الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Ingénierie des Médias

Thème :

Système de Vidéo Surveillance Basé sur les Mixtures de Gaussiennes Adaptatives

---

Encadré par :

M. FAROU Brahim

Présenté par :

HAFFERSSAS Oussama

HIMEUR Chams Eddine

Juin 2015

Résumé

représente

Ce mémoire de recherche ~~vient de présenter~~ le travail de projet de fin d'études (Master), qui a pour ambition d'améliorer la détection des objets en mouvement en utilisant les mixtures de gaussiennes adaptatives.

La première partie est une examination générale des fondamentaux de vision par machine, ainsi qu'une ~~examination~~ détaillée des concepts des systèmes de vidéosurveillance.

Étude

Le rôle de cette partie est la mise au point sur les différents acteurs qui se trouvent dans les systèmes de vidéosurveillance reposants sur la détection automatique des objets en mouvement, précisément : la soustraction du fond.

Les mixtures de gaussiennes adaptatives est une méthode communément utilisée pour détecter les objets en mouvements. Cependant, la performance de cette méthode est susceptible de dégrader face aux plusieurs ~~défis~~ comme : la variation instantanée de luminosité, les objets qui s'arrêtent de bouger, l'ombre et les fonds dynamiques.

facteurs

sera consacré aux différentes techniques

La seconde partie ~~s'applique à décrire les pratiques suivis~~ pour réduire les inconvenants des mixtures de gaussiennes par une modélisation plus fiable et adaptative. Tout en essayant de répondre aux problématiques suivantes :

Comment améliorer les mixtures de gaussienne, tout en maintenant leurs corps et sans l'ajout de post-traitements ?
Beaucoup d'améliorations ont été déjà proposées, pouvons-nous y arriver à proposer une meilleure ?

Acknowledgments

First of all, we are grateful to ***"ALLAH"*** for giving us the ability, the inspiration and the health to complete this work, and all of our studies for the past five years.

We wish to express our sincere thanks and deepest appreciation to our project advisor, Mr Farou Brahim, for the patient guidance, encouragement and advice he has provided throughout this project. Especially, for the incredible support, motivation and the belief and the trust he had on us, which made the completion of this work possible.

Our gratitude goes especially to the jury members, in advance, for their interest in our project, by agreeing to review and enrich it by their proposals.

We must express our gratitude to our parents, for their endless care, support and encouragement, our brothers, sisters, family and beloved ones who experienced all of the ups and downs of us working on this project.

Completing this work would have been difficult without the support and friendship provided by our friends, students of the Media Engineering branch.

Finally, we thank all professors of our Faculty, in particular, professors of the department of computer science for providing us with enough knowledge to undertake this project.

Contents

List of Figures

List of Tables

Abbreviations

MOG	Mixture Of Gaussians
GMM	Gaussian Mixture Model
DRA	Dorsal Rim Area
3D	3 Dimensions
Pixel	Picture Element
PPI	Pixel Per Inch
DPI	Dot Per Inch
SD	Simple Definition
HD	High Definition
FPS	Frame Per Second
RGB	Red Green Blue
YUV	Luma (Y) and two chrominance components (UV)
MPEG	the Moving Picture Experts Group
GoP	Group of Pictures
PS4	PlayStation 4
CCTV	Closed Circuit TeleVision
VSS	Video Surveillance System
UK	United Kingdom
NYPD	New York Police Department
DVR	Digital Video Recorder
HDD	Hard Disk Drive
NI	Noise Image
CJ	Camera Jitter
CA	Camera Automatic-adjustments
TD	Time of the Day
LS	Light Switch

FA	Foreground Aperture
MO	Moved background Objects
IBO	Inserted Background Objects
MB	Multi-model Background
WFO	Walking Foreground Objects
SFO	Sleeping Foreground Objects
GPU	Graphics Processor Unit
FPGA	Field-Programmable Gate Array
SG	Single Gaussian
PDF	Probabilistic Density Array
BGS	BackGround Subtraction
SD	Standard Deviation
WORA	Write Once and Run Anywhere
JDK	Java Development Kit
CPU	Computer Processing Unit
Ghz	Giga Hirtz
RAM	Random Access Memory
IEEE	Institute of Electrical and Electronics Engineers
PoI	Person of Interest
RoI	Region of Interest
TP	True Positive
TN	True Negative
FN	False Negative
FP	False Positive
SE	Shadow Error
CDnet	Change Detection net
Rx-GMM	Region based Gaussian Mixture Model
Bx-GMM	Box based Gaussian Mixture Model

Alhamdulillah... From the bottom of my heart, I dedicate this work to my absolute lovely parents who reached the impossible to make what we do possible.

I dedicate this work to my brothers, and all my family members.

I sincerely dedicate this work also to "Benyahia" family whom I consider as my family.

My special dedication goes to Hiba, the person who always supported me for every mile of this long distance path.

Finally, I dedicate this work to whoever wished for us luck and success.

Oussama

To my mother who supported me and fund me through it all, to a very special person whom without I couldn't get the inspiration and the spirit of creativity to pull my part of work.
And special thanks to both my mate Hafferssass Oussama and Mr. Farou Brahim, it's been great time working with brilliant people like them.

Chems Eddine

General Introduction

We live in the era of the domination of computer vision on every aspect of our lives, from fun use like taking pictures in holidays, to serious stuff like training pilots in simulators using synthetic imagery.

When it comes to video surveillance, which is a very sensitive domain, computer vision plays a very important role, either to offer personal assistance for security managers who can't stay focused on multiple scenes at the same time, or by affording fully automatic surveillance systems that can detect accurately suspicious persons or objects.

In order to offer this accuracy of detection, researchers try always to improve change detection and object detection methods, or even try to come up with new ones to conquer deferent challenges in video surveillance.

From a Variety verity of methods, we are meant to study the Adaptive Mixture of Gaussians method. This method in particular, can detect changes in the monitored location, and can adapt to these changes at the same time. It creates a model for the scene when it's considered empty from any object (called background modeling) and compares new footage with this background model to find the new objects (called foreground detection). After that, it updates the background model to slightly include new changes. défauts defects.

However, original MOG method has its downfalls. As consequence, the aim of this work is to find solutions and improvements to this method.

In this ~~memoir~~ Work, we will follow a logic order to present how deferent domains are linked together, we will walk progressively from computer vision (the field of study), then video surveillance (the field of application), to a brief survey on deferent object detection and tracking methods in video surveillance, where we will explain in details the original Gaussian Mixture Model method. Finally, we will present our proposed improvements, in the form of two deferent new methods (called Rx-GMM and Bx-GMM).

Our improved GMM methods had shown a deferent behavior than GMM's, due to the deferent concept we used to model the background. The tests we performed at the end of this work will show in addition of the points we missed, the degree of improvement we succeeded to make when handling deferent challenges in video surveillance scenes.

Chapter 1

Machine Vision

1 Introduction

Vision has always been the main resource of information that humans, animals, and almost every breathing species on earth has used to discover the exterior world. However there is no single model for vision that is ideal for all circumstances[1], from sophisticated polarized vision of the ant[2] and the locust Figure 1.1, to the special vision system of aquatic mammals, like cetaceans, that functions like a charm both in air and water, keeping in mind that these two media have very different optical features[3].

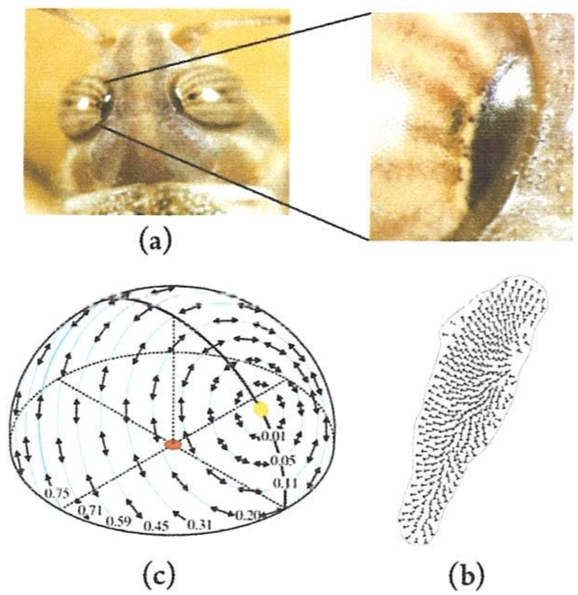


Figure 1.1: Polarized Vision of Locusts : (a) eyes of a locust, (b) polarization-sensitive dorsal rim area (DRA) in the left compound eye of a locust, it gives the locust the ability to get (c) a 3D perception of its position using the pattern of polarized light of the blue sky.[4]

Nowadays, The challenge that faces the modern vision science is not the lack of computing power, nor the lack of inspirational systems, the challenge is how to give the machine the appropriate physical components and the convenient logic that enables such a near-ideal vision system for machines.

2 Human Vision

Human vision system is so far one of the greatest vision systems. It has some physical limitations : we can see just a small portion of electromagnetic spectrum called "visible light"[5], but it has a highly efficient receptor -the eye- that can highly adapt to the changes in this visible spectrum Figure 1.2.

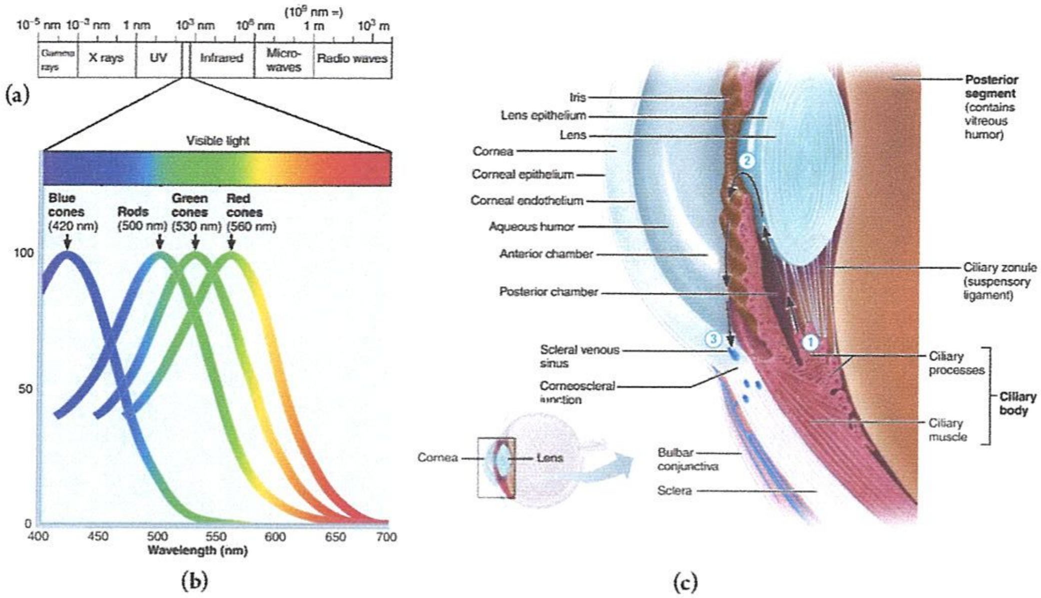


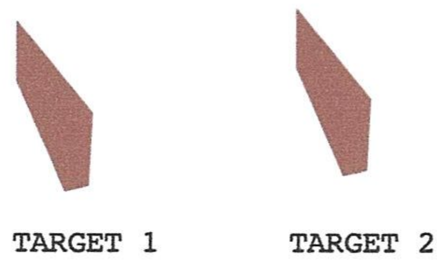
Figure 1.2: Human Vision : (a) The electromagnetic spectrum, of which visible light constitutes only a small portion. (nm 5 nanometers), (b) Sensitivities of rods and the three cone types to the different wavelengths of the visible spectrum. (c) Human eye lenses.[6]

Though scientists has discoved almost the whole architecture of the human eye, yet no one could explain the mysterious process in the human brain that gives us the perception of what eyes receive. Like in Figure 1.3, regions with the same geometry and the same colors, if attributed to deferent scenes, as a consequence they will have deferent meanings.

3 Machine Vision

3.1 Definition

Machine vision is the discipline of giving machines the ability to capture visual information and to understand the context of the received data. Usually this begins by describing the scene, which is captured in one or more images, with deferent types of receptors, and reconstructing its properties, such as shape, illumination, and color distributions[8], to obtain a near-to-reality interpretation.



(a)

(b)

Figure 1.3: Human Perception : (a) Target 1 and Target 2 has the same color, and the same dimensions, for human brain they represent similarity, in (b) They still the same, but for human brain they have very deferent perception, now there is transparency, 3D environment, objects overlapped and all other informations that was added by the brain [7].

(a)

(b)

Figure 1.4: Microsoft HoloLens : (a) A new commercial mixed reality experience, of the real world and a virtual world combined using computer vision techniques of HoloLens, (b) Microsoft's new HoloLens, unveiled in January 21st, 2015[9].

Computer Vision is an important and maturing engineering science. It underpins an increasing variety of applications that require the acquisition, analysis, and interpretation of visual information. It's considered as a young discipline, and despite recent success, it is still a relatively brittle technology[10].

3.2 Machine Vision System

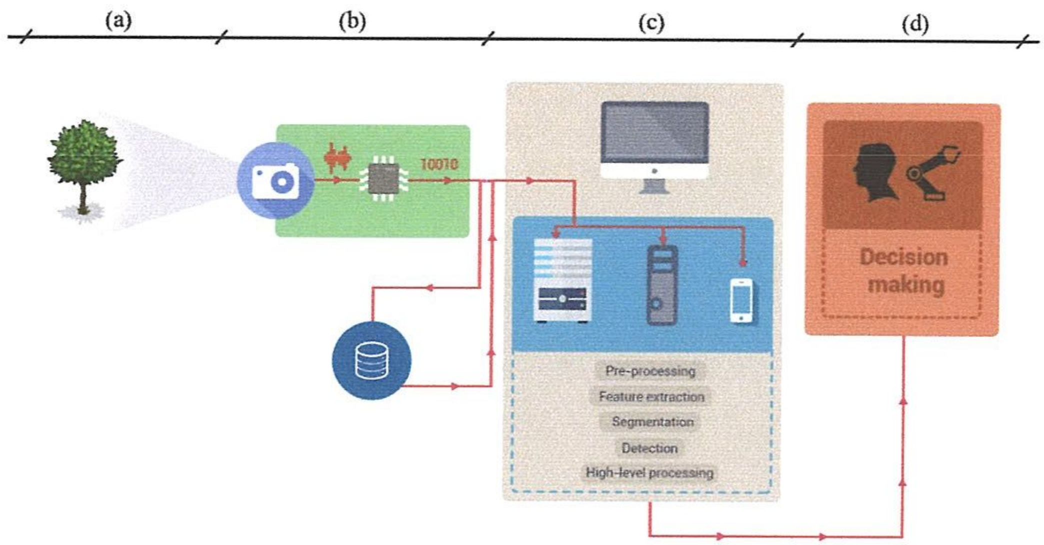


Figure 1.5: Computer Vision System : (a) Real scene. (b) Data resource : Image acquisition or multimedia databases. (c) Image processing techniques can be executed on a verity of processing devices. (c) Decision making for humans and automates.

A Computer Vision systems usually includes[11] :

1. **Real Scene** : The main resource of visual information for computer vision systems. This includes any physical object, environment, or a movement that can be captured or seen. However in some applications, databases are considered as a main resource of information.

2. **Image Acquisition** : A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. All this sensors convert the received signal to a digital representation by sampling and quantization.

3. **Pre-processing** : this includes noise reduction in order to assure that sensor's noise does not falsify original information, and Scale-space representation to enhance image structures at locally appropriate scales (In case where images were captured with a known distortion).

4. **Feature Extraction** : In which, image features at various levels of complexity are extracted from the image data. From extracting interest points such as corners and blobs, to more complex features like lines and shapes, and other zone related features like texture, or motion.

5. **Detection & Segmentation** : The interpretation of the feature extractions results gives the vision system information about object of interest, or regions to select. In

same cases, this will be the end of the computer vision time line, but in others, this step's result would be used for high level processing techniques and algorithms.

6. **High Level Processing** : It includes Image matching and image recognition for classifying results based on a set of data, and Image synthesis for creating new views and scenes based on multiple occurrences for the detected object[12].

7. **Decision making** : which could be the goal of the system in some cases like security and military, or making autonomous actions for robots, or just matching information in case of multimedia databases.

3.3 Digital image

Any visual content that was captured or created, stocked, treated and displayed by a digital device, is considered as a digital image, adding a window of time gives us a digital video.

3.3.1 Image

Digital images are generally classified by the type of numeric representation of its data to two types :

1. **Victor digital image** In which visual content is represented by geometrical objects Figure 1.6. with multiple attributes like shape, color and position. These attributes represent its spacial position in a known landmark, and its appearance.

2. **Raster**[1] **digital image** Visual data are represented by a matrix of n dimensions (generally $n = 2$), each cell of the matrix contains a unified color element (generally square shaped) called "*Pixel*" (for **Pic**ture-**el**ement).
   In addition of dimensions, All raster images have these characteristics :

   - Dimensions : In case of dimension $n = 2$, the image has a width and a height, else if $n - 3$, which is the case of 3D images, pixels are called "*Voxels*"[2] instead.
   - Color Depth : the number of colors that one pixel can have, which must be a finite number obtained by equation 1.1.

   $$Color\ depth = 2^{Bit\ depth} \tag{1.1}$$

   bit depth of images is usually one of these : 2 bits for black and white images, 8 bit for Grey scale images and 16, 24 up to 32 bit for almost real color images.

