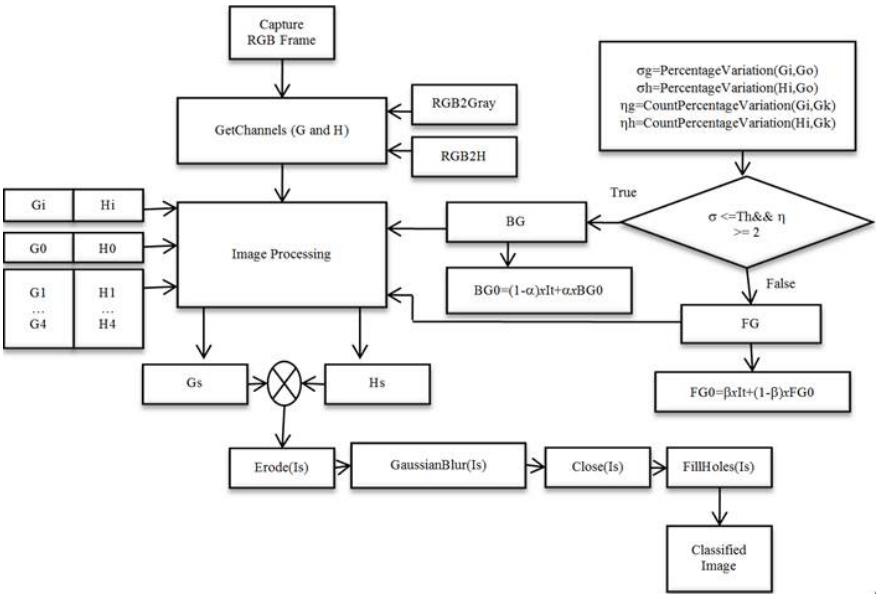


Fig. 4.11 Overview of the implemented background subtraction algorithm

Raspberry Pi CSI connector and is capable of capturing still images as well as high definition videos. When the resolution of 1280x720 is chosen, videos can be captured at 30 frames per second. The designed system is visible in Figure 4.10. The version of the used Raspberry Pi has a Broadcom BCM2835 system on chip, consisting of an ARM1176JZF-S 700 MHz processor, a Video Core IV GPU, 512 MB of RAM, and an SD card for long term storage and booting.

- **Software**: The BS algorithm implemented to perform object's classification as schematized in Figure 4.11. Detection of the objects of interest is based on building a Background model, which will be compared with the current frame to obtain the differences that exist between them.

  1. **Capture RGB frame**

     The Raspberry Pi camera module can capture grayscale or RGB color images. RGB color model with a resolution of 320x240 was used for the embedded system. The first five captured frames are used to initialize the Background model.

  2. **Get channels H and G**

     The color is used as descriptor by considering that the appearance pattern of the object surface can be represented by the color information. Therefore, each captured RGB frame is transformed to Hue color invariant (H) [42] and Gray

scale (G) information with aim of reducing false detections due to illumination changes and/or noises.

3. **Image processing**

   The algorithm analyzes each pixel of each channel separately. To initialize correctly and update the background model, each channel refers to 5 frames four of them are called historical frames, and the fifth frame is called modeled frame.

   The process starts with the building model and for this, the first four successive frames are taken. After that, the fifth frame is created as a replica of the fourth one to initialize the background model. The maintenance of the background model consists in updating the historical frames and the modeled frame. The historical frames are updated by replacing the oldest frame with the new frame. The modeled frame is updated in the analysis process in order to detect moving objects. After initialization, to extract the object of interest, the background model is analyzed with respect to current frame I_t.

   The algorithm analyzes I_t counting for each pixel how many times its percentage variation with respect to the historical frame, is lower than a given threshold. This check is performed for both the H and G channels and the corresponding percentage variation counting, ηg and ηh, are evaluated. At the same time, the percentage variations σg and σh of the pixel value in the current frame with respect to the corresponding pixel in the modeled frame is also calculated for the H and G channel. After that, the channel G recognizes a background pixel whether ηg is greater than 1 and σg is lower than a threshold value Tg=31. Otherwise, the pixel is classified as foreground. Analogously, the channel H recognizes a background pixel if ηh is greater than 1 and σg is lower than Th=41. The threshold values Tg and Th were selected through extensive experimental tests. The modeled frame of the background model is updated according to BG0 (presented in Figure 4.11) in order to determine whether the current pixel is classified as background or as reported in equation FG0.

   It is worth noting that the experimental tests performed for processing several video sequences have shown that α=0.98 and β=0.07 are proper values to guarantee good overall quality.

4. **Post-processing**

   The color transformation from RGB to CI (H) introduces noise to the segmentation process. To cope with such a drawback, the output of the H and G channels are fused together with an AND logic operation. However, the fusion channel

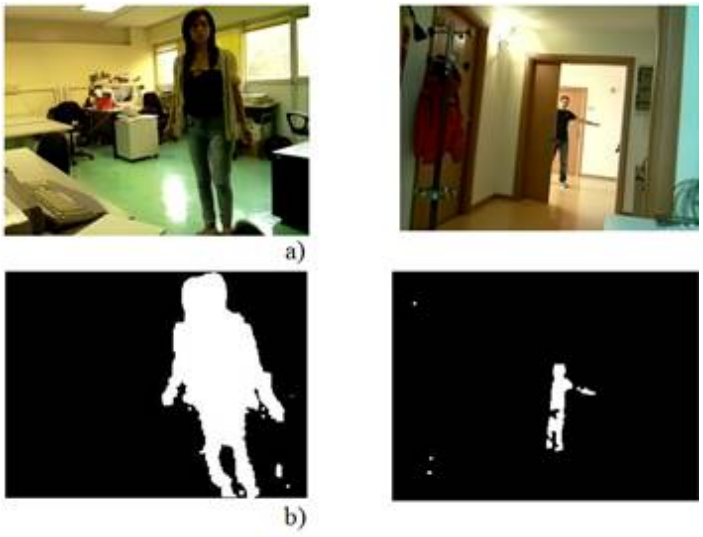


Fig. 4.12 Some results. a) original frame; b) segmented image

is not enough due to the incomplete regions of foreground and noise. Thus, a post-processing step is required to increase the accuracy of the detection. The post-processing section consisting of Gaussian blur, erode and close filters is included to remove random noisy pixels. Finally, the Fill Holes process is also added to extract the shape and the structural information of moving objects.

4.4.2 Experimental results

The proposed embedded system has been first tested in the environment of our lab and some output of the images are depicted in Figure 4.12, 4.13 and APPENDIX B.

The former shows one of the acquired frame and the corresponding segmented image in which the moving object (in this case a person) is clearly visible. On contrary, Figure 4.13 shows one of the acquired frame and the image in which the moving object is bounded within a blob.

The accuracy achieved by the system in classifying background and foreground pixels has been evaluated by processing several benchmark videos [73], [59], [1] and [44] by computing the percentage of correct pixel classifications (PCC).

Table 4.7 exhibits that the embedded novel system achieve higher than GMM color invariants [45] in all the referenced video sequences that are acquired in both indoor and outdoor environments. The segmented binary images obtained for the referred video sequences before the post-processing step are shown in Figure 4.14.

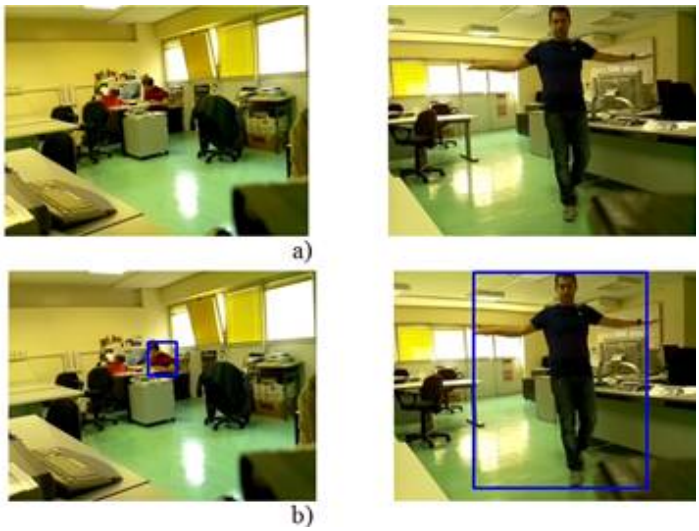


Fig. 4.13 Some results. a) original frame; b) image with blob

Table 4.7 Achieved accuracy and comparison with GMM color invariants [45]

Video sequence	PCC New	PCC [45]
Lobby	98.52	97.85
Fountain	96.71	89.80
WaterSurface	93.26	93.08
Camouflage	89.32	51.79

The accuracy achieved by the new approach is compared to that reached by the algorithm which is explained in GMM color invariants [45] and which uses the same color models exploited in our work. Thus, by combining the color invariant H and the grayscale channel G, we take the advantage of reducing shadows and noisy pixels which are classified as foreground. The novel system is well suited for an embedded implementation in the Raspberry Pi due to its low power consumption for image processing and HD video. Furthermore, experimental tests have demonstrated that color video sequences can be captured at a frame rate up to 126fps.

The Raspberry Pi includes a VFP that can use the hardware unit by improving the performance and by reducing power usage for floating point operations [2], which promotes to build a smaller and portable novel embedded system with lower power consumption. Thus resulting to an approach which is more convenient than PC based surveillance systems. Based on mentioned advantages, the proposed solution works with floating number and does not include optimizations of the numerical operations on the BS algorithm. Obviously, this behavior increases the image processing time.

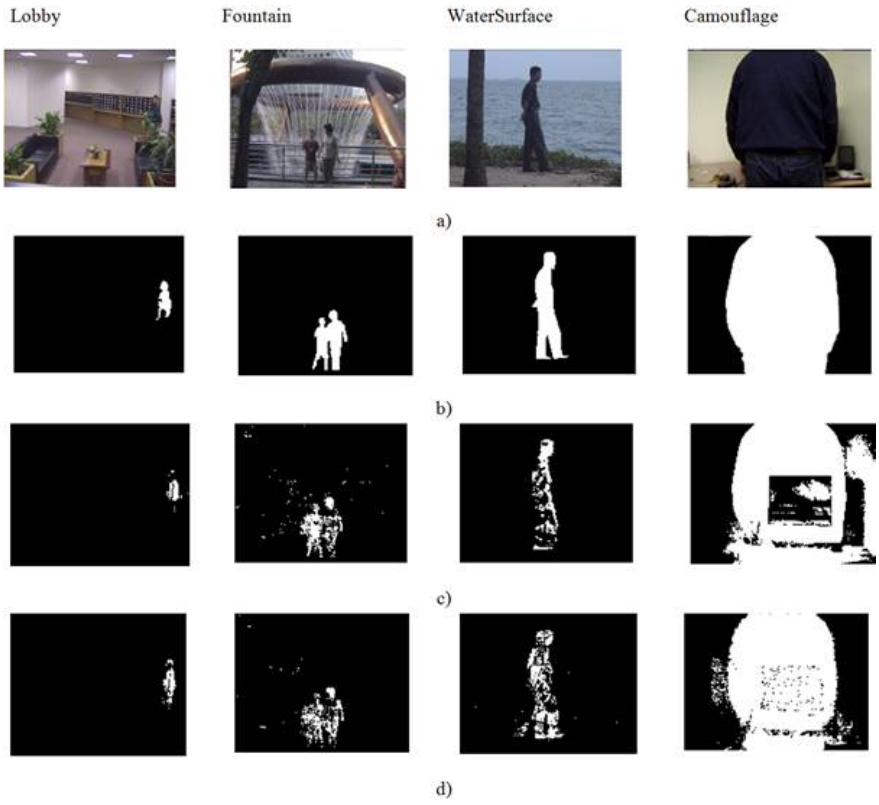


Fig. 4.14 Results related to: a) Original frames; b) ground truths; c) results obtained with [9]; d) results obtained by the proposed embedded algorithm

The computational load of the new approach in terms of the number of multiplications/divisions (MD) and additions/subtractions (AS) required to process all the pixels within the generic analyzed frame is presented in Table 4.8, where Np is the number of pixels and k is the number of Gaussians used to detect foreground objects.

The processing time of the novel embedded algorithm has also been evaluated in terms of number of frames per second (fps). Table 4.9 shows the time consumed by the operations included in the background subtraction algorithm. Here, it is observed that the significant time spent in the color transformation and the segmentation process.

4.5 Multimodal Background Subtraction for high performance embedded systems

In order to achieve low computational cost and high accuracy in real-time applications a novel method for the background subtraction is presented in [25]. It computes the background model using a limited number of historical frames, thus resulting suitable for a real-time

Table 4.8 Computational load

Computation	GMM color invariants [45]		New: Historic with color invariants	
	MD	AS	MD	AS
Initialization	1300xkxNp	2100xkxNp	24xNp	22xNp
Background modeling	$\frac{13}{2}$xkxNp	$\frac{21}{2}$xkxNp	4xNp	2xNp
Foreground detection	0	2x(k-1)xNp	[4x(N-1)]+4xNp	[4x(N-1)]+4xNp

Table 4.9 Processing time

Computation	Time (fps)
Whole algorithm	3.10
Frame captured	107.00
Color gray transformation	214.00
Color invariant transformation H	19.45
Window display	169.50
Background and foreground segmentation	4.28

embedded implementation. To compute the background model as proposed here, pixels Gray scale information and color invariant H are jointly exploited. Differently from state-of-the-art competitors, the background model is updated by analyzing the percentage changes of current pixels with respect to corresponding pixels within the modeled background and historical frames. The comparison with several traditional and real-time state-of-the-art background subtraction algorithms demonstrates that the proposed approach is able to manage several challenges, such as the presence of dynamic background and the absence of frames free from foreground objects, without undermining the accuracy achieved.

With the aim to exploit parallel architecture which can process efficiently heavy data flow in real time providing the advantage of reconfigurable design and low power requirements, different hardware designs have been implemented, for several images resolutions, within an Avnet ZedBoard containing an xc7z020 Zynq FPGA[2] device. Post-place and route characterization results demonstrate that the proposed approach is suitable for the integration in low-cost high-definition embedded video systems and smart cameras. In fact, the presented system uses 32MB of external memory, 6 internal Block RAM, less than 16000 Slices FFs, a

[2]Field-programmable gate array

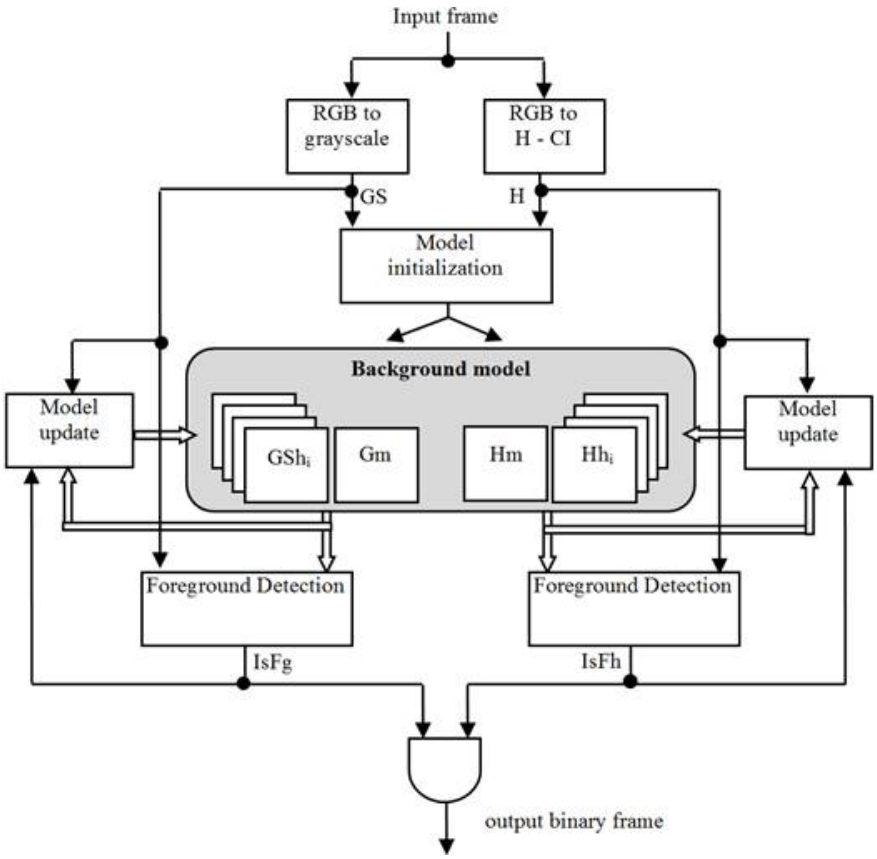


Fig. 4.15 Block diagram of the proposed algorithm

little more than 20000 Slices LUTs and it processes Full HD RGB video sequences with a frame rate of about 74fps.

