REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DJILLALI LIABES DE SIDI BEL ABBES
Faculté de génie électrique
Département d'électronique
Laboratoire : Réseaux de communication, Architecture et Multimédia

N° d'ordre : /2017

THESE

Présentée par

M. Ilias DAHI

Pour l'obtention du Diplôme de Doctorat 3ème cycle
Spécialité : Electronique
Option : Réseaux, Architecture, et Multimédia (RAM)

Une technique robuste et intelligente de détection par vidéosurveillance d'objets abandonnés dans les zones de transit

(A robust and intelligent video surveillance method for abandoned object detection in transit zones)

Devant le jury composé de :

Président :	**BOUNOUA Abdennacer**	Professeur	UDL-Sidi Bel Abbès
Directeur de thèse :	**TALEB Nasreddine**	Professeur	UDL-Sidi Bel Abbès
Co-directeur de thèse :	**CHIKR EL MEZOUAR Miloud**	Maître de Conférences	UDL-Sidi Bel Abbès
Examinateurs :	**BERRACHED Nasr-Eddine**	Professeur	USTO–MB Oran
	BELLOULATA Kamel	Professeur	UDL-Sidi Bel Abbès

ملخص

ازداد نشاط الابحاث في مجال المراقبة بالفيديو، وهدا نظرا لتأثيره في عدة مجالات مثل الامن، المجال العسكري، وتطبيق القانون. تشهد شبكات الكاميرات تزايد كبير في الأماكن الحضرية مثل محطات القطار، المطارات، البنوك، واماكن عمومية لغرض أمني. التدفق الهائل المقدم من طرف هده الكاميرات لا يمكن مراقبته من طرف الانسان لعدم قدرته على التركيز لمدة طويلة من الوقت او عدم المراقبة، الاكتشاف، والابلاغ في الوقت المناسب، ولذلك، عدة طرق للكشف عن العديد من الأحداث عن طريق المراقبة البصرية تدرس لمساعدة الانسان في مهمته.

في هده الأطروحة، سنركز على تطوير طرق اكتشاف الاشياء المتخلي عنها عن طريق المراقبة البصرية. نعتبر الفعالية والتكلفة الحسابية اهتماماتنا الرئيسية نظرا لحساسية هدا النظام. النظام المقترح يكشف عن الاشياء المتخلي عنها مثل الأمتعة في مناطق العبور والأشياء الثابتة في أي من الأشكال في البيئات الداخلية والخارجية، تحت ظروف تغير في الاضاءة، وفي المناطق المزدحمة. الشيء الجديد في طريقتنا هو استخدام حواف بدلا من البيكسل في الكشف عن المناطق الثابتة. اقترحنا ايضا خوارزمية التجميع لتجميع الحواف المستقرة للأشياء المتخلي عنها المرشحة. درجتان احتماليتان مقترحة لتصنيف الاشياء المتخلي عنها.

Résumé

La recherche scientifique dans le domaine de la vidéosurveillance a été très active au cours de la dernière décennie, en raison de son impact énorme sur des domaines tels que la sécurité et les applications militaires. Les réseaux de caméras de vidéosurveillance sont entrain de s'agrandir de jour en jour dans les zones urbaines, comme les gares, les aéroports, les banques et tous les autres zones publiques où la sécurité s'impose. L'énorme quantité de flux vidéo fourni par ces caméras ne peut être contrôlée et vérifiée par un agent humain en raison de son incapacité à se concentrer pendant une longue période de temps ou à repérer un danger et réagir à temps. Par conséquent, plusieurs méthodes pour la détection d'événements suspects dans la surveillance visuelle sont étudiées pour aider l'agent humain dans sa tâche.

Dans cette thèse, nous nous concentrons sur le développement d'une méthode de détection d'objets abandonnés dans la surveillance visuelle. En raison de l'aspect critique d'un tel système, la robustesse et le coût de calcul sont nos principales préoccupations. Le système proposé détecte les objets abandonnés, comme les bagages dans les zones de transit et les objets immobiles de toutes formes dans les environnements intérieurs et extérieurs, avec des changements des lumières brusques et aléatoires, et dans les zones encombrées. La nouveauté de notre méthode est l'utilisation des contours au lieu des pixels dans la détection de régions statiques. Nous avons également proposé un algorithme de clustering pour regrouper les contours stables dans les objets abandonnés candidats. Deux scores robustes sont également proposés pour la classification des objets abandonnés.

Abstract

Research in visual surveillance has been very active in the last decade, because of its huge impact on fields like security, military applications and law enforcement. A big network of CCTV(closed-circuit television) cameras is growing in urban areas, like train stations, airports, banks and other public areas for security purposes. The huge amount of video flux provided by these cameras cannot be monitored by a human operator due to its incapacity to focus for long periods of time or to spot an event on time. Therefore, several methods for several event detections in visual surveillance are being investigated to help the human operator in his task. In this thesis, we focus on developing a method for abandoned object detection in visual surveillance. Robustness and computational cost are our main concerns, due to the critical aspect of such a system. The proposed system detects abandoned objects like luggage in transit zones and static objects of any forms in indoor and outdoor environments, under illumination changes conditions, and in cluttered and crowded areas. The novelty of our method is the use of edges instead of pixels in detecting static regions. We have also proposed a clustering algorithm to group the stable edges into candidate abandoned objects. Two robust scores are also proposed for abandoned object classification.

Acknowledgements

I would like to express my gratitude to all people, who helped me accomplish this work, for their thoughtful advice and encouragement.

In particular, I want to thank both my supervisor Pr. Taleb Nasreddine and my co-supervisor Dr. Chikr El Mezouar Miloud for their support, guidance, and numerous valuable discussions we had together;

All committee members, Pr. Abdennacer BOUNOUA, Prof. Nasr-Eddine BERRACHED and Prof. Kamel BELLOULA for having accepted to examine this thesis;

All the members of the RCAM laboratory, for creating a friendly working atmosphere;

And last but not least, to my parents, brothers and friends for their encouragement, understanding and patience.

Contents

List of Figures

List of Tables

List of Symbols

GMM	Gaussian Mixture Model
CCTV	Closed-Circuit Television
PAN	Pan-Tilt-Zoom
NMS	Non Maximum Suppression
BB	Bounding box
AO	Abandoned object

I dedicate this work to
my dear parents,
my brothers,
and all my friends…

Chapter 1

Introduction

1.1 Overview

In the last decades, the deployment of visual surveillance systems has grown exponentially. National parks, airports, metro stations, streets, campus and public places are all being watched. It was reported that in the united kingdom only, there are 2.5 million CCTV cameras with one camera for every 32 people, and a person is seen 300 times in one day on CCTV cameras. These systems are used for monitoring public places, private homes, industrial areas, etc..., see figure 1.1. Feed of the cameras is usually monitored by a human operator, which performs scene inspection and looks for abnormal events. The human operator may fail in his task, due to its limited capacities in maintaining a continuous watch. Moreover, The big networks of CCTV cameras have resulted in huge feeds and have increased the number of human operators to perform the watch. All these factors have driven toward the deployment of automated video surveillance systems, capable of processing multiple streams at one time and allowing event detection in real time.

FIGURE 1.1: Examples of CCTV cameras deployment in public places, airports, parks, streets

Since two decades, the scientific community in the computer vision field is striving to develop automated smart solutions for visual event detections. For this purpose, many techniques have emerged for the purpose of video analysis such as moving object detection, object tracking, crowd analysis and person identification. Besides that, intelligent visual solutions have been also proposed for other areas of applications such as health care, driving assistance, military purposes, industrial process inspection. The domain of visual events detection in public places is quit vast, ranging from persons behavior analysis, crowd analysis, traffic monitoring, and physical object inspection.

In this thesis, we focus on physical object inspection, more precisely; the aim of this work is the detection of abandoned luggages in public places. Terrorist

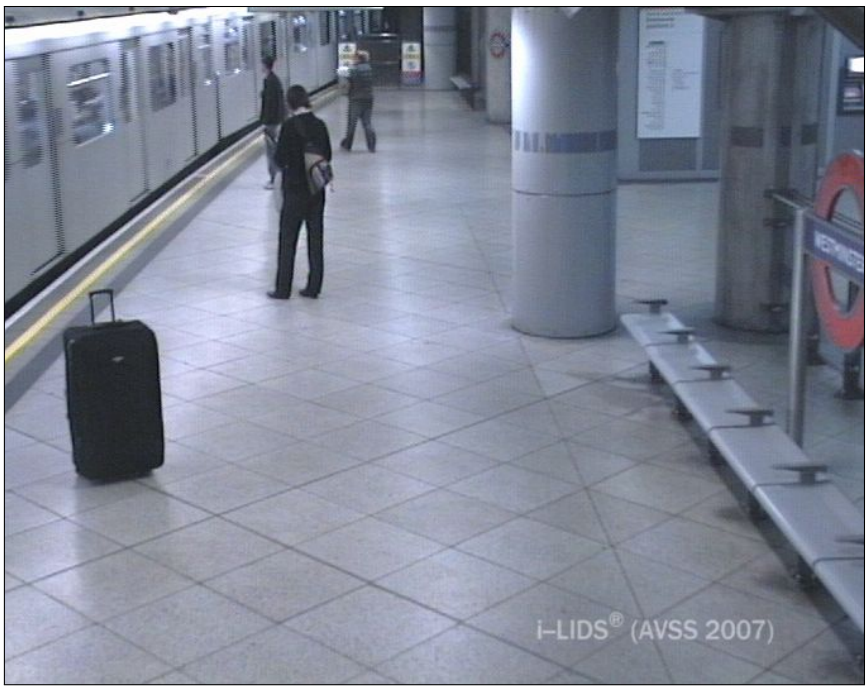


FIGURE 1.2: Example of an abandoned luggage situation

attacks were executed using explosive objects camouflaged in luggage most of times, and left out in public places(transit zones). Figure 1.2 show an example of an abandoned object case. The objective of this work is to detect left out objects or luggages in public places after a defined period of time from the drop event. The difficulty remains in the detection in a crowded environment, which is the case in transit zones. This makes it harder to detect the drop event or the luggage itself. Such algorithms must perform during various conditions, indoor, outdoor, and most of all they must be real time.

1.2 Contribution

This thesis describes methods for static object detection and object class classification for visual surveillance applications. Robustness is our major concern in designing the solution. Computational cost is also a priority because of the critical aspect of such a system. Our contribution is as follow :

Static region detection - In the scope of detecting static regions, a robust technique based totally on edges is described. A temporal accumulation scheme that is robust to spurious movements is described. An algorithm for clustering and static object bounding box forming is also presented.

Object classification - Once a static object candidate is detected, the resulting static region mask is fed to the classification stage, and two features scores are extracted to classify the object. The first score is used to describe the probability that a bounding box enclose an object. The second score is used to verify the staticness of the enclosed object.

Most of the state of the art techniques in the literature use pixels intensities to detect and classify abandoned objects[83, 64, 70]. Hence, many of the problems related to static object detection which is a subtask of an abandoned object system pipeline are caused by the fact that low level features pixels intensities are vulnerable to external conditions like illumination changes. In this work, a novel method is proposed that exploits scenes edges information for both static object detection and classification stages.

1.3 Thesis outline

- Chapter 2:

  Provides a detailed literature review of motion detection, object tracking, abandoned object detection and crowd analysis.

- Chapter 3:

  Provides a detailed description of the proposed method.

- Chapter 4:

  Presents and discusses the obtained results.

- Chapter 5:

  Presents a conclusion providing a summary of the proposed work and possible future works.

Chapter 2

Literature review

2.1 Introduction

Abandoned object detection is one of the most critical tasks in a smart surveillance system. It includes left luggage detection in transit zones, illegally parked vehicles, and left objects in highways. Many definitions have been given to the problem. One definition is that abandoned object detection is the process of identifying objects that have been left by an owner in the scene and have remained static for a period of time. Another definition is that an abandoned object also referred to as static of left object is defined as a foreground object, usually a luggage which has remained static for a defined period of time. The first definition considers that an abandoned object has been left by a person which is the case for luggage in general, while the second definition considers that the object has entered the scene without considering the owner or the person who abandoned the object, and has remained static for a period of time.

The overall scheme of a smart visual surveillance system is composed of preprocessing modules prior for the visual application execution, which is in

this case the abandoned object event detection. Moving object detection, also referred to as foreground object detection is the first stage of a smart visual surveillance system. It is the process of segmenting moving objects from the background scene. The resulting mask is a binary mask with white pixels representing pixels of moving objects while background scene pixels are black. The results are either fed to a high level application as low level features (pixels of moving objects), or in other cases it is preferable to delineate the segmented moving object by a bounding box using a component analysis operation, which is the process of grouping pixels into regions that represent objects. Most of the time, the background subtraction technique is the most used method in vision applications to segment moving objects.

The next stage is usually an object tracking step, it is the process of localization of moving objects through time. Generally objects are given labels as they first enter the scene and the purpose is to keep those labels within these objects till they get out of the scene. It may sound as an easy task, but tracking objects can be a highly complicated task, because of many sources of noise, like the object appearance changing through each frame, occlusions, illumination changes, and noise from the moving object detection step. Object tracking is used in many high level vision applications as a subtask, however it is not always the case for the abandoned object detection task, where it can be either an object tracking step or a static region detection step.

Abandoned object detection is an easy task when the scene is uncluttered and contains no sudden illumination changes, however when it comes to complex scenes with various sources of noise, the task becomes very difficult. Despite the fact that many works have been proposed in literature, the application is not mature enough to be applied in real world situations, and robustness needs to be further improved . Many forms may be confusing and

may be considered as abandoned objects, like static persons, ghosts resulting from a moved background object and illumination spots resulting from sudden illumination changes.

Abandoned object detection problems are related either to moving object detection step noises and imperfections, or to the static region detection step itself.

Background subtraction related problems are:

- Ghosts left by moved background objects.
- Sudden illumination changes.
- Background initialization problems.

Static region detection related problems are:

- Static persons.
- Slow moving crowds and occlusions.

Globally, the assumption is that a robust moving object detection method with noise-free moving object masks for the task of abandoned object detection would reduce the amount of false detections greatly. In other words, a moving object detection method with a minimum of noisy output can improve the overall system performance considerably. Moreover, false detections resulting from the static region detection can be reduced by using a robust classification step that exploits the abandoned object characteristics in a more efficient way.

In this chapter, the main areas of research that are related to abandoned object detection are discussed and will be presented as follow:

- Section 2.2:

  The architecture of video surveillance systems is presented.

- Section 2.3:

  The foreground object detection techniques are presented and discussed.

- Section 2.4:

  Object tracking is discussed.

- Section 2.5:

  Visual event detection representation and detection are presented in detail.

- Section 2.6:

  The stat of the art of abandoned object techniques are reviewed.

- Section 2.7:

  Crowd analysis methods are discussed briefly.

- Section 2.8:

  Recent trends in category independent object detection are discussed.

2.2 Architecture of Surveillance Systems

Surveillance systems play an important role in traffic incident detection, travel time measurement and traffic management. They offer a good solution for helping to solve the present-day security and safety problems in public transportation areas. All over the world, transportation operators, security staffs and the police are under high stress to solve these security and safety problems. Due to this, monitoring costs more than ever before. Furthermore, and as stated before, the huge amount of visual information gathered from public places can no longer be processed through human actors alone without a computer-based assistance. Because of the previously cited importance, such systems have essential requirements that researchers have to take into account to build the needed system and achieve their specified functions and performance. Although, there are many forms of observation and monitoring, e.g. directional microphones, communications interception, listening devices, Closed-Circuit Televisions or GPS tracking, video surveillance is the most popular form of surveillance. In this section, the overall architecture of video based surveillance systems and its applications will be discussed. Then, the major functional, design and performance requirements will be discussed which will help to build a video based surveillance system with a high performance.

2.2.1 Surveillance systems objectives and application forms

Intelligent video surveillance systems deal with the real-time monitoring of static and moving objects within a specific environment. The primary motivation of such systems is to understand, detect, recognize and predict the actions and the interactions of the observed objects autonomously based on

the information acquired by cameras. The main steps of processing in an intelligent video surveillance system are: moving object detection and recognition, tracking, behavioral analysis and retrieval. These steps include the topics of computer vision pattern recognition, artificial intelligence and data management.

There are three main technical evolutions of intelligent surveillance systems. The first generation started with analogue Closed-Circuit Television (CCTV) systems. They gave good performance in specific situations but they had the problem of using analogue supports for image distribution and storage.

The second generation techniques automated visual surveillance by combining computer vision technology with CCTV systems. This combination increased the surveillance efficiency of CCTV systems but they had the problem of robust detection and tracking algorithms required for behavior recognition.

The third generation presents the automated wide-area surveillance systems. They are more accurate than the previous generation due to the combination of different kinds of sensors. They have challenges in information distribution (integration and communication), design methodology, moving platforms, multi-sensor platforms. The typical flow of processing steps in video surveillance systems is illustrated in Figure 2.1. These steps constitute the low-level processing phase which is necessary for any video surveillance system.

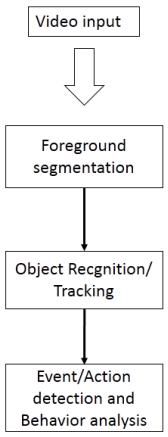


FIGURE 2.1: Processing flow in intelligent video surveillance systems

2.2.2 Surveillance system functions

A video surveillance system has to provide advanced features like remote access and configuration, visual event detection and alerts management. It must be easy to deploy and integrate within an existing surveillance environment, it must be also be flexible, scalable and cost effective.