[1] Also called "Bitmaps".
[2] For **Vo**lume and pixel combined.

- Resolution : the relation between dimensions of an image (in Pixels), and its real representation in real world (on digital display device or on printed paper) in Inches. Digital image's resolution is measured in PPP unit (Pixel Per Inch)

Figure 1.6: Victor Image : This blue star with text is not composed of pixels, instead it's the a result of geometry specifications of the SVG file on the right.

```
<?xml version="1.0" encoding="UTF-8"
      standalone="no"?>
<!--Created with Inkscape http://www.inkscape
      .org/ -->
<svg
   xmlns:svg="http://www.w3.org/2000/svg"
   xmlns="http://www.w3.org/2000/svg"
   xmlns:sodipodi="http://sodipodi.
      sourceforge.net/DTD/sodipodi-0.dtd"
   xmlns:inkscape="http://www.inkscape.org/
      namespaces/inkscape"
   id="svg2"
   height="297mm"
   width="210mm"
   sodipodi:docname="SVG Sample.svg">
  <path
     sodipodi:type="star"
     style="fill:#00adad;fill-opacity:1;
     stroke:#004448;stroke-width:4;stroke-
     miterlimit:4;stroke-opacity:1;stroke-
     dasharray:none"
     id="path5080"
     sodipodi:sides="5"
     sodipodi:cx="86.873116"
     sodipodi:cy="242.21983"
     sodipodi:r1="151.45552"
     sodipodi:r2="75.72776"
     sodipodi:arg1="0.76653248"
     sodipodi:arg2="1.394851"
     inkscape:flatsided="false"
     inkscape:rounded="0"
     inkscape:randomized="0"
     d="M 195.96959,347.2757
     100.12842,316.77847 20.671716,378.44079
      20.059752,277.86627
     -62.138072,221.35316 33.331870,180.6010
      60.362602,93.102657 1
     59.011308,81.008793 100.52603,-3.18238
     -58.62113,81.7262 z"
     inkscape:transform-center-x="16.384462"
     inkscape:transform-center-y="-12.896318"
     transform="matrix
     (2,0,0,2,205.05942,17.038998)" />
</svg>
```

3.3.2 Video

Digital video is a set of bitmap images called frames, shown on a digital display with a constant speed, it can be captured by a digital camera or created by a software, or both. Digital videos have their own characteristics :

1. **Resolution**
   Represents the width x height of the frames, which remains constant for all frames in the video. Some of resolutions are represented by one number (the width), for example a 1080p video means that the resolution is 1920 x 1080 pixel.

2. **Aspect ratio**

   Aspect ratio is the ratio of a frame's width to its height. For example the aspect ration of a video with a resolution of an 1920 x 1080p is obtained by equation 1.2.

$$Aspect\ ration = \frac{width}{height} = \frac{1920}{1080} = \frac{16}{9} \tag{1.2}$$

   Digital videos have different resolutions but they must have one of these standard aspect ratios Figure 1.7, so they can be displayed properly on a digital display device.

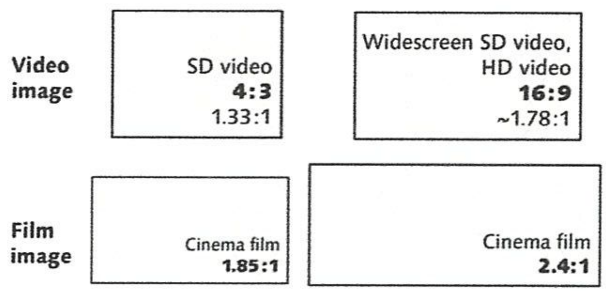


Figure 1.7: Video Aspect Ratio : some standard aspect ratios of videos, for SD videos, HD, and film videos[13].

3. **Frame rate**

   Frame rate of a video is the constant number of frames shown in a period of time, it's usually represented by the number of frames per second FPS. Human vision system can process 10 to 12 separate images per second[14], this means that a video with a 10 FPS does not appear smooth and cannot give us the illusion of continuous feed, this is why videos frame rate is usually set to 24 FPS or more (from 30 to 120 FPS for usual cases).

4. **Compression method**

   For example without compression, if we had a 5 min 720p video with frame rate of a 30 FPS, its size would be :

$$Size = nP \times bitDepth \times FPS \times (60 \times 5) \approx 7,7Gb \tag{1.3}$$

   According to the size equation 1.3, it would be impossible to store a 1 hour film in a DVD, and it takes a large amount of bandwidth to be transfered. Streaming uncompressed videos with the actual Internet speed would be impossible too !
   Digital video compression techniques play an important role to enable video applications. Video compression relies on detecting redundant information and replacing it with a more efficient representation. Video compression also takes into consideration the human psycho-visual system by discarding information that is difficult or impossible for humain eye to see. Almost all video compression techniques are therefore lossy.

There are many different video compression techniques, most commonly used approaches are :

(a) **Color Difference Spaces** : In this approach we convert the color space of the original video from RGB to YUV, specifically the YCrCb format. Frames are divided to 4:4 blocs, after, they can be compressed by down-sampling the resolution of the chrominance Figure 1.8.

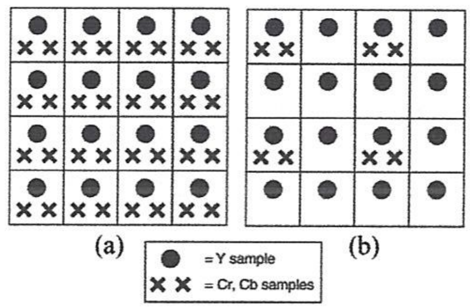


Figure 1.8: Color space compression : the 4:4 down-sampling method, (a) the original video (b) the compressed video wherein there is only one chroma pair sample for every 2 x 2 grid of pixels[15].

(b) **MPEG : Picture coding types (I, P, B)** This approach is the baseline algorithm for MGPEG video Compressing algorithms. A video sequence is partitioned into successive groups of pictures (GoPs).
The first picture in each GoP is coded using a JPEG-like algorithm, independently of other pictures.
I : for Intra or I-picture, is a reference picture available for use in predicting neighboring (non-intra) pictures.
P : for P-picture contains elements that are predicted from the most recent anchor frame. P-pictures are dependent to I-pictures which can have two or more P-pictures.
B : for B-pictures, they are optional. Elements of a B-picture are typically Bipredicted by averaging motion-compensated elements from the past reference picture and motion-compensated elements from the future reference picture.

There are lots of compression methods that can be used to compress videos data. Even though, for every approach we find a standard decoder and a verity of coders, which make it impossible for us to cite them all in this stat-of-the Art Chapter. in Figure 1.10 we find the chronology of the most used video coding standards.

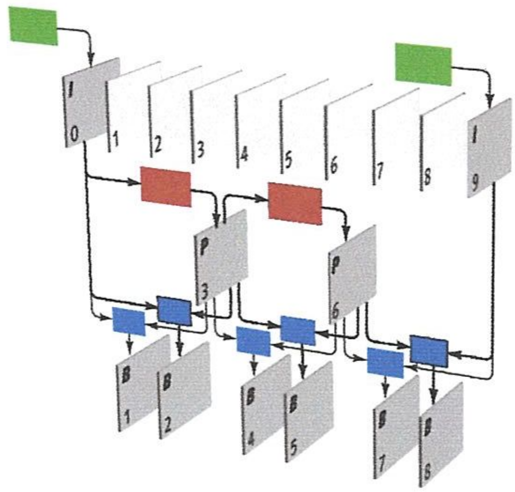


Figure 1.9: The three-level MPEG compression hierarchy :a simple encoder that emits a fixed schedule of I-, B-, and P-pictures (a GoP structure with an I-picture interval of n=9, and a reference P-picture interval of m=3)[13].

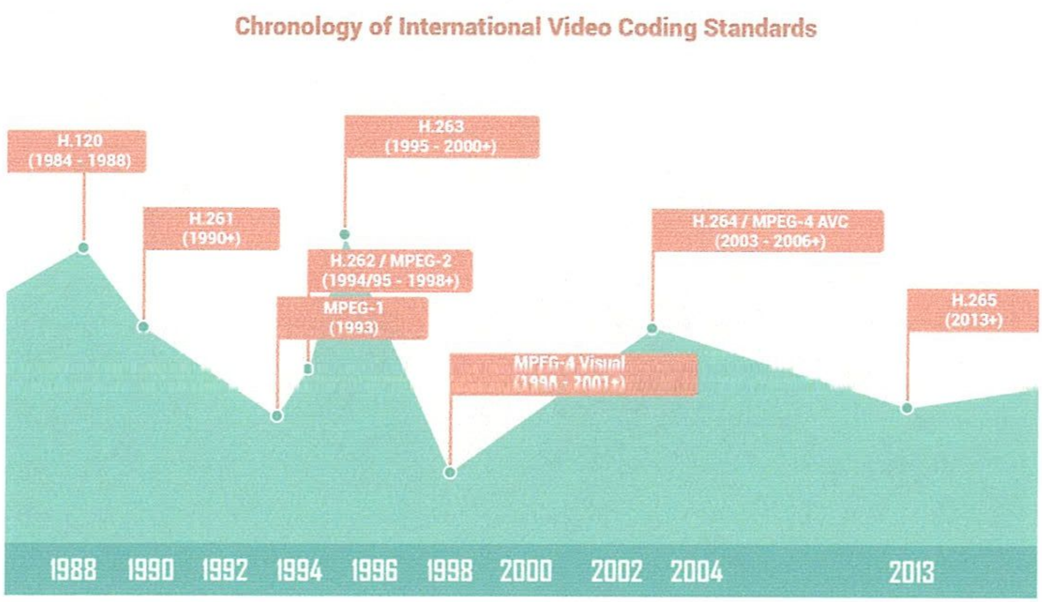


Figure 1.10: Chronology of International Video Coding Standards[16]

4 Application domains of machine vision

4.1 Computer Vision for Visual Effects

Modern blockbuster movies seamlessly introduce impossible characters and action into real-world settings using digital visual effects. These effects are made possible by research from the field of computer vision[17].

Figure 1.11: Computer Vision for Visual Effects : (a) Captured real scene (b) The new scene after adding visual effects, from Game of Throne TV Serie[18].

4.2 Human computer interaction

In order to make Human-Computer interaction more intuitive, machine vision is used to, at least minimize, or replace the click button interaction, with gestures driven interaction which is more natural way of interaction. One of the best examples is using the player's gestures in gaming using machine vision techniques like head tracking Figure 1.12, eyes and hands tracking.

(a)

(b)

(c)

Figure 1.12: Computer Vision In Human Computer Interaction : (a) PlayStation Camera for the PS4 system, with dual wide angle lenses for enabling hands-free navigation of the user interface. (b) Player's natural head movement. (c) Head tracking was used to change the camera view inside the game[19].

4.3 Industry

The industry was one of the first domains that applied computer vision in a massive scale, in order to help employees or replace them. Computer vision in industry is usually accompanied with robotics, to automate industrial tasks Figure 1.13.

(a)

(b)

Figure 1.13: Amazon's Kiva robots : (a) amazon's robots used to handle and transport products in amazon's warehouses. They are equipped with camera on the bottom to detect (b) white squares on the floor used to aide robots to find their way using computer vision[20].

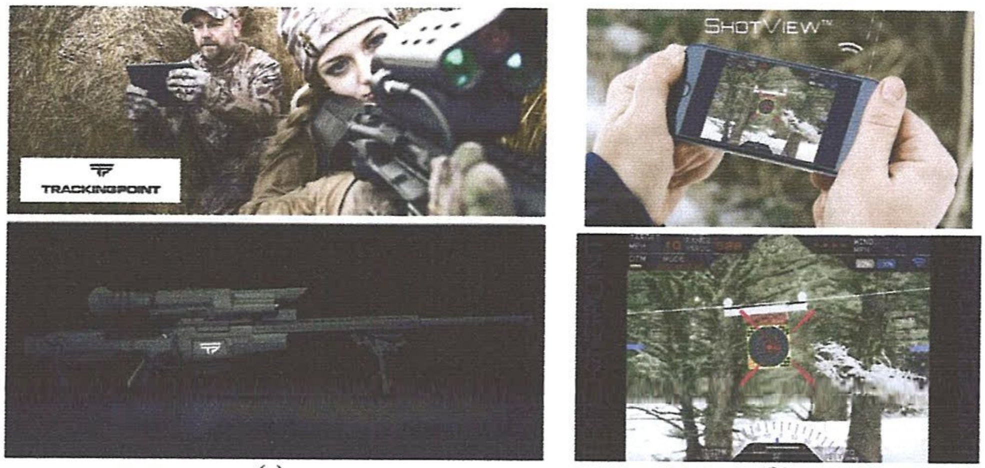
4.4 Medical Imaging

Machine vision plays an essential role in the medical field, this includes computer-assisted diagnosis, image segmentation Figure 1.14, and image-guided therapy, even image annotation and image database retrieval. Integrating machine vision in medical imaging, by using new imaging modalities and methodologies, facilitates the detection and diagnosis of abnormalities in medical images.

Figure 1.14: Computer Vision In Medical Imaging : Brain image segmentation used to classify Dementia[21].

4.5 Military

Computer Vision has long been of interest to, and utilized by sophisticated armies around the world, especially in detecting, targeting and tracking objects to maintain accuracy when shooting. Target lock system in armed planes is the best example. Recent target lock system based on the one on the plane's was added to soldiers' rifles to enable them to shoot long distance targets without experience Figure 1.15.



(a) (b)

Figure 1.15: Machine Vision Precision-Guided Firearm : (a) Machine vision enabled sniper rifles made by Tracking Point Startup. (b) The technology allows a shooter to pinpoint a target, then use object-tracking technology, combined with a variety of variables (temperature, distance, etc.), to determine the most effective place to fire [22].

5 Conclusion

This first chapter allowed us to discover the main themes and the essential ideas to remember in the context of machine vision, as well as the basics of the digital imagery and, especially, digital videos.

Clearly, machine vision has a big impact on our daily lives. However, navigating through machine vision recommends a good knowledge of multiple domains that compose its fundamentals, like Signal Processing, Artificial Intelligence, Geometry and Software Engineering.

For the next chapter, we will dive into digital video surveillance which is our domain of application in this project.

Chapter 2

Video Surveillance

1 Introduction

In our days, watching a camera hanging out from a wall or a roof top is something usual. Because of the popularity of video surveillance, we can find it in banks, parking lots, museums, universities, stadiums, big or small stores, and even in streets.

These cameras usually lead to security rooms, where we often see a matrix of monitors showing cameras streaming, and two or three security agents keeping an eye on them.. This is pretty much what most people think of video surveillance !

In this chapter, we will dive in the science behind video surveillance systems, its component and how, and what's for. In addition, we give an ethical summary about privacy and video surveillance.

2 Definition

According to Collins Dictionary, video surveillance system is a system of monitoring activity in an area or buildings using a television system, in which signals are transmitted from one or more cameras to the receiver's television (a limited set of monitors) by a media (cables or telephone links) forming a closed circuit, for that it's also called : Closed Circuit TeleVision CCTV.

A video surveillance system (also short VSS) offers the possibility of visual surveillance while the observer is not directly on site. Surveillance may be performed not only directly but may also be stored, evaluated and repeated as often as necessary.

3 Chronology

Video surveillance wasn't developed as a science as it is, instead, it had always a relation with its domain of application. These are the big events that shaped video surveillance to its shape that we know today[23] :

- 1942: The first report of using surveillance cameras was for military purposes. Engineer Walter Bruch, Figure 2.1, installed a closed-circuit television CCTV system for Siemens in 1942 at the Test Stand V-II rocket launch site in Peenemunde in Germany to safely monitor any cause of malfunction or problems from the rocket launches.

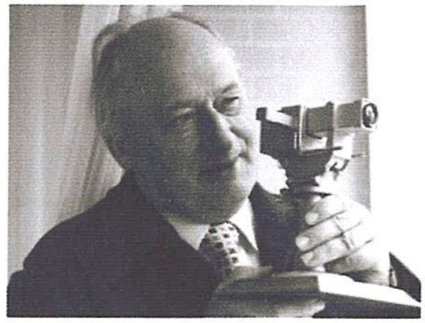


Figure 2.1: Walter Bruch 1908-1990 : The television pioneer Walter Bruch with a model of his iconoscope camera, which was used in the the first live broadcast in 1936. Photo: AEG-Telefunken in February 1978

- 1956 - 1960: was the period when video surveillance starts to be used for public surveillance : The police in Frankfurt, Germany put into service the first ̈photographic and automatic red light-surveillance, in order to investigate violations of traffic regulations. In addition to traffic control, the observation of rallies and public gatherings was the second task delegated to these camera eyes.
  They used it also to monitor crowds attracted to the arrival of the Thai Royal Family in 1960.
- In 1960 - 1970(+) : The video surveillance starts to roll out in United Kingdom in London Transport train station, and some UK companies markets a video surveillance system to retailers to catch shoplifters.