4.5.1 Background subtraction algorithm

The proposed BS algorithm called MBSCIG belongs to the basic model category. The main computational steps performed by the novel algorithm can be summarized as follows. RGB input frames are firstly processed to obtain the Gray scale image GS and the color invariant H. The latter has been chosen because it is more insensitive to image conditions and simpler to be computed than the alternative color invariants C, W, E, as defined in [42]. The adopted background model consists of N frames acquired before the current one, and therefore called historical frames, and one modeled frame containing the current background. As soon as the (N+1)-th frame (the current frame) is acquired, the foreground detection initiates and it is executed pixel-by- pixel on both the Gray scale and CI channels. Each pixel of the current frame is compared to its counterparts in the historical frames to establish whether it significantly varies or not. The pixel is classified as belonging to the foreground only if both

the channels are consistent with such a decision, otherwise the pixel is classified as belonging to the background. After that, the current background is updated for the next computation by taking into account both the current pixel value and its stored history with appropriate weights. The block diagram of the MBSCIG is depicted in Figure 4.15.

The initialization phase of the background model computation is summarized in the pseudo-code reported in Figure 4.16a. The first N frames are captured and stored as historical frames. GShi and Hhi (with i=1, ..., N) indicate the Gray scale and color invariant histories,

```
1.  For i=1 to N
2.      capture the i-th frame
3.      For each pixel Im(x,y) in the frame
4.          GSh_i(x,y)=RGB2Gray(Im(x,y))
5.          Hh_i(x,y)=RGB2H(Im(x,y))
6.      End for
7.  End for
8.  Gm=GSh_N;
9.  Hm=Hh_N;
```

a)

```
1.  capture the current frame
2.  For each pixel Im(x,y) in the frame
3.      GS(x,y)=RGB2Gray(Im(x,y))
4.      Dc=0
5.      For i=1 to N
6.          v=|GS(x,y)-GSh_i(x,y)|
7.          m=max{GS(x,y), GSh_i(x,y)}
8.          D=(v/m)*100
9.          if (D < Tg)
10.             Dc++
11.         End if
12.     End for
13.     vv=|GS(x,y)-Gm(x,y)|
14.     mm=max{GS(x,y), Gm(x,y)}
15.     DD=(vv/mm)*100
16.     if (DD<Tg and Dc>Tgc)
17.         IsFg=0 //a background pixel is detected
18.         Gm(x,y)=(ρB*GS(x,y) + (1-ρB)*Gm(x,y))
19.     else
20.         IsFg=1 //a foreground pixel is detected
21.         Gm(x,y)=(ρF*GS(x,y) + (1-ρF)*Gm(x,y))
22.     End if
23. End for
24. Discard the oldest GSh and replace it with GS
```

b)

Fig. 4.16 The main computational steps of the novel algorithm: a) model initialization; b) foreground detection and model update

respectively, whereas Gm and Hm denote the current background models that are initially filled with GShN and HhN. The foreground detection and the model update phases are then executed. Such processes are identical for both GS and H channels, thus, in the following, only the GS is referred to. As summarized in the pseudo-code of Figure Figure 4.16b, the newest acquired frame GS is compared to all historical frames pixe-by-pixel, GShi and for each pixel at the position (x,y), the percentage variations D between GS(x,y) and the corresponding pixels GShi(x,y) are computed. Then the number Dc of historical frames with respect to which GS(x,y) varies negligibly (i.e. D is less than the threshold Tg) is counted. Similarly, GS(x,y) is compared to the corresponding pixel Gm(x,y) within the current background. Its percentage variation is indicated with DD in the pseudo-code of Figure Figure 4.16b. Then, the final detection step is executed: if DD is less than the threshold Tg and Dc is higher than the threshold Tgc, it can be concluded that the examined pixel belongs to the background of the image and the output flag IsFg is asserted low. Otherwise, IsFg is asserted high to indicate that the pixel is potentially part of the foreground. In both cases, the pixel Gm(x,y) in the current background is updated as shown in lines 18 and 21 of the pseudo-code of Figure 4.16b.

The parameters ρB and ρF are used properly and differently tune this combination in case of a detected background pixel and a recognized foreground pixel, respectively. Details on the values experimentally selected for N, Tg, Tgc, ρB and ρF are provided in the following Section (Experimental results).

As the final step, the historical frames are updated by discarding the oldest one and storing the latest captured frame.

As illustrated in Figure 4.15, the same detection/update process is separately performed on the H channel that generates the flag IsFh. The current examined pixel is recognized as a foreground pixel only if the flags IsFg and IsFh are both high. Also a hardware-oriented version of the proposed algorithm has been investigated. In the following, it is named as MBSCIG Approximated (MBSCIGA) to indicate that it exploits an approximated formulation purposely introduced to make hardware implementation friendlier. The novel formulation here adopted and provided in (4.6) approximates the matrix elements used in color transformation to obtain H (H = $E\lambda/E\lambda\lambda$ defined in T Table 4.2 from RGB in terms of $E\lambda, E\lambda\lambda$o their nearest powers of two.

$$\begin{bmatrix} E \\ E_{\lambda} \\ E_{\lambda\lambda} \end{bmatrix} = \frac{1}{2^7} \cdot \begin{bmatrix} 2^3 & 2^6 & 2^5 \\ 2^5 & 2^2 & -2^5 \\ 2^5 & -2^6 & 2^4 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.6}$$

Table 4.10 Video sequences used as benchmarks

Computation	Resolution	Description
Lobby [73]	160 x 128	The lights are turned on / off, which produce global illumination changes
Fountain [73]	160 x 128	Repetitive background motion
Bootstrap [59]	160 x 128	Parts of the background are mostly obstructed by moving objects
Highway [44]	320 x 240	More than one moving object
Office [44]	320 x 240	Static objects with the same color of the foreground

4.5.2 Experimental results

The proposed MBSCIG algorithm and its approximated version have been characterized in terms of computational complexity and several accuracy metrics, measured referring to datasets available online [73], [59], [1], [44]. As summarized in Table 4.10, the selected benchmarks represent typical situations that can make the extraction of moving objects critical, thus furnishing an effective way to highlight strengths and weakness of each analyzed algorithms. MBSCIG algorithm uses N historical frames, to model the background, and the thresholds Tg, Tgc, Th and Thc, with the weights ρB and ρF, to update it. All these parameters must be properly set to reach the best accuracy.

First, several tests have been performed on the selected video sequences summarized in Table 4.10 with N ranging from 1 to 100. Using the same evaluation approach exploited in [53], the achieved PCC values have been then averaged, thus obtaining the results plotted in Figure 4.17. The latter shows that N=4 leads to an average accuracy quite high and only

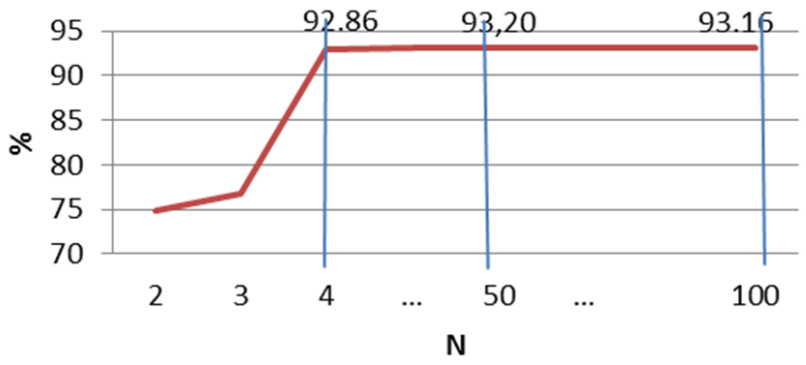


Fig. 4.17 Average PCC versus N

0.3 % lower than that reachable by increasing N to 100. Obviously, since N=4 will ensure significantly low memory requirements, it has been chosen as the best trade-off.

C++ routines of the MBSCIG and MBSCIGA algorithms have been used to evaluate their segmentation quality. To this purpose, their output sequences were compared to the ground truths available within the chosen datasets in terms of PCC, PCB, PCF, Sm and F1.

As in every BS algorithm, threshold and parameter values have been chosen (see Table 4.11). This choice is usually performed by maximizing the accuracy for a given set of benchmark sequences. Experimental tests demonstrated that Tgc=Thc=2 allow good results in the foreground classification process to be achieved, whereas lower threshold values make the model more sensitive to sudden changes in the background.

The experimental tuning process has also shown that the higher (lower) Tg and Th values, the higher false negative (positive) pixels. Then, averaging results on benchmarks depicted in Table 4.10, Tg=25 and Th=20 have been found as values that ensure the best accuracy. Finally, the parameters ρB and ρF have been set to 0.95 and 0.04, respectively, taking into account that experiments have demonstrated that lower (higher) values of ρB (ρF), introduce over-smoothing effects. The BS algorithms presented in [139], [61], [137], [45], [126], [5], [113] and [75] were selected for purposes of comparison. They have been chosen not only because are among the most efficient approaches existing in Literature, but also because, similarly to the proposed MBSCIG algorithm, they perform the foreground detection pixel-by-pixel. Also for these algorithms, several parameters, such as the number of historical frames used to model the background, threshold values, the number of statistical distributions involved in the background model, etc., must be carefully set to guarantee the best accuracy to be achieved for a given set of benchmark scenarios. Table 4.11 provides basic information about all evaluated algorithms, including the adopted color models, parameters and thresholds with corresponding meaning and their experimentally selected values.

Samples of the segmented frames are collected in Figure 4.18 that also shows the ground truths referenced to evaluate their accuracies. From a qualitative analysis, it can be observed that some algorithms fail to identify a sufficient number of pixels of the foreground objects for specific sequences. As an example, this occurs in GMM, GMMHG and FBU for the Highway video, and in CB and FBU for the Lobby benchmark. Outputs are generally noisy for CB and CIHW algorithms. On average, FRA and the MBSCIG algorithms seem to produce the best results, even though somewhat blurred figures are produced by FRA for Highway and Office, and by the proposed MBSCIG algorithms for Lobby and Bootstrap.

C++ routines of the MBSCIG and MBSCIGA algorithms have been used to evaluate their segmentation quality. The quantitative accuracy analysis has been performed by examining all frames contained in the benchmark sequences, their output sequences were compared to

Table 4.11 Parameters used in the compared BS algorithms

Algorithm	Color Model	Parameters	Description
GMM [139]	RGB	N=200	Number of historical frames
		K=4	Number of distributions
		α=0.05	Learning rate
		Th=6.25	Threshold to extract the moving objects
CB [61]	RGB	N=300	Number of historical frames
		L=6.5	Number of codewords in the codebooks
		K=4	Number of distributions
		Maxc=10	Maximum number of codewords
		Minc=3	Minimum number of codewords
CIHW [137]	CIs H,	N=10	Number of historical frames
	Wx, Wy	Th=0.74	Threshold to extract the moving objects
		T=1.2	Portion of variation
GMMHG [45]	CIs H	N=200	Number of historical frames
	and GS	K=4	Number of distributions
		α=0.05	Learning rate
		Th=6.25	Threshold to extract the moving objects
SG [126]	Normalized	N=10	Number of historical frames
	U,V	αv=0.0003	Learning rate to update the variance
		α=0.05	Learning rate
SDM [5]	GS	M=4	Value applied at variance of $\Sigma - \Delta$
		K=3	Number of distributions
		Vinitial=0	Initial value of the variance
		Winitial=0	Initial value of the weights
		H1=4	Upper updating value
		H2=0	Lower updating value
FRA [113]	V of HSV	Th=30	Threshold to extract the moving objects
FBU [75]	GS	Thf=6	Fuzzy threshold to extract the moving objects
MBSCIG,	CI H and	N=4	Number of historical frames
MBSCIGA	GS	Tg=25	Threshold to extract the moving objects in GS channel
		Th=20	Threshold to extract the moving objects in H channel
		Tgc=Thc=2	Counter threshold
		ρB=0.95	Weight for the model update
		ρF=0.04	Weight for the model update

Fig. 4.18 Example of the processed image

the ground truths available within the chosen datasets in terms of PCC, PCB, PCF, Sm and F1. Results are collected in Table 4.12.

Preliminarily, we have to point out that the SG, SDM and FRA algorithms do not include a specific background initialization phase. Instead, they rely on the acquisition of a frame free from foreground objects. In all the benchmarks, the excepted Bootstrap frames fully free from foreground objects exist and were selected during the accuracy tests.