Remote access

The system must be accessible from a remote location for live video screening and system configuration with only authorized different users simultaneously accessing in real time.

Visual event detection and alarms management

Since the main role of a visual surveillance system is to detect suspicious events, it must be able to detect and classify suspicious events automatically

and in real time. besides that, the system must have a policy for alarms management like events priorities and degree of danger.

Easy deployment and possibility for scalability

A video surveillance system must be easy to deploy, this include system modularity and technical abstraction for user ease of installation. Moreover, any existing system must be able to extend if needed. This involve cameras sensor number extension(growing network), storage extension, or software module adds (new event detection module for example).

Surveillance systems installation constraints

When designing a surveillance system for a given situation, many architectural questions arise:

- What types of cameras to deploy?
- What storage technologies to choose?
- What visual software module to include?

For camera sensors many attributes are used to choose the best camera for a situation:

- Camera position:

  A camera can be fixed, and angled toward a specific area of interest, or it can be a PTZ(Pan-Tilt-Zoom), a freely moving camera, with the use of panning, tilting, and zooming to change camera view. Fixed cameras are mostly used in visual surveillance.

- Chroma:

  Color based cameras are mostly used especially in situations with normal conditions. However, in situations with low lighting, infrared or thermal sensor based cameras technologies are used instead, producing a black and white video stream.

- video data transport:

  The video stream output from the camera is first digitized to be viewed on the screen. A user can choose between an analog based or an IP based camera, the main difference between them is that IP based cameras are less expensive and can support high volume mega-pixel video streams.

For the storage question, video surveillance systems have a policy of storing videos data usually for 30 days, it can be shorter(few days) or longer(years) for some specific cases and depending on the user or client requirement. Storage types are :

Internal storage

The Digital video recorder (DVR) or the Network video recorder (NVR) comes with an internal hard drive. It is the most common case in surveillance systems. The DVR is a device with an embedded software dedicated for receiving the video stream from the camera (usually analog) and is responsible for video encoding, storing, and viewing. The difference between a DVR and an NVR is that an NVR receives the video stream through IP based network (Ethernet).

External attached storage

In this case the storage support is an external device attached directly with the recording device.

Network based storage

A network server is dedicated only for data storage and file sharing. It allows system flexibility, scalability and security since the storage is an isolated module.

The last question, is the visual module software to include in the system,for eg. abandoned object detection module, crowd analysis module, behavior analysis module...etc .It depends on the user requirements and on the area being monitored.

Moving object detection

Usually, the main idea of object detection is the segmentation of images in foregrounds and backgrounds. The major two approaches are ”temporal difference” and ”background subtraction”. The first approach consists of the subtraction of two consecutive frames followed by thresholding. The second approach is based on the subtraction of a background followed by a labeling process. Generally, morphological operations are used to reduce the noise and to correct the segmented shapes. The segmentation of images separates the image in two parts, the foreground and the background. The foreground of the image represents the objects to be detected in the scene. After that, different processes can be chosen, starting with the representation and description of the regions shape and ending with processing and analyzing the regions of interest. The results of the previous processes can be used in the field of boundary matching or mathematical models training. The final step is commonly performed to extract the low level features for event detection

systems. Moving object detection will be discussed in details in a subsequent section.

Object recognition

The object recognition and tracking step is normally a model-based technique. Different approaches can be used to classify the new detected objects. For example, Gaussian distribution particle filters, hidden Markov models and Support Vector Machine .

Behavior analysis

The previous steps are important to extract features for event detection where the behavior of the observed object should be analyzed and understood. Furthermore, the analysis of the image and the understanding of the spatial/temporal content is also required to understand the behavior of the object. Suppose the system is detecting a vandalism in a bank, it is not possible to detect the event of vandalism against the Automatic Teller Machine (ATM) without knowing if the object is near by the (ATM) machine or not. Additionally, video streams consist of a sequence of frames (images). Thus the temporal issue should be considered and analyzed to understand which event occurs before another event.
The overall architecture of event detection in surveillance systems has the following three layers:

1. Object detection and tracking: By extracting features using object recognition and object tracking algorithms; this involves image processing and pattern recognition.

2. Primitive events detection: By defining both behavior and rules that are related to objects, simple events can be detected like walking, running, shouting, etc.

3. Complex event detection: By building rules using rule engines acting on simple events, a series of detected simple events can be joined together to form complex events.
   Visual events detection will be discussed in detail in section 2.5.

Database

The final stages in a surveillance system are storage and retrieval. The most used databases are data warehouses which is a database used for reporting and data analysis. It is a central repository which is created by integrating data from and multiple disparate sources (audio or video). The major disadvantage of a data warehouse is its expensive maintenance if it is underutilized.

2.3 Foreground object detection

Foreground object detection is the process of separating moving objects that are of interest for the video surveillance applications like moving persons and vehicles, while ignoring moving objects that are not useful for surveillance applications like waving trees. Foreground object detection is the first step in video surveillance applications in general. Many methods for foreground object detection were proposed in literature such as background subtraction based techniques [48, 95, 92], optical flow and frame differencing. However, background subtraction is the used technique when it comes to

moving object detection compared to other works proposed in literature, due to its robustness and capability of adaption with dynamic scenes.

Background subtraction techniques consist of constructing a background model by learning the scene. Once it is built, the background model is compared with each input frame, to separate moving objects from the background using some defined and tuned parameters. Many problems can affect the robustness of the method, like noise included at initialization step, sudden illumination changes, ghosts occurring from regions that were in the background and then have been moved resulting in false moving object region. Most of the methods are pixel-wise i.e. the intensity of each pixel is learned independently in time. However, since it is a pixel-wise operation, similarities in pixels intensities of a moving object with the background cause the problem of regions fragmentation, which results in fragmented regions of the same object. Another problem that a background subtraction method must handle is the problem of shadow removal. Objects shadows can be missdetected as foreground objects, or can overlap on other foreground objects.

The problem of gradual illumination changes resulting from natural light has been resolved by many methods, however sudden illumination changes caused by action like a light turn on/off and window opening remain a major source of false alarms for pixel intensity based background subtraction methods. This has been partially solved by methods[45, 35, 27] using edge pixels magnitude as feature in constructing the background model.

One of the most popular methods of background subtraction techniques is the Gaussian mixture model (GMM) [95], first proposed by [80]. The method

is robust to periodic motions, scene dynamics such as clutters and waving trees thanks to its multi-modal motions adaption design. However, it has poor performance in situations with sudden illumination changes which results in false detections. GMM has also a high computational cost compared with other techniques in the literature. First we will introduce the basic form of a statistical distribution which is the single Gaussian model.

2.3.1 Single Gaussian model

In [53], the authors proposed to represent each pixel of a background model using arithmetic mean between successive input frames. The model use n pixels of n frames in time to construct the Gaussian distribution representing that pixel. The Gaussian distribution is parameterized by its mean and variance. The background model β_t is updated using running average operator as follow:

$$\beta_t = (1 - \alpha)\beta_{t-1} + \alpha I_t \tag{2.1}$$

Where α is a parameterized learning rate, I_t is the input frame pixel intensity, and β_{t-1} is the previous background model output by the running average previously. Noteworthy that the background model β_t here represent the temporal mean average.

Similarly the variance is updated using the running average operator at the same way using the same parameters :

$$\sigma_t = (1 - \alpha)\sigma_{t-1} + \alpha(I_t - \beta_t)^2 \tag{2.2}$$

The running average allows to store the history of pixel intensity changes without requiring a large amount of memory, so instead of storing every pixel for multiple frames, each pixel has a mean and a variance stored that represent its changing in time.

The foreground mask Fg_t of the moving object is simply obtained by comparing the input frame I_t with the the background model β_t:

$$Fg_t(x,y) = \begin{cases} 1 & \text{if } |I_t - \beta_t| > k\sigma_t \\ 0, & otherwise \end{cases} \tag{2.3}$$

Where k is a constant set to 2.5.

The single Gaussian model allows to cope with scene dynamics, with absorbing temporary consistent pixel intensities. However, when it comes with dealing with short or long term periodic background motions, the single Gaussian model fails to absorb these motions.

2.3.2 Gaussian mixture model

When the scene includes big dynamics or sudden changes, the single Gaussian model fails as stated in the previous section, and a multi-model distribution is required to adapt to the scene changes and describes the scene in a better way. In the GMM method [95] , the background model is represented by a set of Gaussians for each pixel, each one representing a single background dominant motion at that pixel, absorbing dynamic motions into the background model. In other words, the GMM track pixels intensity changes in time by modeling each pixel using a set of Gaussian distributions each having its mean and variance. The input frame pixels are compared with

the Gaussians of the background model corresponding pixels. If an input pixel $p(x, y)$ intensity fits to one of the Gaussians, then it is considered as a background pixel and the corresponding distribution mean and variance are updated and making its weight increased. Otherwise, it is classified as foreground pixel, and a new Gaussian is created for that pixel, with its intensity value as the mean of the Gaussian. Formally, for a pixel X at time T, the probability that this pixel is a background pixel is written as follow:

$$P(I_t) = \sum_{i=1}^{K} w_{i,t}\eta(I_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.4}$$

Where k is the maximum number of Gaussians a pixel can have depending on the quantity of periodic motion present at that pixel. μ_i is the mean of the i_{th} Gaussian, and Σ_i is the covariance matrix of the pixels intensities.

The background model is updated by checking each pixel with the available Gaussians distributions until it fits to one of them. A pixel is considered as belonging to a distribution if it fits to 2.5 of the standard deviation. If the pixel does not match any of the available distributions, then it is considered as a foreground pixel and a new Gaussian distribution with a mean with the pixel value, a weight with minimal value, and a variance is created. Otherwise, if the pixel fits to one of the Gaussians, then this Gaussian component is updated as follows:

$$w_{i,t} = (1 - \alpha)w_{i,t-1} + \alpha \tag{2.5}$$

$$\mu_{i,t} = (1 - \alpha)\mu_{i,t-1} + rhoI_t \tag{2.6}$$

$$\sigma_{i,t}^2 = (1-\alpha)\sigma_{i,t-1}^2 + \rho(I_t - \mu_{i,t})^T(I_t - \mu_{i,t}) \tag{2.7}$$

Where α is the learning rate, which defines how fast a Gaussian distribution must be updated. ρ depends on the learning rate α, the weight of the Gaussian component, and on how likely the input pixel has fitted to the corresponding Gaussian component,and is defined as follows:

$$\rho = \alpha \frac{P(i|I_t, \mu_{i,t}, \Sigma_{i,t})}{w_{i,t}} \tag{2.8}$$

For the other Gaussian distributions that do not have a match with the input pixel, their mean and variance remain unchanged, only their 'a priori' probabilities i.e. their weights are reduced.

$$w_{i,t} = (1-\alpha)w_{i,t-1} \tag{2.9}$$

When a pixel does not match any of the Gaussian components, the component with the lower weight is then replaced with with a new Gaussian with $\mu = p(x, y)$, a large variance Σ_i and a low weight w_i.

2.4 Object tracking

Many methods for object tracking were proposed in literature. Visual object tracking is a crucial step for many vision applications, like video analytics, gesture recognition and augmented reality. Visual tracking in real world scenario is a hard problem, therefore it is still a wide open problem. Targets can

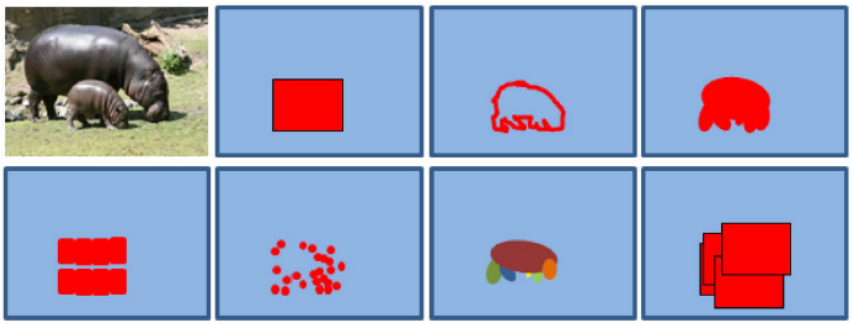


FIGURE 2.2: Target region representations from top left to bottom right: bounding box, contour representation, target blob, patch based, sparse set of salient features, parts, and multiple bounding boxes

be anything of interest like persons or vehicles, it depends on the application. The goal of visual tracking is to estimate the motion state of a target object at each input frame. A typical visual tracking framework is composed of five modules, target region definition, appearance representation, motion representation, object location module, and model updating. For an object to be tracked, it has first to be represented by some visual cues. But first, the region where these visual cues will be extracted has to be defined. The most used region representation is the bounding box [11], its only drawback is that it can include regions for the background, most of the time it is used for general purpose tracking. Some methods use the object contour [82, 93] allowing a high tolerability to changes in the object shape appearance. other works use object blobs[34], patched based[3], set of salient features[52], articulated parts[77], and multiple bounding boxes[61, 42], see figure 2.2.

Visual cues used for tracking in literature are 2D-arrays like image data, 1D histograms and features vectors, and are extracted from the region defined by the target region module. figure 2.3 show an example of cues used in target appearance representation.

Motion representation is the search model for the target in the next frame.

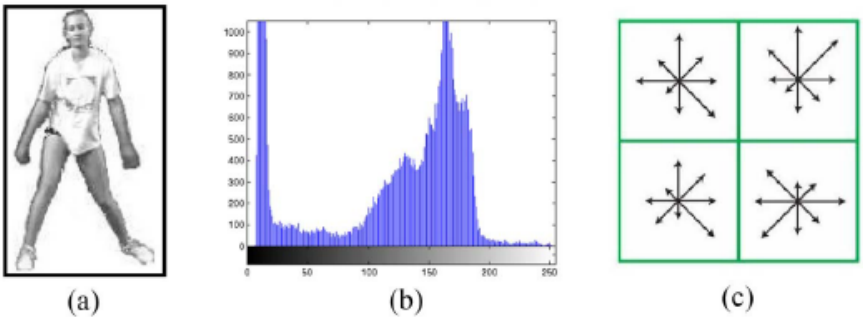


FIGURE 2.3: Target appearance representation: a) 2D pixel array b) Histogram c) Feature vector

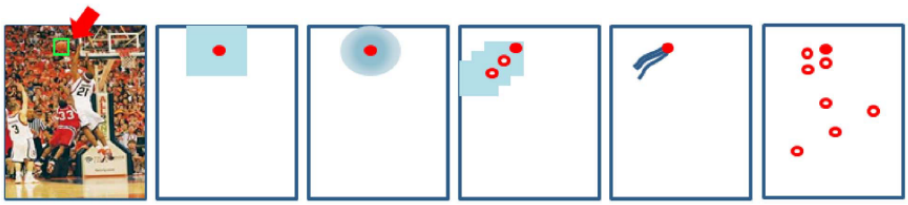


FIGURE 2.4: Motion representation model used by tracking techniques: uniform search, Gaussian motion model, motion prediction, implicit motion model, and tracking and detection model

The most used and simple assumption is that the target is close to its previous position as used in[62, 37]. This assumption is weak when the object moves fast. Another search paradigm is the use of the probabilistic Gaussian motion model used in [74, 61] where locations near the previous object position are assigned more weights, and the weights decrease while getting further from the object previous position. Motion can also be represented by a linear model like Kalman filter [43], and is used in [15]. Another concept is tracking and detection, where many candidates from an object detector are matched with an optical flow tracker object proposal as used in [79]. Figure 2.4 shows the motion representation used in literature.

The matching process, which is the object localization, is the core of the tracking algorithm. It is the process of how to find the target location in the next frame given a motion estimates and target candidates in the neighboring.

The search paradigms can be divided into two categories, direct search and probabilistic search. Direct search is an optimization problem and direct gradient ascent is used to find the best match for the target by locating the maxima as used by KLT[7] and the mean shift[23]. One gradient ascent drawback is that it can get stuck in a local maximum when the tracked object moves too fast. The authors in [51, 74, 52] use a particle filter as a solution to avoid local maxima and at the same time optimize the search space by sampling candidate windows around the target previous position. A distance is then computed between the target visual representation and the visual representation of the candidate window. Tracking by detection is different from the previously mentioned paradigms. Tracking is done by combining a learned detector and an optical flow tracker. The detector learns an appearance model on the bounding box of the target at the first frame using features differentiating the object from a distant background. At each new frame, a set of the best possible target locations are selected using the detector. At the same time, the tracker estimates the object new position and state using optical flow tracker like KLT[7]. Finally, both the tracker and the detector correct each other at each iteration.

2.5 Event detection

2.5.1 Event representation

Event detection is composed of two elements: abstraction, which is the feature extraction operation, and where efficient video event descriptors that characterize uniqueness to events of interest are extracted. The second element is modeling, where these descriptors, or features are used to train a

classifiers to describe those events in order to separate them from spurious or meaningless other events. In the surveillance application context, events represent one of the following cases:

- Human activity (a single person or many people actions);
- Crowd activity (group of persons as one entity behavior);
- Interactions between humans, objects, and their environment;
- Other, such as facial expressions, gesture.