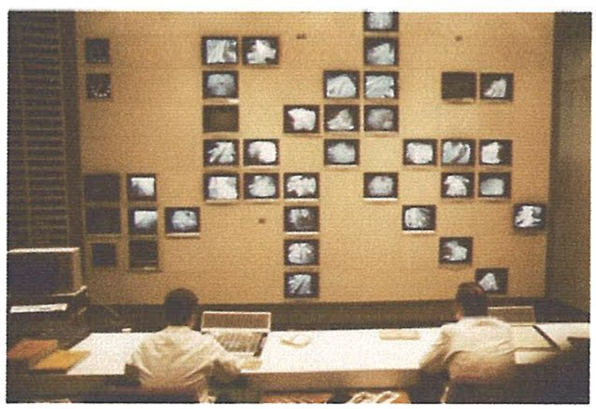


Figure 2.2: Closed circuit TV in Munich, 1973

After that, In United Stats, the NYPD installed cameras in the New York City Municipal Building near City Hall and Times Square.
A nice patent was granted for Marie Van Brittan Brown and her husband, Albert Brown, for the first home security system utilizing television surveillance.

- 1990s : With the early 90s, When digital multiplexer units became affordable, it revolutionized the surveillance industry by enabling recording on several cameras at once.

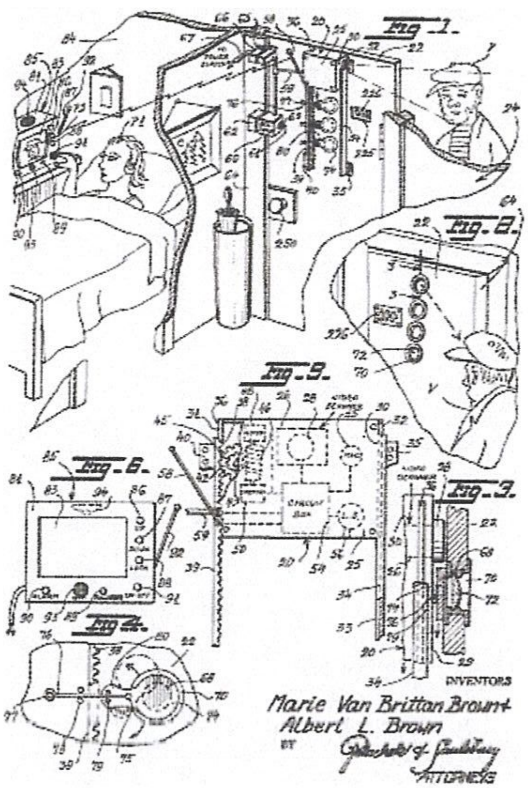


Figure 2.3: First home security system uses video surveillance : with 4 peep holes and a camera that slides up and down to look out each one. Anything picked up was displayed on a monitor, and also featured a remote for unlocking the door.

Digital multiplex also added features like time-lapse and motion-only recording, which saved a great deal of wasted videotape.

- 2000s - Nowaddays : video surveillance can be found in any domain from military to public to private, and for a lot of applications from security and surveillance to more privacy-friendly applications Figure 2.4 like counting people Figure.

Figure 2.4: Passenger Flow and Queue Measurement : A Video surveillance system in London Gatwick Airport for counting passengers flow. (it counts near 10 million passengers per annum).

4 Video Surveillance System Components

4.1 Capture Hardware

Capturing the scene is the first thing any CCTV system must do. In order to capture a scene, equipment must be chosen according to the nature of the scene and the domain of application. Capturing big areas for example recommends cameras with wide lens instead of multiple cameras, indoors video surveillance uses deferent type of cameras which sometimes must be hidden Figure 2.5.

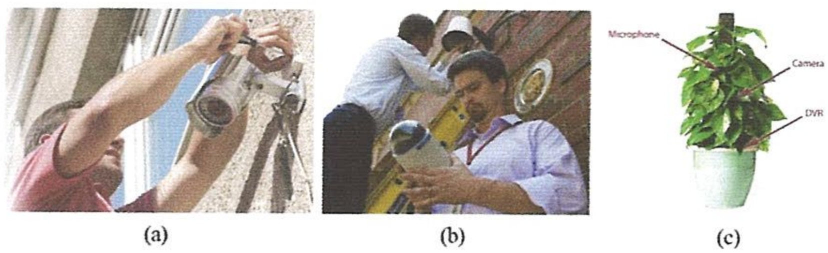


Figure 2.5: Video Surveillance Cameras : (a) CCTV outdoor cameras for night use, (b) for day use. (c) Indoor CCTV hidden hardware

4.2 Transmission

Almost all video surveillance systems use cables to transmit images from cameras to workstation to be displayed or stored. The most used ones are the RG179 and RG59 Coax Cables Figure 2.6 because of their low price and the necessity of long distance cables. However, recent CCTV systems use wireless network to afford more flexibility and online image transmission for distant workstations, using IP-Protocol network.

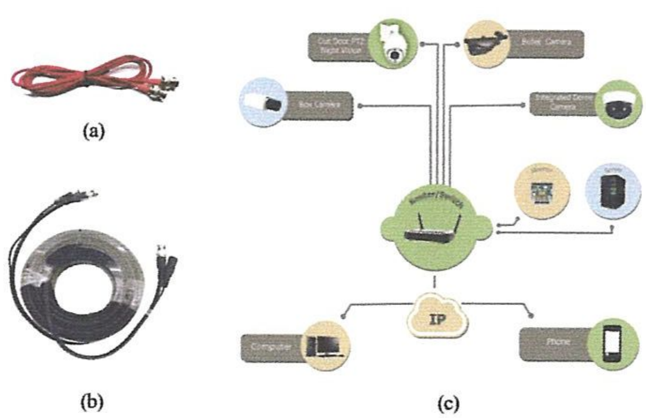


Figure 2.6: Transmission Media in Video Surveillance Systems : (a) Ultra Thin RG179 Coax Cable (b) RG59U plus Coax Cable (c) Recent video surveillance systems using IP network for image transmission

4.3 Visualization

Video surveillance systems needs specific hardware and software for visualizing what is captured by cameras, for the hardware it varies from simple monitors to highly specialized control stations Figure 2.7, and for the software it's all about the Human-Machine Interface that must respect CCTV systems constraints, like giving higher importance for viewing scene and easy-access buttons to save camera images or to zoom in and out etc.

Figure 2.7: Fully equipped monitoring room for a CCTV system.

In case of IP network based CCTV, there is always possibility to show camera feed on smart devices like smartphones and tablets Figure 2.8. this approach has been adopted to lots of recent video surveillance systems called : Remote Mobile Surveillance Systems.

Figure 2.8: Remote Mobile Surveillance Systems : A cross-plateforme remote mobile video surveillance concept.

4.4 Storage

For video surveillance systems it's a necessity to store some of, and sometimes all of cameras' stream in a database, this creates a huge problem of storage capacity especially when using a video surveillance system for a long time and with multiple cameras. Because of that, CCTV systems usually use what's called DVR (Digital Video Recorder Figure 2.9) to store videos. By giving the user the possibility to save videos on HDDs, and the option to replace them when they're full, It provides a convenient way of storage to CCTV systems.

Figure 2.9: Digital Video Recorder : Samsung's DVR with HDD (SRD-1676D)

How much can a DVR save ? to answer this question, lots of factors come in the way, they are :

- Number of HDDs in the DVR
- Capacity of each HDD
- Resolution of cameras
- Number of cameras
- FPS : Number of frames per second considered to store
- Required length of video archive (days)
- Recording motion (Scheduled or non stop)
- Compression of videos

4.5 Analyses

To analyze video feed in video surveillance system, older systems used human based approach, where a person sits in front of monitors and try to recognize any possibility of specious behavior. However, this approach is very limited when using multiple cameras with multiple angles, this is why recent video surveillance systems are equipped with processing unites to analyze, detect, and sometimes recognize objects or persons, and detect and understand their motion.

In order to apply analysis on videos of video surveillance system, Object Detection and Object Tracking Algorithms were developed and implemented in CCTV systems. They can be found in chapter 3.

5 Applications

Visual surveillance has a wide range of potential applications, such as a security guard for communities and important buildings, traffic surveillance in cities a

5.1 Access control in special areas

When somebody is about to enter a restricted area, the system could automatically obtain the visitor's features, such as height, facial appearance and walking gait from images taken in real time, and then decide whether the visitor can be cleared for entry on not.

Figure 2.10: Employee Access Control[24]

5.2 Crowd flux statistics and congestion analysis

Personal identification at a distance by a smart surveillance system can help the police to catch suspects. Having a biometric feature database of suspects, and placing visual surveillance systems at locations where the suspects usually appear, e.g., subway stations. The systems automatically recognize and judge whether or not the people in view are suspects.

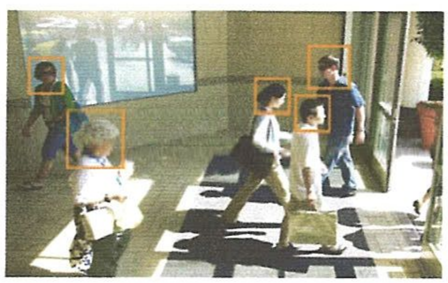


Figure 2.11: Crowd Statistics[25]

5.3 Anomaly detection and alarming

In some circumstances, it is necessary to analyze the behaviors of people and vehicles and determine whether these behaviors are normal or abnormal

Figure 2.12: Density estimation addresses people tracking[26]

5.4 Loss Prevention

In case of expensive equipment or large amounts of merchandise, it is important to protect those assets by monitoring workers and individuals that deal, or get close to these assets.

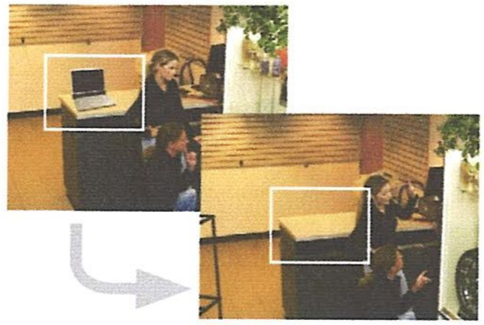


Figure 2.13: Loss Prevention[25]

5.5 Traffic Monitoring

To count cars passing through or to improve the flow of traffic for better travel times, even record fast moving cars, identify them or identify an accident possibility.

Figure 2.14: Bad Conduct Detection[27]

5.6 Interactive surveillance

It's used for social security and cooperative surveillance, one or multiple cameras could be used to ensure the security of an entire community. It can be used for example to trace people's flow and number in public and private areas.

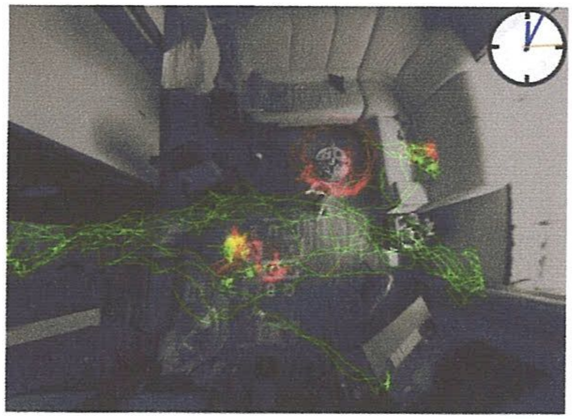


Figure 2.15: Interactive Surveillance : Immersive System for Browsing and Visualizing Surveillance Video[28].

6 Privacy in Video Surveillance

Video surveillance has become commonplace in recent years. By means of closed circuit television (CCTV) technology, individuals are observed without their knowledge in public buildings, train stations, stores, elevators, locker rooms, and school hallways. They're caught at ATMs and when stopped by the police in patrol cars. In London, the average citizen is caught on CCTV cameras 300 times a day[29], and in the United Kingdom alone there are more than 4 million CCTV cameras.

The only problem that comes with the safety of video surveillance systems is that it facilitates the collection of information about an individual and increases the risk of misuse and abuse of surveillance data, which is against the the privacy of the individual.

Far from courts, a solution can be found[30], where private properties or individuals, captured in video surveillance footage, are not accessed freely. Instead the access to the different data types is defined by appropriate privileges.
- Privacy doesn't mean Safety -

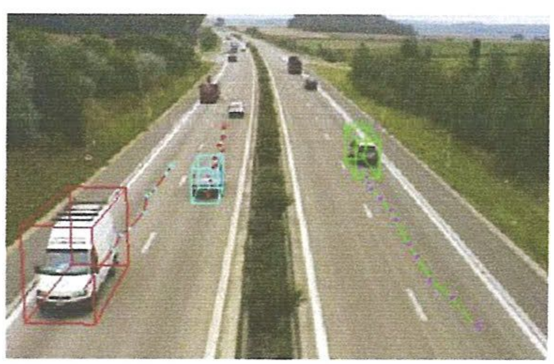


Figure 2.16: Privacy in Video Surveillance : Example of behavioral data. The information provided by the bounding boxes of the vehicles and their trajectories over time is sufficient for traffic monitoring purposes. Instances of the real objects are stored and accesses only by authorized users[31].

7 Conclusion

In this chapter, we discovered the science behind video surveillance, its history and its domains of application, in addition of an ethical view on privacy in CCTV systems.

From this chapter we can conclude that video surveillance systems are very crucial for security applications, which means that developing these systems is a gain for security.

The next chapter will explain how object detection and tracking methods helped to improve and develop the performance of video surveillance systems.

Chapter 3

Object Detection and Tracking in Video Surveillance

1 Introduction

Object tracking is an active research problem in computer vision. It is needed in several areas including video indexing, medical therapy, interactive games, or in this case, surveillance systems. Tracking and detection are very critical in video surveillance systems as their accuracy greatly impacts the eventual success or failure of later scene analysis[32].

In this chapter we will talk about deferent approaches of object tracking, to see how they can help making better video surveillance systems.

Figure 3.1: The surveillance scenario with multiple similar objects[33].

2 Definition

In its simplest form, object tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves in the scene[34]. The algorithm responsible of tracking is called "Tracker", a tracker assigns consistent labels to the tracked objects in different frames of a video. Additionally, depending on the tracking domain, a tracker can also use, and in the same time provide object-centric information, such as orientation, area, or shape.

Tracking objects can be complex[35] due to:

- Loss of information caused by projection of the 3D world on a 2D image.
- Complex object motion.
- Partial and full object occlusions : like tracking a person in crowd.
- Complex object shapes
- Real-time processing requirements.
- Critical Situations : represents general critical situations met in video sequence, defined by computer vision community[36]. They are :
  Noise image due to a poor quality image source (NI), Camera jitter (CJ), Camera automatic adjustments (CA), Time of the day (TD), Light switch (LS), Bootstrapping (B), Camouflage (C), Foreground aperture (FA), Moved background objects (MO),

Inserted background objects (IBO), Multi-modal background (MB), Waking foreground object (WFO), Sleeping foreground object (SFO) and Shadows (S).

Object detection and tracking has become almost axiom in automated and semi-automated video surveillance systems. It affords less human dependency systems, especially for systems with hundreds of cameras, in the same time, it opens the door for machine vision errors that can cost a lot in such systems.
In addition of video surveillance systems, object tracking can be found in :

- Video indexing: automatic annotation and retrieval of videos in multimedia databases.
- Vehicle navigation: video-based path planning and obstacle avoidance capabilities.
- Human-computer interaction: gesture recognition, eye gaze tracking for data input to computers, etc.
- Motion-based recognition: human identification based on gait, automatic object detection, etc

Figure 3.2: Car Tracking In BMW Cars - Used to exclude other cars from BMW car's Laser lights [37].

3 A Survey on Object Detection and Tracking In Video Surveillance

There are lots of surveys about object detection and tracking, with deferent methods of algorithm classifications. According to [38] and [39], Object tracking algorithms in video surveillance are called as follow :

étape

3.1 Motion Detection

Nearly every visual surveillance system starts with motion detection. Motion detection aims to segment regions corresponding to moving objects from the rest of the image. Detecting moving regions provides a focus of attention for later processes such as tracking, object classification and behavior analysis. Because only these regions need be considered in the later processes, they are greatly dependent on it.The process of motion detection usually involves the use of one of these methods :

3.1.1 Background Subtraction

Background subtraction is a popular method for motion segmentation, especially under situations with a relatively static background. It detects moving regions in an image by taking the difference between the current image and the reference background image.
In case of a pixel-by-pixel old fashion method, it is the simplest, but extremely sensitive to changes in dynamic scenes derived from lighting and extraneous events etc.

3.1.2 Temporal differencing

Temporal differencing makes use of the pixel-wise differences between two or three consecutive frames, the current image frame is subtracted either by the previous frame or the next frame of the image sequences to extract moving regions.

Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels.

3.1.3 Optical Flow

Optical flow based motion segmentation uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence.
This method is computationally data intensive and took more time to segment the foreground objects from the scene and that is why there is a need to use specialized hardware (such as FPGA, GPU cards etc.).

The optical flow method is generally used as a feature for both object detection and object tracking.

3.2 Object Classification

Object classification refers to the task of an automatically distinguished moving target of interest from other objects, across successive frames in an image sequence.

For example, the image sequences captured by surveillance cameras mounted in road traffic scenes probably include humans, vehicles and other moving objects such as flying birds and moving clouds, etc. To further track drivers and analyze their behaviors in their cars, it is essential to correctly classify moving objects.
This is why classifying objects is an essential process.

3.3 Object Tracking

After motion detection, surveillance systems generally track moving objects from one frame to another in an image sequence. The tracking algorithms usually have considerable intersection with motion detection during processing.