Table 4.12 Accuracy results in terms of PCC, PCCB, PCCF, F1 and SM

Algorithm	Quality	Lobby	Fountain	Highway	Bootstrap	Office
GMM [139]	PCC	97.06	93.68	87.18	93.44	
	PCCB	99.70	99.45	99.95	99.66	99.98
	PCCF	26.00	37.23	20.15	17.78	20.15
	F1	32.38	47.89	32.78	29.71	30.41
	SM	20.81	32.31	20.08	17.44	20.11
CB [61]	PCC	73.43	20.17	95.77	30.54	71.39
	PCCB	73.63	17.06	98.70	18.61	67.98
	PCCF	63.21	90.57	56.43	96.92	95.81
	F1	28.01	9.39	67.35	29.83	48.74
	SM	17.44	4.96	50.97	17.53	35.41
CIHW [137]	PCC	95.32	90.06	87.30	84.74	85.16
	PCCB	96.72	191.98	92.30	98.89	89.33
	PCCF	18.07	41.19	24.05	5.98	40.87
	F1	10.27	23.88	19.58	10.67	27.26
	SM	5.54	13.70	10.87	5.64	16.06
GMMHG [45]	PCC 98.26	96.31	92.96	84.74	93.49	
	PCCB	99.92	99.92	99.99	98.89	99.97
	PCCF	3.73	5.89	1.90	5.98	12.05
	F1	6.22	10.69	3.70	10.67	18.94
	SM	3.36	5.72	1.89	5.64	11.94
SG [126]	PCC	92.54	95.22	88.83	82.54	92.31
	PCCB	93.82	97.96	93.29	89.77	95.78
	PCCF	21.11	25.78	31.24	42.32	50.72
	F1	8.54	28.58	26.88	42.48	49.60
	SM	4.52	16.98	15.87	26.97	33.54
SDM [5]	PCC	98.38	96.89	95.48	87.84	94.02
	PCCB	99.56	99.37	99.45	98.85	99.58
	PCCF	32.72	35.34	42.93	26.60	32.04
	F1	39.15	44.07	56.47	40.00	43.74
	SM	25.26	29.46	39.46	25.00	29.96
FRA [113]	PCC	98.35	95.10	93.73	87.92	93.13
	PCCB	99.62	97.08	96.06	95.16	97.80
	PCCF	28.32	44.88	62.39	47.59	42.37
	F1	35.20	41.84	61.21	54.55	48.29
	SM	22.38	27.28	45.19	37.50	34.26
FBU [75]	PCC	97.06	95.70	90.34	79.87	90.96
	PCCB	98.70	99.40	96.61	90.93	97.18
	PCCF	4.35	2.91	7.34	18.32	17.84
	F1	4.73	4.68	9.83	21.71	24.62
	SM	2.46	2.42	5.28	12.18	14.45

Algorithm	Quality	Lobby	Fountain	Highway	Bootstrap	Office
MBSCIG	PCC	97.86	95.60	93.90	85.34	92.99
	PCCB	98.93	97.62	95.81	89.41	97.93
	PCCF	38.64	45.15	66.61	62.70	40.37
	F1	37.27	43.83	58.32	56.58	44.03
	SM	23.04	28.35	41.88	39.45	30.49
MBSCIGA	PCC	97.86	95.30	93.88	85.80	92.98
	PCCB	98.93	96.84	95.67	89.33	97.75
	PCCF	38.91	56.67	68.21	62.91	42.25
	F1	37.43	47.79	58.77	56.60	45.12
	SM	23.16	31.93	42.34	39.47	31.22

However, in real environments, such a condition is not always guaranteed. Therefore, an appropriate training-phase is mandatory [13]. For these reasons, we believe that the precision of the above algorithms could be somewhat overestimated.

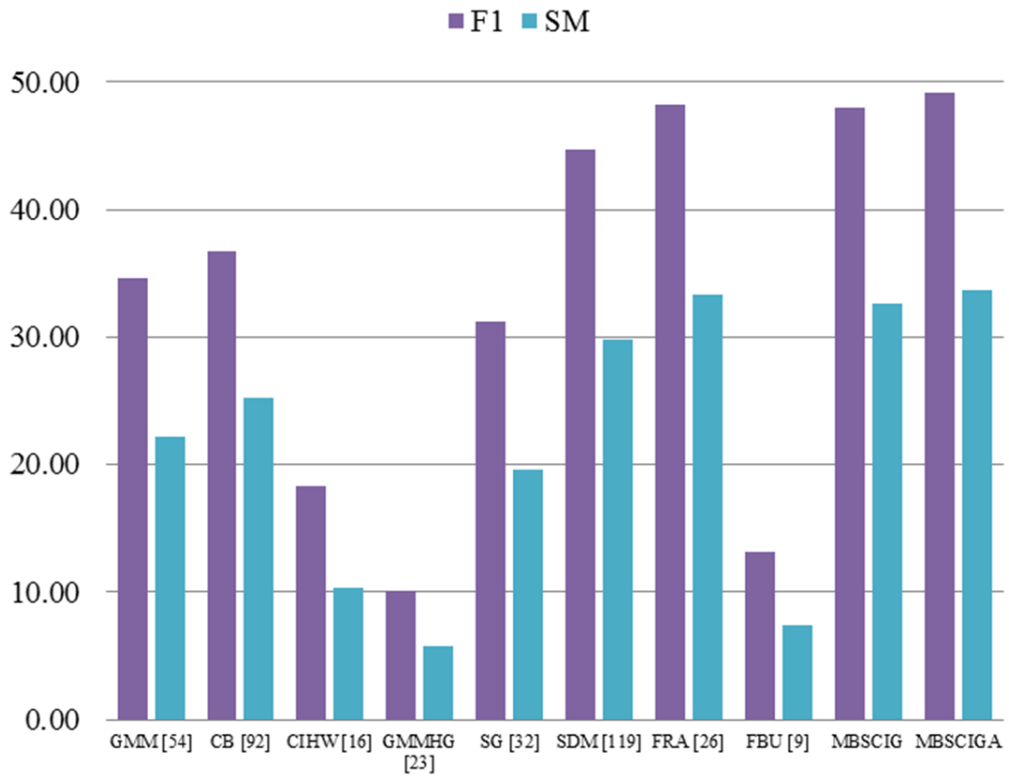


Fig. 4.19 Comparison results in terms of F1 and SM metrics

Averaging all F1 and SM measures in Table 4.12, it can be seen that the proposed MBSIGA algorithm reaches the highest accuracy score. As also shown in the histogram of Figure 4.19 , the closest competitor is represented by the FRA algorithm. The proposed algorithms are also characterized by very low sensitivity to the scene. In fact, by considering the standard deviation σ and the mean value μ computed for F1 and SM, they show σ/μ(F1) and σ/μ(SM) as low as 0.18 and 0.22, respectively. Only the SDM algorithm does better, but with lower average precision.

However, each algorithm of Table 4.12 has its own strengths and weakness. In fact, the SDM algorithm seems to be the most efficient to cope with the global illumination changes in the Lobby benchmark. GMM and MBSCIG are the most precise when repetitive background motions are present in the scene (Fountain). Even with the limit of the above mentioned ideal setting of the initial background, FRA and CB show the better reaction to more than one moving objects (Highway). Pixels classification when the background is obstructed by moving objects (Bootstrap) is better afforded by the MBSCIG algorithms. Whereas, SG and FRA excel when static and moving objects have the same color (Office). The nice property of the novel algorithms is that when they do not win the comparison with competitors, they have a precision level very close to the highest one for all the benchmark scenes, thus resulting in the highest F1 and SM averages.

The computational complexities of all the above algorithms is now evaluated in terms of the number of addition/subtraction (AS) and multiplication/division (MD) operations which are required for their main computational steps that can be summarized as: i) the pre-processing step, eventually needed to compute color transformations, Gray scale intensities and/or CIs from the RGB pixels; ii) the initialization of the background model; iii) the updating of the background model; iv) the foreground detection. We recall that the SG, SDM and FRA algorithms do not perform any initialization phase, but in practical applications they could require further operations to be executed for an appropriate training phase [13].

Table 4.13 Computational cost

Algorithm	PreProcessing	Initialization	Updating	Foreground Detection
	$AS+MD$	$AS+MD$	$AS+MD$	$AS+MD$
GMM [139]	0	6×k×N×Np+11×k×N×Np	6×k×Np+11×k×Np	(k-1)×Np+0
CB [61]	0	18×N×L×Np+17×N×L×Np	18×L×Np+17×L×Np	8×L×Np+3×L×Np
CIHW [137]	12×Np+24×Np	6×(N-1)×Np+3×(N+2)×Np	6×(N-1)×Np+3×(N+2)×Np	6×Np+3×Np
GMMHG [45]	12×Np+21×Np	21/2×N×Np×k +13/2×N×Np×k	21/2×Np×k +13/2×Np×k	2×(k-1)×Np+0
SG [126]	6×Np +11×Np	0	4×Np +5×Np	2×Np +4×Np
SDM [5]	2×Np +3×Np	0	17×Np +4×Np	1×K×Np+0
FRA [113]	6×Np +9×Np	0	2×Np +2×Np	1×Np+0
FBU [75]	2×Np +3×Np	0	4×Np +5×Np	17×Np +2×Np
MBSCIG	6×Np+10×Np	(4×N+2) ×Np+(4×N+4) ×Np	4×Np+4×Np	(4×N-2) ×Np+4×N×Np
MBSCIGA	6×Np+4×Np	(4×N+2) ×Np+(4×N+4) ×Np	4×Np+4×Np	(4×N-2) ×Np+4×N×Np

Table 4.13 collects computational complexities data, where Np represents the number of pixels within each frame. The meaning of all other parameters is reported in Table Table 4.11. From Table 4.13, it is clear that higher computational complexities does not always lead to higher accuracies. As an example, GMM [139], CB [61] and GMMHG [45] algorithms, though being among the most complex, show relatively low F1 and SM performances. On the contrary, the FRA algorithm [113] , which has the lowest number of operations achieves accuracies well higher than most of the compared algorithms, provided that initial background frame choice has been correctly performable. Undoubtedly, the ability of the MBSIG and MBSIGA algorithms to efficiently manage the above critical conditions also makes them suitable for much wider actual applications contexts.

4.5.3 Hardware architecture

The proposed algorithm could be implemented using DSP-, GPU- or FPGA-based hardware platforms. All these solutions are viable, and the most suitable platform will depend on the specific project constraints in terms of image size, throughput, latency, power consumption, cost, and development time. The FPGA technology has been selected as the target, since modern FPGA devices have frequencies compatible with RT applications and sufficient logic resources to support complex processing. Moreover, they can achieve computational speed higher than DSPs, power consumption lower than GPUs and guarantee high flexibility with relatively low development time [120], [60].

The proposed algorithm has been implemented by using an Avnet ZedBoard that contains an xc7z020 Xilinx Zynq FPGA chip. Xilinx Zynq System-On-Chip devices allow the design of a complex embedded system to be efficiently realized exploiting its embedded Processing System based on a two-cores Cortex A9. Such powerful processor is equipped with 32/32KB I/D Caches, 256KB on-chip RAM and several interfaces on AMBA buses.

The xc7z020 SOC has a FPGA-Fabric with 85K Logic Cells and 140 36Kb memory blocks available for specific user design.

Two different implementations of the proposed algorithm have been evaluated. In the former, the software code of the MBSCIGA algorithm is executed by one of the available Cortex A9 cores running at 800 MHz clock frequency. In the meantime, the second core takes care of the Linaro 14.04 "Trusty" Operating System. In such an implementation, the FPGA-Fabric is used just for realizing sub-systems for testing purpose and the DDR3 memory available on the Zed Board is exploited. Input video sequences with QQVGA and QVGA resolutions have been processed with frame rates up to 57 and 15.2 fps, respectively. Obviously, larger images sizes can be processed, but the low frame rates achieved would be inappropriate for RT applications.

The latter implementation is the design of a specific stand-alone architecture dedicated to the background subtraction and fully implemented in the FPGA-Fabric of the SOC. To this purpose, the novel background subtraction algorithm has been coded in VHDL with a limited use of IP cores, thus minimizing the efforts required to retarget, if needed, the design onto different hardware platforms. The top level architecture of the whole circuit is depicted in Figure 4.20 where the two computational channels above described are clearly visible. For each pixel in the generic input frame the modules RGB2H and RGB2G compute the color invariant H and the Gray scale data GS, respectively. Gray scale and H data of the four historical frames (GShi and Hhi, with i=1,...,4) and of the current background frame (Gm and Hm) are separately stored within the memory modules also depicted in Figure 4.20.

The circuits CheckAndUpdateH and CheckAndUpdateG compute the flags IsFh and IsFg that are then logically ANDed to produce the final output IsF, which is asserted to indicate that the processed pixel is recognized as a foreground pixel.

To implement the module RGB2G, the IP core color space converter available within the Xilinx design libraries has been exploited, whereas the circuit illustrated in Figure 4.20 has been purpose-designed for the module RGB2H. It can be seen that, equation (4.6), applied in the dashed box to compute $E\lambda$ and $E\lambda\lambda$, is easily hardware implemented through right-shifts, 2's complements and additions. Then, $E\lambda$ and $E\lambda\lambda$ are divided through a 21-stage pipelined IP core fixed-point divisor that computes at each clock cycle a novel 16-bit value of H, with an 8-bit fractional part.

Figure 4.22 shows in details the architecture of the CheckAndUpdateG module. Its computations can be divided within three main steps underlined in dashed boxes: Historical

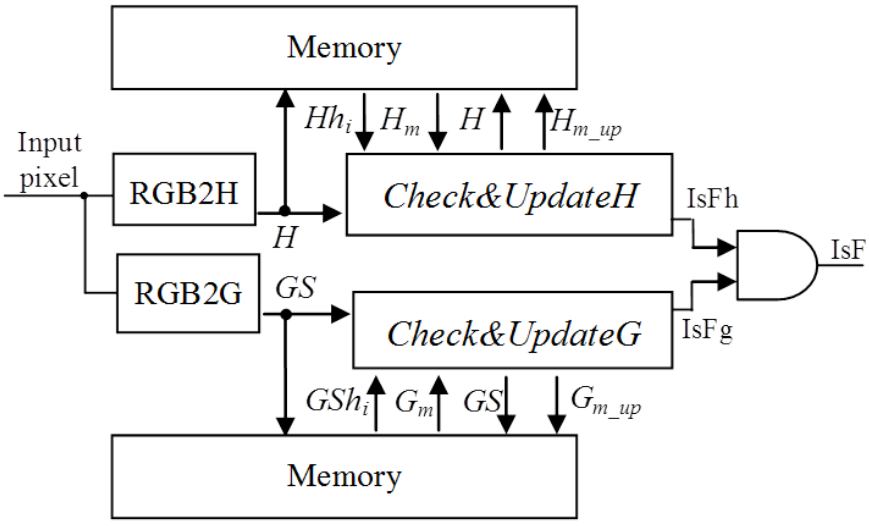


Fig. 4.20 The top-level hardware architecture

check, Current check and Update. The Historical check sub-module compares the Gray scale input pixel GS(x,y) with the corresponding pixels belonging to the N historical frames in parallel. Thus, N instances of the CC sub-circuit establish if the input pixel differs significantly from previously stored ones or not. The binary outputs obtained in this way are then added and compared to the threshold Tgc. The Current check sub-module compares the input pixel GS(x,y) with the corresponding pixel of the current background Gm(x,y), in a similar manner. Such information is then processed to detect whether the pixel is recognized as part of the background or not, and to compute its updated value Gmup(x,y) to store in place of Gm(x,y) for the next computation. The same architecture is utilized within the color invariant channel.

A different view of the processing system in which memory banks are explicated is illustrated in Figure 4.23. N+1 frame buffers (RAM1-RAM5) are used to store the four historical frames and the current background model. The Initialization System manages the storing of the first four frames in the buffers. Then, when the first pixel GS(x,y) of the (N+1)-th frame arrives the computation of the updated background pixel starts. After one clock cycle, all GShi(x,y) terms are consumed. Therefore, the oldest one (e.g. GSh1(x,y)) can be substituted with the value of the current processed pixel GS(x,y).