Event representation is a feature extraction task consisting of extracting spatial and motion cues from the video that are discriminative with respect to particular activities within a scene. Description of an event or activity always starts from extracting low-level features. Low level information in a two-dimensional video frame consists of shape, color or texture depicted in that frame. If a sequence of video frames is available, differences between the consecutive frames provide motion information about the objects present in the captured scene. Using a combination of the static information in each frame and the differences between the frames that capture dynamic information, spatio-temporal descriptors are formed. Due to the temporal nature of video events, descriptors from consecutive frames have to be grouped into the meaningful event representations. For this task, temporal segmentation techniques are employed to identify boundaries of events in video data. The final video event representation is then used in the higher level event modeling and classification steps.

Four types of low-level feature extraction methods exist in the literature: background subtraction, optical flow, point trajectory, and filter responses. Background subtraction is a popular method for identifying the moving parts of the scene. The shape of the resulting object silhouette is often used to

describe objects and their activities using global methods such as moments [10]. Although silhouettes provide strong cues for action recognition and are insensitive to color, texture and contrast change, they fail in detecting self occlusions and depend on robust background segmentation. Optical flow provides a concise description of both the regions of the image undergoing motion and the velocity of that motion. Optical flow often serves as a good approximation of the motion projected onto the image plane. Optical flow based representations do not depend on background subtraction, but they are sensitive to changes in color intensities of the pixels due to variation of light, camera jittering, or camera motion.

Trajectories of moving objects have been used as features in many applications to infer the activity of the object. The trajectory itself is not very useful as it is sensitive to translations, rotations and scale changes. Alternative representations such as trajectory velocities, trajectory speed, spatio-temporal curvature or relative motion have been proposed to acquire invariance to some of these variabilities. Extracting unambiguous point trajectories from video is complicated because of several factors such as occlusions, noise and background clutter. Temporal filtering is an alternative approach to the region-of-interest detection in image sequences. These approaches usually represent actions using bag-of-features (BOF) which are histograms that count the occurrences of the vocabulary-features within a video segment. The practical advantage of this approach is that filter responses show consistency for similar observations, but can account for outliers. Filtering is useful in scenarios with low-resolution or poor quality videos where it is difficult to extract other features such as optical flow or silhouettes.

There are three types of methods for visual action detection: nonparametric, volumetric and parametric. The non-parametric approach extracts a set of

features from each video frame and compares them to a predefined template. Examples of non-parametric methods are dimensionality reduction(PCA), template matching, 3D object matching, and manifold learning. This approach needs a background subtraction step to extract the shape of the moving object with accuracy. The volumetric approach does not extract features on the frame by frame way. Instead, it considers a video stream as a 3-D volume of pixel intensities and extends the standard image features to the 3-D space. Examples of volumetric methods are parts constellation, space-time filtering and sub-volume matching. This approach is robust for capturing the motion of the events that are difficult to define. The parametric approach imposes a model on the temporal dynamics of motion, from which the parameters for a class of actions are estimated. Parametric methods include hidden Markov models (HMM) and linear dynamic systems (LDS). This approach is often used for complex actions such as dancing, juggling, and other actions with complex dynamics.

The action segmentation is the task of separating single visual actions from streams of video data. In the stats of the art, action recognition results are often demonstrated on segmented video clips and each video clip represents a single action from start to finish. In real-world surveillance case, videos cannot be segmented by hand. In [91], temporal action segmentation can be classified into three categories : boundary detection, sliding windows and grammar concatenation. Motion boundaries are usually detected as a preprocessing step before event classification. Boundary detection methods delineate the temporal limits of an action without dependence on the action classes, but can present errors in the recovery of motion fields and are affected by the presence of multiple, overlapping, and simultaneous movements. Video sequence can be divided into multiple, overlapping segments using a sliding

window. Classification is performed on all the segments, and peaks in the resulting classification scores are interpreted as action locations.

The sliding window approach, when compared to motion boundaries, produces much more segments that need to be evaluated by the classifier, thus are usually more computationally intensive. However, sliding window methods based on fewer assumptions can be integrated with any action classifier. Grammar concatenation techniques require action representation that involves grammars, which give a model of transitions between states and actions. Concatenative grammars can be build by joining all models in common start and end node and by adding a loop-back transition between these two nodes. Typically these approaches are hand crafted to specific scenarios and do not generalize well to other scenes.

2.5.2 Event classification

Event classification consists of learning statistical models from the action representations and using those models to classify new observations. A major challenge for the algorithms is dealing with the large variability of events that belong to the same class. Objects participating in the same class events can exhibit different sizes, speeds, and styles. Event classification approaches can be broadly grouped into four groups:

- Logic based methods.
- Graphical models.
- Support vector machines.
- Clustering approaches.

Logic based methods rely on formal logic rules to describe activities. Several works have proposed ontologies for specific domains of visual surveillance. For example, the authors in [21] proposed an ontology for analyzing social interaction in nursing homes. In [36] it is proposed an ontology for videos of meetings. [33] proposed ontologies for activities in bank monitoring settings. Though empirical constructs are fast to design and work well, they are limited in their utility to specific deployments for which they have been designed.

A graphical model is a probabilistic model for which a graph represents a conditional dependence form between many random variables. Graphical models can be divided into two categories: Bayesian networks and Markovian networks. A Bayesian network (BN) is a graphical model that represents complex conditional dependencies between a set of random variables.

Dynamic belief network (DBN) is a generalization of BN where temporal dependencies are incorporated between random variables. Usually the structure of DBN is provided by the domain expert, and to learn local conditional dependence relations, it requires very large amounts of training data or extensive hand-tuning by experts both of which limit the applicability of DBNs in large scale settings. A Markov network is represented by an undirected graph and is based on a set of random variables having Markov properties. Hidden Markov model (HMM) is a widely used method in speech recognition and is increasingly used for visual event recognition. The authors in [85] modeled visual activities by representing each activity by a distinct HMM and achieved 90% to 95% recognition rate for waking, running, skipping, sitting down and standing up activities. In comparison to DBNs, HMM encodes less complex conditional dependence relations.

A Support Vector Machine (SVM) is a machine learning technique that is

used for solving problems in classification, regression and novelty detection. An important characteristic of support vector machines is that the model parameters determination is a convex optimization problem so any local solution is also a global optimum. The basic idea of a linear SVM is to find a suitable hyperplane that divides a given dataset into two parts with maximum margin, this step is called the learning phase. The obtained SVM model is utilized to classify unlabeled datasets. However, in practice, many data are not linearly separable, and no hyperplane may exist that can split the data into two parts. A non-linear SVM can be achieved by using Kernels. An SVM is not only capable of learning in high dimensional spaces but can also provide high performance with limited training data.

Clustering analysis is the operation of grouping of data instances in order to figure out the structure in the data. The results of a cluster analysis may produce an identifiable structure that can be used to generate a hypothesis [90, 9] applied spectral clustering algorithm with a Median Hausdorff distance to get a grouping of the dataset. The authors in [89] used a spectral clustering algorithm to find the classes of actions.

2.5.3 Unusual event detection

Detecting unusual activities in visual surveillance applications is of major interest in practice. Algorithms able to single out abnormal events within streaming or archival videos can serve a range of applications - from monitoring surveillance feeds, or frames of interest proposals in scientific visual data that an expert has to analyze, to summarizing interesting visual content on a day's worth of web-cam feeds. In general, automatic detection of

anomalies should significantly improve the efficiency of video analysis, saving valuable human attentiveness for only the most relevant content [44]. Despite the problem's practical appeal, abnormal event detection remains technically difficult. The big challenge is that in real world conditions, unusual events occur with unpredictable and different variations, making it difficult to discriminate a true abnormal event from noisy observations of normal observation. Furthermore, the visual context in a scene tends to change over time. This implies that a model that represents what is normal has to be updated as soon as new observations are available.

The goal of abnormal event detection is to detect, recognize and learn relevant events. Many reviews are conducted for unusual event detection in different domains. In signal processing, outliers detection is a useful task for fault detection, radar target detection, masses detection in mammograms, control of stochastic process and many other tasks. In the literature, this task has been identified using terms such as suspicious, irregular, uncommon, unusual, abnormal, novelty, anomaly activity/event/behavior.

[59, 60] reviewed important aspects related to novelty detection, like robustness and trade-offs, parameter optimization, generalization and computational complexity. It was found that assumptions on the nature of the data have to be made before proceeding to modeling with statistical approaches. Moreover, the quantity and quality of the training data is very important for a precise determination of the model parameters. [39] conducted a comparative study of techniques for outliers detection. The authors have divided the outliers detection techniques set into three groups: clustering, classification, and a novelty approach. The conclusion was that algorithm designers should choose a modeling technique depending on the data type, the available ground-truth labeling, and how they want to detect and process

the outliers. Most of the earlier work in abnormal event detection has been conducted for the control systems domain.

Network anomaly detection for cyber attacks prevention has received a lot attention in the last decade due to the advances in networking technologies that allowed Internet to expand as a global support in communications and transactions. The studies in this area show that most of the remedies for computer intrusions are still based on the intrusion signatures [76, 66, 54], but there is an increasing trend to employ techniques that create models of usual acceptable behaviors and identify unknown and abnormal threats by detecting the deviations from these models [49, 41, 8].

Many surveys for anomaly detection in a variety of areas have been conducted by Chandola et al.[19, 17, 18]. In [19], Chandola et al. classified outliers detection techniques based on input data, type of supervision and type of outliers. In a subsequent survey by Chandola et al.[20], a comparative evaluation of a large number of anomaly detection techniques was discussed. The conclusions from the conducted experiments were that, for anomaly detection, the nearest neighbor (KNN)based techniques perform slightly better than clustering techniques based methods. Finite state based methods are the most consistent methods, while probabilistic suffix trees and the sparse Markovian techniques performance is not satisfying. It is also noted that the performance of a technique depends tightly on the nature of the data. In their latest survey [18], the anomaly detection methods were categorized based on the problem formulation that they are trying to tackle: sequence based, contiguous subsequence based, and pattern based anomaly detection techniques.

Surveys of anomaly detection in automated surveillance applications were conducted by authors in [78] and [69]. In [69], an abnormal behavior task was

presented as a general task in visual surveillance applications and conducted a broad overview of the visual event detection area. The anomaly detection task is regarded as a pattern learning problem that deals with object behavior classification by finding matches either with an already known template of objects behavior or learning and forming statistical models of object behavior types from the spatio-temporal feature data.

2.6 Abandoned object detection

Automatic abandoned object detection is an important problem in smart video surveillance, considering its huge impact on security in public places. Such system must detect objects left by persons in the monitored area. Many works were dedicated to abandoned object detection in last decades. We can distinguish methods that focus on the detection of the drop-off event [5, 73]. Detecting the drop-off event can be feasible in simple scenarios with no clutters. However, in a crowded and cluttered scenarios like an airport or a train station, it is not possible to detect such event, since it will be occluded in most of the time, especially in rush hours. On the other hand, there are works that focus on analyzing the scene to detect new objects that had not been present before and that are static for a defined period.

Almost all methods proposed in the literature are based on the background subtraction model to detect abandoned object. We can divide them into three categories; dual background based [70, 87, 30, 88], foreground sampling [56] based, and the last category uses the GMM [95] properties to detect static pixels[83]. Noteworthy that most the background subtraction based techniques use the GMM as a background subtraction module.

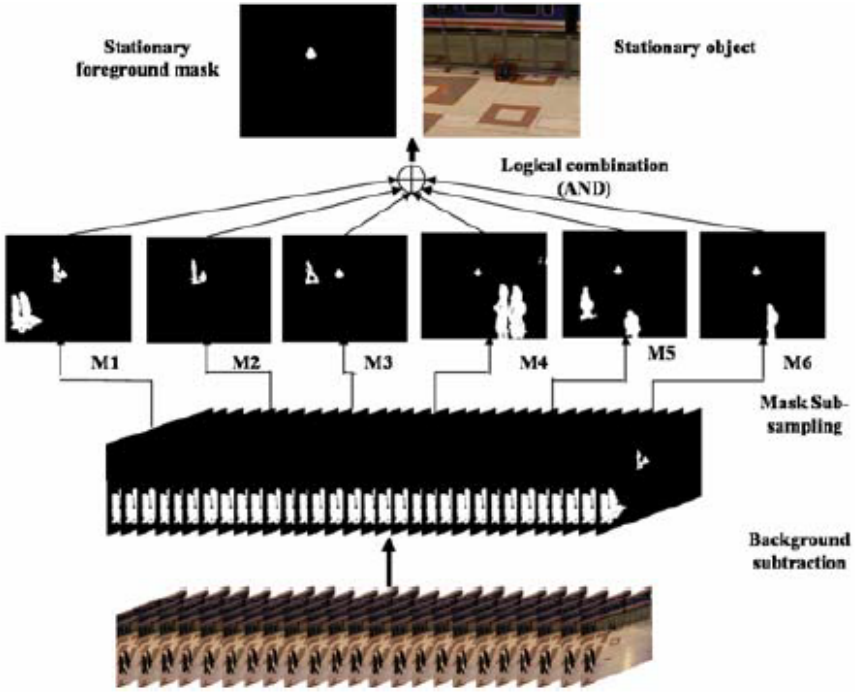


FIGURE 2.5: Foreground sampling method

Dual background based methods use a long-term background model with a slow learning rate, combined with a short-term background model with a high learning rate. This way, pixels with value 1 in the long-term model foreground mask and value 0 in the short-term model foreground mask are classified as static pixels. The resulting mask is the mask of static objects. In [30], the authors have added a finite state machine to track the pixel class in time. Foreground sampling methods[56, 25] exploit the temporal consistency of static object pixels in the foreground mask, by accumulating the foreground masks every T frames for N seconds to identify static regions, see figure 2.5, however the authors provide no post processing step to verify the Abandoned object candidate. The GMM state transition from one Gaussian component to another can be very informative about the pixel state (moving, static, background), see figure 2.6. In [83], the authors use the GMM property

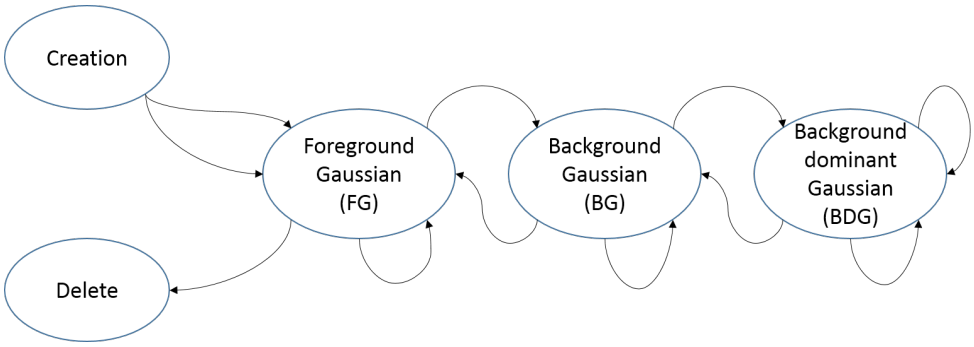


FIGURE 2.6: Pixel state transition in GMM

to identify static pixels. Their assumption is that in a GMM, a pixel migrating from a low weighted Gaussian to a high weighted Gaussian is a pixel that is part of a region becoming static. Moreover, identifying static regions at pixel levels generate noisy static object masks, and results in false detections. Without a high-level visual interpretation of the candidate static region, illumination spots, immobile people, and ghosts can be miss-detected as true abandoned objects. Therefore, several works focused recently on the abandoned candidate validation[64, 81, 31, 47, 63]. A region analysis based on multiple scores is proposed in[64], each score is intended for a type of noise. the authors defined a Foregroundness score intended for background model maintenance by checking whether a region is caused by a ghost or an illumination spot. In addition, they used an Abandonness score to check whether the candidate is a true abandoned object or it is caused by static persons or other spurious detection. In [81], a technique is proposed that checks foreground pixels stability in time. Then, a clustered region of these stable pixels is compared with contours of moving objects resulting from the background subtraction step to reduce false alarms resulting from spurious detections.

In [31] three attributes where used: The Foregroundness, the Staticness, and the Abandonment which define respectively the likelihood that an object be a foreground object, the likelihood that an object be static for a defined period

of time, and the likelihood that an object be left by its owner. These attributes were used as high level features to detect true abandoned objects. Each one of these attributes was trained using low-level features. Using these quantified attributes, noise was reduced and adaptability was increased. Similarly, the authors in [47], used three features (intensity, motion and shape) to identify abandoned objects, achieving a good detection rate. They also proposed in [63] a spatio-temporal set of features to check the stationarity of a region.

Other works focused on dealing with illumination changes related problems, particularly in outdoor scenarios. the authors in [38] proposed a method for that deal with illumination changes , he defined a contour score that tracks the AO candidate contour stability in time to decide if it is a true abandoned object. In [46], the authors proposed to intervene at the background subtraction stage by using edge pixels magnitude instead of pixels intensities in separating foreground from the background, the resulting moving edges are accumulated in time to extract stable edges. The method was tested on the publicly available dataset PETS2006 [67] only and presented some false alarms in the second scenario.

2.7 Crowd analysis

Among computer vision applications used in smart video surveillance systems, crowd analysis is an important branch. It involves the tasks of crowd counting, crowd density estimation and crowd behavior analysis, for the purpose of crowd density and flow control to prevent crowd disasters as that of the Muslims Pilgrimage at Mena in 2006.