3.3.1 Region-based Tracking

According to the first paper about Region-based tracking[40], it's a method that can track objects according to variations of image regions corresponding to moving objects. For these algorithms, the background image is maintained dynamically

3.3.2 Contour-based Tracking

Also boundary-based tracking algorithms, they track objects by representing their outlines as bounding contours and updating these contours dynamically in successive frames. Previous works on object tracking showed theoretically and experimentally that this method is insensitive to illumination changes, it can describe objects more simply and more effectively and reduce computational complexity. Even under disturbance or partial occlusion, these algorithms may track objects continuously. However, the results of this method are inaccurate if the image wasn't well prepared before.

3.3.3 Feature-based Tracking

Feature-based tracking algorithms perform recognition and tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images. Tracking of the object is based on the features, requires selecting the right features, which plays a critical role in tracking. In general, the features used for tracking must be unique so that the objects can be easily distinguished in the feature space. The following various features are used for object tracking[41]:

- Color
- Edges
- Centroids
- Texture

3.3.4 Model-based Classification

Model-based tracking algorithms track objects by matching projected object models, produced with prior knowledge, to image data. The models are usually constructed before tracking with manual measurement. It is an example of feature-based tracking. The reason why it is independently described is due to the requirement of grouping, reasoning, and rendering, which may defer it from the feature based tracking. In addition, prior knowledge about the investigated models is normally required.

3.3.5 Optical Flow-based Tracking

Object tracking with optical flow uses the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between objects and the camera and the scene. It uses a vector field which describes how the image changes with time, to keep tracking of the object of interests, which is in this case an ensemble of vectors having almost the same velocity and the same direction.

Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware.

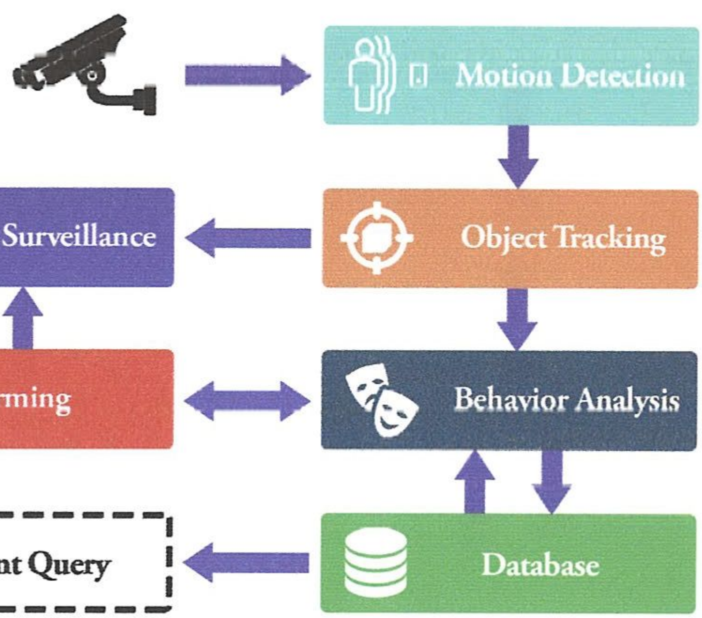


Figure 3.3: Architecture of Video Surveillance System Equipped with Object Detection and Tracking Capabilities[42].

4 Gaussian Mixture For Background Subtraction

4.1 Background Modeling for Video Surveillance

Background subtraction is based on the assumption that the difference between the background image and a current image is caused by the presence of moving objects. Pixels that have not changed are considered as "background" and pixels that have changed are considered as "foreground" that contains "moving objects"[43].

In background subtraction, we have to use background modeling (or representation) that describes the kind of model used to represent the background. It essentially determines the ability of the video surveillance system to deal with uni-modal (static) or multi-modal (dynamic) backgrounds.

4.2 Gaussian Model

In Gaussian Background modeling, to represent the background we model the history pixel's intensity values over time by a Gaussian. The first approach of this method of modeling[44] used a single Gaussian (SG). Authors proposed fitting a Gaussian Probabilistic Density Function (PDF) on the most recent n frames. In order to avoid fitting the PDF from scratch at each new frame time t, a running (or on-line cumulative) average is computed.
However, representing changes with one Gaussian as unimodal model cannot handle dynamic backgrounds when there are waving trees, water rippling or moving algae. To solve this problem, the Mixture of Gaussians (MOG) or Gaussian Mixture Model (GMM) has been used to model dynamic backgrounds[45]

4.3 Background Modeling using Mixture Of Gaussians

In the next steps, we will explain in detail, and step by step, how GMM algorithm works[46], on the most used case : RGB images.

(A) Each pixel is characterized by its intensity in the RGB color space. Then, the probability of observing the current pixel value is considered given by the following formula in the multidimensional case:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}.\eta\left(X_t, \mu_{i,t}, \sum_{i,t}\right) \tag{3.1}$$

Where :

K : is the number of distributions

$\omega_{i,t}$: is a weight associated to the i^{th} Gaussian at time t with :

$\mu_{i,t}$: The mean, and

$\sum_{i,t}$: the standard deviation

(B) We calculate η the Gaussian probability density function :

$$\eta\left(X_t, \mu, \sum\right) = \frac{1}{(2\pi)^{n/2} |\sum|^{1/2}} + e^{-\frac{1}{2}(X_t-\mu)\sum^{-1}(X_t-\mu)} \tag{3.2}$$

(C) The next step is defining the covariance matrix. For computational reasons, we assume that the RGB color components are independent and have the same variances[45]. So, the covariance matrix will be :

$$\sum_{i,t} = \sigma_{i,t}^2 I \tag{3.3}$$

For now, for each pixel we have a mixture of K Gaussians, combined define the background Mixture Of Gaussians Model.

(D) Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized.
The parameters of MOG's model are : K, $\omega_{i,t}$, $\mu_{i,t}$, and $\sum_{i,t}$.

K determines the multi-modality of the background, it's determined by the available memory and computational power[45] [43], and it's usually set :

$$3 \leq K \leq 5 \tag{3.4}$$

The initialization of the weight $\omega_{i,t}$, the mean $\mu_{i,t}$ and the covariance matrix $\sum_{i,t}$ is better be made using an EM algorithm. To avoid huge amount of calculations, we can use the K-mean algorithm, else they can be initialized using intensities of pixels from previous frames of an empty background.

(E) After initializing and updating parameters, we can make the first foreground detection and then update the parameters.
First we order the K Gaussians according to this ratio :

$$r_j = \omega_j / \sigma_j \tag{3.5}$$

Following this ratio. This ordering supposes that a background pixel corresponds to a high weight with a weak variance due to the fact that the background is more present than moving objects and that its value is practically constant.

(F) The first B Gaussian distributions which exceed certain threshold T are retained for a background distribution using this formula :

$$B = argmin_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right) \tag{3.6}$$

The other distributions are considered to represent a foreground distribution.

(G) When a new frame incomes at times $t+1$, a match test is made for each pixel. It calculates if a pixel matches a Gaussian distribution if the *Mahalanobis distance*[1] :

$$\sqrt{(X_{t+1} - \mu_{i,t})^T \cdot \sum_{i,t}^{-1} \cdot (X_{t+1} - \mu_{i,t})} < k\sigma_{i,t} \tag{3.7}$$

where k is a constant threshold :

$$k = 2.5 \tag{3.8}$$

The boolean result represent two cases :

- The TRUE Case : A match is found with one of the K Gaussians.
  In this case, if the Gaussian distribution is identified as a background one, the pixel is classified as background else the pixel is classified as foreground.
- The FALSE Case : No match is found with any of the K Gaussians.
  The pixel is classified as foreground.

At this step, we obtain the binary mask of the foreground.

(H) To make the next foreground detection, the parameters must be updated.
Using the match test 3.7, we can obtain two cases :

- The TRUE Case : A match is found with one of the K Gaussians.
  For the matched component, the update is done as follows :

$$\omega_{i,t+1} = (1-\alpha)\,\omega_{i,t} + \alpha \tag{3.9}$$

$$\mu_{i,t+1} = (1-\rho)\,\mu_{i,t} + \rho . X_{t+1} \tag{3.10}$$

$$\sigma_{i,t+1}^2 = (1-\rho)\,\sigma_{i,t}^2 + \rho\,(X_{t+1} - \mu_{i,t+1}) . (X_{t+1} - \mu_{i,t+1})^T \tag{3.11}$$

where α : is a constant learning rate and ρ :

$$\rho = \alpha . \eta \left(X_{t+1}, \mu_i, \sum_i \right) \tag{3.12}$$

[1]The Mahalanobis distance is a measure of the distance between a point P and a distribution D[47], introduced by *P. C. Mahalanobis* in 1936.

For the unmatched components, μ and $\sum$ remain unchanged, only the weight is re-calculated by :

$$\omega_{j,t+1} = (1-\alpha)\,\omega_{j,t} \tag{3.13}$$

- The FALSE Case : No match was found with any of the K Gaussians.
  In this case, the least probable distribution k is replaced with a new one with parameters :

$$\eta_{k,t+1} = LowPriorWeight \tag{3.14}$$

$$\mu_{k,t+1} = X_{t+1} \tag{3.15}$$

$$\eta^2_{k,t+1} = LargeInitialVariance \tag{3.16}$$

(I) Once the parameters maintenance is made, foreground detection can be made and so on, we continue the same steps for every frame.

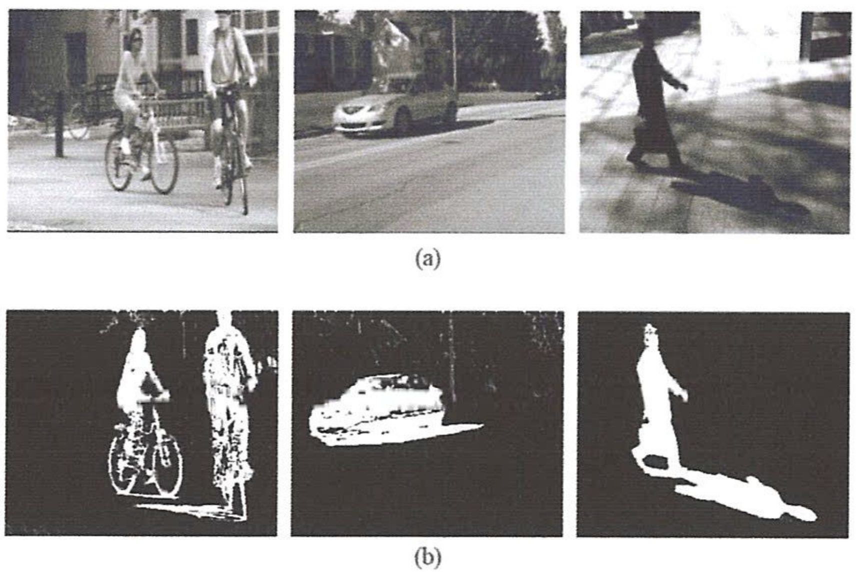


Figure 3.4: Background Subtraction Using MOG : (a) The first row presents original scenes[48]. (b) The second row shows the corresponding foreground masks obtained by the MOG[43]

5 Conclusion

In this chapter, we presented a preamble on the definition and the most used methods of object detection and tracking in video surveillance in general. In addition, we highlighted the deferent challenges where these methods could lose their efficiency.

We explained also, in detail, the concept of background modeling, and how it was established by the original Gaussian Mixture Model method. Based on this explanation, and the result of GMM method, we will propose an improved GMM approach for background modeling in the next chapter.

Chapter 4

Improved Adaptive Gaussian Mixture Model for Background Subtraction

1 Introduction

In this chapter, we will discuss the limitations of the original Mixture Of Gaussians model algorithm, and the reason why we should improve it. Then, we will focus on our new approach, and we'll discuss in details ~~the~~ steps we followed to improve the original MOG model method.

2 Limitations of GMM algorithm

The MOG model deals with the movement in the background (MB), and the gradual illumination changes (TD) greatly, due to the multi-modality in the representation step. However, the original GMM approach has its disadvantages that come along with assumptions and parameters that build the core of the algorithm. These are some :

- Handling sleeping foreground objects (SFO) : The original GMM model detects foreground objects, then removes them if they stay a while in the scene, as a part of the process of updating background model.
- Handling critical situations, light switch (LS) : Quick illumination changes in the background are detected as a foreground which implies a false detection[46].
- Bootstrapping problem (B) : For BGS it is valuable to be able to initialize the algorithm quickly, even as fast as using an only single frame, which is known as bootstrapping[49]. Unfortunately, original GMM needs time to initialize the first model of the background.
- Pre and post-processing : Furthermore, some critical situations need pre-processing or post-processing; the case of noise image (NI), camera jitter (CJ) and Camera automatic adjustments (CA).
- Using RGB : this can permit to make well shadows detection (S) like in Figure 3.4.

3 Literature of GMM Improvements

To solve these different limitations, many improvements can be found in the literature of BGS based on this model. Proposed improvements are classified to two major classes

(A) Intrinsic Model Improvements : concern directly the MOG model like the initialization and the maintenance of the parameters, and the foreground detection, by improving the core of the algorithm.

(B) Extrinsic Model Improvements : concern the conditions that surround the algorithm, using the knowledge of temporal and spatial information we add one or more external processes without making any changes to the core algorithm.

Intrinsic Model Improvements

- Number of Components - K
- Initialization of the Weight, the Mean and the Variance
  - By using another algorithm for the initialization
  - By allowing presence of foreground objects in the training sequence
- Maintenance of the Weight, the Mean and the Variance
- Learning rates a and p
  - By using better settings
  - By adapting the learning rates
- Threshold T
- Foreground Detection
  - By using a different measure for the matching test
  - By using a Pixel Persistence Map (PPM)
  - By using the probabilities
  - By using a foreground model
  - By using some matching tests
  - By using the most dominant background model
- Feature Size
  - Block-wise approaches
  - Cluster-wise approaches
- Feature Type
  - Color features
  - Edge features
  - Texture features
  - Stereo features
  - Spatial Features
  - Motion Features
  - Video Features

4 Our Method

4.1 Hypothesis

In order to improve GMM algorithm in video surveillance, we tend to develop a new algorithm based on both : the original GMM and the following hypothesis :

> In a very small temporal window $(1/24)s$, between the frame used to generate the Gaussian and the frame which subtraction is being applied on, if the density of a pixel shifts very far from the mean, then it's more likely to be "camera noise", and it's considered as so.

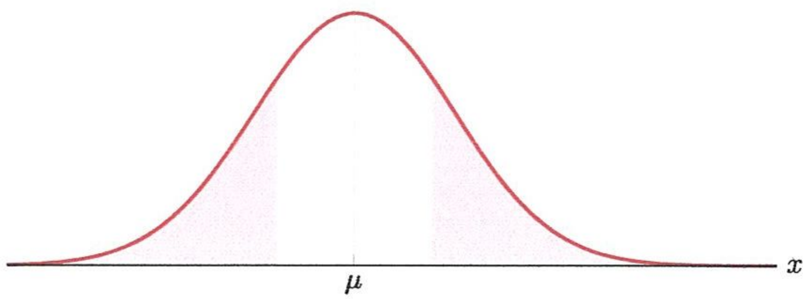


Figure 4.1: Camera noise in a Gaussian : following the previous hypothesis, the zone considered as camera noise is in red.

This means, that pixels that doesn't shift too far from the mean using the old Gaussian, are more likely to be background, and they are considered as so.

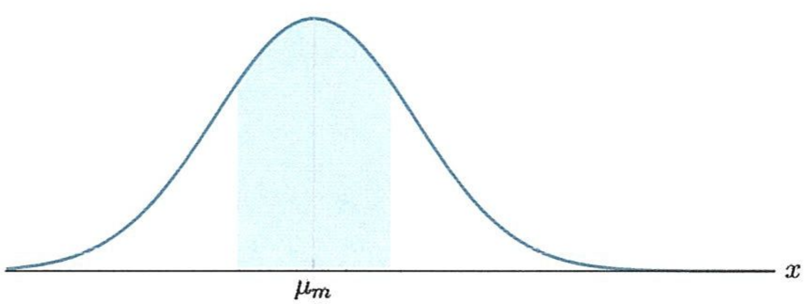


Figure 4.2: Background intensities in a Gaussian : following the previous hypothesis, the zone used for background modeling is in blue.

The hypothesis of the algorithm can be summarized in the following Figure 4.3 :

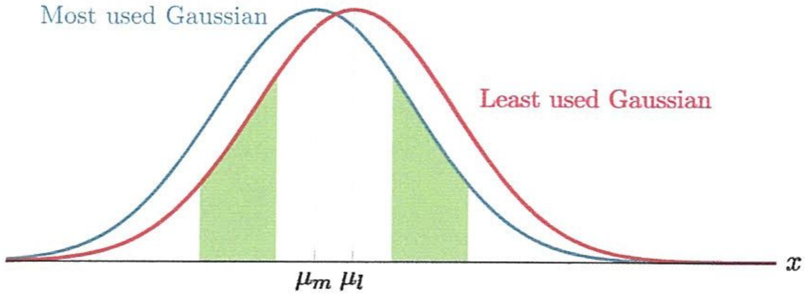


Figure 4.3: Target Zone of the Algorithm

4.2 Core Algorithm

4.2.1 Initialization

- After acquiring the first frame, we begin by the process of image segmentation. In order to apply segmentation we used two deferent approaches :
    1. **Color Segmentation** : Segmenting image to zones having same or close color density.
    2. **Bloc Segmentation** : By dividing the frame into equal square surfaces (or boxes).
- Each zone contains pixels coordination, in addition to μ_{ideal} a value called "ideal mean". The ideal mean μ_{ideal} is meant to represent the intensity of the zone, when it's all covered by background. And it's calculated by finding the average values of every Gaussian's mean associated to the zone in all times.