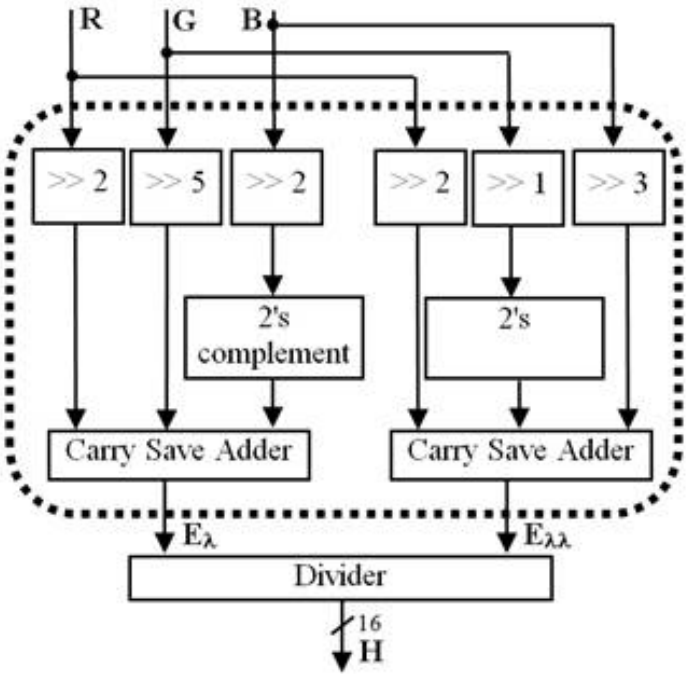


Fig. 4.21 The structure of the module RGB2H

Table 4.14 Post-place and route implementation results

SG		**SDM**		**FRA**		**BFPGA**	**MBSCIGA**				
Technology											
Spartan-3A		Spartan-3A		Spartan-3A		Virtex 6	Zynq-700				Virtex
Resolution											
QCIF	QVGA	QCIF	CIF	QCIF	CIF	FULL HD	QQVGA*	QVGA*	QQVGA	Full HD	Full HD
Processor											
0	0	0	0	0	0	0	Cortex A9	Cortex A9	0	0	0
External RAM											
0	0	0	0	0	0	54MB	512MB**	512MB**	0	32MB	32MB
BRAM18											
44	125	26	100	20	71	44	0	0	0	0	0
BRAM36											
0	0	0	0	0	0	57	0	0	75	6	12
DSP											
8	8	2	2	2	2	35	0	0	0	0	0
Slices LUTs											
415	868	329	386	418	715	16594	0	0	1868	20156	14141
Slices FFs											
288	297	174	184	175	181	22301	0	0	1376	15898	14648
Freq (MHz)											
100.6	84.3	92.4	72.2	81.2	63.3	124.4	800*	800*	154	154	129
Fps											
3969	1098	3645	711	3202	624	60	57	15.2	7520	7520	62

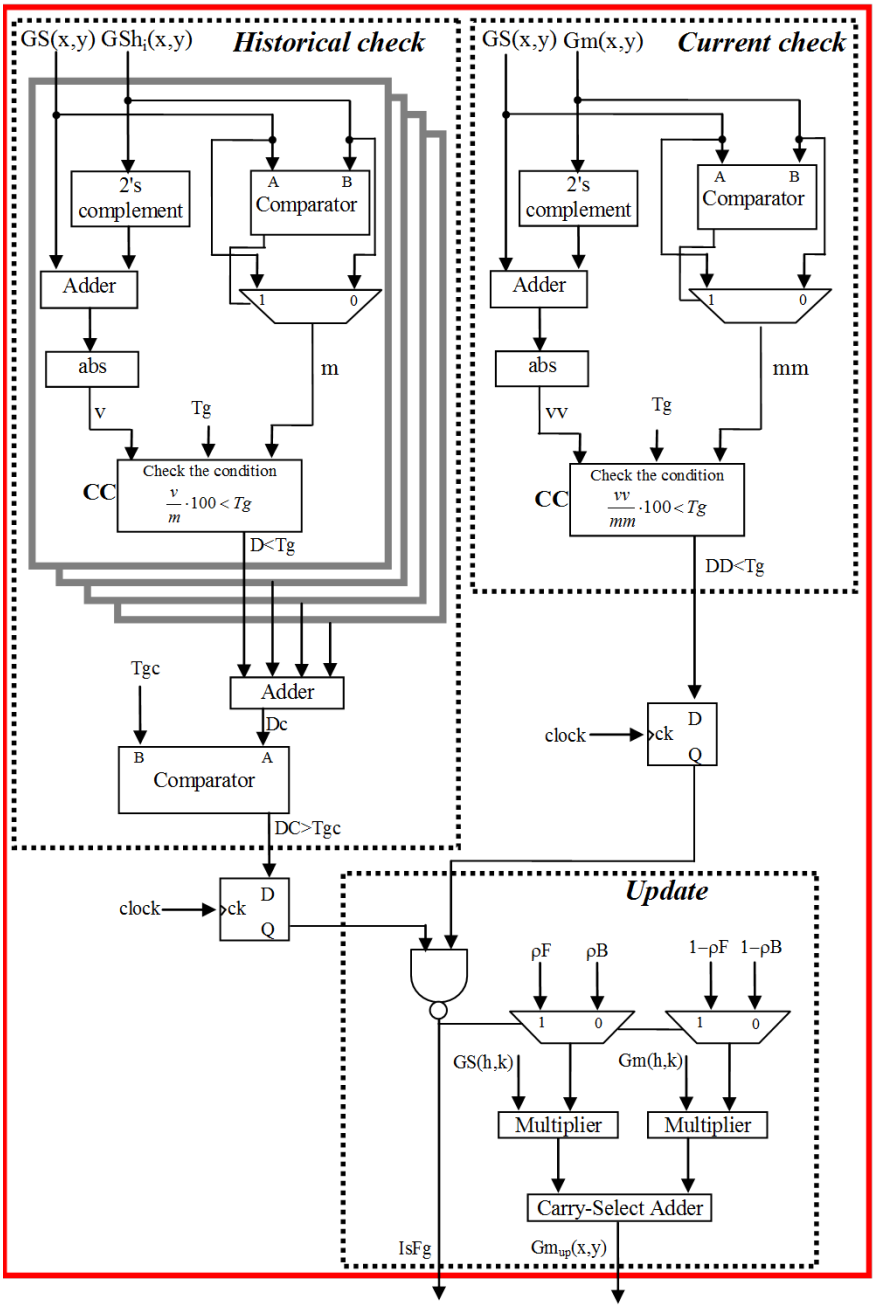


Fig. 4.22 The CheckAndUpdateG module

A rotating register maintains the information of which buffer detains the oldest frame, that is used to control the writing in the correct memory bank. After one more clock cycle, the updated background is computed and it is written in the RAM5 block.

When realized on the 85K Logic Cells xc7z020 FPGA chip to process RGB video sequences with a frame resolution of 128?160 pixels, the proposed system occupies 1868

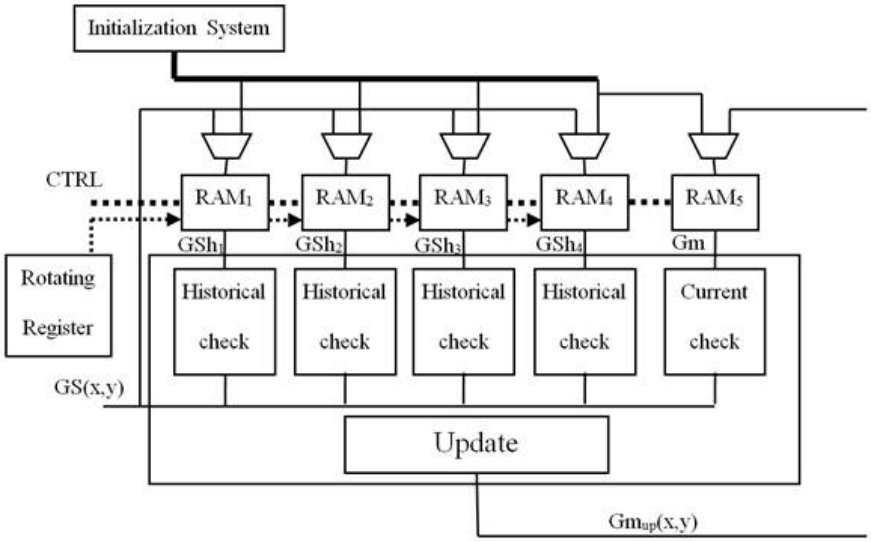


Fig. 4.23 The memory module

slice LUTs, 1376 slice Registers and 75 internal 36Kb Block RAMs. It has a latency of 81955 clock cycles and reaches a maximum clock frequency of 154MHz, thus producing an output frame each 0.13ms. At a parity of the frame resolution, the frame rate reached by the proposed hardware design is more than 132 times higher than the pure software execution above described.

Due to the limited amount of internal RAM resources, if higher resolutions are adopted, external memories have to be used. Following the same approach shown in Figure 4.23, the memory buffers could be realized by using the external DDR3 memory resources available on the ZedBoard. In such a case, the proposed system is made able to process Full HD RGB video sequences with a frame rate up to 74fps, and occupies 20156 Slice LUTs, 15898 Slice FFs, and 6 36Kb Block RAMs.

Post-place and route characteristics of all the designs implemented using the algorithm here proposed are summarized in Table 4.14 that also shows data related to the hardware designs presented in [16] for the SDM [5] , FRA [113], FBU [75] algorithms and Real-time background generation for high-definition video stream in FPGA device (BFPGA) [66].

The hardware system presented in [66] is also included in the comparison since it represents a good touchstone for the hardware design proposed here to process high resolutions video streams, although the original paper provides accuracy results obtained with a different set of benchmark sequences.

To guarantee a fair comparison also with [66], the novel algorithm has been implemented also within the XC6VLX 240T-1FF1156 FPGA device and referring to the ML605 board equipped with a 512MB DDR3 SRAM chip to be used as the external memory resource.

To process Full HD video sequences, apart 32MB of the external memory, the proposed circuit occupies 14141 Slice LUTs, 14648 Slice FFs, 12 36Kb Block RAMs and reaches a 129MHz running frequency, thus achieving a frame rate of about 62fps. The design implemented in [66] reaches a similar frame rate of 60fps, but it requires more than six times internal memory, 69 % more external memory, 17 % more slices and 52 % more flip-flops. Accuracy tests, performed on the same benchmarks used in [66], demonstrated that such an architecture reaches average F1 and SM only 2.3 % and 3.8 % higher than the proposed MBSCIGA. It is worth noting that, all the above resources counts take into account the memory controllers required to interface the background subtraction engine to the external memories.

4.6 Gaussian Mixture Model and MBSCIG evaluation for Real-Time Background Subtraction

The real-time BS algorithms demand a relatively low computational load and should be highly efficient to detect moving object in diverse environments at common video sequences rates. Therefore, with the aim to establish the efficiency given by GMM modifications [41] and MBSCIG [25], which are focused on high-performance for real-time segmentation, several experimental analysis have been performed and implemented in C++ with Open CV to evaluate them in conjunction with their optimized variations. In order to reduce efficiently the computational cost required for MBSCIG, two updating process variations have been proposed which are described in the following text. It is notably that while original techniques provide high robustness, herein, the experimental test determines that quite tunings achieve good performance with a scheme pixel-by-pixel in terms of accuracy, percentage of correct classification, computational load, and additionally are comparable with the version of GMM presented in [41].

4.6.1 GMM Background subtraction algorithm

The GMM known as statistical background modeling presented in [118] and its optimizations for hardware implementations presented in [41] are considered for software evaluation purposes in terms FNR, FPR, F1, PCC and computational complexity. The reported GMM algorithm leads the effectiveness in real-time applications with a good deal between constraints of low computational load and memory requirement, robustness and the ability to cope critical situations like illumination variation and added or removed objects. By this approach, some optimizations for hardware implementations are proposed in [41].

1. capture the current frame
2. For each pixel *It*(x,y) in the frame
3. …
4. if (*DD*<*T* and $\hat{\lambda}$>=*2*)
5. *IsFg*=0 //a background pixel is detected
6. $BG_{t+1} = (1-\propto) . I_t + \propto . BG_{t+1}$
7. else
8. *IsFg*=1 //a foreground pixel is detected
9. $FG_{t+1} = \beta . I_t + (1-\beta) . FG_{t+1}$
10. …
11. End for

a)

1. capture the current frame
2. For each pixel *It*(x,y) in the frame
3. …
4. if (*DD*<*T* and $\hat{\lambda}$>=*2*)
5. *IsFg*=0 //a background pixel is detected
6. $BG_{t+1} = (1-\propto) . I_t + \propto . BG_{t+1}$
7. else
8. *IsFg*=1 //a foreground pixel is detected
9. **if (DD > T)**
10. $\boldsymbol{FG_{t+1} = I_t}$
11. …
12. End for

b)

1. capture the current frame
2. For each pixel *It*(x,y) in the frame
3. …
4. if (*DD*<*T* and $\hat{\lambda}$>=*2*)
5. *IsFg*=0 //a background pixel is detected
6. $BG_{t+1} = (1-\propto) . I_t + \propto . BG_{t+1}$
7. else
8. *IsFg*=1 //a foreground pixel is detected
9. …
10. End for

c)

Fig. 4.24 The updating process of the MBSCIG: a) original version; b) MBSCIG v1; c) MBSCIG v2

- **GMM Optimized (GMM v1)**: The GMM algorithm [118] implemented in Open CV is able to work with one or three channels and its execution involves floating point operations, that is becoming a complex statistical model which provides good accuracy with a lot of computational cost and that also challenges its use on hardware implemen-

tations for real-time applications. Therefore, in order to reduce the computational cost, the author in [41], examines the algorithm [118], and proposes some optimizations, herein called GMM v1, which are based on the following characteristics:

- Handle the algorithm processing with video frames in Gray scale
- Use fixed value for mean (μ) and variance (σ) instead of floating value because of much greater computational power that is consumed by floating values. The floating point operation uses more internal circuitry and requires at least 32-bit data paths to manage two parts, one part of 24 bits for integer values (base of the real number) and the other one of 8-bit for the exponent.
- Establish the word length for each parameter so to reduce the error rate inserted with the diminution of number of bits.
- Define the number of mixture of Gaussian distributions to K=3 as suggested in [68]. Quantize the learning rates αw and $\alpha(k,t)$ as power of two

$$\alpha_w = 2^{n_w} \quad \alpha_{k,t} = 2^{n_{k,t}} \tag{4.7}$$

$$IF_{k,t} = (1/F_{k,t})^2 \tag{4.8}$$

$$IF_{k,t} = \sigma_{k,t}^2 . 2^{2(n_{k,t}-n_w)} \tag{4.9}$$

$$where\ n_{k,t} = log_2(\alpha_{k,t}) \ \text{ and } \ n_w = log_2(\alpha_w) \tag{4.10}$$

- **MBSCIG Optimized**: The algorithm MBSCIG was explored and tested through two different updating behaviors when a pixel is classified as foreground with the target to limit the number of operations by reducing the computational load as is showed in Figure 4.24. In order to incorporate gradual changes quickly in the background model, the first one (MBSCIG v1 Figure 4.24b), particularly updates the foreground pixels with the value of the current pixel when the percentage variation is higher than T. The second one (MBSCIG v2 Figure 4.24c), consists in discard the updating operation when a pixel corresponds to the set of moving objects.