Methods for crowd density estimation can be divided into two categories [1], direct based approaches based on person detection and indirect approaches based on features. Direct based approaches try to simply estimate a crowd density by detecting every single person in the scene, and calculate their position on the real plane. However, due to occlusion and high crowded environment problems, detecting people in complex scenes can be a real challenge, and thus affecting the density estimation. Some works tried to overcome such obstacles by detecting only the exposed part of a person which is the superior part, as head based detection [58] and head and shoulders shape detection [55]. However, these attempts are only applicable for low and medium density crowds but not for high density crowds.

Direct based approaches can further be divided into two categories, model-based and trajectories based methods. In the first category, persons are detected and segmented individually, and then counting is performed using a model of human shapes [55] by applying some pedestrian classifier, with every method proposing its own detector. In the second category, people are detected, by clustering every single motion. The assumption is that two points moving in the same direction are more likely to be of the same entity.Figure 2.7 shows the results obtained by both methods described in [12] and [71]. In [12], the authors tried to cluster points with the same movement direction into one entity using a Bayesian clustering method. Points are described using low level features, and tracked then grouped into an independent moving entity. The space-time proximity and the trajectory were used as constraints in the clustering. As in [71], Kanade Lucas Tomasi [84] tracked low level feature points and combined then with a clustering method for the motion segmentation purpose and to count the number of people.

FIGURE 2.7: Trajectory clustering for people counting with results output from [12] on the left, and results output from [71] on the right.

Indirect based approaches describe the relation between a set of features from foreground pixels such as foreground areas [40], histograms of edge orientations [75], edge counts [16] and the number of people using a regression function[65] or a learning based model like neural network as proposed in [50]. Crowd density related problems are occlusions and geometrical distortions from the ground plane to the image. Direct based approaches are better at dealing with occlusions than indirect approaches. On the other hand, the indirect based approaches are more attractive from the direct based approach in term of efficiency. Geometrical distortions are corrected by camera calibration. Occlusions caused by people self-occlusion can cause underestimation of crowd density.

2.8 Category independent object detection

Object proposals methods work with the assumption that all objects share visual properties. The main motivation behind this research is to improve the traditional classifier based object detection, where the classifier proceeds all over the image with a sliding window traversing the image. The classifiers have increased greatly in detection accuracy, at cost of a high computational

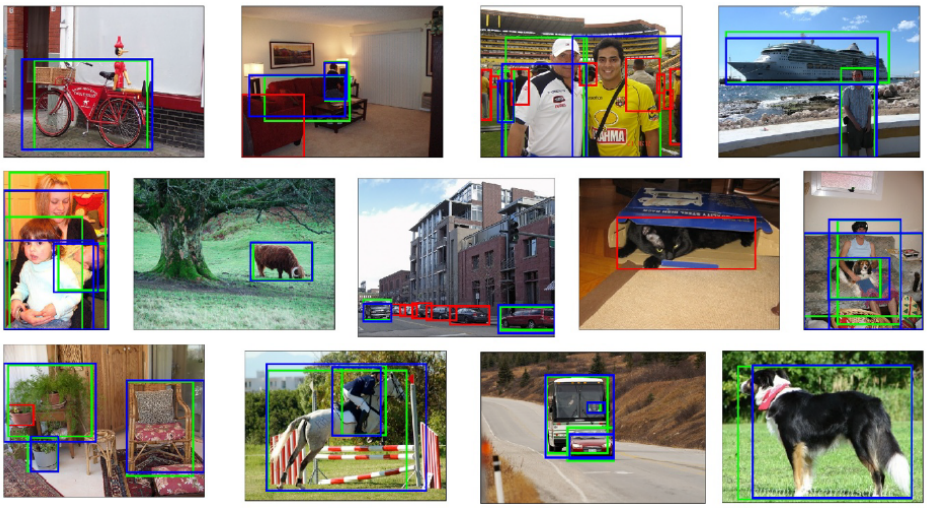


FIGURE 2.8: Object proposals generation examples

time per window. Object proposals can be used to improve the computational speed while keeping the detection accuracy of the classifier. The idea is to use these common visual properties (shape, textures...) to design a learning based or a score based method that given an input image, it outputs a set of windows that are likely to contain objects. This way, the search process is optimized and the classifier has no need to traverse all the image at different window scales, but it will apply only on the proposals delivered by the object proposals module. This can allow more sophisticated classifiers to be used without caring about computational time. We distinguish three general paradigms for object proposals: seed segmentation, grouping method(super pixel merging) and window scoring (Objectness).

2.8.1 Window scoring

In [4], the authors used a combination of a number of cues in a classification framework to define an Objectness score to each window candidate. Among those cues, [72] used the classification framework idea with edges distribution near window boundary as a cue, and learn an efficient cascade quick

and accurate window candidate ranking. In [4], in addition to using edges distribution near window boundary as a cue, the authors use also a super-pixel straddling measure, which penalizes candidates that contain pixel segments that overlap the window boundaries. The authors in [94] proposed a method for object proposals that score a bounding box by the quantity of edges enclosed within. They developed a technique that weights edges according to their af g edges by grouping edge pixels of the same orientation into groups, then affinities are computed between these edge groups. Finally the edge groups are weighted according to their affinities to edges groups which directly overlap the bounding box boundaries. Figure 2.8 shows some detection examples obtained in [94].

2.8.2 Seed segmentation

[13, 29] proposed to generate a random seed regions and generate a foreground-background segmentation for each seed. The resulting segmentation masks are very accurate but the computational time is very high.

2.8.3 Super pixel merging

One popular method is the selective search[86], super-pixels are merged to generate segments that are more likely to represent objects. No learning is required, instead super-pixels are merged using features and similarity functions that are previously designed. Selective search owns its success to its high recall and fast speed object proposal generation.

Chapter 3

Proposed method

3.1 The abandoned object event representation

In defining the abandoned object event, we have taken into consideration the scene complexity i.e the event representation depends on the capability of the sub-steps like moving object detection and tracking to perform in crowded and cluttered situations. Areas like airports and train stations present high amount of crowds and clutter especially in rush hours, thus, moving object detection and tracking steps output noisy information about the scene. Starting from this, if the event rules are complex and the low level information from the preprocessing steps(background subtraction, tracking) are noisy, then the output of logical model built using the logical rules will be erroneous. We use a simple but effective model to describe the event of abandoned object detection, capturing the motion stability at low level i.e. edges, without incorporating the object level.

The proposed method uses edges to detect and classify AO objects [26]. Our focus is on reducing false alarms. We detect stable edges by applying temporal accumulation on the output of a foreground edge detection method. These stable edges are then grouped using a clustering algorithm to form

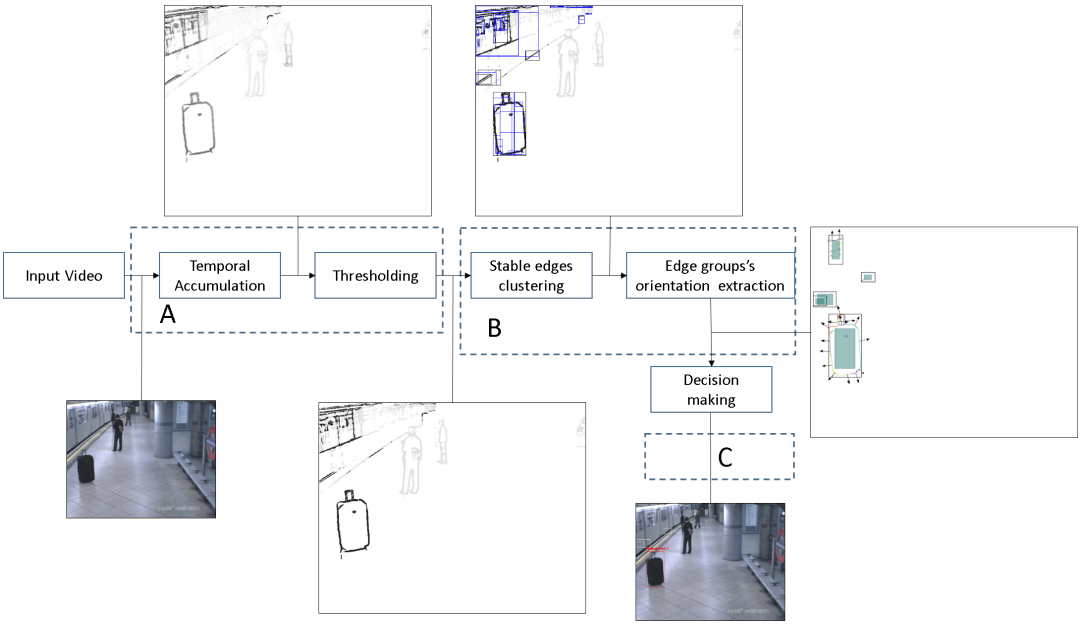


FIGURE 3.1: Flowchart of the proposed method. A:Stable edges detection. B: Stable edges clustering and orientations extraction. C: AO candidates classification.

bounding boxes of AO. Next, in order to classify the AO candidates, we propose a robust score based on the configuration and orientation of the AO candidates' edges to check whether the bounding box really encloses an object or not. An- other score based on static edges consistency is proposed to check the object stability in time, and to reject candidates with small and internal movements like still persons. Figure 3.1 depict the main stages of the proposed system.

3.2 Moving object detection

Background subtraction is the most used technique for separating foreground object from the background. It is the process of constructing a background model by learning the scene using a number of frames, then the foreground areas are simply obtained by comparing this background model with the incoming frame. In order to maintain the background up to date, a background

updating is performed at each frame input to absorb the scene background changes. The speed of updating differs according to the application, and is controlled by a parameter called the learning rate. Moreover a shadow removal step is performed on the resulting foreground mask, this is usually done by comparing the texture of the foreground region with the textures of the background image.

Most background subtraction based methods proposed in the literature[95, 48, 92] use pixels intensities to estimate the background model. There are a number of problems related to the background subtraction phase that can affect the abandoned object detection robustness like sudden illumination and ghost effects generated by a moved background object. In this work we assumed that these problems can be overcome using edges instead of pixel intensities in estimating the background model. These are the main reasons why we opted for the use of edges instead of pixel intensities in estimating our background model. Edge magnitudes are invariant to gradual and sudden illumination changes which according to[32] cause 38% of false alarms in abandoned object detection applications. Moreover, no shadow removal step is needed when using edges since intensities difference between a shadow zone and its outside is negligible.

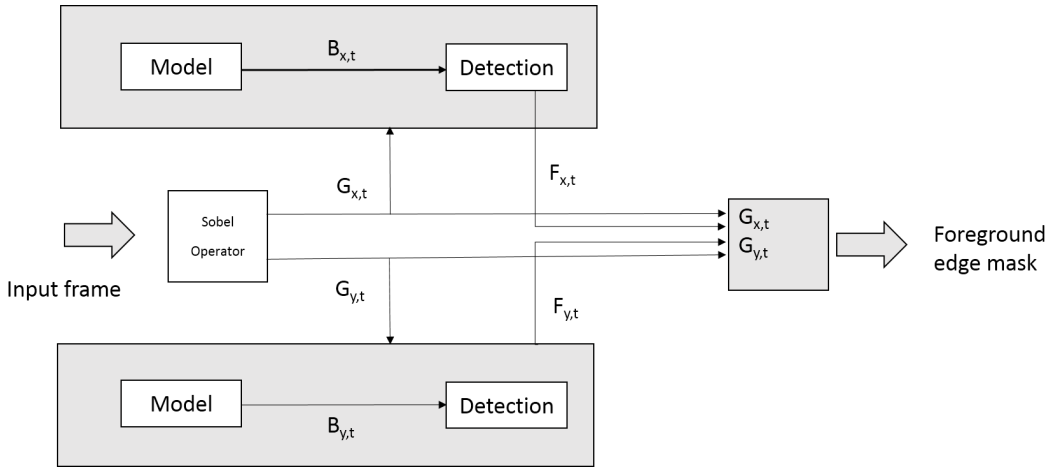


FIGURE 3.2: Block diagram of the edge based background subtraction method

3.2.1 Moving edges extraction

Few methods for moving edges detection were proposed in the literature [35, 45, 27]. In this work, to separate foreground edges from the background, we use a slightly modified version the method proposed in [35], see figure 3.2. The method operates in the X and Y directions independently, i.e. the background model is estimated in the X and Y directions independently. At each frame input, the Sobel operator is applied to compute edges magnitudes in the X and Y directions. Then running average is applied to each direction to estimate the background model gradient in both directions. The difference between the actual frame and the estimated background model is used for update, see equations 3.1 and 3.2. Two foreground masks are obtained, one for the X direction and one for Y direction by comparing the gradient of the input frame with the background model for the corresponding, direction see equation 3.3.

$$B_{x,t}(x,y) = B_{x,t-1}(x,y) + \alpha D_{x,t}(x,y) \tag{3.1}$$

$$D_{x,t}(x,y) = G_{x,t} - B_{x,t}(x,y) \tag{3.2}$$

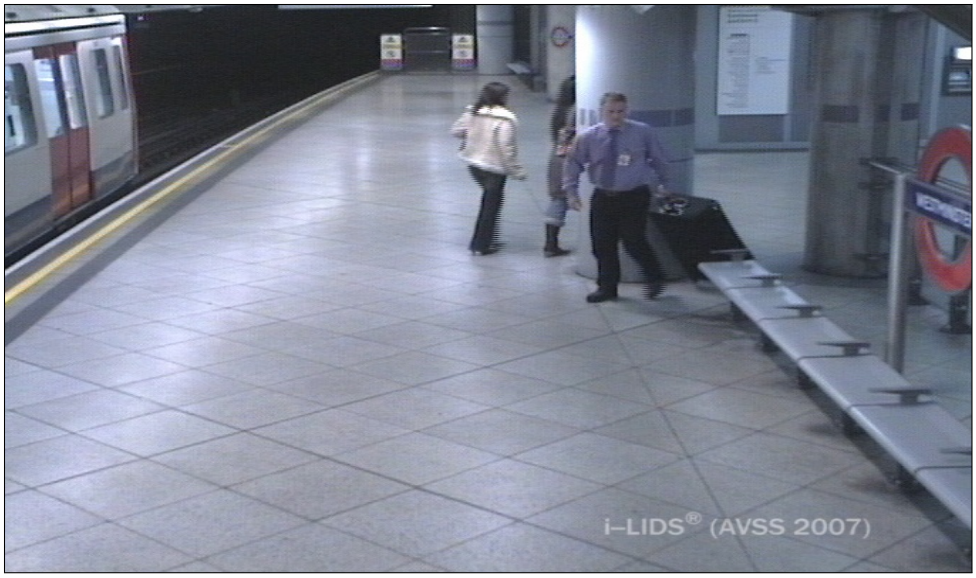
$$F_{x,t}(x,y) = hyst(|D_{x,t}|, T_{low}, T_{high}) \tag{3.3}$$

where $B_{x,t}$, $D_{x,t}$, $F_{x,t}$, $G_{x,t}$ and α are respectively the background model in the X direction, the difference between the background model and the current gradient image, the binary foreground edges mask, the gradient difference in the X direction, and the learning rate(the same applies for the Y direction).

$F_{x,t}$ and $F_{y,t}$ are calculated using a hysteresis thresholding function. Pixel values higher than T_{high} are set to 1, while those lower than T_{low} are set to 0. However pixel values between T_{low} and T_{high} are set to 1 if one of the pixels in the eight-connected neighbors has a value greater than T_{high}, otherwise they are set to 0. Finally, the foreground edges mask F is obtained using an OR operator between $F_{x,t}$ and $F_{y,t}$. Unlike the authors in [35] who use a short-term model with a high learning rate combined with a long-term model with a low learning rate to suppress noisy and unwanted moving edges, in this work, and for simplicity, we use only one model to detect the moving edges. Figure 3.3 shows the built background model and the output foreground edges mask.

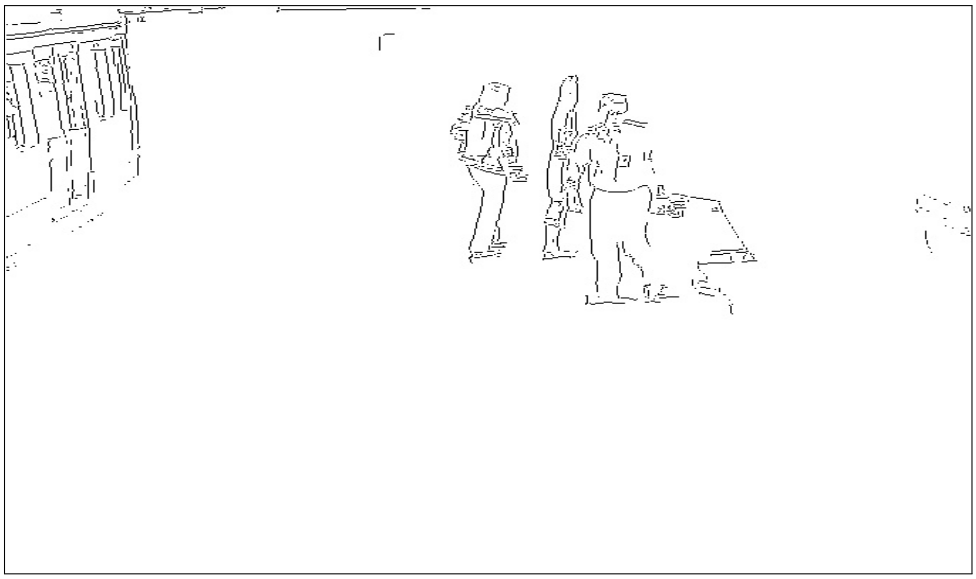
3.2.2 Stable regions detection

This task consists of identifying relatively static regions, which means objects or regions that move and then become static for a period of time. We use extracted moving edges and statistically check their motion stability in



(A)

(B)



(C)

FIGURE 3.3: Moving edges detection a). Input frame b). Background image c). Foreground image

time. Formally, at each frame, the resulting moving edges mask is temporally accumulated, then we identify static edge pixels simply by applying a threshold operation on the temporal accumulation mask. Simultaneously, the grouping algorithm attributes a label to the new static edges and group them with existing object edges or creates new object labels consisting only of these new edges. In order to give the temporal accumulation mask a short memory aspect, at each frame, static edge pixels are incremented by a value of 2, and all edge pixels, static or not, are decreased by a value of 1, ensuring that edges that have been static for a short period of time will be forgotten.