4.2.2 Background Modeling

Before getting into foreground detection, any subtraction algorithm should have a reference model representing the background, which the system will use to deduct whether there are new objects presented into the scene or not.

- To obtain the background model, first, we generate one Gaussian for every zone in the first frame.
- For the next frame, we do the same thing, but we gather the generated Gaussians of every zone in a set of Gaussian Mixture. (for 9 frames we get 9 Gaussians for every zone). K
- The number of Gaussians in the Gaussian Mixture is limited $K = 10$, so, generating Gaussians by time will follow one of these two cases :

1. $k_{gen} < K$: The number of generated Gaussians k_{gen} didn't reach the limit, so we add the new Gaussian to the set of Gaussian Mixture.
2. $k_{gen} = K$: The ensemble of Gaussians is complete, before adding the new Gaussian, we merge the two least used Gaussians into one, then we add the new one.

- The merge of Gaussians is based on the following conditions :
  - We keep the Gaussian with the closer mean to the ideal mean.
  - If both Gaussians in are not very far from the ideal mean, then we keep the one with the smaller standard deviation SD.

Since using single frame model (in other BGS Algorithms) helps detecting any new object and keep them highlighted no matter how long they stay (including lighting changes and noise.. etc.), and using a set of renewable frames (like in original GMM Algorithm) helps getting a more adaptive model (eliminates noise and fast light changes) but removes objects that stay too long in the scene. To achieve best of both, the new algorithm relies on a value previously mentioned in the document by the term "ideal mean", which helps creating an adaptive background model represented in one single image.

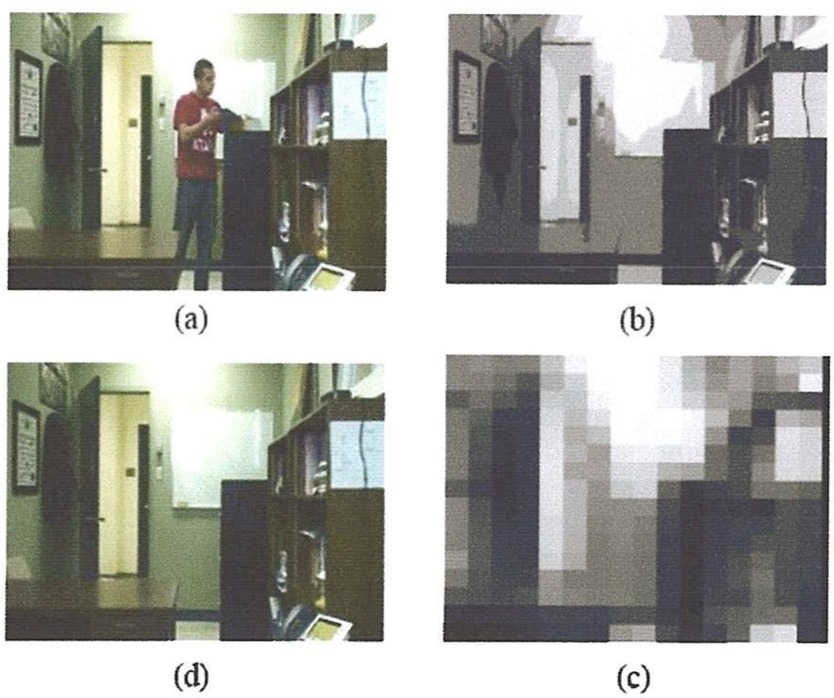
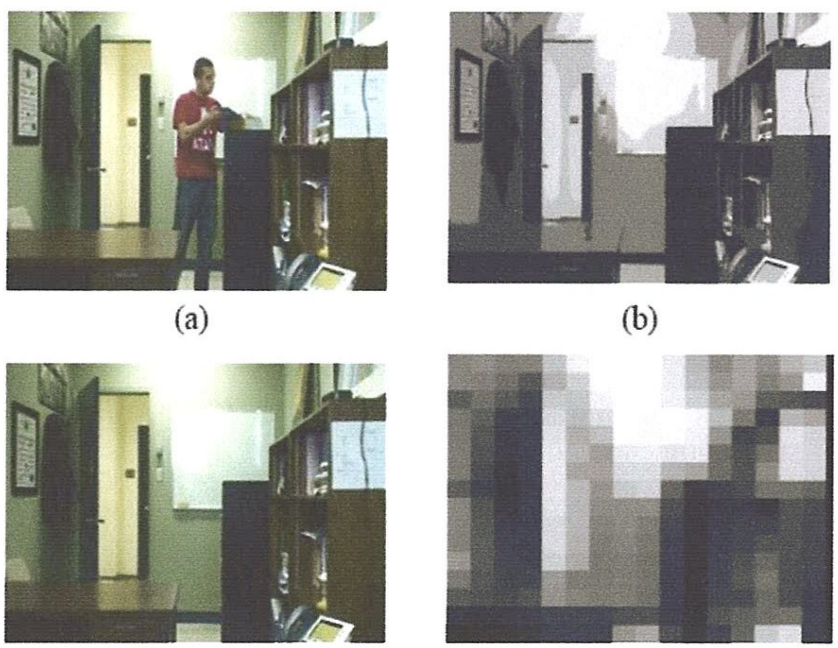


Figure 4.4: Adaptive Background Modeling in Our Approach : (a) represents a frame at time t, (b) and (c) represent the background model for the frame at time t, (d) represents the very first frame at time t_0

1. $k_{gen} < K$: The number of generated Gaussians k_{gen} didn't reach the limit, so we add the new Gaussian to the set of Gaussian Mixture.
2. $k_{gen} = K$: The ensemble of Gaussians is complete, before adding the new Gaussian, we merge the two least used Gaussians into one, then we add the new one.

- The merge of Gaussians is based on the following conditions :
  - We keep the Gaussian with the closer mean to the ideal mean.
  - If both Gaussians in are not very far from the ideal mean, then we keep the one with the smaller standard deviation SD.

Since using single frame model (in other BGS Algorithms) helps detecting any new object and keep them highlighted no matter how long they stay (including lighting changes and noise.. etc.), and using a set of renewable frames (like in original GMM Algorithm) helps getting a more adaptive model (eliminates noise and fast light changes) but removes objects that stay too long in the scene. To achieve best of both, the new algorithm relies on a value previously mentioned in the document by the term "ideal mean", which helps creating an adaptive background model represented in one single image.



(a) (b)

(d) (c)

Figure 4.4: Adaptive Background Modeling in Our Approach : (a) represents a frame at time t, (b) and (c) represent the background model for the frame at time t, (d) represents the very first frame at time t_0

4.2.3 Background Maintenance

In this step we calculate both overall weight of the GM, and their current weight as well using the following algorithm :

Algorithm 1 Weight Updating Algorithm

for each zone **do**
  for each Gaussian associated to the zone **do**
    for each pixel in the zone **do**
      if | pixelIntensity - μ | $< SD$ **then** $\triangleright$ $SD = 2.5$
        overallWeight ++
        currentWeight ++
      else
        overallWeight –
      end if
    end for
  end for
end for

4.2.4 Background Subtraction

In order to apply subtraction, we start by finding the most used Gaussian, and least used Gaussian in the Gaussian Mixture according to their current weight calculated in the step before.
Then, pixels are classified "background / foreground" using the following algorithm :

Algorithm 2 Background Subtraction Algorithm

for each zone **do**
  for each Gaussian associated to the zone **do**
    for each pixel in the zone **do**
      if | pixelIntensity - μ_m | $> X_{SD}$ **and** | pixelIntensity - μ_l | $< Y_{SD}$ **then**
        pixel = background
      else
        pixel = foreground
      end if
    end for
  end for
end for

Where :

μ_l : is the mean of the least used Gaussian, μ_m : is the mean of the most used Gaussian, and X_{SD} and Y_{SD} : are predefined values.

To better understand how least and most Gaussians are affecting the foreground/background segmentation, Figure 4.5 shows the result we get using only the most used Gaussian to determine the camera noise, to set it as background. And Figure 4.6 shows the result of using only the least used Gaussian to find the background, wheres Figure 4.7 shows the result of using least and most used Gaussians combined in the whole algorithm.

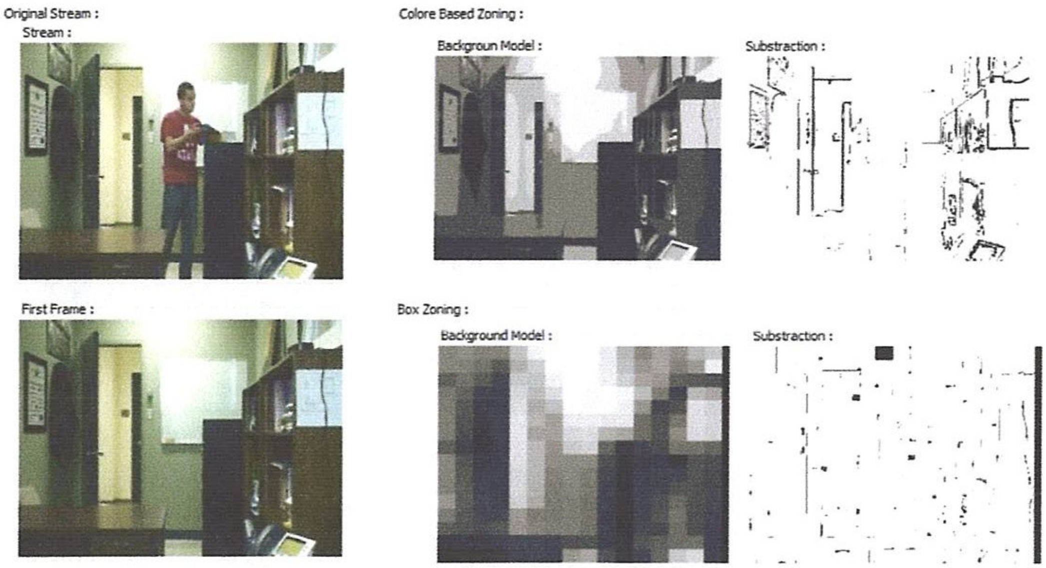


Figure 4.5: Using Most Used Gaussian to Determine the Camera Noise

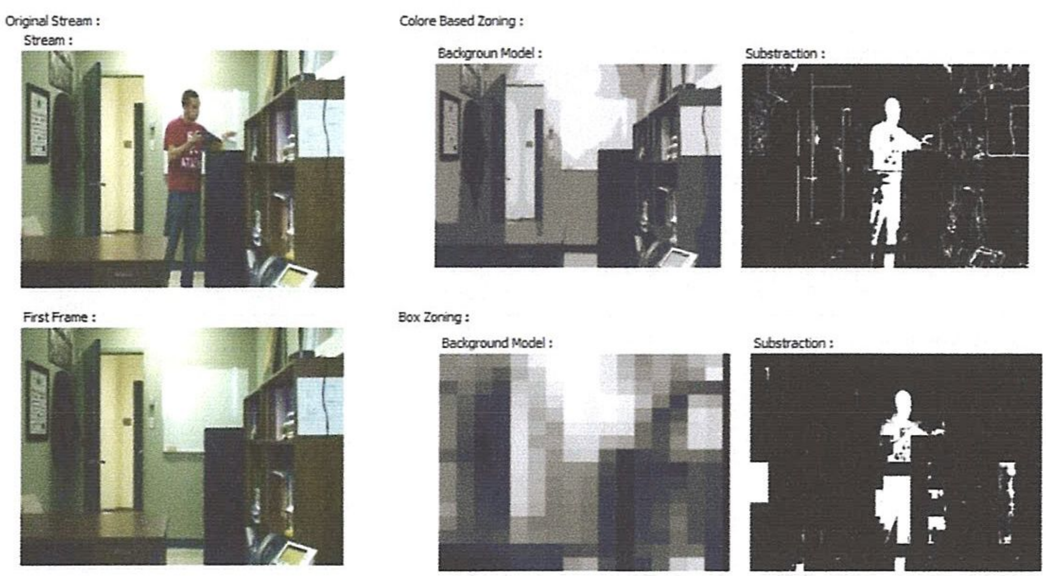


Figure 4.6: Using Most Used Gaussian to Determine the Background

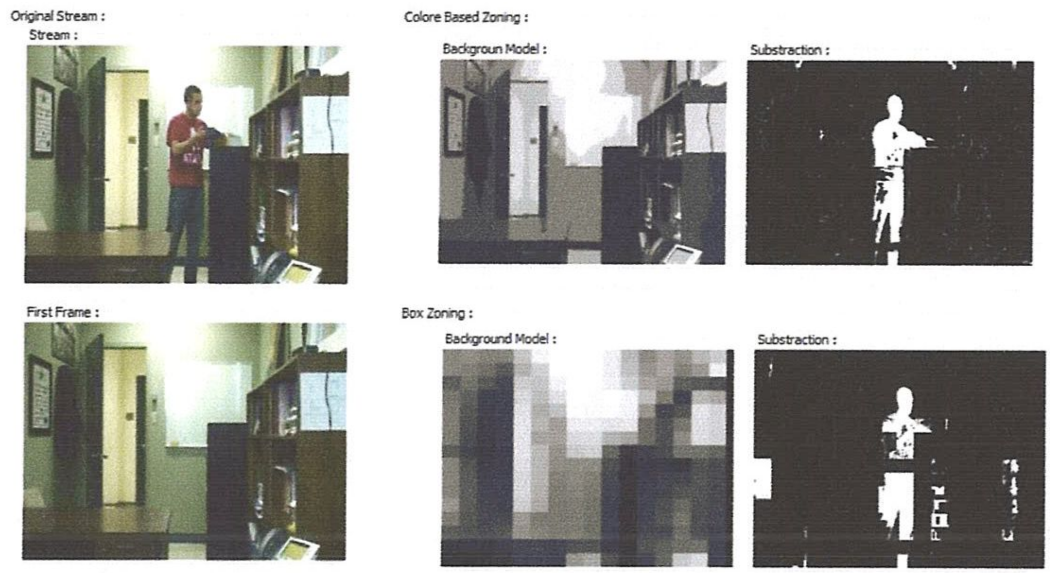


Figure 4.7: The Final Result Using Least and Most Used Gaussians Combined

Our Method is summarized in the next flow chart in Figure 4.8.

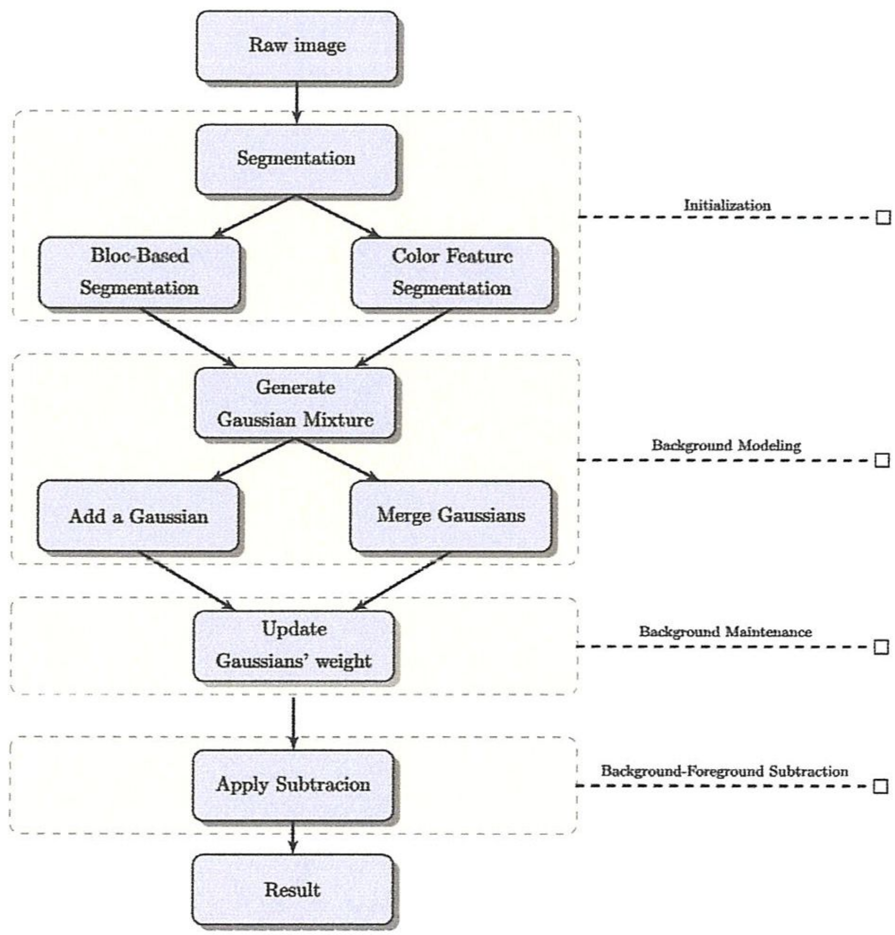


Figure 4.8: General Flow Chart of Our Approach for the Improved GMM Algorithm

Chapter 5

Implementation and Tests

1 Introduction

After presenting the deferent steps to apply both original GMM method and our new Adaptive GMM method in the precious chapter, it is time in this chapter to present our own implementation of our method, besides the detailed tests and results on deferent international well designed data sets to get better and rational judgments on the efficiency of the proposed Algorithms.