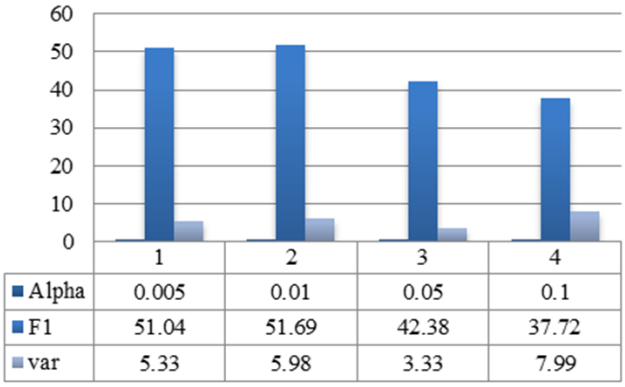


Fig. 4.25 Performance of learning rate in GMM

Table 4.15 Average of false positive and false negative rate

Algorithm	Lobby		WavingTree		Bootstrap		Highway		Office	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
GMM	0.64	1.02	0.28	18.14	2.08	14.33	0.32	5.47	0.26	7.06
GMM v1	0.71	1.07	10.98	25.17	4.80	14.37	1.45	5.87	2.80	4.71
MBSCIG	0.87	1.23	33.18	9.69	7.15	8.46	1.48	4.39	1.16	6.90
MBSCIG v1	1.07	1.19	32.88	7.85	6.54	6.70	2.16	3.16	2.42	3.16
MBSCIG v2	7.73	1.21	23.62	8.02	18.97	4.88	2.46	3.37	2.73	1.50

4.6.2 Experimental results

Since the learning rate (α) has a fundamental impact on the overall classification in algorithms based on GMM, the established value of α takes a significant interest to achieve high performance. Therefore, range of values between [0.01 to 0.05] is evaluated in [25]. In order to provide good classification, herein, the limits of the range [0.01 to 0.05] and values of 0.1 and 0.005 suggested by [25] and [30] have been measured by computing the F1 metric in five benchmark video sequences.

The quantitative evaluations are depicted in Figure 4.25, where it can be seen that the value of $\alpha = 0.05$ gives a lower variation (± 3.33) of F1 with respect to the average of the results and which means that 0.05 are well suited for all tested sequences and can be applied in both indoor and outdoor environments to achieve a good object identification. Therefore, in the next experiments α is established to 0.05.

The versions of GMM and MBSCIG were tested on I2R [113], Wallflower [92], 2012 and 2014 dataset [80]. Lobby is part of I2R dataset, which is defined by illumination changes and complex background, and contains twenty ground-truth images for evaluation

target. Wallflowers Dataset includes video sequences with dynamic motions and movement of background objects for which we have tested Waving Trees by considering its ground-truth provided. 2012 and 2014 Datasets contain outdoor and indoor environments respectively, where Bootstrapping is evaluated based on its one ground-truth, while Office and Highway video sequence have been tested by comparing the segmented results with respect to ten ground-truth given.

In order to establish the performance over the tested video sequences, the average of the numerical results achieved in the analyzed metrics is computed for each algorithm. Table 4.15 presents the percentage of FPR and FNR, where it can be seen that the GMM obtains the lowest FPR for all video sequences, following their optimized version. However the FNR have been reduced in Waving Tree, Bootstrap and Highway after that tuning the updating process in MBSCIG.

Figure 4.26 illustrates qualitative results for GMM, MBSCIG and its optimized versions, where it depicts that the original version of GMM (row b), works better than another algorithms in dynamics backgrounds with small movements. However, unfortunately the use of only three Gaussian Mixtures in both versions diminishes the overall accuracy in all experiments. On the other hand, the variants of MBSCIG algorithm perform much better than original MBSCIG, but all of them are still weak for the dynamic backgrounds.

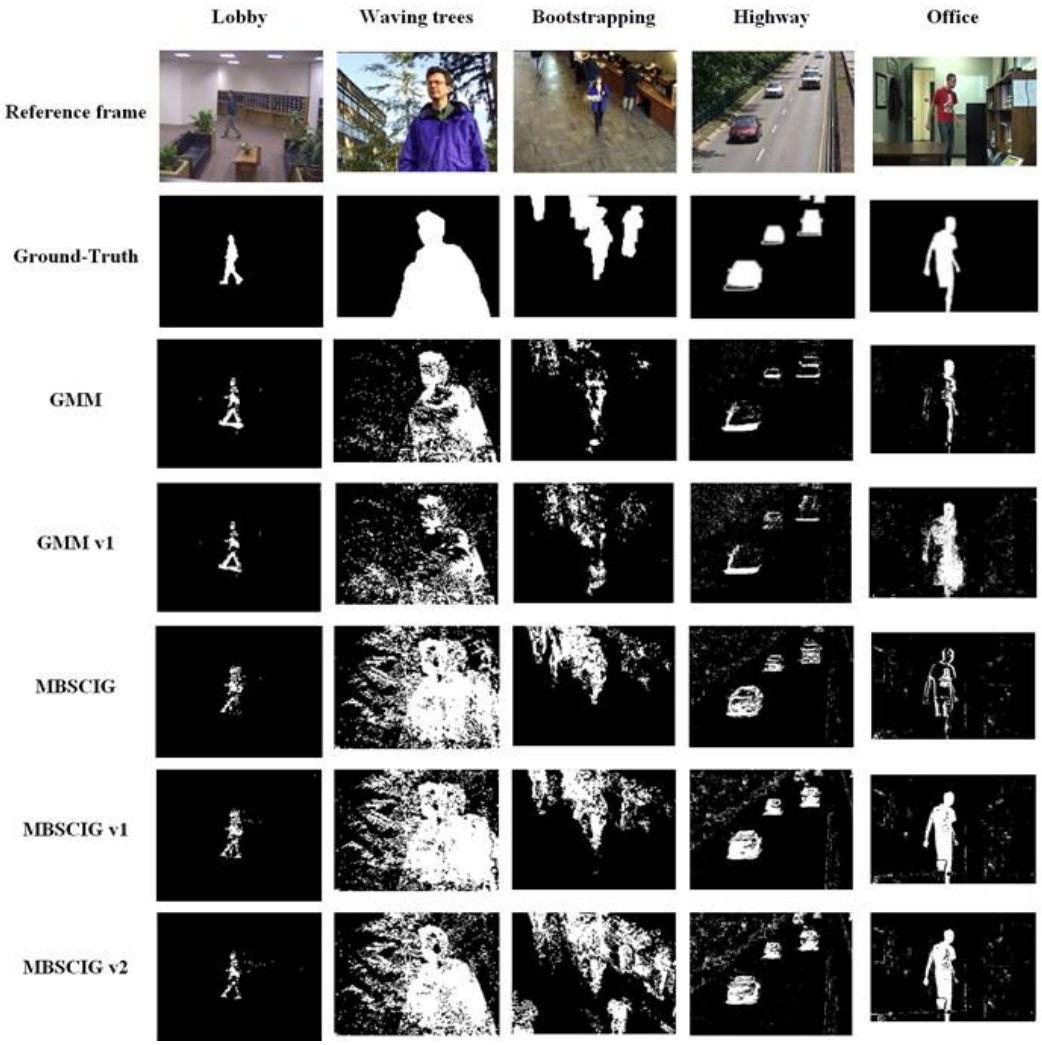


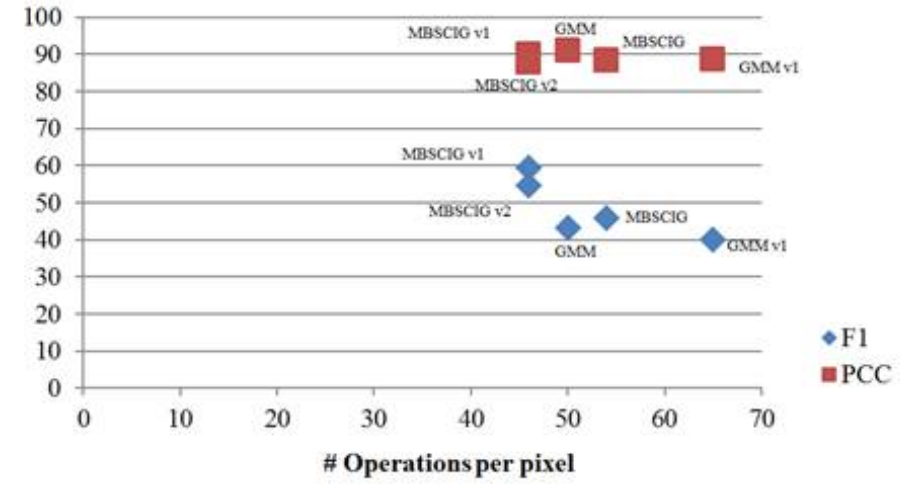
Fig. 4.26 Image segmented image

Table 4.16 Accuracy in terms of F1 and PCC

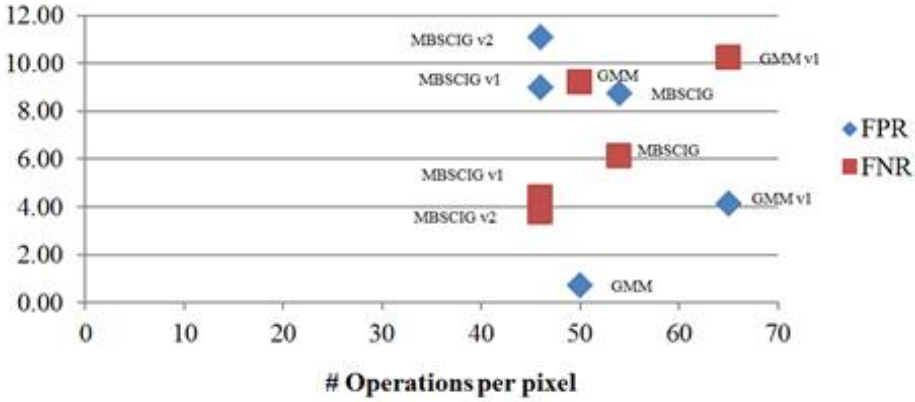
Algorithm	Lobby		WavingTree		Bootstrap		Highway		Office	
	F1	PCC	F1	PCC	F1	PCC	F1	PCC	F1	PCC
GMM	47.62	98.37	67.40	82.54	30.71	86.08	38.96	94.67	31.27	93.32
GMM v1	43.57	98.25	51.22	74.92	27.30	83.74	28.91	93.27	49.21	93.10
MBSCIG	33.58	97.93	61.66	70.27	54.93	86.77	48.36	94.60	30.49	92.63
MBSCIG v1	34.17	97.78	64.06	71.74	62.99	88.78	58.74	95.09	77.68	96.41
MBSCIG v2	20.18	91.21	69.55	78.05	52.31	79.78	55.66	94.63	75.03	96.00

Table 4.17 Computational Load

Algorithm	Color Model	Channels	Size	Background Model	Foreground Segmentation	Total
GMM	Gray scale	1	K=3	(27AS+21MD) x Np	2AS x Np	(29AS+21MD) x Np
GMMv1	Gray scale	1	K=3	(30AS+33MD) x Np	2AS x Np	(32AS+33MD) x Np
MBSCIG	Gray scale+H	2	N=4	(8AS+8MD) x Np	(18AS+20MD) x Np	(26AS + 28MD) x Np
MBSCIGv1	Gray scale+H	2	N=4	(4AS+4MD) x Np	(18AS+20MD) x Np	(22AS + 24MD) x Np
MBSCIGv2	Gray scale+H	2	N=4	(4AS+4MD) x Np	(18AS+20MD) x Np	(22AS + 24MD) x Np



a)



b)

Fig. 4.27 Accuracy vs. complexity

To present the quantitative accuracy of the tested methods, several experiments compare F1 and PCC. Their values are reported in Table 4.16, which confirms that the variations of MBSCIG are robustly capable of detecting moving objects. While GMM that is implemented is robust for environments with illumination changes and sudden small movements introduced in the background.

The computational load of the evaluated algorithms is presented in Table 4.17 for segmentation and modeling steps. As can be seen, the computational load in terms of Additions-Subtractions (AS) and Multiplications-Divisions, where Np is the number of pixels of each Frame, is related with the number of channels and the number of distributions or the number

of historical frames in the case of MBSCIG. It can be noted that the higher computational load of GMM does not promise the higher accuracy scores of F1 and PCC metrics, as is showed in Figure 4.27a. On the contrary, Figure 4.27b shows that the tuning of MBSCIG maintains lower values of FPR and FNR while reducing the computational load. From accuracy and computational complexity analysis, we can observe that the conjunction between H and Gray scale provides a soft and efficient method with a low computational load for BS.

4.7 Deep auto-encoder for Background Subtraction

Modeling the background for extracting the moving objects (objects of interest) has been widely applied [46]. For instance [45], a statistical algorithm based is focused on reducing misclassified objects exploiting GMM with advantages of combining two channels color invariant H and Gray scale. Taking into account the computational constrains for real-time applications, authors in [25] have proposed a novel method for the background subtraction to achieve low computational cost and high accuracy in real-time applications which computes the background model by using a limited number of historical frames. Thus, resulting more suitable for a real-time embedded implementation and being able to perform efficiently without any requirement of having clean background images (without moving objects) during initialization phase.

Beware that in real environments, it is not possible having clean backgrounds during initialization phase and to handle dynamic backgrounds. To overcome this, auto-encoders have emerged as a useful framework for unsupervised learning of internal representations [101]. Taking into account that extended denoising auto-encoders inject noise before the nonlinearity with the aim of reconstructing the input by extracting the noise and based on the assumption that foreground is processed as "noise" data and background as "clean" data. Authors in [128] proposed a background subtraction algorithm which is based on deep auto-encoder networks, where after the training network, the background images are extracted through a deep auto-encoder called BEN (Background Extraction Network) from input sequences (input). However, considering that after this process the extracted background image (output) is not completely “clean” (when foreground objects remain static for long periods of time), a preprocessing step is applied in order to improve the data of background image. Then a second auto-encoder called BLN (Background Learning Network) is performed to learn the dynamic background.

It is important to mention that when the foreground movements are fast, the output of the BEN provides a good representation of the background image. Therefore, below there is an evaluation of auto-encoder network architecture based on one and two auto-encoders.

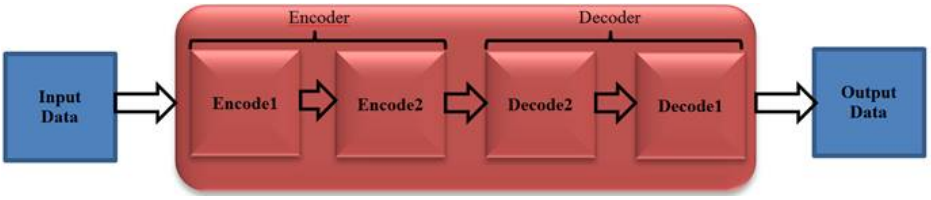


Fig. 4.28 Auto-encoder architecture for background subtraction

4.7.1 Auto-encoder Background Subtraction

The network architecture is based on mnist auto-encoder with Caffe [132], which is depicted in Figure 4.28. The same architecture has been used for the second auto-encoder when it is used.

The auto-encoder network is learned by back-propagation with cross-entropy cost function as is defined in (4.11) [128], [136].

$$\varepsilon(x) = -\Sigma_{k=1}^{N}(x_i log\widehat{x_i} + (1-x_i)log(1-\widehat{x_i})) \tag{4.11}$$

The bias are initially set to 0.1, while weights are randomly initialized through "xavier" algorithm. For training purposes, 10.000 iterations are performed with stochastic gradient descent for fast convergence, mini-batch size of 50 training samples, and a learning rate of 0.001.

At the end, an absolute difference is applied in order to obtain the final segmented image that contains the moving object.

$$\begin{aligned} B &= 0 \; if \; |output - I_t| < Th \\ B &= 1 \; \; if \; |output - I_t| > Th \end{aligned} \tag{4.12}$$

4.7.2 Experimental results

The evaluation has been performed on Lobby, Highway and Office as Gray scale input sequence. The first half of input sequence is used for the training network and the second one is used as a test set. Samples of the segmented frames with two auto-encoders are depicted in Figure 4.29, which also showed the ground truths reference to evaluate their accuracies.