3.2.3 Temporal accumulation

As stated above, stable edges are extracted using a temporal accumulation on the moving edges mask. At every frame input, a pixel in the temporal accumulation mask is incremented by a value of 2 if the pixel at the same location in the moving edges mask has value of 1, see equation 3.4.

$$ACC_t(x,y) = \begin{cases} ACC_{t-1}(x,y) + 1 & \text{if } (F_t(x,y) = 1 \;\&\; i\%\Delta = 0) \\ ACC_{t-1}(x,y) - 1 & \text{if } (F_t(x,y) = 0 \;\&\; i\%\Delta = 0) \end{cases} \tag{3.4}$$

Frequency of the temporal accumulation

To avoid taking into account temporarily static and slow moving objects, ACC is incremented every Δ frames instead of doing frame by frame accumulation. The Δ value is defined in a way to avoid noisy accumulation from slowly moving objects, while not incrementing it too high to avoid missing static edges accumulation. Experiments were done to identify the right value

for Δ, when setting it a lower value of 1, the resulting temporal accumulation mask ACC contains very noisy information. On the other hand, when the value of Δ is set to 10, the mask captures only relevant information i.e. true edge pixels of static objects.

Stable edges extraction

Stable edges are obtained by performing a hysteresis thresholding on the ACC mask instead of single valued thresholding to recover under-accumulated stable edges due to partial occlusions and slow moving objects that result in low values in ACC. The output mask $SEMask$ contains the resulting stable edges, and AO_{time} is the threshold to say if an edge belongs to a stable object, see equation 3.5.

$$SEMask_t(x, y) = hyst(ACC_t(x, y), AO_{time}/2, AO_{time}) \tag{3.5}$$

Finally stable edges magnitudes are extracted by performing an AND operation between Gx and Gy masks, then by applying non maximum suppression (NMS) in order to obtain thinned edges, needed further for classification using the following formulas:

$$Sg_{x,t}(x,y) = \begin{cases} G_{x,t}(x,y) & SEMask(x,y) = 1 \\ 0, & otherwise \end{cases}$$

$$Sg_{y,t}(x,y) = \begin{cases} G_{y,t}(x,y) & SEMask(x,y) = 1 \\ 0, & otherwise \end{cases} \tag{3.6}$$

Illumination changes handling

Both sudden and gradual illumination changes are a big challenge for video surveillance, particularly in outdoor scenarios. Noise occurring from illumination changes was one of the motivations in choosing the use of edges for motion detection. Defining the scene as a sharp local change reduces the illumination changes impact, since in theory an illumination spot effect will have the same amplitude on the affected zone. In practice, in easy scenario situations, the illumination changes will not affect the background model, however when the illumination changes is strong, there may be changes in edges thickness, but this will not cause erroneous output since we deal with edges as segments and not at as pixel intensities.

Ghost effect problem handling

The ghost effect is a background subtraction related problem, and can lead to false detections for the stable regions module. It is caused by a moving of

a region/object that was belonging to the background. In pixels based background subtraction methods like the Gaussian Mixture Model [95], ghost regions in the foreground object binary mask, are the result of a removed region that belongs to the background. Thus, the system can process it as true foreground object. In our method, when a region/object belonging to the background model is moved away, it results in random edges that are accumulated in time, then results in the stable region mask. The advantage of our method is the representation of the ghost object. Since we use edges, the resulting edges from the ghost effect are randomly organized, and thus we can easily identify those ghost regions.

3.2.4 Edges clustering

After extracting stable edges, we need to identify to which object belongs every edge segment. To do this, we apply a clustering algorithm that uses two constraints to group two edges; stability time and spatial distance between two edges. At its first apparition in $SEMask$, an edge is enclosed within a rectangle, and is tracked, and its lifetime is recorded. The distance between two rectangles is simply computed by calculating the minimum distance of four corners of the rectangles. The algorithm sweeps the unlabeled rectangles recursively verifying the distance and time difference between two rectangles to propagate the labels.

As stated above, stable edges are tracked in time; when a static edge is accumulated in the stable edges mask $SEMask$, it keeps changing since its pixels will not reach the AO_time threshold simultaneously, then the algorithm must keep knowing that the same edge is growing. Moreover, a stable edge in the $SEMask$ is attributed a lifetime counter.

The algorithm is both iterative and recursive; the algorithm sweeps all the rectangles in the mask. However, when the algorithm gets to a rectangle, it checks all the rectangles in its neighborhood, and if the the neighboring rectangle is unlabeled, then it propagates the label to that rectangle if it satisfies the time and distance conditions. Then in a recursive way, the algorithm jumps to that new labeled rectangle to perform the same operation. See algorithm1.

Even if the threshold distance D_th is a parameter that is easily defined, it must be defined based on the scene being monitored since it depends on the distance of the camera from the scene ground. On the other hand, the time threshold T_th may be a little tricky to define , since it depends on how much occlusion is present in the scene.If T_th is defined at a low value , stable edges resulting from objects with partial occlusion will not be labeled together, since they will not accumulate and not appear at the same time as $SEMask$.

The proposed grouping algorithm handles situations of two static objects superpositions with different abandonment time by detecting each one in a different bounding box separately. Moreover, our algorithm is perfect for edge clustering since it uses label propagation which can group circular edges layout which is the case for convex objects boundaries.

Figure 3.4 shows an example of the edge segments grouping; the bottom row shows the $SEMask$ map evolution in time at frames 1110, 1570, 2070. Static edges are enclosed in the blue boxes, and the black boxes are the abandoned objects candidates, resulting from clustering edge boxes with the same label. When using the T_{th} threshold for Staticness time, the algorithm works well in cases of close abandoned objects or even overlapping objects with different times of drop, by generating two bounding boxes for each object.

Algorithm 1 Pseudo code listing of the proposed clustering algorithm

```
1: counter ← 0
2: Initialize all n Rectangles labels to 0
3: procedure GROUPING
4:     for k ← 0, n do
5:         PROPAGATE(k)
6:     end for
7: end procedure
8: procedure PROPAGATE(i)
9:     for j ← 0, n do
10:        if (j ≠ i && Rect_j.label = 0) then
11:            if (dist(i, j) <D_th && Rect_i.time − Rect_j.time < T_th) then
12:                if Rect_i.label = 0 then
13:                    Rect_i.label ← counter
14:                    Rect_j.label ← counter
15:                    Counter + +;
16:                else if Rect_i.label > 0 then
17:                    Rect_j.label ← Rect_i.label
18:                end if
19:                PROPAGATE(j)
20:            end if
21:        end if
22:    end for
23: end procedure
```

3.2.5 Classification

In order to reduce the false alarms rate, a classification step is necessary after static object detection. False alarms result from spurious detections of stable edges from sudden illuminations, ghost effect and other noise sources. When a new object is formed in $SEMask$, it is checked whether it is a true abandoned object or a false detection. Abandoned objects do not have a regular shape, so applying a learning based object detection to verify the Objectness of the AO candidate is not feasible. Therefore, a more general model is needed to classify these candidates. We have been inspired by the recent works on category independent object detection [94, 22, 28]. The idea is to

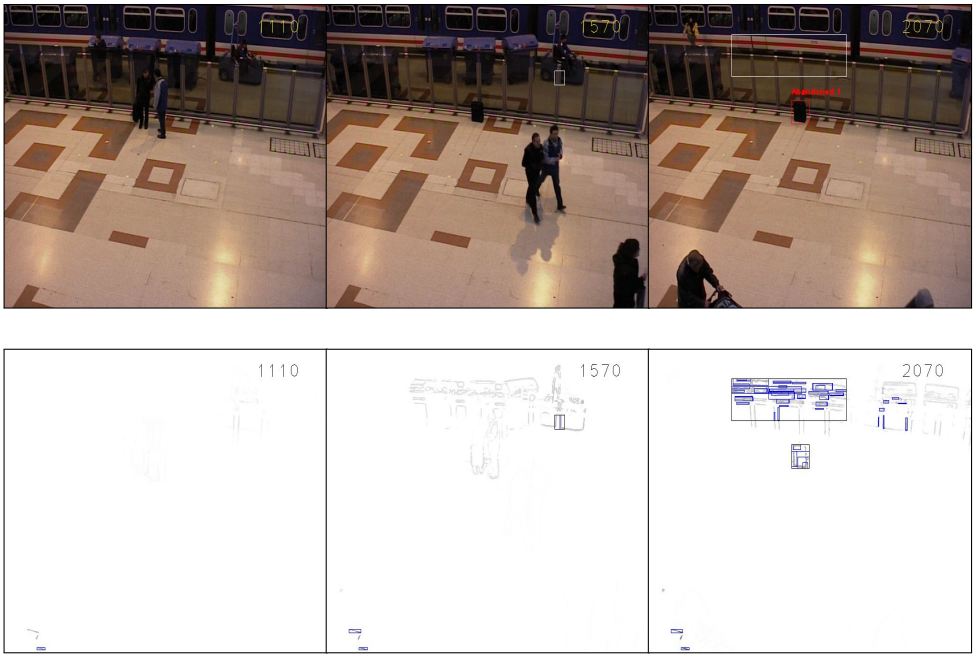


FIGURE 3.4: Static edges accumulation in time at frames 1110, 1570 and 2070.

detect an object without caring about to which category it belongs. It is intended to speedup the learning based object detection by applying the classifier only on a set of object candidates (1000 proposals for example), instead of sweeping the entire image. Despite the fact that these techniques have a high Recall, their precision is not high enough. However, with a prior knowledge, which is the set of AO candidates output in the $SEMask$, the scores are applied on these candidates to maximize the precision of our detection method.

Two scores are proposed to filter out false detections [26]. The first score is the Objectness score, and it is used to verify whether the edges enclosed in the bounding box represent the boundary of a true object or not by using the stable edges configuration and orientation within their bounding box. The second one is the Staticness score and it is based on the stable edges enclosed in the bounding box, connectivity and defects to check and filter out objects with inner motion that results in these defects like an immobile person.

Objectness score

We take as assumption that an abandoned object or luggage has in general simple convex boundaries, in defining our Objectness score. The Objectness score proposed in [94] works well for simple high scale images with clear background, but it is not suited for complex surveillance videos, because of the complexity of the scenes and sizes of AO candidates that can be very small.

The proposed Objectness score is based on the assumption that objects are things with well-defined circular boundaries i.e. we use the object boundary convexity characteristic. To achieve this, we verify the candidate object convexity by checking its edges orientations near the bounding box boundaries. First edge segments are represented using edge group representations and their average gradient directions are proposed in [94], where edge groups are formed by grouping connected edge pixels according to their orientations until the sum of their gradient orientations difference exceeds $\pi/2$. Four sides of the candidate bounding box are defined for edges orientations checking, top, bot, right and left, see figure 3.5. Next for each bounding box, the orientation of each edge segment is then calculated. This is done by computing the mean of all edge pixels gradient directions that are composing the edge segment. It is to note that edge segments that are far from the bounding box boundaries are neglected. Given the edge segments orientations, we compare each edge segment orientation with its corresponding bounding box boundary, and then edge segments satisfying this condition are taken into consideration in scoring, in other words, the quantity of edge segments in

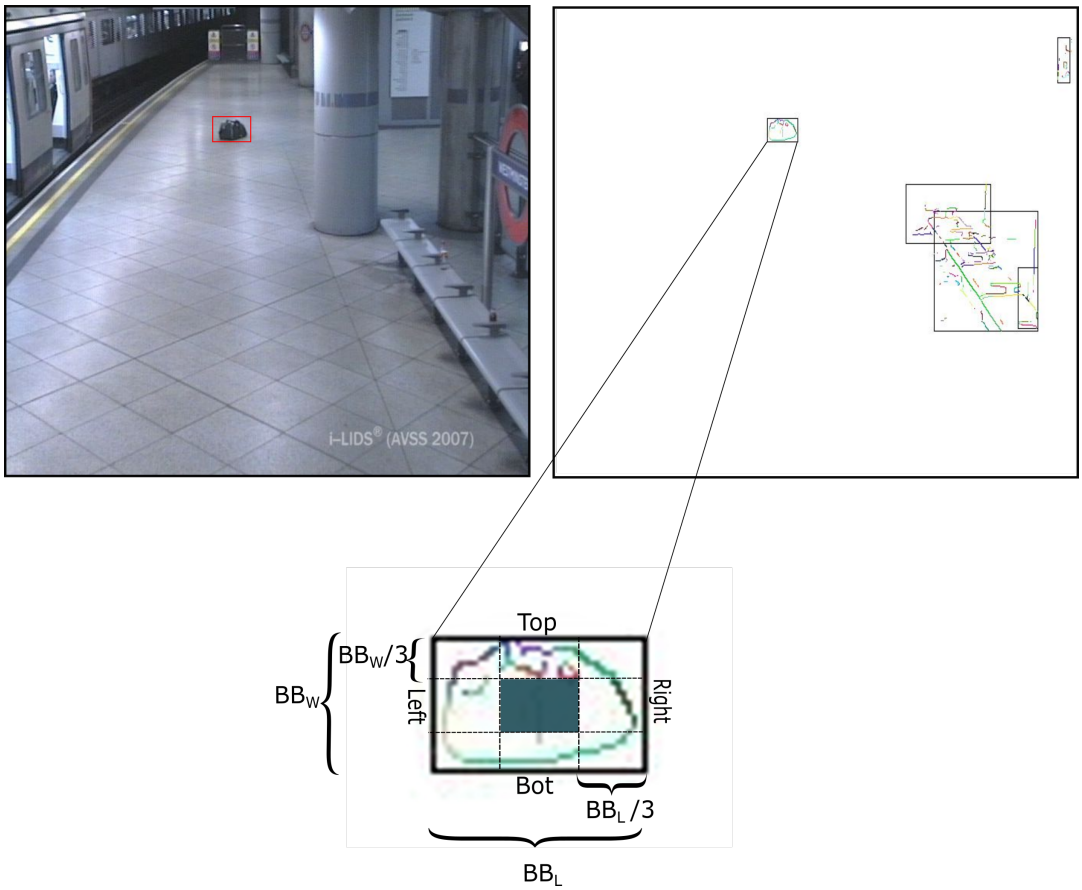


FIGURE 3.5: Dividing of Bounding box in regions

each side satisfying the convexity conditions.

$$T_{reg} = T_{reg} + segLength_i \quad \text{if } |\sin(\theta_i)^2 - 1| < \sigma \tag{3.7}$$

$$B_{reg} = B_{reg} + segLength_i \quad \text{if } |\sin(\theta_i)^2 - 1| < \sigma \tag{3.8}$$

$$R_{reg} = R_{reg} + segLength_i \quad \text{if } |\sin(\theta_i)^2| < \sigma \tag{3.9}$$

$$L_{reg} = L_{reg} + segLength_i \quad \text{if } |\sin(\theta_i)^2| < \sigma \tag{3.10}$$

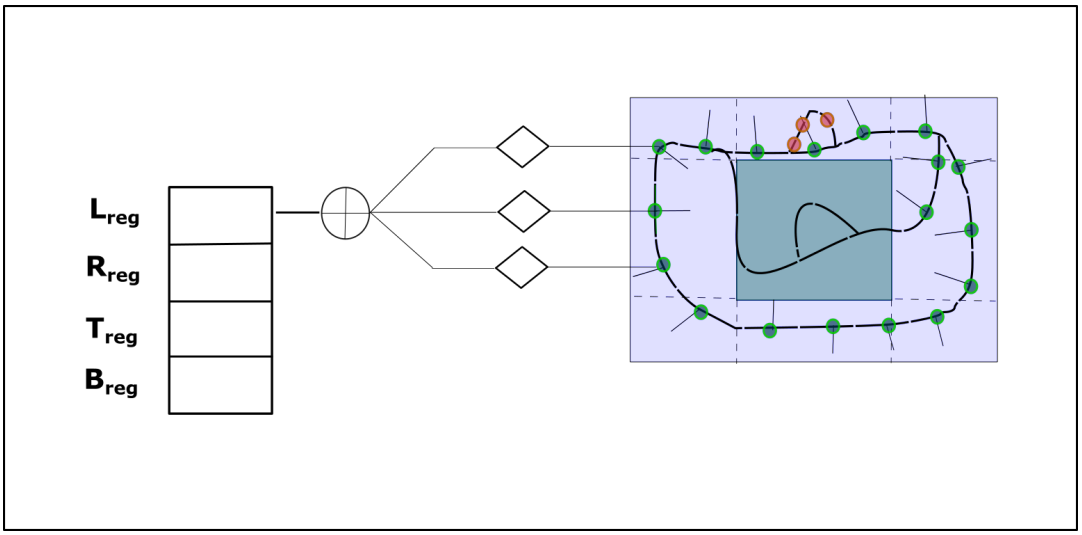


FIGURE 3.6: Illustration of the gradient direction of the AO candidate edges groups. Edge groups with green circle on the left region satisfy equation. 3.10

Where θ_i and $segLength_i$ are respectively the mean orientation and the length of the i^{th} edge group in each region.