2 Conception

2.1 Application's Logic

In order to maintain a comprehensive programming approach for the implementation of the proposed Gaussian Mixture Model methods, we used the Oriented Object design pattern. Using Oriented Object Programming helps enforcing good software design principles like encapsulation, cohesion, low coupling, etc. It uses the concept of Class-Object and makes the source code more readable and understandable, and therefore ultimately more maintainable[50]. These are the classes of our application project grouped by package :

Application Project		
Package	**Class**	**Description**
appcore	BxGMM	Handling Block based GMM method.
	Engine	The main thread, feeds frames to the running method and updates the main frame.
	RxGMM	Handling Region based GMM method.
appcore.helper	MapGenerator	Segmentation of images for Rx-GMM.
	Pixel	A better definition of a pixel with its coordinates.
	Seeker	Loops through image seeking for unclassified pixels for region-based segmentation.
	UnZipper	Unzipping zipped files of deferent integrated datasets.
	ZoneBoudaries	Defining a zone for Rx-GMM.
benchmark	BenchmarkedDataset	Contains deferent metrics after benchmarking a specific dataset.
	BenchFrame	Frame for selecting result and groundtruth frames.

	Benchmarker	Calculating deferent benchmark metrics after comparing groundtruth frames with result frames.
dataset	No - Class	Contains zipped files of ready datasets
images	No - Class	The images used to enhance user experience in deferent graphical user interfaces.
main	Main Statics	Running the main program Contains objects that must be known across all classes and packages of the application.
ui	DatasetChooser MainFrame	Offering multiple input methods for selecting a database. The main frame of the application.

Table 5.1: Deferent Packages and Classes used in the application's project

2.2 Application's Graphical User Interface

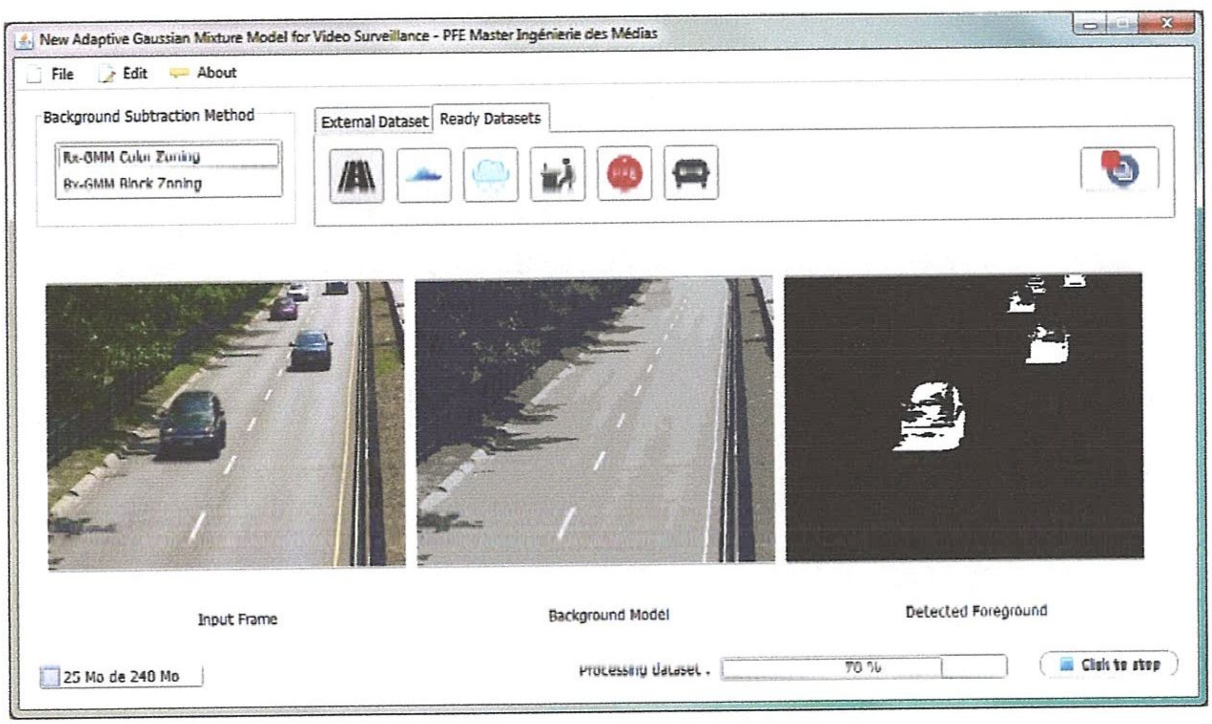


Figure 5.1: Application's Main GUI

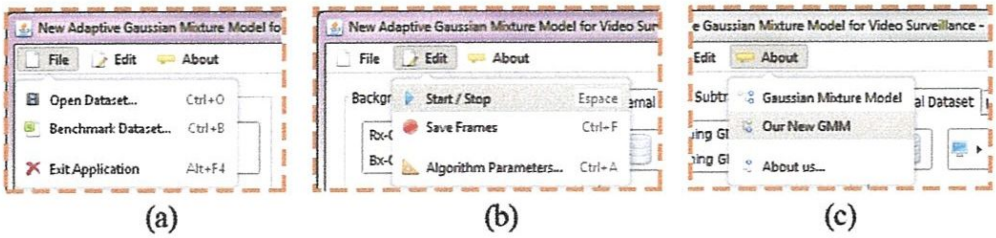


Figure 5.2: Ergonomic Menus : (a) File menu with the possibility of saving benchmark result to an Excel file, (b) Edit menu to control the run of the algorithm, (c) About Menu for more information

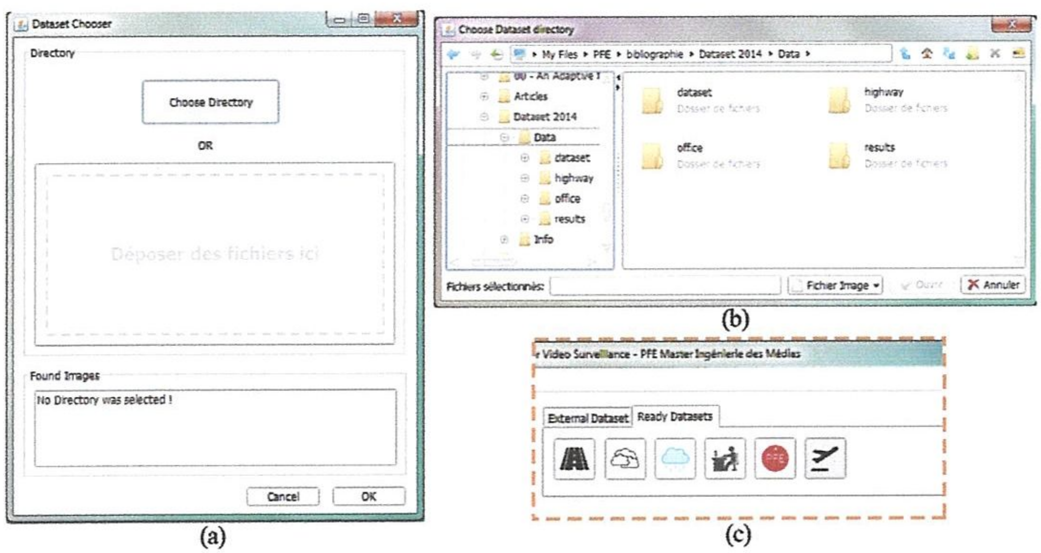


Figure 5.3: Multiple Handy Dataset Input methods : (a) Easy to use method by dragging and dropping image files to import them, (b) import datasets using the file chooser, (c) Ready datasets to select and use on the go

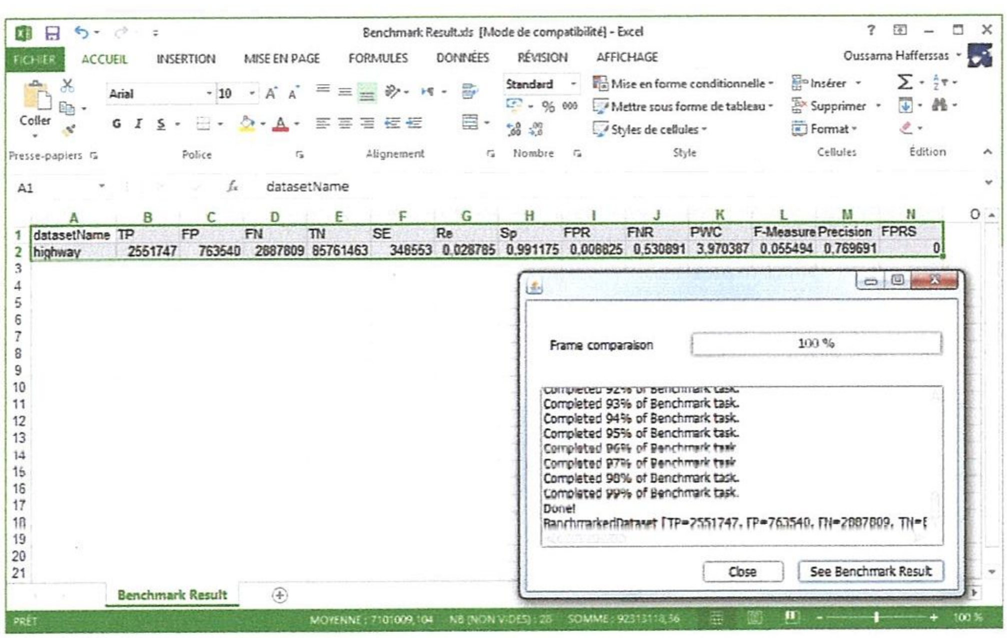


Figure 5.4: Excel-Ready Benchmark Results : After Benchmarking, the results are automatically saved in an Excel file for better interpretation later.

3 Implementation

3.1 Software

3.1.1 Programming Language

In order to implement our method, we chose the *Java programming language.* In addition of having a good programming experience with it, We have chosen *Java* because it's one of the best cross-platform programming languages.
Developed by *James Gosling* at *Sun Microsystems* in 1990, *Java* is an object-oriented programming language available for an array of development requirements; right from mobile application to enterprise web applications, web services to desktop applications. Due to its long standing formula (WORA) "Write Once and Run Anywhere", Java remains a very popular programming language.
We used the Java Development Kit JDK version 8 update 45[51].

3.1.2 Integrated Development Environment

As a development environment, we used *IntelliJ IDEA Ultimate 14.1.2* to manage our source code with more efficiency. *IntelliJ IDEA Ultimate* is a paid IDE, but it's one of the most intelligent *Java* IDEs in current time. It's used by *Google*, and main big software companies for developing their own deferent and powerful software solutions. One of the greatest things about this IDE, is that it supports all the latest modern technologies and frameworks available out of the box[52].

3.1.3 Third-Party Libraries

- **JavaCV 0.6** : A Java library that uses wrappers from the *JavaCPP* Presets of commonly used libraries by researchers in the field of computer vision (*OpenCV*, *FFmpeg*, *libdc1394*, *PGR FlyCapture*, *OpenKinect*, *videoInput*, *ARToolKitPlus*, and *flandmark*), and provides utility classes to make their functionality easier to use on the *Java* platform[53]. *JavaCV* also comes with hardware accelerated full-screen image display, we used it to better manage image feed from both camera and dataset.

- **JavaFX 8 SDK** : *JavaFX* affords a very customizable, beautiful, and rich user interface in addition of a great tools to draw graphs for statistics and math plots[54]. We used this library for plotting Gaussians.

- **WebLaF 1.29** : *WebLookandFeel* is a *Java Swing* Look and Feel and extended components library for cross-platform applications[55]. We used it to give our application's user interface a simple and stylish cross-platform default theme.

- **Apache Commons IO 2.4** : One of the best of Iava libraries which assists with developing input / output functionalities with its verity of utilities[56].

- **Apache POI 3.12** : a Java API for manipulating various file formats based upon the Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound Document format (OLE2)[57]. We used it to save benchmark results in *Microsoft Excel* file (.xls)

3.1.4 Test Operating Systems

In order to really test the behavior of our application, we tested it on two major operating systems; a commercial one : *Microsoft Windows* and an open source OS : *Linux.*
On *Windows* we tested on *Windows* 7 SP1 x64 and the latest *Windows* 8.1 x64.
On *Unix* we selected it's popular distribution called *Ubuntu* 14.04 LTS 64 bit (Long Term Service) which delivered the shortest execution time.

3.2 Hardware

We used 3 deferent machines to test the application :

- **Costum Desktop** : CPU : Intel i3 - 3210 @ 3.20 GHz, RAM : 4 Go DDR3, GPU : AMD Radeon HD 6570 (4 Go GRAM).
- **HP Laptop** : CPU : Intel i5 - 3230M @ 2.60 GHz, RAM : 4 Go DDR3, GPU : NVIDIA GeForce 820M (3 Go GRAM).
- **Lenovo Laptop** : CPU : Intel i7 - 3632QM @ 2.20 GHz, RAM : 6 Go DDR3, GPU : NVIDIA GeForce

4 Tests and Results

In order to test our method, we followed the scientific community of video surveillance discipline for testing new algorithms, in which, they tend to test algorithms applied in video surveillance on multiple specialized datasets. Except for basic datasets, each dataset contains input frames and ground-truth frames. Input frames represent a challenge of real life in video surveillance systems, and ground-truth frames are made manually and they represent what should the perfect result look like, so we can avoid relativity when we evaluate the results.

In order to do so, we selected a new change detection benchmark dataset that was evaluated for the IEEE Change Detection Workshop 2012 [58], and its expanded version of the IEEE Change Detection Workshop of 2014 [59], but we used also another video of our own. All Dataset characteristics are given in Table 5.2 :

Dataset Name	**Video Categories**	**Frame Size**	**Number of Frames**	**Groundtruth Frames**
Highway[58]	Baseline	320 × 240	1700	Yes
Office[58]	Baseline	360 × 240	2050	Yes
Sofa[58]	Intermittent Object Motion	320 × 240	2750	Yes
Hallway	Random Dataset	640 × 480	3402	No
Snowfall[59]	Bad Weather	720 × 480	6500	Yes
Turbulence[59]	Turbulence	720 × 480	5000	Yes

Table 5.2: Dataset Characteristics

Our tests are classified by dataset type as follow :
For all next Figures : Figure 5.5, Figure 5.6, Figure 5.7, Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11, and Figure 5.12, Figure 5.13 :
(a) is the original frame (n°n),
(b.1) and (b.2) in order, represent the background model obtained by our method when using color feature-based segmentation, and the result frame of the subtraction,
(c.1) and (c.2) in order, represent the background model obtained by our method when using bloc-based segmentation, and the result frame of the subtraction,
(d) the result of subtraction in original GMM[1].

4.1 Baseline

These sets are fairly easy, but not trivial to process. They are provided mainly as reference[58].

4.1.1 Highway

This sequence represents a basic example of traffic in a highway in daytime in good weather and light conditions. The background has a static part -which is the road-, and a dynamic part -which is the trees- with shadow on the road[58].

[1]We obtained Background subtraction results of original GMM from this proposed massive evaluation of BS algorithms called : BGSLibrary [60] [61], Current version[62] is composed of 37 BGS methods, provided in an easy-to-use C++ framework based on OpenCV, it was referenced in [43] for evaluation and benchmarking change detection algorithms.