Table 4.18 shows the quantitative evaluation that has been characterized by taking average of F1 in order to summarize accuracy achieved for each evaluated video sequence. It can be seen that the architecture with one auto-encoder is such that it identifies a sufficient number

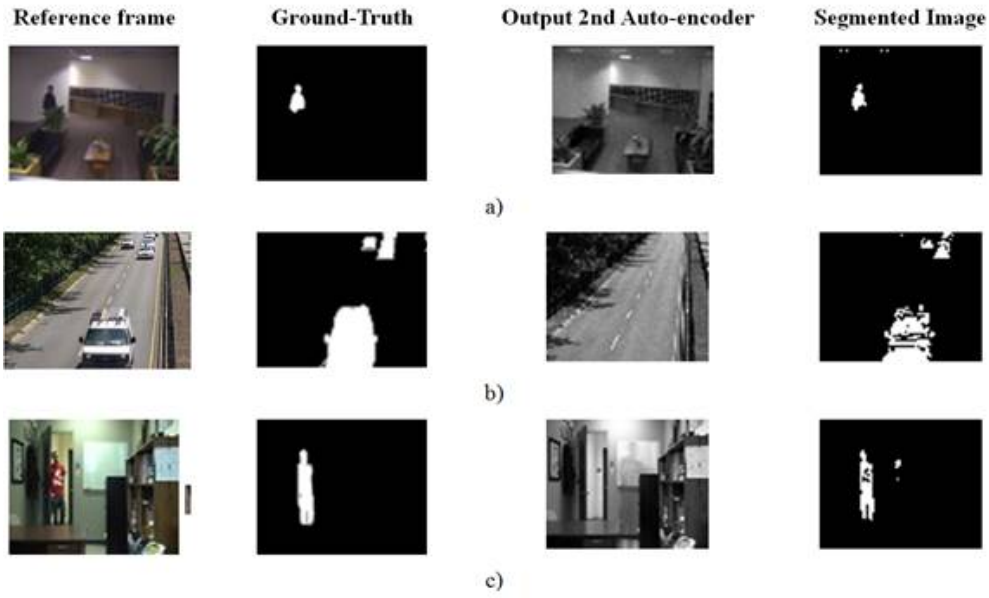


Fig. 4.29 Auto-encoder results for a) Lobby; b) Highway; and c) Office video sequences

Table 4.18 Accuracy in terms of F1

Video sequnce	F1	
	1 auto-encoder	2 auto-encoder
Lobby	88.40	49.10
Highway	85.70	88.00
Office	89.70	88.10

of pixels of the foreground objects for evaluated sequences. It can be successfully applied by taking into account that training network requires higher computational resources. Accuracy reduction in the architecture with two auto-encoder is due the threshold dependency applied in the preprocessing step to get a clean background before second auto-encoder called BLN according [128].

Chapter 5

Summary and Conclusions

5.1 Summary

This dissertation was focused on performing background subtraction algorithms for moving object detection. The proposed approaches analyze individual pixel information and color descriptor which takes the advantages of combining color invariant [42] with Gray scale to build the background model in order to reduce the misclassified pixels and be less sensitive to noise.

Color invariant study for background subtraction [46] is performed to evaluate the possibility and advantages of combining the complete set of color invariants (H, N, C and W) with Gray scale information. Several combinations refer for both indoor and outdoor experimental environments to demonstrate that the efficiency of extracting moving objects depends on the selected descriptors and combined through different logical operators.

GMM based on color invariants and Gray scale levels [45] is focused on reducing the sensitivity to noise ratio through the characterization of each frame by two channels (color Hx and Gray scale). Each pixel of each input frame is then modeled by using mixture of Gaussians represented in terms of the mean (μ), the weight (w) and the variance (σ). Thresholding is then separately applied to the channels to recognize both background and foreground pixels. Background pixels are updated based on a random process. The independent results obtained in this way are properly combined by using logical operator AND to generate the final binary image.

Embedded surveillance system using background subtraction and raspberry PI [22] is proposed as a low-cost solution for embedded video surveillance applications. The implemented algorithm reduces the number of historical frames with the use of two channels based on the invariant color H and the Gray scale information to achieve high performance and good quality also within the Raspberry-Pi platform.

Multimodal background subtraction for high performance embedded systems [25] is presented as basic model solution where RGB input frames are firstly processed to obtain the Gray scale and the color invariant H channels. It uses a limited number of historical frames thus, being reliable to achieve low computational cost and high accuracy in real-time embedded applications. The background model is updated by analyzing the percentage changes of current pixels with respect to the corresponding pixels within the modeled background and historical frames. The proposed approach is able to manage the presence of dynamic background and the absence of clean frames (frames free from foreground objects) without undermining the accuracy achieved. Additionally, different hardware designs have been implemented for several image resolutions within an Avnet ZedBoard containing an xc7z020 Zynq FPGA device have demonstrated that the proposed approach is suitable for the integration in low-cost high-definition embedded video systems and smart cameras.

A comparative evaluation of the original and optimized versions of the Gaussian Mixture Model (GMM) and the Multimodal Background Subtraction (MBSCIG) is performed in terms of computational complexity and numerical accuracy metric (F1 and PCC). Both real-time background subtraction algorithms were selected due to their low computational load and high efficiency to detect moving object in diverse environments at common video sequences rates. While original techniques provide high robustness, experimental test determines that quite tunings allow achieve good performance with a scheme pixel-by-pixel.

A deep networks based on auto-encoder architecture to extract moving objects from dynamic background has been evaluated in terms of F1 and qualitative results. This kind of architecture is based on the assumption that foreground is processed as "noise" data and background as "clean" data. Therefore, background frame is the expected output of the trained auto-encoder. However, the output with only one auto-encoder include some foreground pixels especially in video sequences where the moving objects remain static for long period of time. In such cases, pre-processing task is required before performing the following auto-encoder training.

5.2 Conclusion for Color Invariant study

Sets of CI combinations for BS have been empirically compared and some of them include Gray scale. The tests measured the performance of the combinations by referring to indoor and outdoor experimental environments by demonstrating that the Gray scale insertion mitigates the problem of misclassified pixel. H and Gray scale combination provides the highest performance with respect to other combinations with the benefit of including only two channels.

Gray color model leads to background with less noise. On the contrary, CIs increase the noise due to the transformational operations but, the combination with Gray color space allows in achieving high effectiveness in the BS. These characteristics can be efficiently introduced in the algorithms for the image segmentation.

5.3 Conclusion for Gaussian Mixture Model with color invariant and gray scale

A novel algorithm has been proposed for the background subtraction, which combines color invariant H and gray color on Gaussian mixture model. In this way, the problem of misclassified foreground objects is mitigated. Gray colors lead to background with less noise but include shadows. On the contrary, color invariants reduce the shadow pixels detected as foreground.

Although the algorithm reduce the misclassified pixels, a post-processing step could be useful to overcome the problems of apertures and discontinuities, thus improving the overall result. Tests and comparisons with codebook, GMM, and novel color invariants competitors have demonstrated that the proposed algorithm can reach upto higher quality in detecting foreground objects. A possible architecture suitable for hardware implement the novel algorithm has been presented and discussed.

5.4 Conclusion for Embedded surveillance system using BS and Raspberry Pi

The novel embedded system that implements an innovative background subtraction algorithm using Raspberry Pi board is focused on offers portability, low cost, and high accuracy in detecting moving objects in both indoor and outdoor environments.

To mitigate the noise problem added in the fusion of segmented images, the novel surveillance system applies some post-processing operations in order to improve the overall results and to get a pure segmented binary image.

A disadvantage with use of color invariants is that the color coefficients after transformation process from RGB to CI are floating point numbers, which reduces the performance while increasing hardware complexity. However, the numerical comparison and experimental tests reveal that the proposed system can reach up to higher quality in detecting objects of interest such as people.

The proposed embedded algorithm can be improved by changing the formulation in the algorithm from floating point to fixed point representation which will allow in improving the overall performance with respect to the proposed system.

5.5 Conclusion for Multimodal Background Subtraction for high performance embedded systems

The novel algorithm called MBSCIG exploits Gray scale information and color invariant H at the pixel-level and uses a very simple background model which is consisting of only four historical frames. The background model is unconventionally updated by analyzing the percentage changes of current pixels with respect to their counterparts within the historical frames.

An approximated version of the novel algorithm, named MBSCIGA, has also been proposed having an efficient hardware implementation as the target. In this version, complex operations are avoided to reduce logic and memory resource requirements and computational time.

Referring to several testbench video sequences, the achieved accuracy has been analyzed in terms of percentage of correctly classified background and foreground pixels, as well as in terms of the F1 and Similarity metrics that evaluate the overall accuracy of the computed segmented images. Obtained results demonstrated not only that the introduced approximations do not compromise the achieved accuracy but also that the novel algorithm efficiently trade-off the strength of several state of the art competitors.

With the main objective of demonstrating that the novel algorithm is suitable for the integration within low-cost embedded video systems and smart cameras oriented to real-time applications, several hardware implementations have been characterized for different images sizes. Reconfigurable FPGA devices have been selected as the target hardware platform, but also ASIC-, DSP- and GPU-based implementations can be easily carried out. Moreover, since the VHDL has been exploited with a limited use of specific IP cores, the proposed design can be retarget to different platforms with reduced efforts.

When realized within an xc7z020 FPGA chip, the proposed system can process Full HD (1920x1080) RGB video sequences with frame rates up to 74fps, occupying, apart 32MB of external RAM, only 38%, 19% and 4% of the Slices LUTs, the Slices FFs and the 36Kb Block RAMs, respectively, available within the chip.

5.6 Conclusion for Gaussian Mixture Model and MBSCIG evaluation for Real-Time Background Subtraction

Two efficient real-time approaches for Background Subtraction have been tested that are based on accuracy metrics in terms of FPR, FNR and F1. They have demostrated that the efficiency is very close between GMM implemented in OpenCV and MBSCIG with their variations. However, considering the high robustness as the convergence between a good effectiveness with a low computational cost, MBSCIG and their variations are established as affordable for real-time applications and particularly suitable on hardware platforms with on-board memory and limited computational resources and FPGA-based hardware accelerators.

5.7 Conclusion for Deep auto-encoder for Background Subtraction

Moving objects are extracted in a binary image through absolute difference after applying an architecture based on deep auto-encoder network instead of build complex Background Subtraction algorithms to cope with dynamic scenes. Experimental results measured by average of F1 for each evaluated input sequence show that an architecture based on one auto-encoder provides higher performance than one based on two auto-encoders particularly for scenes with objects of fast movements.

References

[1] (2016). Changedetection.net (cdnet) a video database for testing change detection algorithms.

[2] (2016). Raspberry pi.

[3] (2016). Techtarget-whatis.com.

[4] Abramovich, Y. A., Aliprantis, C. D., and Burkinshaw, O. (1995). Another characterization of the invariant subspace problem. *Operator Theory in Function Spaces and Banach Lattices.* The A.C. Zaanen Anniversary Volume*, Operator Theory: Advances and Applications*, 75:15–31. Birkhäuser Verlag.

[5] Abutaleb, M., Hamdy, A., Abuelwafa, M., and Saad, E. (2009). Fpga-based object-extraction based on multimodal σ-δ background estimation. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–7. IEEE.

[6] Ancey, C., Coussot, P., and Evesque, P. (1996). Examination of the possibility of a fluid-mechanics treatment of dense granular flows. *Mechanics of Cohesive-frictional Materials*, 1(4):385–403.

[7] Aupetit, B. (1991). *A Primer on Spectral Theory*. Springer-Verlag, New York.

[8] Barnich, O. and Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6):1709–1724.

[9] Boninsegna, M. and Bozzoli, A. (2000). A tunable algorithm to update a reference image. *Signal Processing: Image Communication*, 16(4):353–365.

[10] Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11:31–66.

[11] Braham, M. and Van Droogenbroeck, M. (2016). Deep background subtraction with scene-specific convolutional neural networks. In *International Conference on Systems, Signals and Image Processing, Bratislava 23-25 May 2016*. IEEE.

[12] Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer.

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

[14] Butler, D., Sridharan, S., and Bove, V. J. (2003). Real-time adaptive background segmentation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–349. IEEE.

[15] Butler, D. E., Bove Jr, V. M., and Sridharan, S. (2005). Real-time adaptive foreground/background segmentation. *EURASIP Journal on Advances in Signal Processing*, 2005(14):1–13.

[16] Calvo-Gallego, E., Brox, P., and Sánchez-Solano, S. (2014). Low-cost dedicated hardware ip modules for background subtraction in embedded vision systems. *Journal of Real-Time Image Processing*, pages 1–15.

[17] Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P. (2003). Free-viewpoint video of human actors. In *ACM transactions on graphics (TOG)*, volume 22, pages 569–577. ACM.

[18] Chen, R.-C., Dreossi, D., Mancini, L., Menk, R., Rigon, L., Xiao, T.-Q., and Longo, R. (2012). Pitre: software for phase-sensitive x-ray image processing and tomography reconstruction. *Journal of synchrotron radiation*, 19(5):836–845.

[19] Chen, Z. and Ellis, T. (2014). A self-adaptive gaussian mixture model. *Computer Vision and Image Understanding*, 122:35–46.

[20] Cheung, S.-C. S. and Kamath, C. (2005). Robust background subtraction with foreground validation for urban traffic video. *EURASIP Journal on Advances in Signal Processing*, 2005(14):1–11.

[21] Chiranjeevi, P. and Sengupta, S. (2014). Detection of moving objects using multichannel kernel fuzzy correlogram based background subtraction. *IEEE transactions on cybernetics*, 44(6):870–881.

[22] Cocorullo, G., Corsonello, P., Frustaci, F., Guachi, L., and Perri, S. (2015). Embedded surveillance system using background subtraction and raspberry pi. In *2015 AEIT International Annual Conference (AEIT)*, pages 1–5. IEEE.

[23] Cong, D. N. T., Khoudour, L., Achard, C., and Flancquart, A. (2011). Adaptive model for object detection in noisy and fast-varying environment. In *International Conference on Image Analysis and Processing*, pages 68–77. Springer.

[24] Conway, J. B. (1990). *A Course in Functional Analysis*. Springer-Verlag, New York, second edition.

[25] Corsonello, P., Cocorullo, G., Frustaci, F., Guachi, L., and Perri, S. (2016). Multimodal Background Subtraction for high performance embedded systems. *Journal of Real-Time Image Processing*.

[26] Cristani, M., Bicego, M., and Murino, V. (2003). Multi-level background initialization using hidden markov models. In *First ACM SIGMM international workshop on Video surveillance*, pages 11–20. ACM.

[27] Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1337–1342.

[28] Culibrk, D., Marques, O., Socek, D., Kalva, H., and Furht, B. (2007). Neural network approach to background modeling for video object segmentation. *IEEE Transactions on Neural Networks*, 18(6):1614–1627.

[29] Deng, R., Yang, D., Liu, X., and Liu, S. (2014). A background subtraction algorithm based on pixel state. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 251–254. ACM.

[30] Dhar, J., Kurele, R., Arora, S., and Sinha, S. (2015a). Background subtraction in surveillance systems-a neural fuzzy approach. *International Journal of Imaging and Robotics™*, 15(4):29–42.

[31] Dhar, J., Kurele, R., Arora, S., and Sinha, S. (2015b). Background subtraction in surveillance systems-a neural fuzzy approach. *International Journal of Imaging and Robotics™*, 15(4):29–42.