T_{reg}, B_{reg}, R_{reg} and L_{reg} are respectively the sums of the edge groups lengths that satisfy the convexity condition in the Top, Bot, Right and Left regions (figure3.6, and figure 3.7). For Top and Bot regions, lengths of edge groups with a mean orientation near β=($\pi/2$ or $3\pi/2$) are accumulated when ($|\sin(\theta_i)^2 - \sin(\beta)^2| < \sigma$) (eq. 3.7 and 3.8). For Right and Left regions, lengths of edge groups with a mean orientation near β=(0 or π) are accumulated when ($|\sin(\theta_i)^2 - \sin(\beta)^2| < \sigma$).(eq.3.9 and 3.10). Finally we compare T_{reg} and B_{reg} with the bounding box length BB_L, and compare R_{reg} and L_{reg} with the bounding box height BB_W, by defining the following Objectness score S_b:

$$S_b = \frac{\lambda}{((L_{reg} - BB_W) * (R_{reg} - BB_W) * (T_{reg} - BB_L) * (B_{reg} - BB_L))^2} \quad (3.11)$$

λ being a constant.

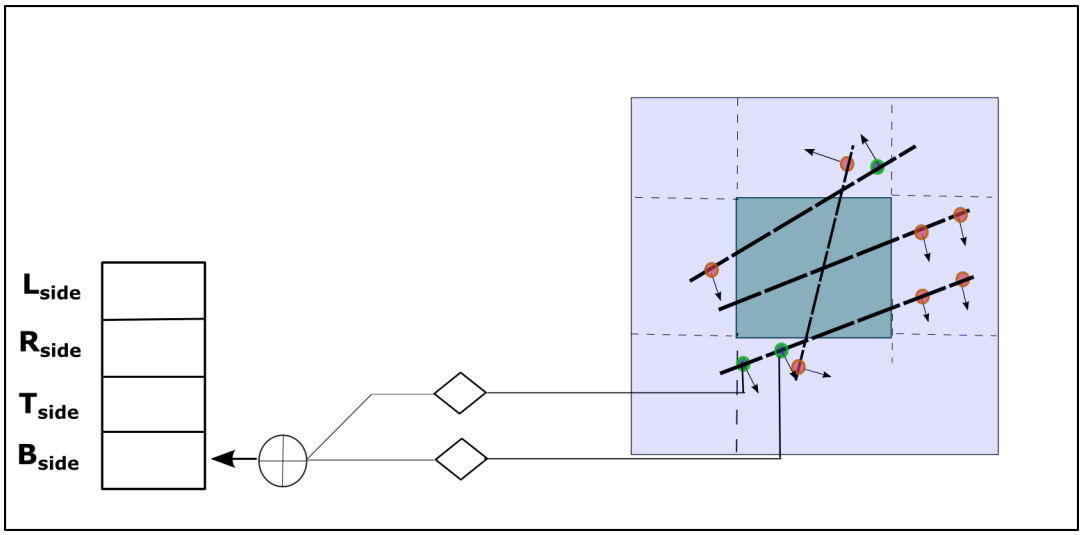


FIGURE 3.7: Illustration of the gradient direction of a non object candidate edges group

Staticness score

The Objectness score is intended to filter false alarms based on their shapes like illumination spots, ghosts random objects and other spurious detections. There are other types of false alarms that do not apply to the abandoned object motion pattern like still persons for example. Still persons can have high Objectness in some cases like a sitting person. An idea is to detect internal small movements. We define a Staticness score to capture the objects motions based on the object edges consistency and connectivity. i.e. the observation is that static objects with small and internal movements result in fragmentation and non-consistency in their edges.

The approach to compute the score is simple. A true abandoned object has no internal motion like still people, thus when temporal accumulation is applied, the resulting stable edges are consistent with no defects neither fragmentation. On the hand, a static object with small internal movements would result in small stable edges fragments and defects. The score is calculated using the edge segments representation. The assumption is for a complete

object boundary, an edge segment has at least two connections with its neighboring edge segments. The Staticness score C_b is defined as follows:

$$C_b = \prod \frac{|\phi(i)|}{2} \tag{3.12}$$

$\phi(i)$ is the set of inter-edges connections for the i_{th} edge group.

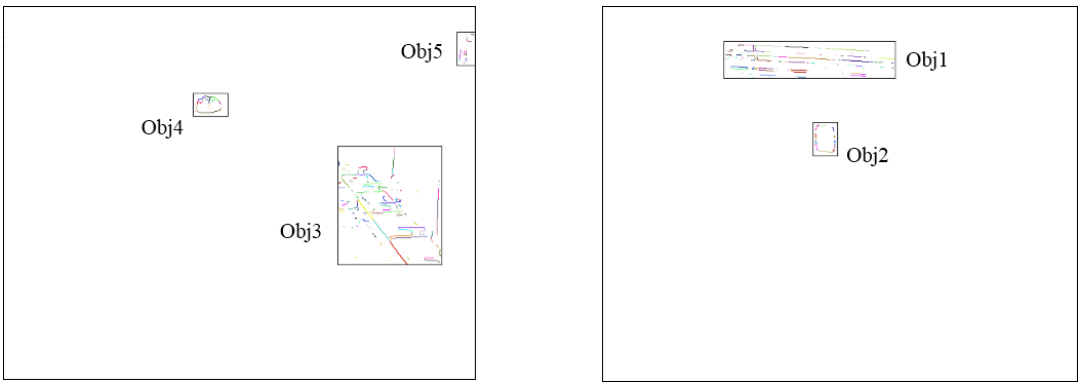


FIGURE 3.8: Scores of different AO candidates from PETS2006 S2 scenario (right) and AVSS2007 Medium video(left) :Obj1 (S_b=0,126.10-12, C_b=0) Obj2 (S_b =0,0008, C_b =0,00018) Obj3 (S_b =2,123.10-5, C_b =0) Obj4 (S_b =8,92, C_b =3,796) Obj5 (S_b =1,656.10-21, C_b =1,5652.10-5)

Fig.3.8 shows some Objectness and Staticness scoring examples of some AO candidates. $Obj1$ is a ghost left by the garbage removed from the scene and therefore highly fragmented random edges with no defined boundaries are clustered in a bounding box, the Objectness and Staticness are too low and $Obj1$ is rejected from being an abandoned object. On the other hand, $Obj4$ and $Obj2$, which are both objects and have remained static for a defined period of time, have high S_b and C_b scores. Thresholds for both scores will be defined in the next section. The C_b score is effective in cases of static humans seen as AO candidates, with the fact that even in a static state, a human makes some small internal movements. This results in highly fragmented edges because edges are sensitive to small movements.

Ghost effect classification

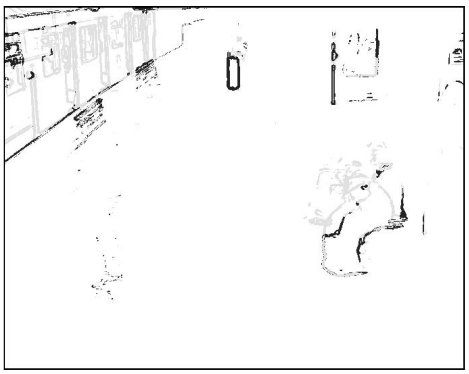
An abandoned object candidate resulting from a ghost effect consists of a random edges configuration inside the AO bounding box. Most methods employ a dedicated score to filter this false alarm source. Their scores are based on the shape resulting from the moving edges mask; the contours pixels of the foreground object are compared with the edges pixels of the incoming frame at the same position, similarity is quantified, and the object is said a ghost if the similarity of the two contours is low. We consider this method costly, and not effective since it depends on the foreground mask precision, and edges from moving objects overlapping the presumed ghost position can be introduced in the similarity calculation. As stated before, our Objectness score is inspired by the human perception, we do not need to identify each false alarm, we only need to see things of interest which are objects. By applying the Objectness score to each resulting AO candidate, we check if there is an object inside, and when an object belonging to the background is moved away, it results in random and sparse edges configuration, and thus can be classified by our Objectness score as a non object. In figure 3.7 , an illustration of a false detection is depicted. As shown edge segments no satisfying the convexity condition defined by the Objectness condition are neglected resulting in a low Objectness score value.

Static human

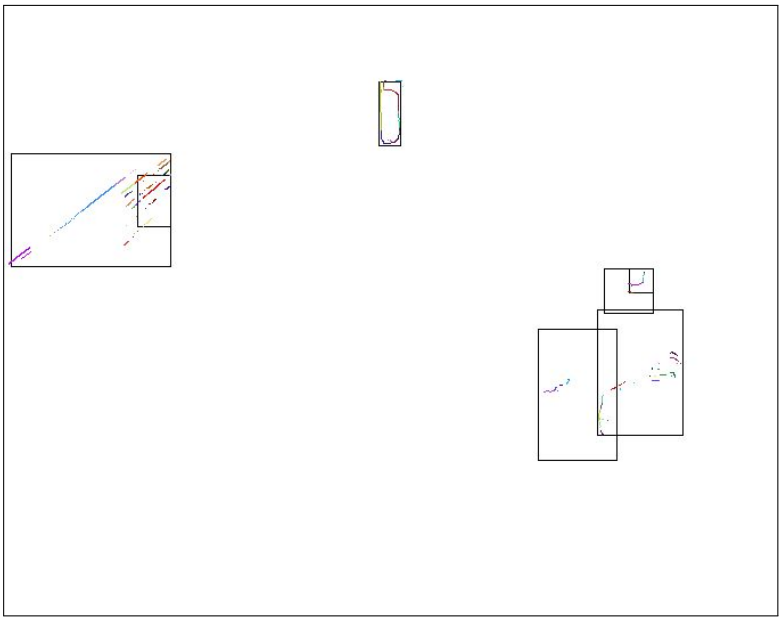
Static human, is a major source of false alarms especially in situations of crowded areas. For example, in a train station, a person enters the scene and is waiting in a standing or sitting position for the train arrival. The

edges of the static person will be accumulated and a bounding box grouping those static edges is created on $SEMask$. The Objectness can be high since a human has somehow a convex object shape especially for sitting persons. However, the resulting stable edges are highly fragmented, and that is what our Staticness score captures. Persons have high internal movements, these movements are captured even if they are of small amplitudes since we use edges.In figure 3.9, a situation of a sitting person is encountered, we observe that the stable edges in the $SEMask$ mask and after edges clustering, do not represent the fully shape of the sitting person, this is due to the body movements of that person. The Staticness score detects the degree of fragmentation of the stable edges and then filters out the still human as a false detection.

(A) Input frame



(B) Stable edges mask $SEMask$ after thresholding static edges



(C) edges groups representation for AO classification

FIGURE 3.9: Static human false alarm detection.

Decision making

We use two thresholds $T1$ and $T2$ for the two scores S_b and C_b, respectively. If S_b is greater than T1 and C_b is greater than T2, then the inside object will be validated and considered as a true abandoned object. To avoid taking into account moving objects edges occluding the AO candidate, only the edge distributions with an accumulation value higher than $(AO_{time}/2)$ are considered in scoring.

Chapter 4

Experiments and results

4.1 Experimental setup

For experiments purpose, we use a general-purpose laptop with an I7 processor. We use C++ as the programming language to fit the real time requirements of the visual surveillance application. The OpenCV computer vision library is used for acquisition and preprocessing purposes. The method is evaluated on the publicly available datasets used in the abandoned object detection literature [24]: I-LIDS's AVSS2007 [6] , PETS2006[67], and PETS2007[68]. Moreover, the method was also tested on some newly introduced state of the art benchmarks as CDNET2014[14], and ABODA[2] datasets. The evaluation is performed using Recall, Precision and F-measure. Recall is a ratio of true detection to the ground truth, while Precision is a ratio between the number of true detections and the number of all detections. F-measure is the harmonic mean between Recall and Precision.

$$\text{F-Measure} = \frac{2xRecallxPrecision}{Precision + Recall}$$

4.2 Parameters & thresholds

Table 4.1 summarizes the parameter values used in the algorithm. $T_{low} = 40$, $T_{high} = 70$, and $\alpha = 0.005$ are for the background maintenance. They were chosen in a way that the absorption time of an object in the background model is greater than the abandonment time duration AO_{time}. AO_{time} is a user-defined parameter. The σ threshold for an edge group to be taken into account in scoring was set to 0.5, which means that the difference in orientations between an edge group, and its relative bounding box's bound should be inferior than $\pi/4$. λ is a constant and is set to 10^8. Thresholds $T1$ and $T2$, for S_b and C_b respectively, were decided by conducting experiments on AVSS2007 videos, to find the optimal cutoff between the true positives class and the false positives class; see Fig.4.1. Five threshold configurations were chosen to decide the best cutoffs for both scores. Performance of each configuration is measured using Recall, Precision, and F-measure.

Fig.4.1 shows that when the values decrease for T1 and T2, Recall is high, but Precision decreases, which means that all positives are classified correctly, but there is a high false positive rate, so decreasing T1 and T2 means more tolerability. On the other hand, for higher values of T1 and T2, Recall decreases because not all true positives are detected, while precision is high and consequently the classification is conservative. We observe that the configuration T1=10^{-5} and T2=10^{-5} results in the best performance, with Precision and Recall both equal to 1.0. Consequently, these values were chosen as thresholds for the algorithm. It is to note that the thresholds for classification are the same for all datasets and needed no further tuning. For the classification step, the value for threshold σ was set to 0.5, allowing to take

TABLE 4.1: Summary of the parameters used in the algorithm

Parameter	α	T_{low}	T_{high}	σ	T1	T2	λ
value	0.005	40	70	0.5	10^{-5}	10^{-5}	Constant

into account only edge groups that have orientations difference with the corresponding bounding box boundary less than $\pi/4$. The constant λ is set to 10^8. The T1 and T2 threshold values used for the Objectness score S1 and the Staticness score T2 respectively, are determined by conducting experiments on videos from the AVSS2007 dataset. The optimal cutoff between the true positives class and the false positives class is determined.

We use five configurations for the (T1, T2) thresholds to determine the best cutoff, by measuring each configuration performance using Recall, Precision and F-measure performance metrics.From the performance results of each configuration depicted in figure 4.1, we observe that for higher values of T1 and T2, we have a low Recall, due to the missed true positives, while precision is at maximum since no false positives are detected as true detections, so higher values of T1 and T2 lead to a conservative classification.

On the other hand, when we decrease T1 and T2, Recall is at maximum, which mean that all true positives are correctly classified. However, this is on the cost of precision, which decreased drastically allowing for too much tolerability. The ($T1 = 10^{-5}$, $T2 = 10^{-5}$) configuration has resulted in the best performance, with Recall, and Precision both equal to 1.0. This configuration has been adopted and used in the algorithm for the evaluation on all datasets without any further tuning.