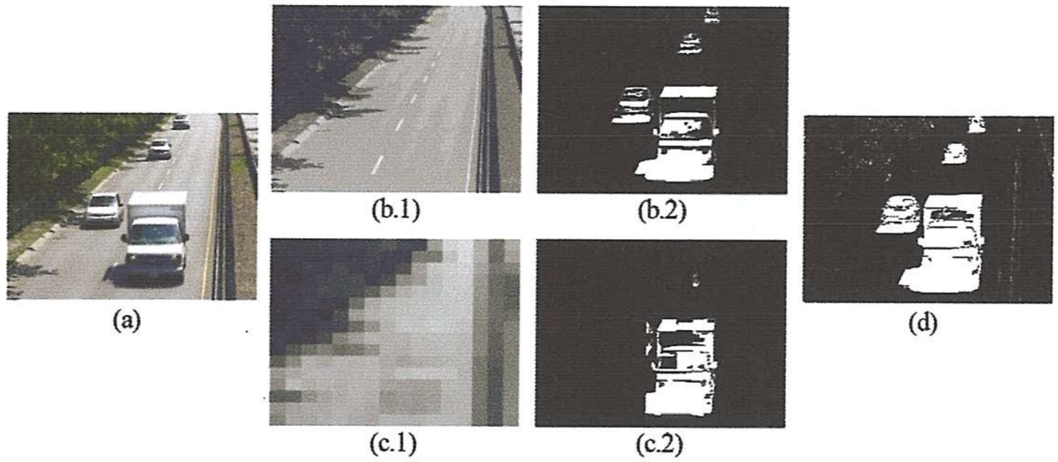


Figure 5.5: First Test on Highway set : Frame N°271, Real Size : 320×240 *px*

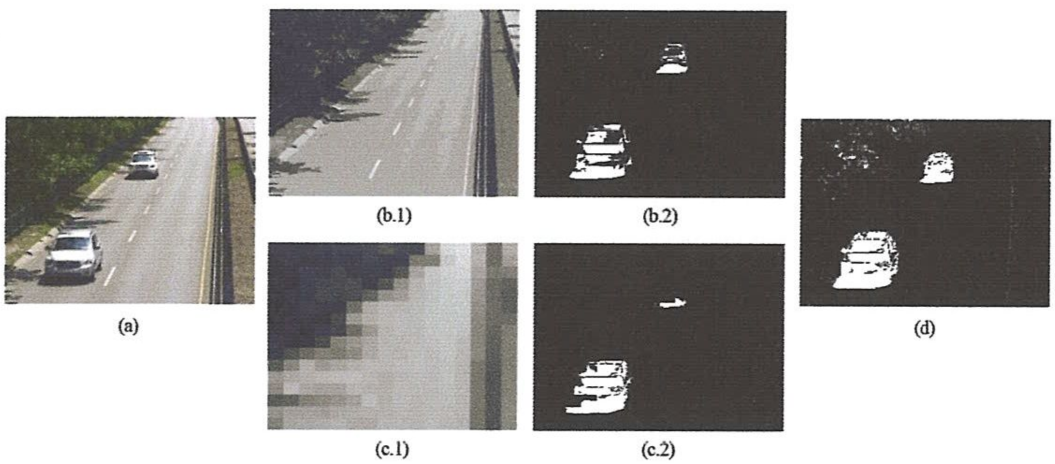


Figure 5.6: Second Test on Highway set : Frame N°375

Rx-GMM	**Bx-GMM**	**Original GMM**
(+) All new objects are highlighted.	(-) NOT all new objects are highlighted.	(+) All new objects are highlighted.
(-) some objects have some missing parts.	(-) some objects have some missing parts (Black Box effect).	(-) some objects have some missing parts.
(-) hard shadows are detected.	(-) hard shadows are detected.	(-) hard shadows are detected.
(+) Dynamic out of interest regions are set as background with minimal noise.	(+) Dynamic out of interest regions are set as background with minimal noise.	(-) Dynamic out of interest regions are set as background with lot of noise.

Table 5.3: Comparison Between Algorithms' Performance in the Highway Dataset

4.1.2 Office

This sequence represents indoor surveillance example in an office, where the person of interest (PoI) is a student. This set contains two challenges : Light switch (LS) challenge, when the student enters the office, and affects its illumination, and the second challenge is the sleeping foreground object (SFO) when the students stand still reading the book[58].

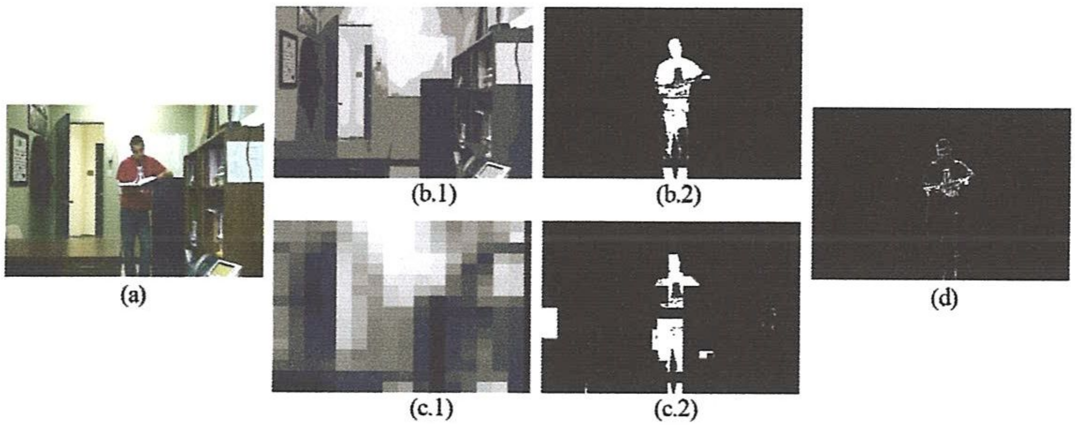


Figure 5.7: Second Test on Highway set : Frame N°1159, Real Size : 320×240 *px*

Rx-GMM	Bx-GMM	Original GMM
(+) PoI is highlighted when he enters the scene.	(+) PoI is highlighted when he enters.	(+) PoI is highlighted when he enters.
(+) PoI is always detected even if he was totally stable.	(+) PoI is always detected even if he's not moving.	(-) PoI is almost NOT detected if he's not moving.
(+) RoI are detected without missing parts.	(-) Some parts are missing from PoI. (Black Box effect).	(-) Some parts are missing from PoI.
(+) Light distortions are not detected when PoI enters.	(-) Light distortions are detected when PoI enters .	(+) Light distortions are not detected when PoI enters.
(+) Background is set with minimal noise.	(+) Background is set with minimal noise.	(+) Background is set with minimal noise.

Table 5.4: Comparison Between Algorithms' Performance in the Office Dataset

4.1.3 Sofa

This set is under the *intermittent object motion* dataset. In this sequence, people come and sit for a short period of time in the sofa, they go and they leave small objects in the scene.

This set represents the challenge of abandoned objects and objects stopping for a short while and then moving away[59].

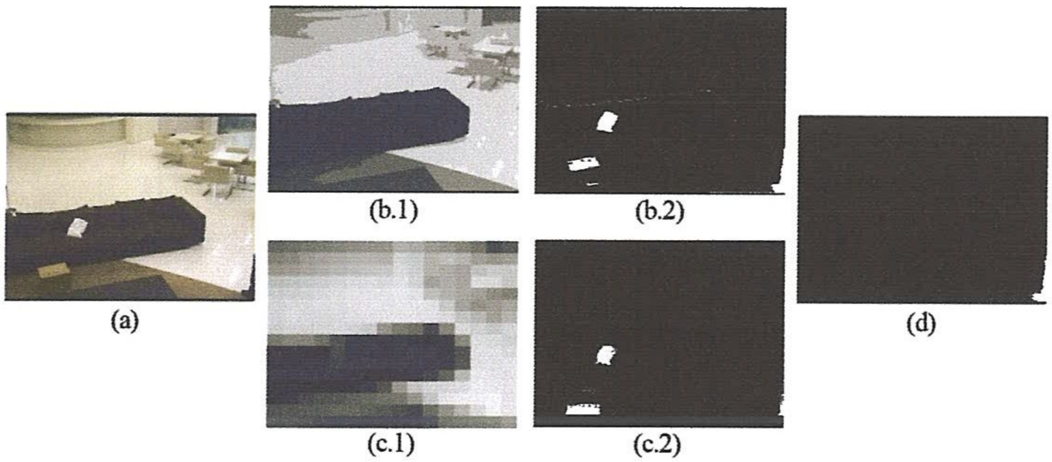


Figure 5.8: Test on Sofa set : Frame N°1124, Real Size : 320×240 px

Rx-GMM	Bx-GMM	Original GMM
(+) RoI are highlighted when they enter the scene. (+) Small abandoned objects are always detected. (-) Background is set with noise.	(+) RoI are highlighted when they enter the scene. (+) Small abandoned objects are always detected. (+) Background is set with minimal noise.	(+) RoI are highlighted when they enter the scene. (-) Small abandoned objects are not detected. (+) Background is set with minimal noise.

Table 5.5: Comparison Between Algorithms' Performance in the Sofa Dataset

4.1.4 Our dataset - Hallway

This set represents a randomly selected scene, that we filmed in order to test algorithms on non manipulated frames. This scene was filmed inside our computer science department. It represents multiple challenges : Long term vertical camera jitter(CJ), High Definition camera (1280×720 px), Camera with automatic adjustments (CA), and foreground sleeping object (FSO).

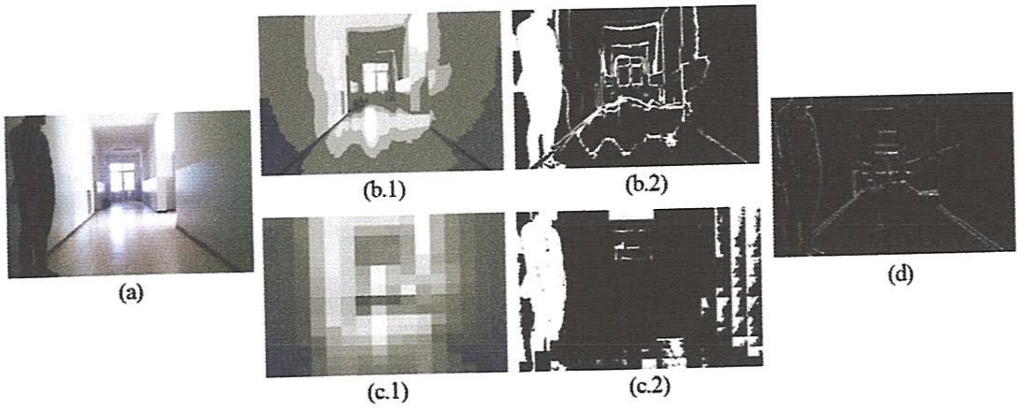


Figure 5.9: First Test on our Hallway set : Frame N°1991, Real Size : 640×480 *px*

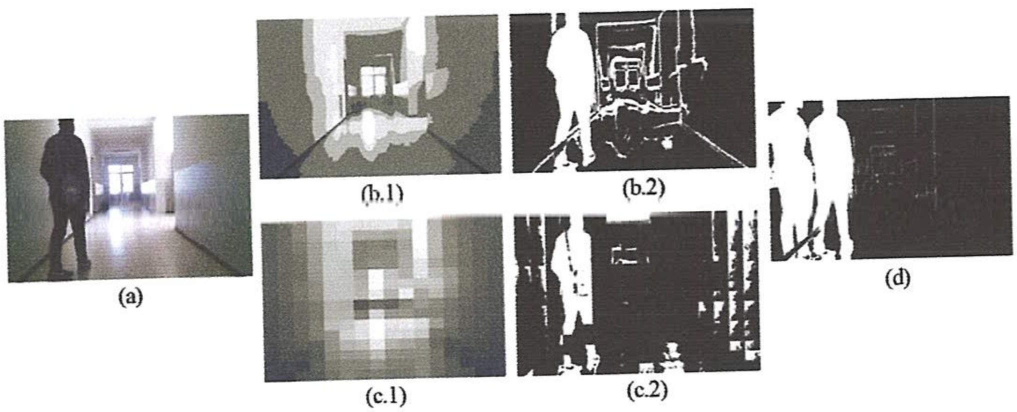


Figure 5.10: Second Test on our Hallway set : Frame N°3003

Rx-GMM	Bx-GMM	Original GMM
(+) PoI is highlighted when he enters the scene. (+) PoI is always detected when he is totally stable. (+) No ghosting effect when PoI leaves his stable position. (-) Sensitive to camera jitter.	(+) PoI is highlighted when he enters. (+) PoI is always detected when he is totally stable. (+) No ghosting effect when PoI leaves his stable position. (-) Sensitive to camera jitter.	(+) PoI is highlighted when he enters. (-) PoI is almost NOT detected if he's not moving. (-) Ghosting effect when PoI leaves his stable position. (+) Not sensitive to camera jitter.

Table 5.6: Comparison Between Algorithms' Performance in the Hallway Dataset

4.2 Bad weather

A sequence that represents outdoor videos showing low-visibility winter storm conditions.

4.2.1 Snow fall

This set includes traffic scene in a heavy snow fall. It represents a double challenge: in addition to snow accumulation, the dark tire tracks left in the snow have potential to cause false positives, and the heavy snow causes a strong noise in images[59].

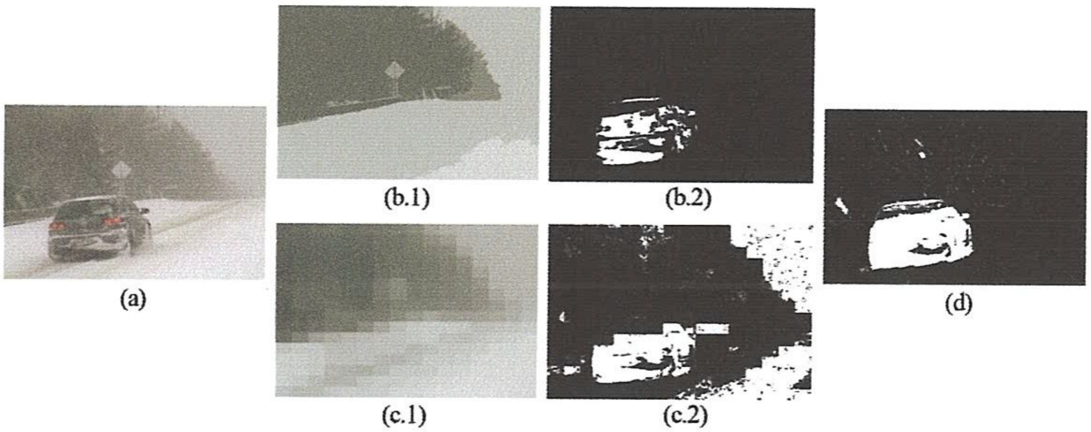


Figure 5.11: Test on Snowfall set : Frame N°816, Real Size : 540×630 px

Rx-GMM	Bx-GMM	Original GMM
(+) RoI are highlighted when they enter the scene. (+) Tire traces are not detected. (+) Background is set with minimal noise.	(+) RoI are highlighted when they enter. (+) Tire traces are not detected. (-) Background is set with strong noise.	(+) RoI are highlighted when they enter. (+) Tire traces are not detected. (-) Background is set with noise.

Table 5.7: Comparison Between Algorithms' Performance in the Snowfall Dataset

4.2.2 Blizzard

4.3 Air Turbulence

This set is moving objects at noon during a hot summer day. The scene is filmed at a distance 5 km with a telephoto lens, the heat causes constant air turbulence and distortion in frames. This results in false positives. The size of the moving objects is small[59].
The air turbulence category presents very similar challenges to those arising in long-distance remote surveillance applications.

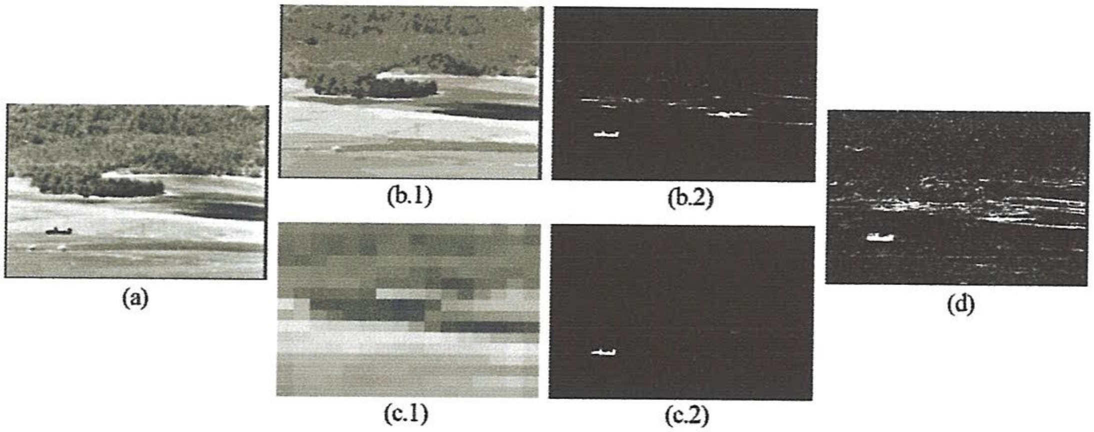


Figure 5.12: Test on Turbulence set : Frame N°368, Real Size : 720×480 px

Rx-GMM	Bx-GMM	Original GMM
(+) RoI are highlighted when they enter the scene. (+) Small objects are detected. (-) Lots of false positives caused by air turbulence.	(+) RoI are highlighted when they enter. (+) Small objects are detected. (+) Fewer false positives.	(+) RoI are highlighted when they enter. (+) Small objects are detected. (-) Lots of false positives caused by air turbulence.

Table 5.8: Comparison Between Algorithms' Performance in the Turbulence Dataset

4.4 Dynamic Background

This sequence represents scenes with strong (parasitic) background motion. In which we find boats on shimmering water[58].

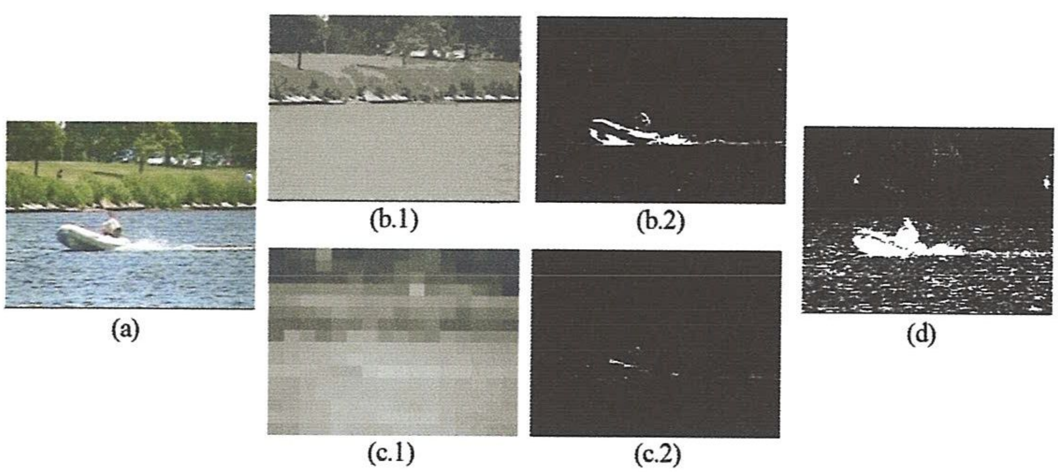


Figure 5.13: Test on Boats set : Frame N°2000, Real Size : 320×240 px

Rx-GMM	Bx-GMM	Original GMM
(+) RoI are highlighted when they enter the scene. (-) Detected objects have missing parts (+) Dynamic background (water) is set with minimal noise. (+) Far objects are detected.	(+) RoI are highlighted when they enter. (-) Detected objects have lots of missing parts (+) Dynamic background (water) is set almost without noise noise. (+) Far objects are detected.	(+) RoI are highlighted when they enter. (+) Detected objects don't have missing parts (-) Dynamic background (water) is usually detected as foreground. (+) Far objects are detected.

Table 5.9: Comparison Between Algorithms' Performance in the Boats Dataset

5 Benchmarking Results

5.1 Evaluation Metrics

After testing our two deferent GMM improvements, comparing them against other GMM methods is a necessity. By benchmarking the methods, we can use the right metrics that accurately measure the ability of a method to detect the right foreground and the right background. These metrics must be obtained after testing on the same scenes or datasets. The next steps clarify how exactly we benchmarked deferent GMM methods :

- We used the same datasets [58] [59].
- These datasets have, for each set, and for each input frame, a groundtruth frame that contains precise ground truth labels for every pixel Figure 5.14 (All pixels ware manually labeled a number of times by deferent researchers).
- Each pixel has a label that represents one of these evaluation metrics[58] :
  - **Static** : assigned grayscale value of 0,
  - **Shadow** : assigned grayscale value of 50,
  - **Non-ROI** : assigned grayscale value of 85,
  - **Unknown** : assigned grayscale value of 170,
  - **Moving** : assigned grayscale value of 255.
- Performance of each method is obtained by calculating the next evaluation metrics across all frames of one set[58] :
  - TP : True Positive for correct foreground detection,

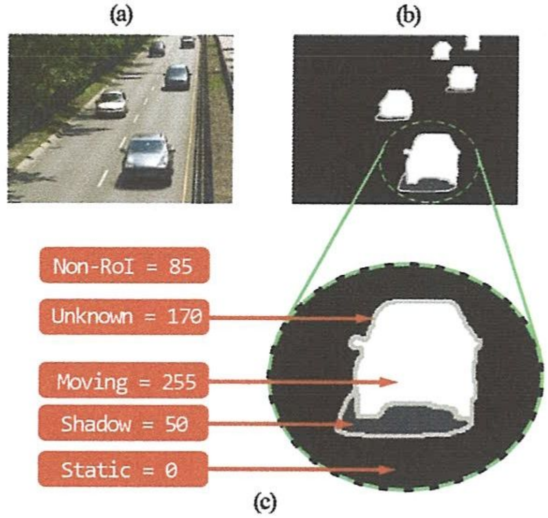


Figure 5.14: Evaluation Metrics in Groundtruth Frames : (a) Input frame N°1065 (b) Its groundtruth frame (c) Deferent groundtruth pixel labels

- TN : True Negative for correct background detection,
- FN : False Negative for false background detection,
- FP : False Positive for false foreground detection,
- SE : Shadow Error.

• The correctness of the results is expressed by two measures[63] :

- *Recall* : High recall means that the method returned most of the relevant results. It's calculated as follow :

$$Recall = \frac{TP}{TP + FN} \tag{5.1}$$

- *Precision* : High precision means that the method returned substantially more relevant results than irrelevant. it's obtained as follow :

$$Recall = \frac{TP}{TP + FP} \tag{5.2}$$

5.2 Results and interpretation

We have evaluated the performance of five GMM based methods : Rx-GMM (our region based GMM), Bx-GMM (our box based GMM), The Original GMM[45], and two improved GMMs : the improved GMM of Zivkovic et al.[64] and a block-based GMM RECTGAUSS-tex[65].

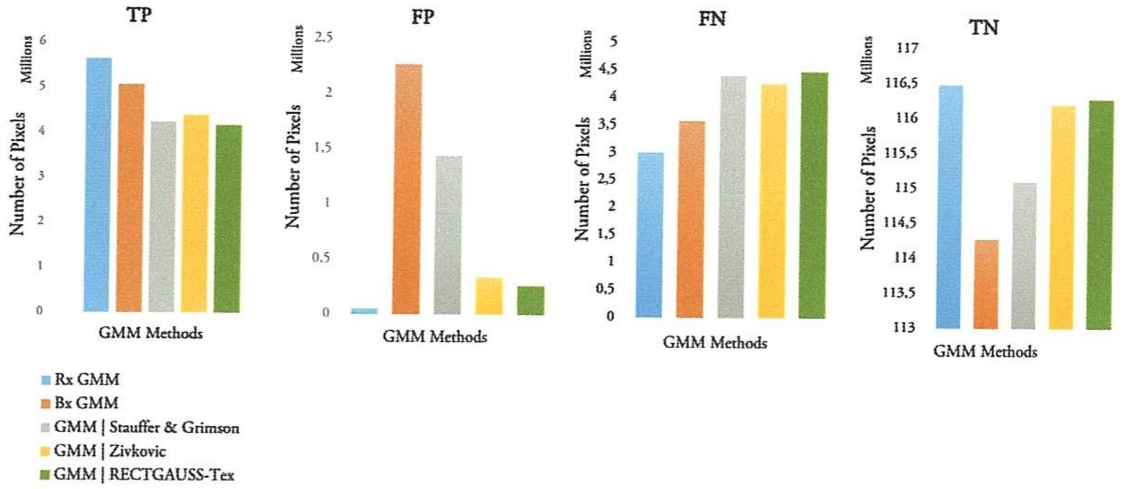


Figure 5.15: Benchmarking Resluts of GMM Methods on Office Dataset

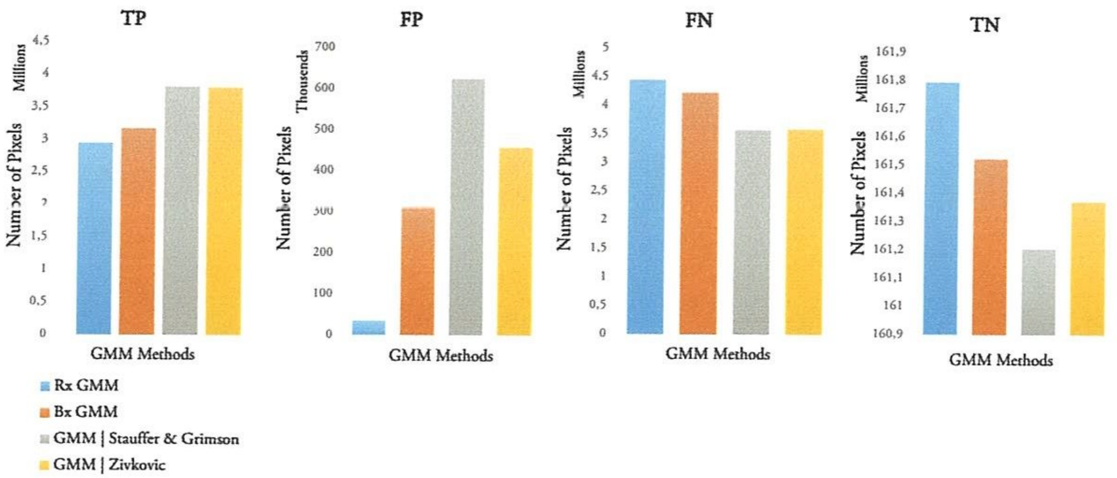


Figure 5.16: Benchmarking Resluts of GMM Methods on Sofa Dataset

Bibliography

[1] JonathanT. Erichsen and J.Margaret Woodhouse. Human and animal vision. In BruceG. Batchelor, editor, *Machine Vision Handbook*, pages 89–115. Springer London, 2012.

[2] Rudiger Wehner. Polarization vision: A discovery story. In Gabor Horvath, editor, *Polarized Light and Polarization Vision in Animal Sciences*, volume 2 of *Springer Series in Vision Research*, pages 3–25. Springer Berlin Heidelberg, 2014.

[3] AlexanderYa. Supin, VladimirV. Popov, and AllaM. Mass. Vision in aquatic mammals. In *The Sensory Physiology of Aquatic Mammals*, pages 229–284. Springer US, 2001.

[4] Uwe Homberg, Stanley Heinze, Keram Pfeiffer, Michiyo Kinoshita, and Basil el Jundi. Central neural coding of sky polarization in insects. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1565):680–687, 2011.

[5] NASA : National Aeronautics and Space Administration. What Wavelenght Goes With a Color ? http://science-edu.larc.nasa.gov/EDDOCS/Wavelengths_for_Colors.html, 2011. Accessed: 03-04-2015.

[6] Elaine Marieb. The eye and vision. In *Human anatomy and physiology*, pages 545–565. Pearson, 2013.

[7] Beau Lotto. TEDGlobal : optical illusions show how we see. http://www.ted.com/talks/beau_lotto_optical_illusions_show_how_we_see, 2009. Accessed: 07-03-2015.

[8] Richard Szeliski. Introduction. In *Computer Vision*, Texts in Computer Science, pages 1–25. Springer London, 2011.

[9] Microsoft. The era of holographic computing : microsoft hololens official site. http://www.microsoft.com/microsoft-hololens/en-us, 2015. Accessed: 08-03-2015.

[10] ECVision: The European Research Network for Cognitive Computer Vision Systems. A Research Roadmap of Cognitive Vision. www.ecvision.org, 2005. Accessed: 12-03-2015.

[11] Wiki Authors. Computer vision. http://en.wikipedia.org/wiki/Computer_vision, 2015. Accessed: 25-03-2015.

[12] Machine vision. In E.R. DAVIES, editor, *Machine Vision (Third Edition)*, Signal Processing and its Applications. Morgan Kaufmann, Burlington, third edition edition, 2005.

[13] Raster images. In Charles Poynton, editor, *Digital Video and HD"*, The Morgan Kaufmann Series in Computer Graphics, pages 3 – 18. Morgan Kaufmann, Boston, second edition, 2012.

[14] Paul. Read and Mark-Paul Meyer. In *Restoration of Motion Picture Film*, Conservation and Museology, pages 24–26. Butterworth-Heinemann, 2000.

[15] Video compression. In LING LIU and M.TAMER ÃZSU, editors, *Encyclopedia of Database Systems*, pages 3271–3271. Springer US, 2009.

[16] Gary J Sullivan. Overview of international video coding standards (preceding h. 264/avc). In *ITU-T VICA workshop, Geneva*, 2005.

[17] Richard J. Radke. *Computer Vision for Visual Effects.* Cambridge University Press, New York, NY, USA, 2012.

[18] RodeoFX. Demo RODEO FX : Game of Thrones, Season 4 â VFX breakdown. `http://www.rodeofx.com/showreels-fr/feature-reel-fr`, 2014. Accessed: 09-03-2015.

[19] SONY PlayStation. PlayStation Camera. `https://youtu.be/pLKGhCr4dQw?t=25s`, 2014. Accessed: 14-03-2015.

[20] CNET UK. Meet Amazon's busiest employee – the Kiva robot. `http://www.cnet.com/uk/news/meet-amazons-busiest-employee-the-kiva-robot/`, 2014. Accessed: 11-05-2015.

[21] Esther Bron, Marion Smits, John van Swieten, Wiro Niessen, and Stefan Klein. Feature Selection Based on SVM Significance Maps for Classification of Dementia. In Guorong Wu, Daoqiang Zhang, and Luping Zhou, editors, *Machine Learning in Medical Imaging*, volume 8679 of *Lecture Notes in Computer Science*, pages 272–279. Springer International Publishing, 2014.

[22] Computer Vision Software. Computer Vision steps up soldiers game. `http://computervisionsoftware.org/computervision-steps-up-soldiers-game/`, 2014. Accessed: 19-03-2015.

[23] VinTech Systems. Back to Basics: Where Did the Video Security System Come From? `http://www.vintechnology.com/journal/uncategorized/back-to-basics-where-did-the-video-security-system-come-from/`, 2011. Accessed: 15-04-2015.

[24] Ionu Inc. Internet Protocol (IP) CCTV. `http://ionu.co.uk/ip-cctv/`, 2014. Accessed: 18-04-2015.

[25] SAMSUNG. Intelligent Video Analytics in Samsung Total Security Solutions. `https://www.samsung-security.com/SAMSUNG/upload/Product_Specifications/Samsung_Full_Catalog_2014.pdf`, 2014. Accessed: 11-05-2015.

[26] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.

[27] SAMSUNG. Intelligent Video Analytics for Efficient Monitoring. `https://www.samsung-security.com/SAMSUNG/upload/Product_Specifications/Samsung_2013_Full_LineUp_Catalog.pdf`, 2013. Accessed: 24-03-2015.

[28] Philip DeCamp, George Shaw, Rony Kubat, and Deb Roy. An immersive system for browsing and visualizing surveillance video. In *Proceedings of the international conference on Multimedia*, pages 371–380. ACM, 2010.

[29] Angela Merkel. Privacy in video surveillance. 2007.

[30] Andrea Cavallaro. Adding privacy constraints to video-based applications. In *EWIMT*, 2004.

[31] Andrea Cavallaro. Adding privacy constraints to video-based applications. In *EWIMT*, 2004.

[32] Isaac Cohen and Gerard Medioni. Detecting and tracking moving objects for video surveillance. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[33] Zdenek Kalal, Krystian Mikolajczyk, and Jiří Matas. Face-tld: Tracking-learning-detection applied to faces. 2010.

[34] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[35] Karan Gupta and Anjali V Kulkarni. Implementation of an automated single camera object tracking system using frame differencing and dynamic template matching. In *Advances in Computer and Information Sciences and Engineering*, pages 245–250. Springer, 2008.

[36] Hanzi Wang and David Suter. A novel robust statistical method for background initialization and visual surveillance. In P.J. Narayanan, ShreeK. Nayar, and Heung-Yeung Shum, editors, *Computer Vision â ACCV 2006*, volume 3851 of *Lecture Notes in Computer Science*, pages 328–337. Springer Berlin Heidelberg, 2006.

[37] Epicroads. Consumer Electronic Show CES 2015 : BMW Laser Headlights. `https://www.youtube.com/watch?v=-WvK5WC4ns0`, 2015. Accessed: 26-04-2015.

[38] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.

[39] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, Aug 2004.

[40] François Meyer and Patrick Bouthemy. Region-based tracking in an image sequence. In *Computer VisionâECCV'92*, pages 476–484. Springer, 1992.

[41] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system.

[42] Ping-guang Cheng and Zeng Zheng. Moving object tracking in intelligent video surveillance system. In Zhicai Zhong, editor, *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, volume 217 of *Lecture Notes in Electrical Engineering*, pages 195–202. Springer London, 2013.

[43] Thierry Bouwmans, Fatih Porikli, Benjamin Höferlin, and Antoine Vacavant. *Background Modeling and Foreground Detection for Video Surveillance.* CRC Press, 2015.

[44] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[45] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[46] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.

[47] G.J. McLachlan. Mahalanobis distance. volume 4, pages 20–26. Springer India, 1999.

[48] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection. net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.

[49] Kevin Hughes. Subspace bootstrapping and learning for background subtraction. 2013.

[50] Udacity. Software Development Process. `https://www.udacity.com/course/software-development-process--ud805`, 2015. Accessed: 06-06-2015.

[51] Oracle. Java SE Development Kit 8 Downloads. http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html, 2015. Accessed: 10-05-2015.

[52] JetBRAINS. IntelliJ IDEA. https://www.jetbrains.com/idea/, 2015. Accessed: 10-05-2015.

[53] Bytedeco.org. JavaCV Introduction. https://github.com/bytedeco/javacv, 2015. Accessed: 10-05-2015.

[54] Oracle. JavaFX: Getting Started with JavaFX. http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm, 2015. Accessed: 10-05-2015.

[55] Mikle Garin. Java Look and Feel for cross-platform Swing applications. http://weblookandfeel.com/, 2015. Accessed: 10-05-2015.

[56] Apache. Commons IO. https://commons.apache.org/proper/commons-io/, 2015. Accessed: 05-06-2015.

[57] Apache. Apache POI - the Java API for Microsoft Documents. https://poi.apache.org/, 2015. Accessed: 05-06-2015.

[58] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection. net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.

[59] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 393–400. IEEE, 2014.

[60] Andrews Sobral. BGSLibrary: An opencv c++ background subtraction library. In *IX Workshop dc VisÃ£o Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun 2013.

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

[62] Andrews Sobral and Antoine Vacavant. BGSLibrary (Version 1.9.2) : A Background Subtraction Library. https://github.com/andrewssobral/bgslibrary, 2015. Accessed: 13-05-2015.

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

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

[65] Dorra Riahi, P St-Onge, and G Bilodeau. Rectgauss-tex: Blockbased background subtraction. *Dept. génie informatique et génie logiciel, École Polytechn. de Montreal, Montreal, QC, Canada, Tech. Rep. EPM-RT-2012-03*, 2012.