[32] Ding, J., Li, M., Huang, K., and Tan, T. (2010). Modeling complex scenes for accurate moving objects segmentation. In *Asian Conference on Computer Vision*, pages 82–94. Springer.

[33] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163.

[34] Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer.

[35] Elgammal, A., Harwood, D., and Davis, L. (2015). Lecture Notes in Computer Science Non-parametric Model for Background Subtraction. (November).

[36] Elhabian, S. Y., El-Sayed, K. M., and Ahmed, S. H. (2008). Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32–54.

[37] Farcas, D., Marghes, C., and Bouwmans, T. (2012). Background subtraction via incremental maximum margin criterion: a discriminative subspace approach. *Machine Vision and Applications*, 23(6):1083–1101.

[38] Feng, Y., Luo, S., Tian, Y., Deng, S., and Zheng, H. (2014). Comprehensive analysis and evaluation of background subtraction algorithms for surveillance video. *Sensors & Transducers*, 177(8):163.

[39] Fernandez-Sanchez, E. J., Rubio, L., Diaz, J., and Ros, E. (2014). Background subtraction model based on color and depth cues. *Machine vision and applications*, 25(5):1211–1225.

[40] Florack, L. M., ter Haar Romeny, B. M., Koenderink, J. J., and Viergever, M. A. (1992). Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388.

[41] Genovese, M. and Napoli, E. (2014). Asic and fpga implementation of the gaussian mixture model algorithm for real-time segmentation of high definition video. *IEEE transactions on very large scale integration (VLSI) systems*, 22(3):537–547.

[42] Geusebroek, J.-M., Van den Boomgaard, R., Smeulders, A. W. M., and Geerts, H. (2001). Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350.

[43] Gevers, T. and Smeulders, A. W. (1999). Color-based object recognition. *Pattern recognition*, 32(3):453–464.

[44] Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., and Ishwar, P. (2012). Changedetection. net: A new change detection benchmark dataset. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE.

[45] Guachi, L., Cocorullo, G., Corsonello, P., Frustaci, F., and Perri, S. (2014). A novel background subtraction method based on color invariants and grayscale levels. In *2014 International Carnahan Conference on Security Technology (ICCST)*, pages 1–5. IEEE.

[46] Guachi, L., Cocorullo, G., Corsonello, P., Frustaci, F., and Perri, S. (2016). Color Invariant Study for Background Subtraction. In *CENICS 2016 : The Ninth International Conference on Advances in Circuits, Electronics and Micro-electronics*.

[47] Guo, J.-M., Liu, Y.-F., Hsia, C.-H., Shih, M.-H., and Hsu, C.-S. (2011). Hierarchical method for foreground detection using codebook model. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(6):804–815.

[48] Gupte, S., Masoud, O., Martin, R. F., and Papanikolopoulos, N. P. (2002). Detection and classification of vehicles. *IEEE Transactions on intelligent transportation systems*, 3(1):37–47.

[49] Haritaoglu, I., Harwood, D., and Davis, L. S. (1998). W4s: A real-time system for detecting and tracking people in 2 1/2d. In *European Conference on computer vision*, pages 877–892. Springer.

[50] Horprasert, T., Harwood, D., and Davis, L. S. (2000). A robust background subtraction and shadow detection. In *Proc. ACCV*, pages 983–988.

[51] Hsiao, H.-H. and Leou, J.-J. (2013). Background initialization and foreground segmentation for bootstrapping video sequences. *EURASIP Journal on Image and Video Processing*, 2013(1):1–19.

[52] Javed, O., Shafique, K., and Shah, M. (2002). A hierarchical approach to robust background subtraction using color and gradient information. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 22–27. IEEE.

[53] Javed, S., Oh, S. H., Bouwmans, T., and Jung, S. K. (2015). Robust background subtraction to global illumination changes via multiple features-based online robust principal components analysis with markov random field. *Journal of Electronic Imaging*, 24(4):043011–043011.

[54] Javed, S., Oh, S. H., Heo, J., and Jung, S. K. (2014). Robust background subtraction via online robust pca using image decomposition. In *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems*, pages 105–110. ACM.

[55] Jayamanne, D. J., Samarawickrama, J., and Rodrigo, R. (2013). Appearance based tracking with background subtraction. In *Computer Science & Education (ICCSE), 2013 8th International Conference on*, pages 643–649. IEEE.

[56] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

[57] Joshi, K. A. and Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48.

[58] Karmann, K.-P., Brandt, A. V., and Gerl, R. (1990). Moving object segmentation based on adaptive reference images. In *5. European Signal Processing Conference.*, volume 2, pages 951–954.

[59] Kentaro Toyama, John Krumm, Barry Brumitt, B. M. (2016). Test images for wallflower paper.

[60] Kestur, S., Davis, J. D., and Williams, O. (2010). Blas comparison on fpga, cpu and gpu. In *2010 IEEE computer society annual symposium on VLSI*, pages 288–293. IEEE.

[61] Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2004). Background modeling and subtraction by codebook construction. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 5, pages 3061–3064. IEEE.

[62] Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2005). Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185.

[63] Kleeman, L. (1996). Understanding and applying kalman filtering. In *Proceedings of the Second Workshop on Perceptive Systems, Curtin University of Technology, Perth Western Australia (25-26 January 1996)*.

[64] Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., and Russell, S. (1994a). Towards robust automatic traffic scene analysis in real-time. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 126–131. IEEE.

[65] Koller, D., Weber, J., and Malik, J. (1994b). Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision*, pages 189–196. Springer.

[66] Kryjak, T., Komorkiewicz, M., and Gorgon, M. (2014). Real-time background generation and foreground object segmentation for high-definition colour video stream in fpga device. *Journal of Real-Time Image Processing*, 9(1):61–77.

[67] Kumar, K. and Agarwal, S. (2013). An efficient hierarchical approach for background subtraction and shadow removal using adaptive gmm and color discrimination. *International Journal of Computer Applications*, 75(12).

[68] Kumar, M. S. and Venugopal, S. (2015). Robust technique for background subtraction using k-means algorithm.

[69] Kumar, P., Rout, D. K., Kumar, A., Verma, M., and Kumar, D. (2015). Detection of video objects in dynamic scene using local binary pattern subtraction method. In *Intelligent Computing, Communication and Devices*, pages 385–391. Springer.

[70] Kumar, S. and Yadav, J. S. (2016). Video object extraction and its tracking using background subtraction in complex environments. *Perspectives in Science*.

[71] Le, Q. V. et al. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks.

[72] Lee, J. and Park, M. (2012). An adaptive background subtraction method based on kernel density estimation. *Sensors*, 12(9):12279–12300.

[73] Li, L., Huang, W., Gu, I. Y.-H., and Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472.

[74] Li, P., Page, T., Luo, G., Zhang, W., Wang, P., Zhang, P., Maass, P., Jiang, M., and Cong, J. (2014). Fpga acceleration for simultaneous medical image reconstruction and segmentation. In *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on*, pages 172–172. IEEE.

[75] Lijun, X. (2011). Moving object segmentation based on background subtraction and fuzzy inference. In *Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on*, pages 434–437. IEEE.

[76] Ljubič, J. I. and Macaev, V. I. (1965). On operators with a separable spectrum. *Amer. Math. Soc. Transl. (2)*, 47:89–129.

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

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

[79] Mahadevan, V. and Vasconcelos, N. (2008). Background subtraction in highly dynamic scenes. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6. IEEE.

[80] Mahapatra, A., Mishra, T. K., Sa, P. K., and Majhi, B. (2013). Background subtraction and human detection in outdoor videos using fuzzy logic. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–7. IEEE.

[81] Manzanera, A. and Richefeu, J. (2004). A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'04)*.

[82] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56.

[83] Melin, P. and Castillo, O. (2014). A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition. *Applied soft computing*, 21:568–577.

[84] Messelodi, S., Modena, C. M., Segata, N., and Zanin, M. (2005). A kalman filter based background updating algorithm robust to sharp illumination changes. In *International Conference on Image Analysis and Processing*, pages 163–170. Springer.

[85] Mittal, A. and Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302. IEEE.

[86] Mohamed, S. S., Tahir, N. M., and Adnan, R. (2010). Background modelling and background subtraction performance for object detection. In *Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on*, pages 1–6. IEEE.

[87] Monari, E. and Pasqual, C. (2007). Fusion of background estimation approaches for motion detection in non-static backgrounds. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 347–352. IEEE.

[88] Monnet, A., Mittal, A., Paragios, N., and Ramesh, V. (2003). Background modeling and subtraction of dynamic scenes. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1305–1312. IEEE.

[89] Murshed, M., Ramirez, A., and Chae, O. (2010). Statistical background modeling: an edge segment based moving object detection approach. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 300–306. IEEE.

[90] Network, T. (2016).

[91] Nguyen, V.-T., Vu, H., and Tran, T.-H. (2015). An efficient combination of rgb and depth for background subtraction. In *Some Current Advanced Researches on Information and Computer Science in Vietnam*, pages 49–63. Springer.

[92] Noldus, L. P., Spink, A. J., and Tegelenbosch, R. A. (2002). Computerised video tracking, movement analysis and behaviour recognition in insects. *Computers and Electronics in Agriculture*, 35(2):201–227.

[93] Ohya, J., Kurumisawa, J., Nakatsu, R., Ebihara, K., Iwasawa, S., Harwood, D., and Horprasert, T. (1999). Virtual metamorphosis. *IEEE Multimedia*, 6(2):29–39.

[94] Oliver, N. M., Rosario, B., and Pentland, A. P. (2000). A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):831–843.

[95] Panda, D. K. and Meher, S. (2016). Detection of moving objects using fuzzy color difference histogram based background subtraction. *IEEE Signal Processing Letters*, 23(1):45–49.

[96] Park, D., Zitnick, C. L., Ramanan, D., and Dollár, P. (2013). Exploring weak stabilization for motion feature extraction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2882–2889.

[97] Parks, D. H. and Fels, S. S. (2008). Evaluation of background subtraction algorithms with post-processing. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 192–199. IEEE.

[98] Parmar, M. P. and Pandi, M. G. S. (2015). Performance analysis and augmentation of k-means clustering, based approach for human detection in videos. In *International Journal of Engineering Development and Research*, volume 3. IJEDR.

[99] Petrosino, A. (2016). Deep Learning and Neural Networks E CASES INDUSTRIAL USE CASES INDUSTRIAL USE CASES INDUSTRIAL Machine learning.

[100] Piccardi, M. (2004). Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE.

[101] Poole, B., Sohl-Dickstein, J., and Ganguli, S. (2014). Analyzing noise in autoencoders and deep networks. *arXiv preprint arXiv:1406.1831*.

[102] Power, P. W. and Schoonees, J. A. (2002). Understanding background mixture models for foreground segmentation. In *Proceedings image and vision computing New Zealand*, volume 2002, pages 10–11.

[103] Radford, D. (1995). Background subtraction from in-beam hpge coincidence data sets. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 361(1):306–316.

[104] Read, C. J. (1985). A solution to the invariant subspace problem on the space l_1. *Bull. London Math. Soc.*, 17:305–317.

[105] Reddy, V., Sanderson, C., and Lovell, B. C. (2013). Improved foreground detection via block-based classifier cascade with probabilistic decision integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):83–93.

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

[107] Sanin, A., Sanderson, C., and Lovell, B. C. (2012). Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition*, 45(4):1684–1695.

[108] Sankari, M. and Meena, C. (2011). Estimation of dynamic background and object detection in noisy visual surveillance. *International Journal of Advanced Computer Sciences and Applications*, 6(2).

[109] Sebastian, P., Voon, Y. V., and Comley, R. (2010). Colour space effect on tracking in video surveillance. *International Journal on Electrical Engineering and Informatics*, 2(4):298.

[110] Sen-Ching, S. C. and Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. In *Electronic Imaging 2004*, pages 881–892. International Society for Optics and Photonics.

[111] Sepulveda, J. and Velastın, S. A. (2015). F1 score assesment of gaussian mixture background subtraction algorithms using the muhavi dataset. In *6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)*, pages 1–6. IET.

[112] Shah, M., Deng, J. D., and Woodford, B. J. (2015). A self-adaptive codebook (sacb) model for real-time background subtraction. *Image and Vision Computing*, 38:52–64.

[113] Sigari, M., Mozayani, N., and Pourreza, H. (2008). Fuzzy running average and fuzzy background subtraction: concepts and application. *International Journal of Computer Science and Network Security*, 8(2):138–143.

[114] Silva, C., Bouwmans, T., and Frélicot, C. (2015). 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 2015*.

[115] Sivanantham, S., Paul, N. N., and Iyer, R. S. (2016). Object tracking algorithm implementation for security applications. *Far East Journal of Electronics and Communications*, 16(1):1.

[116] Sobral, A. and Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21.

[117] St-Charles, P.-L. and Bilodeau, G.-A. (2014). Improving background subtraction using local binary similarity patterns. In *IEEE Winter Conference on Applications of Computer Vision*, pages 509–515. IEEE.

[118] Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE.

[119] Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):747–757.

[120] Tessier, R., Pocek, K., and DeHon, A. (2015). Reconfigurable computing architectures. *Proceedings of the IEEE*, 103(3):332–354.

[121] Tian, Y.-L., Lu, M., and Hampapur, A. (2005). Robust and efficient foreground analysis for real-time video surveillance. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1182–1187. IEEE.

[122] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE.

[123] Varcheie, P. D. Z., Sills-Lavoie, M., and Bilodeau, G.-A. (2010). A multiscale region-based motion detection and background subtraction algorithm. *Sensors*, 10(2):1041–1061.

[124] Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., and Ishwar, P. (2014). Cdnet 2014: an expanded change detection benchmark dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 387–394.

[125] Wikipedia (2016). real-time application (rta).

[126] Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785.

[127] Wu, H., Liu, N., Luo, X., Su, J., and Chen, L. (2014). Real-time background subtraction-based video surveillance of people by integrating local texture patterns. *Signal, Image and Video Processing*, 8(4):665–676.

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

[129] Xue, G., Sun, J., and Song, L. (2012). Background subtraction based on phase feature and distance transform. *Pattern Recognition Letters*, 33(12):1601–1613.

[130] Yang, M., Zhang, L., Shiu, S. C., and Zhang, D. (2013). Gabor feature based robust representation and classification for face recognition with gabor occlusion dictionary. *Pattern Recognition*, 46(7):1865–1878.

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

[132] Yeager, Luke (2016). Training lenet on mnist with caffe.

[133] Zhang, S., Yao, H., and Liu, S. (2008). Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *2008 15th IEEE International Conference on Image Processing*, pages 1556–1559. IEEE.

[Zhehuo] Zhehuo, W. Chapter 3 Wiener Filters. In *Adaptive Signal Processing Course*, chapter 3.

[135] Zhong, J. and Sclaroff, S. (2003). Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 44–50. IEEE.

[136] Zhou, G., Sohn, K., and Lee, H. (2012). Online incremental feature learning with denoising autoencoders. *Ann Arbor*, 1001:48109.

[137] Zhou, H., Chen, Y., and Feng, R. (2013a). A novel background subtraction method based on color invariants. *Computer Vision and Image Understanding*, 117(11):1589–1597.