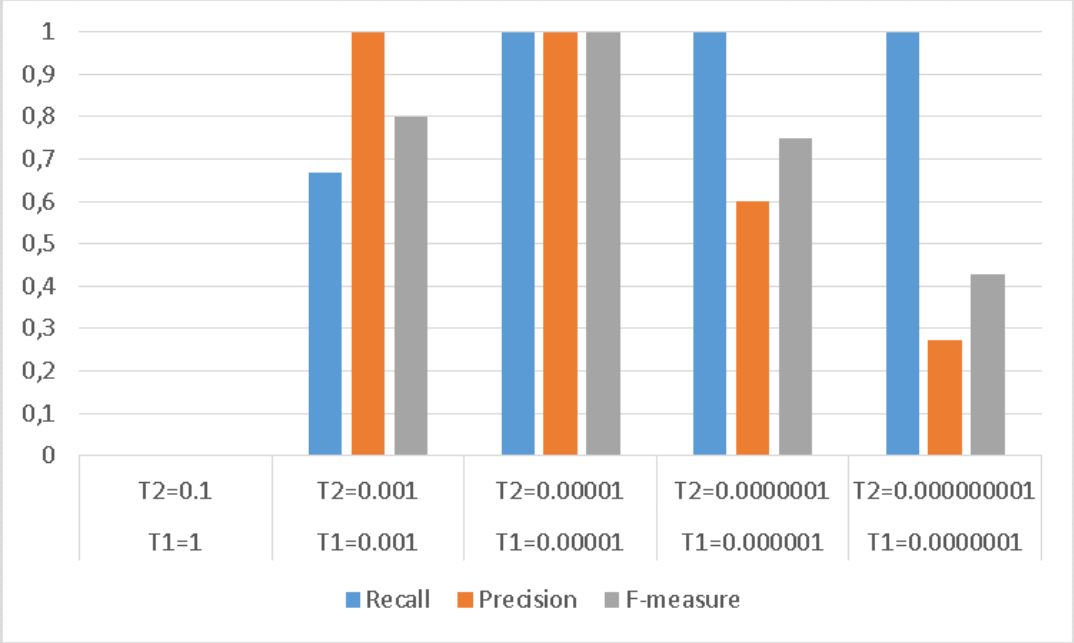


FIGURE 4.1: Algorithm evaluation on AVSS2007 using different threshold values

4.3 Results and discussion

4.3.1 Datasets-Qualitative results

PETS2006 - The dataset contains seven scenarios of a train station, each one with a different scenario and its own difficulty level. Each scenario has four videos, each one with different view angles. The abandoned objects include luggage, suitcases, and ski gears. The third view angle is chosen because it offers a better field of view of the scene. The video contains an event of people dropping their luggage on the floor and leaving. The challenges are moving crowds and background object moving (resulting in ghosts). Figure 4.2 shows our method results on scenarios 3, 5, and 7. The proposed method detects all abandoned objects with precision while filtering false alarms like in scenario 2, where a garbage object belonging to the background image is moving. In this scenario, our classification module filters the resulting ghost

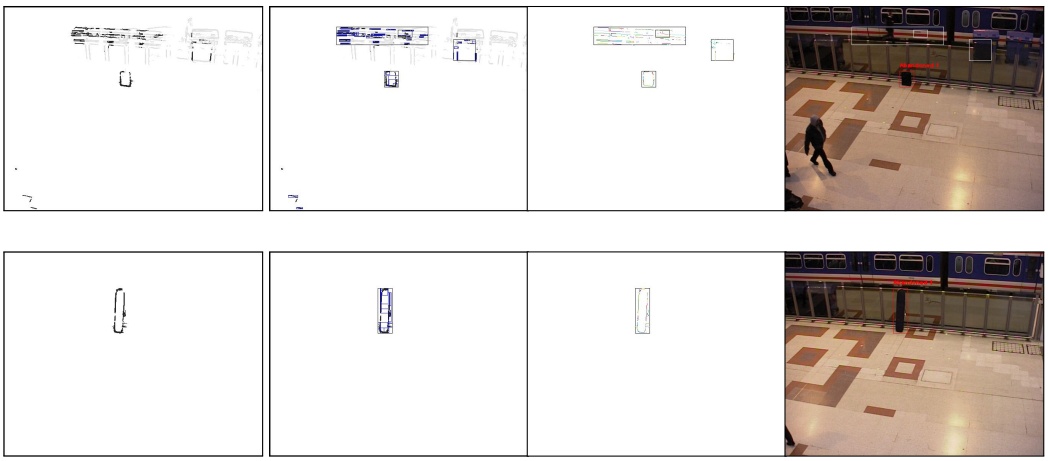


FIGURE 4.2: PETS2006 abandoned object detection examples

perfectly, unlike the method proposed in [46] where the resulting ghost from the moving garbage is classified as true abandoned object. Moreover, the crowds occluding the AO candidate in video 7 do not affect the robustness of our algorithm .

PETS2007 -

This dataset contains eight videos with different scenarios: loitering people, left object and abandoned object scenarios. This dataset also has four view angles and we have used the third one. Video 7 and 8 contain the abandoned objects scenarios. This dataset is very challenging in term of sudden illumination changes. Figure 4.3 shows the obtained results. We have noted that, due to high noise represented by the strong sudden illumination changes and high moving crowds' quantity, the AO candidates output by the static region detection module is very high. However, using the proposed scores, our classification module filters the false alarms correctly and detects only true positives.

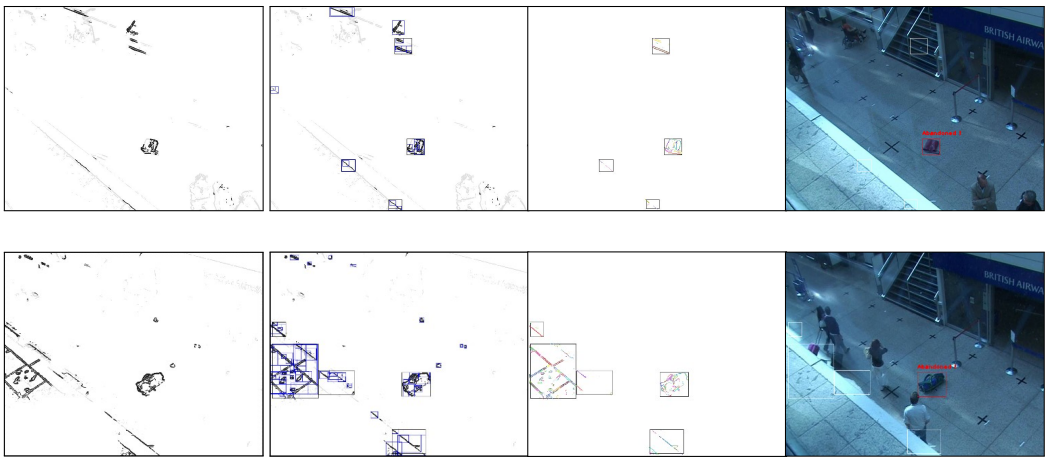


FIGURE 4.3: PETS2007 abandoned object detection examples

AVSS2007 -

The dataset contains three videos of the same metro terminal scene, with different difficulty levels: easy, medium and hard. The difficulty level is characterized by the distance of the dropping zone from the camera, the people staying still, and the quantity of crowds occluding the dropping zone. Figure 4.4 shows the obtained results on the three scenarios. Our proposed algorithm identifies successfully the abandoned object in the three videos.

ABODA -

Authors in[57] proposed their own dataset for evaluation, it contain eleven videos, each one with different environments and challenges; outdoor and indoor footage, crowded scenes and night detection. There are also videos with sudden light switching, which is very challenging to the background subtraction module. The obtained results on some videos are depicted in figure 4.5. The first column shows the static region detection masks after stable edges clustering, the second column shows the edges groups representation for score computing, and the last column shows the output image. The first

FIGURE 4.4: AVSS2007 abandoned object detection examples

scenario is an outdoor scene where a person leaves a bag on the floor and leaves the scene. The bag is correctly detected as an abandoned object.

The second row shows an outdoor nighttime scenario with a light switching resulting in changing the scene luminosity. Since we use edges in modeling the background scene, this method is not affected by the light switching, while abandoned objects were detected successfully without any false detection. The third scenario is in a classroom, where the light is turned off and the IR mode is turned on resulting in a complete changing in the background model. The static region detection module outputs high amount of AO candidates, the classification module detects the abandoned object successfully, nevertheless there were some false detections. Table 4.2 shows a comparison of the results of the eleven videos of our method with the method proposed in [57] and the method proposed in [87] by counting the number of TPs and

TABLE 4.2: Evaluation and Comparison on the ABODA dataset

Sequence	Scenario	Difficulty	GT	[57]		[87]		Ours	
				TP	FP	TP	FP	TP	FP
Video1	Outdoor	*	1	1	0	1	0	1	0
Video2	Outdoor	**	1	1	0	0	1	1	0
Video3	Outdoor	**	1	1	0	0	1	1	0
Video4	Outdoor	**	1	1	0	0	1	1	0
Video5	Night-time	***	1	1	1	1	0	1	0
Video6	Light Switching	*****	2	2	0	-	-	2	0
Video7	Light Switching	*****	1	1	1	-	-	1	2
Video8	Light Switching	*****	1	1	1	-	-	1	2
Video9	Indoor	*	1	1	0	1	0	1	0
Video10	Indoor	*	1	1	0	1	0	1	0
Video11	Crowded	****	1	1	3	-	-	0	1

FPs for each method. From the table, we note that our method clearly outperforms the method proposed in [87] in most of the videos. However, comparing our method to the results obtained in [57], our method did not present any false alarm in video 5 unlike lin's method [57]. Nevertheless, there was a miss detection in video 11 where lin's method[57] detected the AO successfully.

CDNET2014 -

We have also evaluated our method on the CDNET2014 state of the art dataset from the CVPR 2014 change detection challenge [14]. The dataset is for the evaluation of foreground object detection methods, it contains many motion pattern categories, we have used the intermittent object motion category that contains videos with abandoned/static objects scenarios for evaluating our method. We have compared our results with those obtained in [88]. To do so , we have applied the component analysis on the binary mask obtained in [88] to compare at bounding box level, because it is not possible to compare at pixel level since we use edges instead of pixels. Figure 4.6 shows the obtained results by both our method and the Wang one. The first row shows a situation

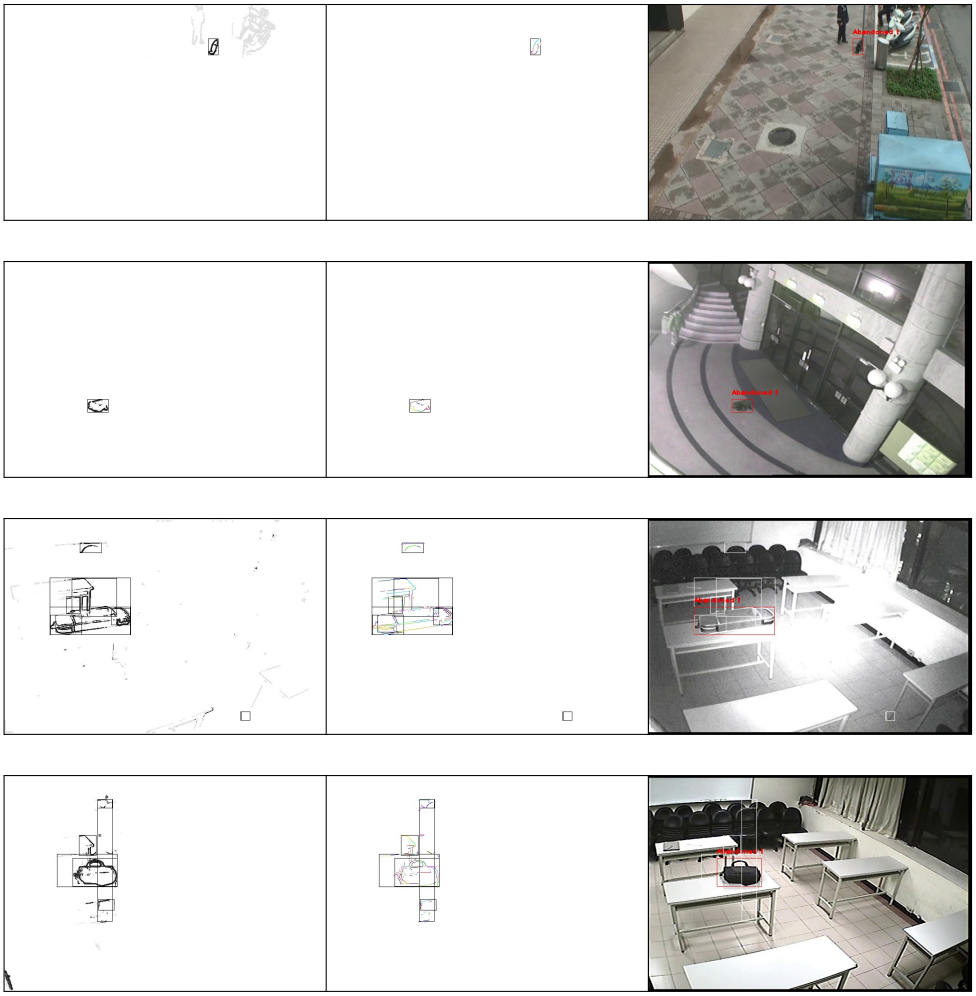
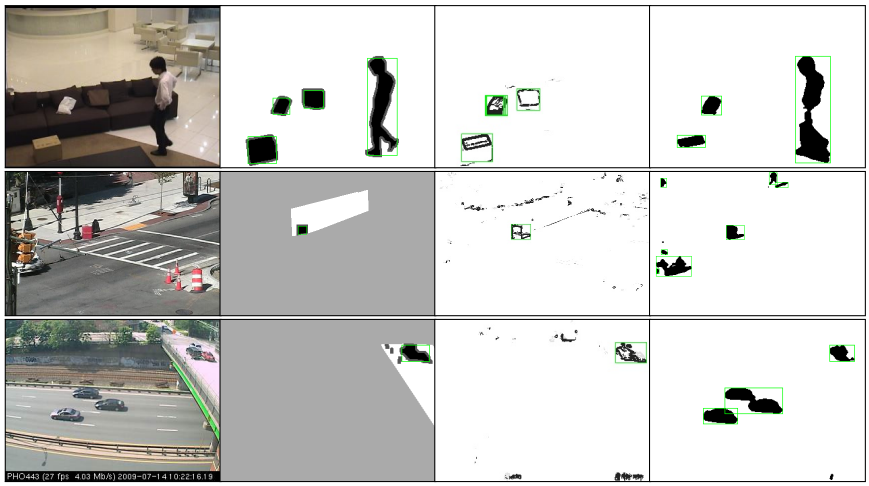


FIGURE 4.5: ABODA Abandoned object detection examples

with three objects and two people. The three objects are left on a sofa and the two persons leave the scene. Our method detects the three objects accurately. In [88], the method presented one missed detection, and another object is not detected accurately because of the low gradient with the carpet. The second row is an outdoor scene with strong illumination changes and waving trees noise. The scenario is a box moved from one position to another. Both our and Wang's methods detected the box accurately. The last row is an outdoor scenario also, but the static object is a cars stopping a streetlight, both methods detected the stopped cars as static objects. It is to note however, that the method proposed in [88] does not make a difference between still people and

static objects.



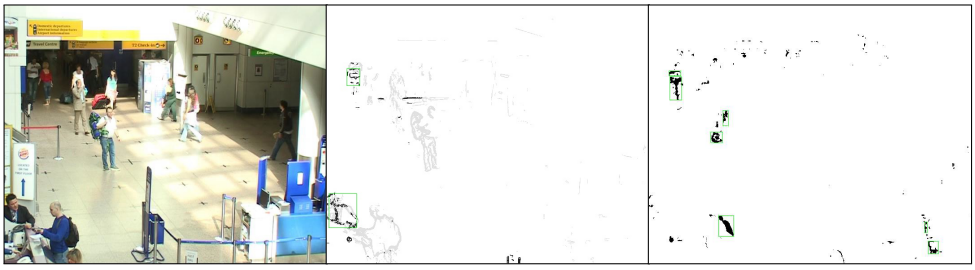
(A) (B) (C) (D)

FIGURE 4.6: Detection results on CDNET2014: first row: Sofa Scenario, second row: Abandoned box scenario and the third scenario: Streetlight scenario. a). Input frame b). Ground-Truth image c). Proposed method static mask d). [88] detection mask image

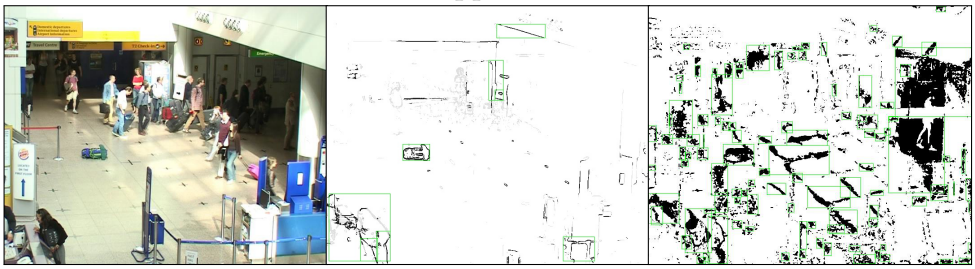
4.3.2 Subjective comparison with the GMM-based method

Most of the proposed methods are based on the pixel intensity based GMM background subtraction technique. To justify our choice for edges instead of pixel intensities in the background subtraction step, we have performed a comparison with the method in [83] which uses the GMM property to identify static regions. The comparison is done at static region detection level i.e. without applying the classification step. We use video 7 of dataset PETS2007 for the comparison due the background related challenges present in these videos like strong sudden illumination changes and the quantity of crowds. Figure 4.7 shows the output of both our method and the method proposed in [83]. The first row shows the output static region masks at frame 705 for both methods. The second row shows the output of the masks at frame 1780.

Compared to the method proposed in [83], our stable region detection module produces a mask with a minimum noise. For the method proposed by Tian [83], the static region detection module yields, a noisy mask affected by strong global and local sudden illumination changes, and the quantity of crowds present in the scene. Consequently, it results in many static region candidates, most of them false alarms. The method proposed in [83] uses three Gaussians set to model the background and relies on the second Gaussian to represent temporally static objects. In a crowded environment, different objects and slow moving crowds with similar color distributions will cross the same regions, which will result in increasing the weight of the Gaussian representing the foreground pushing it in the Gaussians set of the background model. Thus, those regions will be considered as static regions by the static region detection step of the method in [83].



(A) Input image at frame 705 (B) Static objects mask of P.Approach (C) Static objects mask from Tian et al.



(D) Input image at frame 1780 (E) Static objects mask of P.Approach (F) Static objects mask from Tian et al.

FIGURE 4.7: PETS2007's Results comparison of our approach with [83]

4.3.3 Quantitative results and comparison

In order to investigate the robustness of our method compared to other methods, we have performed a comparative study using the performance metrics Recall, precision and F-measure. The experiments are conducted on the PETS2006 and PETS2007 datasets with the state of the art methods [64, 83, 31, 81, 57]. From Table 4.3, we have noted that almost all methods detect abandoned objects accurately, except methods in [31] and in [81], which present missing detections in PETS2006, with a Recall equal to 0.8 for [31] and 0.86 for [81]. On the other hand, the method proposed in [83] detected all objects. Yet, there were false detections in the AVSS2007 scenarios, consequently their Precision and F-measure are lower than those obtained in the other methods. Both our method and the one in [57] have performed perfectly well in both datasets, with F-measure=1.0. However, in [57] the authors have restricted their detection area to the train station platform in AVSS2007, and to the waiting zone area in PETS2006 only. In contrast, our algorithm performs all over the scenes in both datasets. Thanks to our Objectness score and Staticness score, our algorithm is able to filter both shape and motion related false alarms. The method presented by Pan's method [64] have rough results in AVSS2007 datasets, however it was not evaluated on PETS2006 or PETS2007.

TABLE 4.3: Comparison results on PETS2006 and AVSS2007

Method	PETS2006			AVSS2007		
	R	P	FM	R	P	FM
[83]	1.0	0.85	0.92	1.0	0.35	0.52
[64]	-	-	-	1.0	1.0	1.0
[81]	0.86	1.0	0.98	1.0	1.0	1.0
[31]	0.8	0.95	0.87	1.0	0.97	0.98
[57]	1.0	1.0	1.0	1.0	1.0	1.0
P.Approach	1.0	1.0	1.0	1.0	1.0	1.0

4.3.4 Processing time

To demonstrate the real time aspect of our system, we have evaluated the processing time of the algorithm. To do so, we have used the AVSS2007 dataset for which the resolution is 720x576, and the dataset CDNET2014 which resolution is 320x240. The processing time of each module of the system has been evaluated. From Table 4.4, we have observed that the clustering module is the less time consuming module, noting that it depends tightly on the the number of edges in the stable edges mask $SEMask$. On the other hand, edge groups orientations extraction is the most time consuming module. Finally, the stable edges extraction module depends only on the input frame resolution since it is Matrix wise computations. The overall computational speed is 108 fps for 320x240 resolution, and 18 fps for 720x576.

TABLE 4.4: Processing time complexity of the proposed algorithm

Module	720x576	320x240
Stable edges detection	20 ms	4 ms
Clustering	2.4 ms	0.7 ms
Orientations extraction	32 ms	4.5 ms
Overall System	54.4 ms	9.2 ms

TABLE 4.5: Comparison of processing time with other methods on 320x240 resolution

Method	Processing speed
Ours	108 Fps
[81]	49 Fps
[57]	29 Fps

Table 4.5 shows a comparison of our method with the methods proposed in [57, 81] in term of the number of frames processed per second (FPS). From the table, we observe that our method clearly outperforms both methods proposed by Lin [57] and Szwoch [81], with an FPS of 108. We justify this

by the following reasons: the employed edge based background subtraction method in our system is much more time efficient than the GMM used in most techniques in the literature. Another reason is the use of edges, which saves us from including additional processing like the shadow removal module. Moreover, our stable edges extraction module outputs a $SEMask$ with a minimum quantity of noise, and thus less bounding boxes, and therefore low time consuming.

Chapter 5

Conclusion and future works

In this thesis, we have developed a system for automatic abandoned object detection. An edge based background subtraction method was proposed for moving object detection. We have also proposed a grouping algorithm for stable edges clustering. For classification stage, we have proposed two robust probabilistic scores; Objectness and Staticness for abandoned object candidates classification. Our method proved its robustness against classical video surveillance problems (illumination changes, cluttering areas) and effectiveness in real world scenarios. Our system was designed to handle real time video processing, so ideally, it can be deployed in real world situations and assists human operators to detect the abandoned object events in videos and to take actions in a timely manner.

Future works

Our approach can be improved and extended in the following manners:

- The Objectness score can be further improved by including a combination of cues like super-pixels straddling instead of using only the edges distribution cue.

- An owner tracking system can be further included to track the owner from the object dropping time. Considering the crowded scenes, the tracking module would be a head or a head-shoulder based detection based tracking approach.
- Since edges are sensitive to camera jittering that can affect the background subtraction process, a method can be proposed as a preprocessing step to cope with that problem.

Bibliography

[1] Sami Abdulla et al. “Recent survey on crowd density estimation and counting for visual surveillance”. In: *Engineering Applications of Artificial Intelligence* 41 (), pp. 103–114.

[2] "ABODA". URL: http://imp.iis.sinica.edu.tw/ABODA/index.html.

[3] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. “Robust fragments-based tracking using the integral histogram”. In: *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 798–805.

[4] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. “Measuring the objectness of image windows”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2189–2202.

[5] Edouard Auvinet et al. “Left-luggage detection using homographies and simple heuristics”. In: *Proc. 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS'06)*. 2006, pp. 51–58.

[6] "AVSS2007". URL: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.

[7] Simon Baker and Iain Matthews. “Lucas-kanade 20 years on: A unifying framework”. In: *International journal of computer vision* 56.3 (2004), pp. 221–255.

[8] Mario Berger et al. "Adaptive load allocation for combining Anomaly Detectors using controlled skips". In: *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE. 2014, pp. 792–796.

[9] Moshe Blank et al. "Actions as space-time shapes". In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 2. IEEE. 2005, pp. 1395–1402.

[10] Aaron F. Bobick and James W. Davis. "The recognition of human movement using temporal templates". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.3 (2001), pp. 257–267.

[11] Kai Briechle and Uwe D Hanebeck. "Template matching using fast normalized cross correlation". In: *Aerospace/Defense Sensing, Simulation, and Controls*. International Society for Optics and Photonics. 2001, pp. 95–102.

[12] Gabriel J Brostow and Roberto Cipolla. "Unsupervised bayesian detection of independent motion in crowds". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 594–601.

[13] Joao Carreira and Cristian Sminchisescu. "Cpmc: Automatic object segmentation using constrained parametric min-cuts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7 (2012), pp. 1312–1328.

[14] "CDNET2014". URL: http://changedetection.net/.

[15] Luka Čehovin, Matej Kristan, and Alesš Leonardis. "An adaptive coupled-layer visual model for robust visual tracking". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1363–1370.

[16] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people

models or tracking". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–7.

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[18] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection for discrete sequences: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 24.5 (2012), pp. 823–839.

[19] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Outlier detection: A survey". In: *ACM Computing Surveys* (2007).

[20] Varun Chandola, Varun Mithal, and Vipin Kumar. "Comparative evaluation of anomaly detection techniques for sequence data". In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE. 2008, pp. 743–748.

[21] Datong Chen, Jie Yang, and Howard D Wactlar. "Towards automatic analysis of social interaction patterns in a nursing home environment from video". In: *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*. ACM. 2004, pp. 283–290.

[22] Ming-Ming Cheng et al. "BING: Binarized normed gradients for objectness estimation at 300fps". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 3286–3293.

[23] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. "Real-time tracking of non-rigid objects using mean shift". In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Vol. 2. IEEE. 2000, pp. 142–149.

[24] Carlos Cuevas, Raquel Martínez, and Narciso García. "Detection of stationary foreground objects: A survey". In: *Computer Vision and Image Understanding* (2016).

[25] Ilias Dahi et al. "Abandoned object detection using blind motion history analysis". In: *Système Conjoint de Compression et d'Indexation Basé-Objet pour la Vidéo, Proceedings of the SCCIBOV Conference*. 2015, pp. 148–151.

[26] Ilias Dahi et al. "An edge-based method for effective abandoned luggage detection in complex surveillance videos". In: *Computer Vision and Image Understanding* 158 (2017), pp. 141–151.

[27] James W Davis and Vinay Sharma. "Background-subtraction using contour-based fusion of thermal and visible imagery". In: *Computer Vision and Image Understanding* 106.2 (2007), pp. 162–182.

[28] Ian Endres and Derek Hoiem. "Category independent object proposals". In: *European Conference on Computer Vision*. Springer. 2010, pp. 575–588.

[29] Ian Endres and Derek Hoiem. "Category-independent object proposals with diverse ranking". In: *IEEE transactions on pattern analysis and machine intelligence* 36.2 (2014), pp. 222–234.

[30] Rubén Heras Evangelio and Thomas Sikora. "Static object detection based on a dual background model and a finite-state machine". In: *EURASIP Journal on Image and Video Processing* 2011.1 (2010), p. 1.

[31] Quanfu Fan, Prasad Gabbur, and Sharath Pankanti. "Relative attributes for large-scale abandoned object detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 2736–2743.

[32] Quanfu Fan and Sharath Pankanti. "Robust foreground and abandonment analysis for large-scale abandoned object detection in complex surveillance videos". In: *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE. 2012, pp. 58–63.

[33] Benoit Georis et al. "A video interpretation platform applied to bank agency monitoring". In: (2004).

[34] Martin Godec, Peter M Roth, and Horst Bischof. "Hough-based tracking of non-rigid objects". In: *Computer Vision and Image Understanding* 117.10 (2013), pp. 1245–1256.

[35] Sebastian Gruenwedel, Peter Van Hese, and Wilfried Philips. "An edge-based approach for robust foreground detection". In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2011, pp. 554–565.

[36] Asaad Hakeem and Mubarak Shah. "Ontology and taxonomy collaborated framework for meeting classification". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 4. IEEE. 2004, pp. 219–222.

[37] Sam Hare et al. "Struck: Structured output tracking with kernels". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2016), pp. 2096–2109.

[38] Waqas Hassan et al. "Illumination invariant stationary object detection". In: *IET Computer Vision* 7.1 (2013), pp. 1–8.

[39] Victoria Hodge and Jim Austin. "A survey of outlier detection methodologies". In: *Artificial intelligence review* 22.2 (2004), pp. 85–126.

[40] Ya-Li Hou and Grantham KH Pang. "People counting and human detection in a challenging situation". In: *IEEE transactions on systems, man, and cybernetics-part a: systems and humans* 41.1 (2011), pp. 24–33.

[41] Antti Juvonen and Tuomo Sipola. "Combining conjunctive rule extraction with diffusion maps for network intrusion detection". In: *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE. 2013, pp. 000411–000416.

[42] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. "Pn learning: Bootstrapping binary classifiers by structural constraints". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 49–56.

[43] Rudolph Emil Kalman et al. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.

[44] Jaechul Kim and Kristen Grauman. "Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2921–2928.

[45] Jaemyun Kim et al. "Edge-segment-based background modeling: Non-parametric online background update". In: *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*. IEEE. 2013, pp. 214–219.

[46] Jaemyun Kim et al. "Unattended object detection based on edge-segment distributions". In: *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE. 2014, pp. 283–288.

[47] Jiman Kim and Daijin Kim. "Accurate static region classification using multiple cues for ARO detection". In: *IEEE signal processing letters* 21.8 (2014), pp. 937–941.

[48] Kyungnam Kim et al. "Background modeling and subtraction by codebook construction". In: *Image Processing, 2004. ICIP'04. 2004 International Conference on*. Vol. 5. IEEE. 2004, pp. 3061–3064.

[49] Robert Koch. "Towards next-generation intrusion detection". In: *Cyber Conflict (ICCC), 2011 3rd International Conference on*. IEEE. 2011, pp. 1–18.

[50] Dan Kong, Douglas Gray, and Hai Tao. "Counting Pedestrians in Crowds Using Viewpoint Invariant Training." In: *BMVC*. 2005, pp. 1–6.

[51] Junseok Kwon and Kyoung Mu Lee. "Tracking by sampling trackers". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1195–1202.

[52] Junseok Kwon and Kyoung Mu Lee. "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 1208–1215.

[53] Andrew HS Lai and Nelson HC Yung. "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence". In: *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*. Vol. 4. IEEE. 1998, pp. 241–244.

[54] Aleksandar Lazarevic, Vipin Kumar, and Jaideep Srivastava. "Intrusion detection: A survey". In: *Managing Cyber Threats*. Springer, 2005, pp. 19–78.

[55] Min Li et al. "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection". In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE. 2008, pp. 1–4.

[56] Huei-Hung Liao, Jing-Ying Chang, and Liang-Gee Chen. "A localized approach to abandoned luggage detection with foreground-mask sampling". In: *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*. IEEE. 2008, pp. 132–139.

[57] Kevin Lin et al. "Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance". In: *IEEE Transactions on Information Forensics and Security* 10.7 (2015), pp. 1359–1370.

[58] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. "Estimation of number of people in crowded scenes using perspective transformation". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 31.6 (2001), pp. 645–654.

[59] Markos Markou and Sameer Singh. "Novelty detection: a review—part 1: statistical approaches". In: *Signal processing* 83.12 (2003), pp. 2481–2497.

[60] Markos Markou and Sameer Singh. "Novelty detection: a review—part 2:: neural network based approaches". In: *Signal processing* 83.12 (2003), pp. 2499–2521.

[61] Xue Mei and Haibin Ling. "Robust visual tracking using 1 minimization". In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 1436–1443.

[62] Hieu T Nguyen and Arnold WM Smeulders. "Robust tracking using foreground-background texture discrimination". In: *International Journal of Computer Vision* 69.3 (2006), pp. 277–293.

[63] Diego Ortego and Juan C SanMiguel. "Multi-feature stationary foreground detection for crowded video-surveillance". In: *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2014, pp. 2403–2407.

[64] Jiyan Pan, Quanfu Fan, and Sharath Pankanti. "Robust abandoned object detection using region-level analysis". In: *2011 18th IEEE International Conference on Image Processing*. IEEE. 2011, pp. 3597–3600.

[65] Nikos Paragios and Visvanathan Ramesh. "A MRF-based approach for real-time subway monitoring". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2001, pp. I–I.

[66] Animesh Patcha and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer networks* 51.12 (2007), pp. 3448–3470.

[67] "PETS2006". URL: http://www.cvg.reading.ac.uk/PETS2006/data.html.

[68] "PETS2007". URL: http://www.cvg.reading.ac.uk/PETS2007/data.html.

[69] Oluwatoyin P Popoola and Kejun Wang. "Video-based abnormal human behavior recognition—A review". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 865–878.

[70] Fatih Porikli, Yuri Ivanov, and Tetsuji Haga. "Robust abandoned object detection using dual foregrounds". In: *EURASIP Journal on Advances in Signal Processing* 2008.1 (2007), pp. 1–11.

[71] Vincent Rabaud and Serge Belongie. "Counting crowded moving objects". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 705–711.

[72] Esa Rahtu, Juho Kannala, and Matthew Blaschko. "Learning a category independent object detection cascade". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1052–1059.

[73] Jesús Martínez-del Rincón et al. "Automatic left luggage detection and tracking using multi-camera ukf". In: *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS'06)*. 2006, pp. 59–66.

[74] David A Ross et al. "Incremental learning for robust visual tracking". In: *International journal of computer vision* 77.1 (2008), pp. 125–141.

[75] David Ryan et al. "Crowd counting using multiple local features". In: *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.* IEEE. 2009, pp. 81–88.

[76] Farzad Sabahi and Ali Movaghar. "Intrusion detection: A survey". In: *Systems and Networks Communications, 2008. ICSNC'08. 3rd International Conference on*. IEEE. 2008, pp. 23–26.

[77] Guang Shu et al. "Part-based multiple-person tracking with partial occlusion handling". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1815–1821.

[78] Angela A Sodemann, Matthew P Ross, and Brett J Borghetti. "A review of anomaly detection in automated surveillance". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 1257–1272.

[79] Xuan Song et al. "An online approach: Learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 739–746.

[80] 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.* Vol. 2. IEEE. 1999.

[81] Grzegorz Szwoch. "Extraction of stable foreground image regions for unattended luggage detection". In: *Multimedia Tools and Applications* 75.2 (2016), pp. 761–786.

[82] Demetri Terzopoulos and Richard Szeliski. "Tracking with Kalman snakes". In: *Active vision*. MIT press. 1993, pp. 3–20.

[83] YingLi Tian et al. "Robust detection of abandoned and removed objects in complex surveillance videos". In: *IEEE Transactions on Systems, Man,*

and Cybernetics, Part C (Applications and Reviews) 41.5 (2011), pp. 565–576.

[84] Carlo Tomasi and Takeo Kanade. "Detection and tracking of point features". In: (1991).

[85] Md Zia Uddin, Tae-Seong Kim, and Jeong-Tai Kim. "A spatiotemporal robust approach for human activity recognition". In: *International Journal of Advanced Robotic Systems* 10.11 (2013), p. 391.

[86] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[87] Wahyono Wahyono, Alexander Filonenko, and Kang-Hyun Jo. "Unattended Object Identification for Intelligent Surveillance System Using Sequence of Dual Background Difference". In: *IEEE Transactions on Industrial Informatics* ().

[88] Rui Wang et al. "Static and moving object detection using flux tensor with split gaussian models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 414–418.

[89] Yang Wang et al. "Unsupervised discovery of action classes". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2006, pp. 1654–1661.

[90] Andrew R Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.

[91] Daniel Weinland, Remi Ronfard, and Edmond Boyer. "A survey of vision-based methods for action representation, segmentation and recognition". In: *Computer vision and image understanding* 115.2 (2011), pp. 224–241.

[92] Jian Yao and Jean-Marc Odobez. "Multi-layer background subtraction based on color and texture". In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, pp. 1–8.

[93] Xue Zhou et al. "Markov random field modeled level sets method for object tracking with moving cameras". In: *Computer Vision–ACCV 2007* (2007), pp. 832–842.

[94] C Lawrence Zitnick and Piotr Dollár. "Edge boxes: Locating object proposals from edges". In: *European Conference on Computer Vision*. Springer. 2014, pp. 391–405.

[95] Zoran Zivkovic. "Improved adaptive Gaussian mixture model for background subtraction". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 2. IEEE. 2004, pp. 28–31.