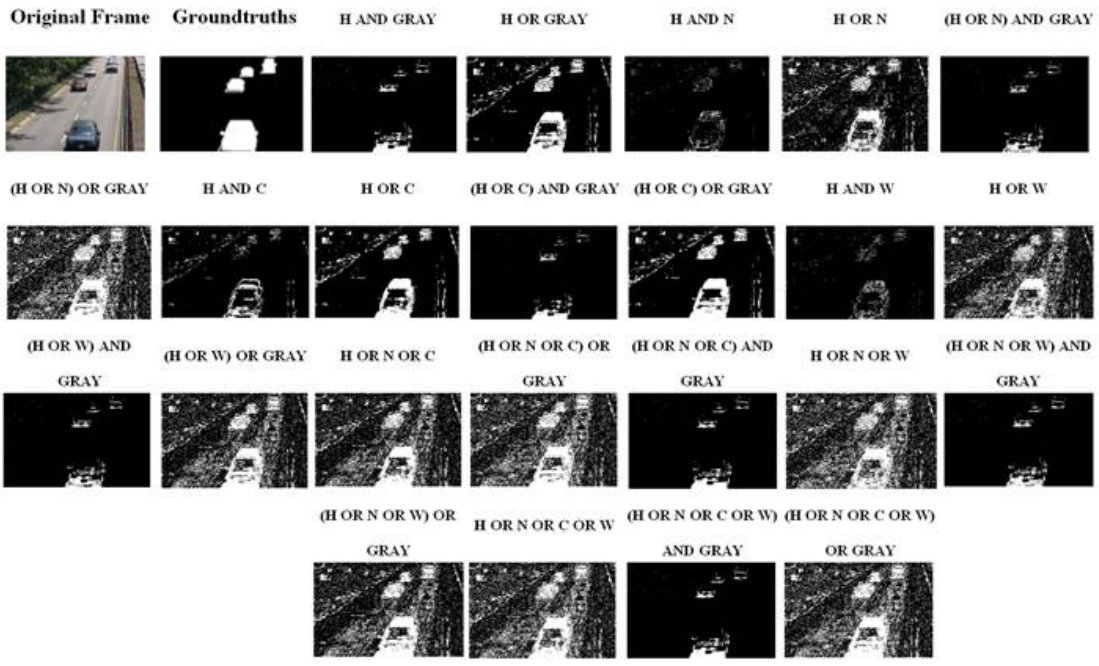
[138] Zhou, X., Yang, C., and Yu, W. (2013b). Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610.

[139] Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE.

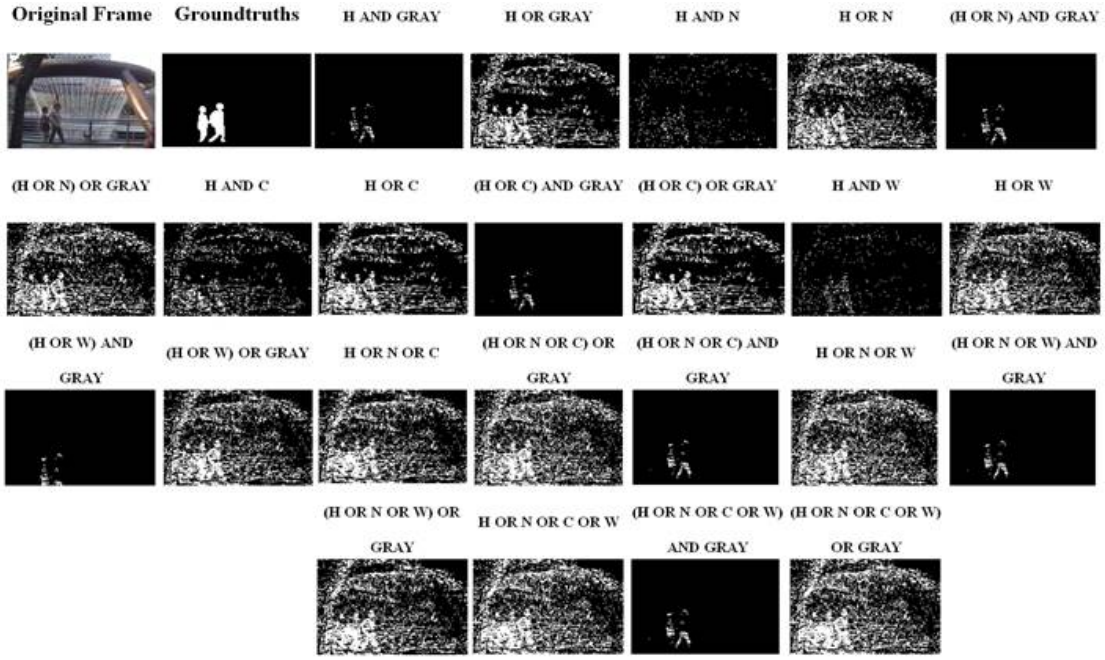
Appendix A

Segmented images obtained with color combinations for Background Subtraction

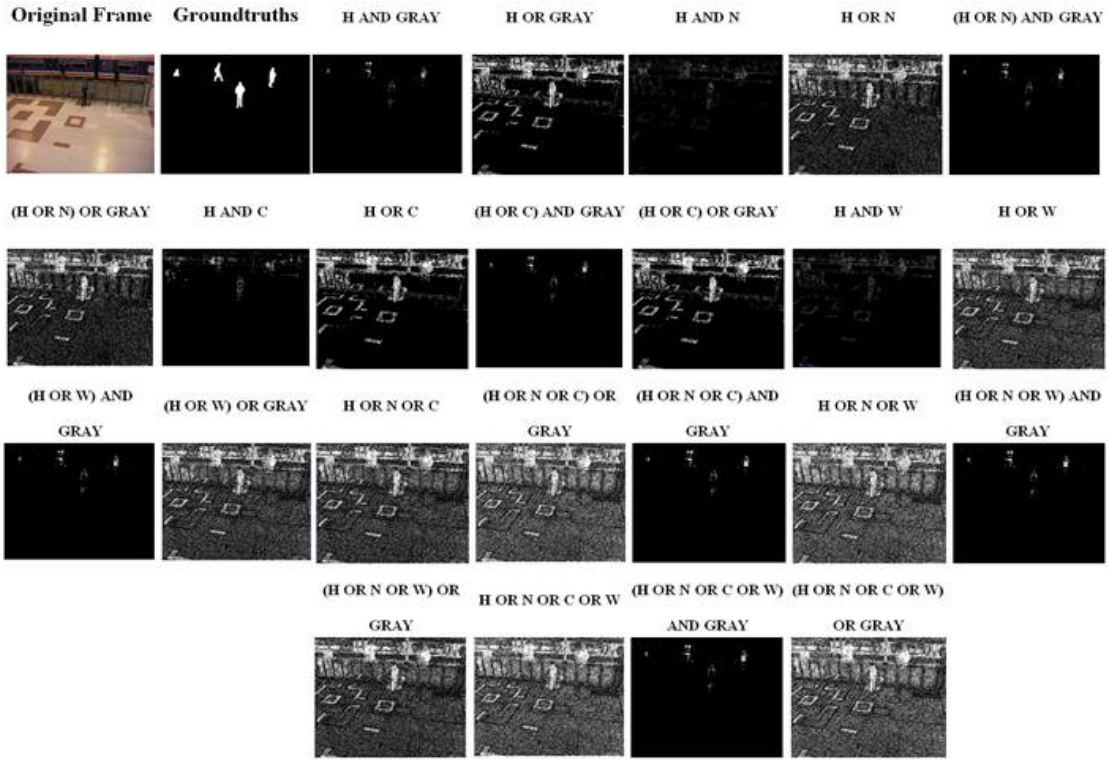
Segmented images related to highway



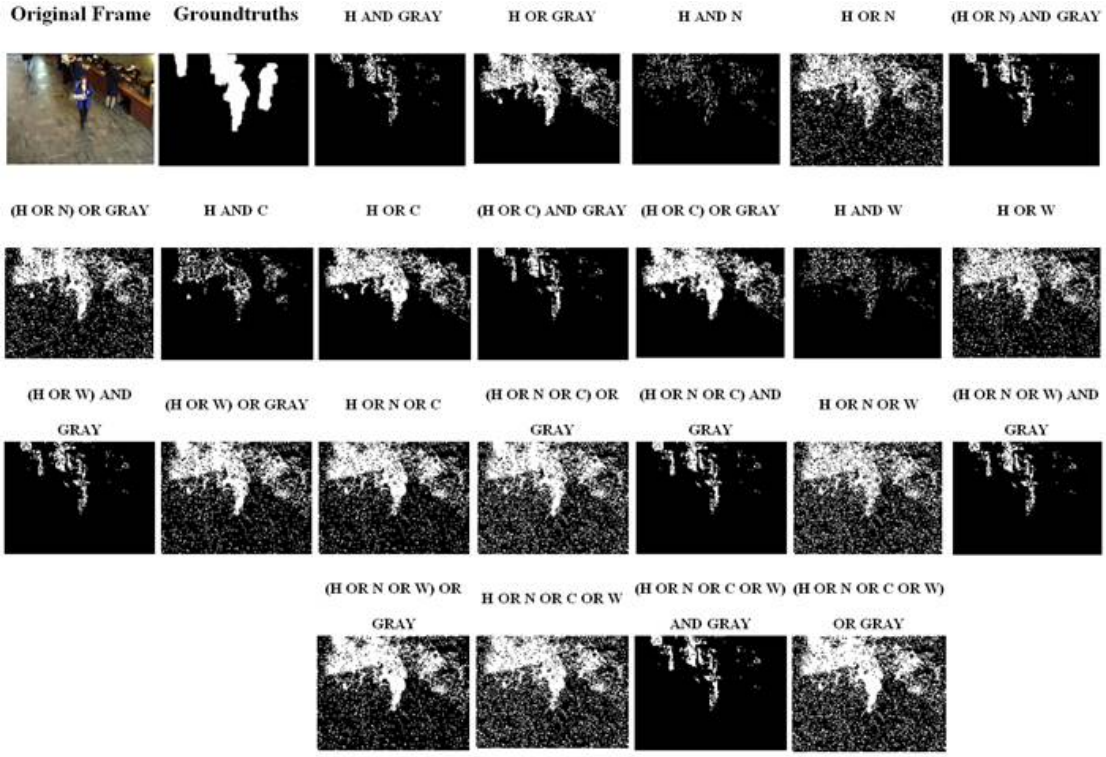
Segmented images related to fountain



Segmented images related to Pets2006



Segmented images related to Bootstrap

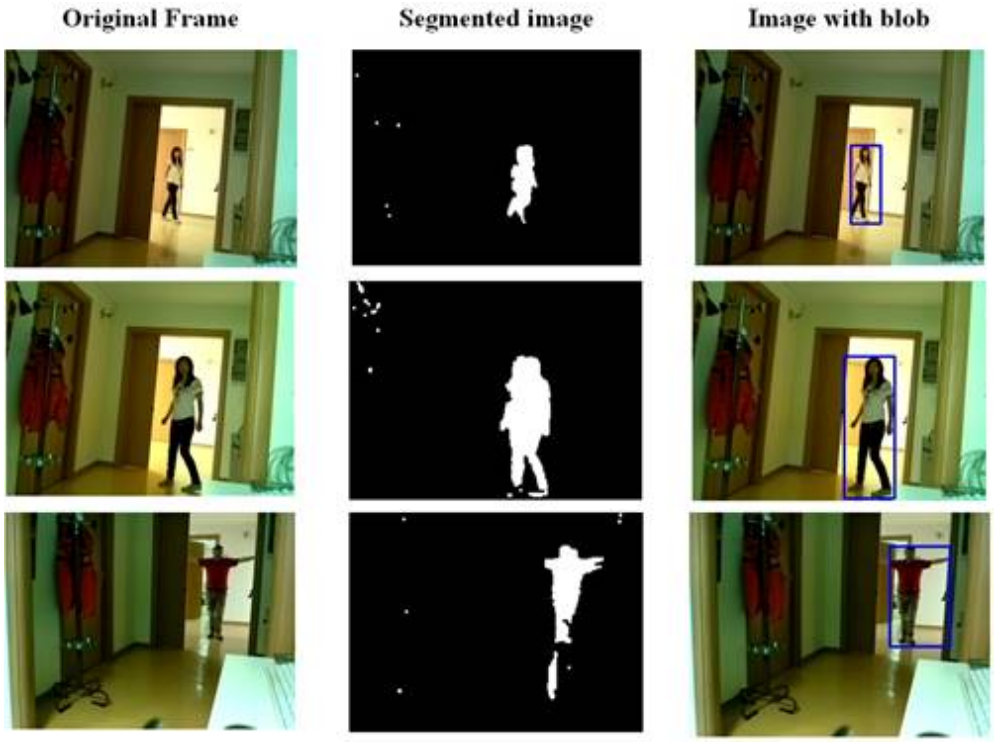


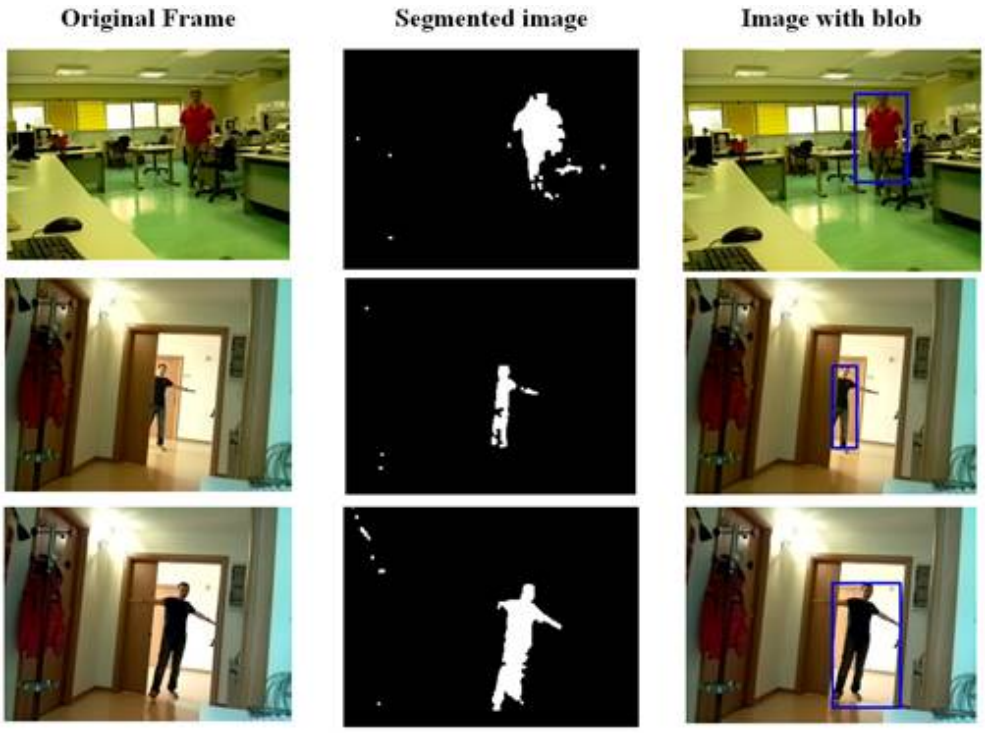
Segmented images related to Office

Appendix B

Resulting images of moving detection with embedded surveillance system

Resulting images of moving detection in DIMES lab




Original Frame
Segmented image
Image with blob

BACKGROUND SUBTRACTION FOR MOVING OBJECT DETECTION

Lorena de los Angeles Guachi Guachi

Supervisors:

Prof. Giuseppe Cocorullo
Prof. Stefania Perri
Prof. Pasquale Corsonello

Rende, Italy
November, 2016

UNIVERSITÀ DELLA CALABRIA

SENESCYT
Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación