

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



**CONTRIBUCIONES A LA DETECCIÓN
DE OBJETOS ROBADOS Y
ABANDONADOS EN SECUENCIAS DE
VÍDEO-SEGURIDAD**

-PROYECTO FIN DE CARRERA-

Luis Alberto Caro Campos
Julio 2011

CONTRIBUCIONES A LA DETECCIÓN DE OBJETOS ROBADOS Y ABANDONADOS EN SECUENCIAS DE VÍDEO-SEGURIDAD

Autor: Luis Alberto Caro Campos

Tutor: Juan Carlos San Miguel Avedillo

Ponente: José María Martínez Sánchez

email: {Luis.Caro, Juancarlos.Sanmiguel, JoseM.Martinez}@uam.es



Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2011

Abstract

In this work, we present an exhaustive analysis of a video-surveillance system aimed at the detection of abandoned and stolen objects. A formalization of the problem is presented, followed by a description of the different stages of analysis required for the detection, focusing our attention on the discrimination of stationary regions between abandoned and stolen objects. For this stage, we propose three new discrimination methods based on active contours adjustments, and a discriminator based on extracting color contrast information. Additionally, fusion schemes are studied to combine information from multiple discriminators. These novel approaches are then evaluated and compared against state-of-the-art approaches on an heterogenous dataset.

Resumen

En este Proyecto, se presenta un análisis exhaustivo de un sistema de videovigilancia cuyo objetivo es la detección de objetos robados y abandonados. Se presenta una formalización del problema, seguida de una descripción de las distintas etapas de análisis requeridas para detección; centrando nuestra atención en la discriminación de regiones estáticas entre objetos abandonados y robados. Para esta etapa, proponemos tres nuevos métodos de discriminación basados en ajustes de contornos activos, y un discriminador basado en la extracción de información el contraste de color. Adicionalmente, se estudian distintos esquemas de fusión para combinar la información proveniente de múltiples clasificadores. Posteriormente, los métodos propuestos son evaluados y comparados con discriminadores existentes, sobre un conjunto de datos heterogéneo.

Keywords

Video analysis, video-surveillance, abandoned and stolen objet detection, unattended luggage, stationary regions, active contours

Agradecimientos

The reward of our work is not what we get, but what we become.

Paulo Coelho, O Aleph

En primer lugar, quiero agradecer a mi tutor, Juan Carlos, por su inestimable ayuda en la realización de este proyecto; por su paciencia y sus consejos.

A mis profesores, Jesús Bescós y José María Martínez, por haberme brindado la oportunidad de realizar este proyecto en VPU-Lab, así como por su permanente orientación a lo largo de toda la carrera.

A mi familia, por su apoyo incondicional y su cariño. A mi padre, por estar siempre dispuesto a ayudarme; a Soledad, a quien debo tantísimas cosas; y a mi hermanos, que siempre llevo presentes allá donde esté.

A mis compañeros de VPU-Lab, en especial a Laura, Loreto, Alfonso e Isabel, por haber hecho más amenas aquellas tardes de trabajo.

A todas aquellas personas con las que he tenido el placer y el honor de compartir estos últimos años en la EPS. Me temo que me faltaría espacio para nombraros a todos.

A Irene, por los buenos recuerdos que guardo de mi etapa en la ORI, por sus consejos y su apoyo.

A mi prima María; y a mis amigos, en especial a Alberto, Marta, Santiago y Cristina, por sus ánimos, su apoyo, por aquellas excursiones a bibliotecas... Gracias por estar ahí.

Por último, me gustaría dedicar este trabajo a la memoria de mi abuela María Elena y a la de mi madre, Alejandra.

Luis Caro Campos

Julio 2011

Contents

Abstract	v
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Document Structure	4
2 Related work	7
2.1 Introduction	7
2.2 Problem statement	8
2.3 Detection of objects of interest	9
2.3.1 Foreground segmentation	10
2.3.2 Stationary object detection	13
2.3.3 Object classification	14
2.4 Discrimination of abandoned and stolen objects	16
2.4.1 Edge-based approaches	17
2.4.2 Color-based approaches	18
2.4.3 Hybrid approaches	18
2.5 Existing datasets	19

3	Base system	23
3.1	Introduction	23
3.2	Foreground segmentation module	24
3.2.1	Background subtraction	24
3.2.2	Shadow removal	25
3.2.3	Noise removal	27
3.2.4	Blob Extraction	27
3.3	Blob Tracking	28
3.4	Stationary object detection	29
3.4.1	Stationary blob detection	29
3.4.2	Object classification	29
3.5	Abandoned and stolen object discrimination	30
3.5.1	Features	31
3.5.2	Evidence model	35
3.5.3	Hybrid abandoned and stolen object discrimination	35
4	Single-feature discrimination for abandoned and stolen objects	37
4.1	Introduction	37
4.2	Based on active contours	38
4.2.1	Overview of the discrimination scheme	38
4.2.2	Parametric, edge-based active contours	41
4.2.3	Geometric active contours	43
4.3	Based on pixel color contrast	48
4.3.1	Overview of the discrimination scheme	48
4.3.2	Boundary Spatial Color Contrast	48
5	Multi-feature discrimination for abandoned and stolen objects	51
5.1	Introduction	51
5.2	Motivation for multi-feature fusion	52

5.2.1	Observed limitations of single feature discrimination	53
5.2.2	Advantages of multi-feature fusion	54
5.3	Structure of the multi-feature discriminator	55
5.4	Selected combination techniques	56
5.4.1	Naïve Bayes	57
5.4.2	Support Vector Machines	59
5.4.3	K-Nearest Neighbor	62
6	Experimental validation	65
6.1	Introduction	65
6.2	Setup	66
6.2.1	Dataset	66
6.2.2	Performance metrics	69
6.2.3	Implementation	71
6.2.4	Parameter selection	71
6.3	Single-feature evaluation	73
6.3.1	Annotated data	73
6.3.2	Real data	78
6.3.3	Computational cost comparison	82
6.4	Multi-feature evaluation	82
6.4.1	Feature selection	83
6.4.2	Annotated data	84
6.4.3	Real data	84
7	Conclusions and future work	85
7.1	Summary of work	85
7.2	Conclusions	86
7.3	Future Work	87
	Bibliography	89

A Support vector machines	97
B Automatic generation of foreground masks from video annotations	103
B.1 Introduction	103
B.2 ViPER-GT	103
B.3 Foreground mask extraction	104
C Publications	107
D Introducción	115
D.1 Motivación	115
D.2 Objetivos	117
D.3 Organización de la memoria	118
E Conclusiones y trabajo futuro	121
E.1 Resumen del trabajo realizado	121
E.2 Conclusiones	123
E.3 Trabajo futuro	124
F Presupuesto	127
G Pliego de condiciones	129

List of Figures

1.1	Examples of abandoned (left) and stolen (right) objects.	2
2.1	Example of abandoned (first row) and stolen (second row) object event definitions.	9
2.2	Common stages of analysis for the detection of objects of interest.	10
2.3	Foreground segmentation example	11
2.4	Example of the stationary object detection method proposed in [1], figure taken from [2]	15
2.5	Examples of features of object classification [3]	16
2.6	Stolen/abandoned discrimination analysis stage	17
2.7	Examples of extracted edges for abandoned and stolen objects	18
2.8	Frame samples from selected datasets	21
3.1	Block diagram of the existing video analysis system for abandoned and stolen object detection.	24
3.2	Block diagram of the <i>Foreground Segmentation Module</i>	25
3.3	Foreground mask at different stages: initial foreground mask (a), after shadow removal (b) and after noise removal (c)	26
3.4	Object classification examples	30
3.5	Stolen/abandoned object discriminator	31
3.6	First row: Bounding-box in foreground mask (a), current frame (b) and background (c). Second row: Color histograms: R1 in background frame (H1), R2 in current frame (H2), R2 in background frame (H3).	32
3.7	Gradient detectors example for an abandoned object	35
3.8	Classification by fusing evidence data	36
4.1	Scheme for abandoned and stolen object detection by active contours adjustment.	39

4.2	Examples of contour adjustments for abandoned objects (left) and stolen objects (right).	40
4.3	Contour adjustment comparison using the <i>Sobel</i> gradient operator (rows 2&3), and <i>Canny</i> edge detector (rows 4&5). Both operators combine information from all color channels.	44
4.4	Contour adjustments for an abandoned object, performed on the current frame (row 1) and the background image (row 2). (a) Initial contour and adjustments using approaches (b) PE, (c) GR and (d) GE.	46
4.5	Scheme for abandoned and stolen object detection using color contrast features.	49
4.6	Pixel color contrast detector: (a) static foreground object, (b) analyzed points along the boundary and (c) analyzed contour point.	50
5.1	Levels of information fusion in a visual system [4]	52
5.2	Fusion levels in the stolen/abandoned discrimination problem	53
5.3	Scheme of the proposed multi-feature discriminator	56
5.4	Scheme of Naïve Bayes classifier using normal distributions	57
5.5	Maximum margin hyperplane and support vectors	60
5.6	Non-linear transformation example	61
5.7	KNN classification example	63
6.1	Examples of video sequences with abandoned (1) and stolen (2) objects, from all three categories (C1, C2, C3).	66
6.2	Examples of extracted frames and foreground masks.	69
6.3	ROC analysis for single-feature discrimination on annotated data.	74
6.4	Examples of problematic scenarios for the color histogram discriminator of the base system (CHIST).	75
6.5	Scores of the single-feature discriminators of the base system for the annotated data	76
6.6	Examples of problematic scenarios for the gradient-based discriminators of the base system (GH, GL and GRD).	77
6.7	Scores of the proposed single-feature discriminators for the annotated data	77
6.8	Example of contour problematic adjustments for a stolen object using the GE discriminator.	78
6.9	ROC analysis for single-feature discrimination on real data.	79
6.10	Scores of the single-feature discriminators of the base system for the real data	80
6.11	Scores of the proposed single-feature discriminators for the real data	81

A.1	Possible decision boundaries for linearly separable data in the 2-dimensional space	98
A.2	Maximum margin hyperplane and support vectors	99
A.3	Non-linear transformation example	100
B.1	Foreground masks extraction process	106

List of Tables

6.1	Dataset description.	68
6.2	Layout of the confusion matrix.	70
6.3	Single-feature discrimination results for annotated data. (Key. ACC:accuracy, AUC:Area Under Curve).	74
6.4	Single-feature discrimination results for real data. (Key. ACC:accuracy, AUC:Area Under Curve).	79
6.5	Comparative computational cost	83
6.6	Correlation coefficients between discriminators scores	84
6.7	Multifeature-feature discrimination results. (Key. ACC:accuracy).	84
B.1	Descriptor attributes	104

Chapter 1

Introduction

1.1 Motivation

Nowadays, the demand for automatic video-surveillance systems is growing as a consequence of increasing global security concerns [5]. Traditionally, the monitoring task is performed by human operators who have to simultaneously analyze information from different cameras. A reduction of efficiency is expected as operators have to process large amounts of visual information generated by these cameras. For this reason, real-time automatic video interpretation is emerging as a solution to aid operators in focusing their attention on specific security-related events.

In this situation, the detection of abandoned and stolen objects has become one of the most promising research topics especially in crowded environments such as train stations and shopping malls. For example, a useful application of abandoned object detection could be to detect unattended packages in a subway station. For stolen object detection, an interesting application could be the monitoring of specific items in an office, showroom or museum. This detection aims to provide a continuous supervision of the information captured by the camera so that the appropriate actions can be taken. Figure 1.1 shows some examples of these application scenarios.

In general, the detection of abandoned and stolen objects is achieved by developing a system that comprises the following analysis stages: foreground segmentation, stationary region de-



Figure 1.1: Examples of abandoned (left) and stolen (right) objects.

tection, blob classification and stolen/abandoned discrimination. Firstly, moving objects (foreground) are differentiated from the background of the scene in the foreground segmentation stage. Then, stationary regions are detected by analyzing foreground objects over time. Each detected region is then classified by type (person, group of people, luggage, ...). For those regions classified as stationary objects, additional analysis is performed in order to determine whether the object has been abandoned or stolen from the scene.

Each stage in the system has different challenges that affect their performance. Changes in lighting conditions and non-stationary backgrounds may result in incorrect foreground segmentation, hindering the detection of objects of interest. Moreover, in crowded environments where occlusions are more frequent, static regions may not correctly be detected, and object tracking becomes more complex as it has to cope with an increased number of objects and interactions. Blob classification may be affected by the variability of object appearance in the video sequences. Furthermore, since potential abandoned or stolen objects may have arbitrary shape and color, specific object recognition methods can not be applied. Finally, low-complexity algorithms have to be employed if real-time alarms are required.

Many methods have been proposed for abandoned and stolen object detection. Examples include approaches that focus on the stabilization of the image sequence from a moving camera [6], based on the static foreground region detection [7], based on blob classification (e.g., people vs. objects) [3] or discriminating static regions between abandoned and stolen [8]. These approaches yield acceptable results in simple scenarios where objects of interest can accurately be detected.

However, this is not always valid for complex situations in which a performance decrease is expected. In particular, the discrimination of stationary regions between abandoned and stolen objects has not been fully explored under different characteristics of the video sequence.

1.2 Objectives

The main objective of this project is the study of the last stage of analysis of a video-surveillance system that is capable of detecting abandoned and stolen objects in video sequences. This stage is in charge of determining whether stationary objects correspond to abandoned or stolen objects. The goal of this project is to improve an existing system currently in development at Video Processing and Understanding Lab at Universidad Autónoma de Madrid (VPU-Lab).

The above-mentioned goal can be further specified in the following general objectives:

1) Study of the state of the art

For each of the aforementioned stages of analysis, the related literature is reviewed. Special emphasis is given to the study of the different techniques for the discrimination between abandoned and stolen objects.

2) Study of the existing abandoned and stolen object detection system in VPU-Lab

A comprehensive study of the existing video analysis system provided by VPU-Lab is performed, with the aim of identifying challenges in the detection of objects of interest and the discrimination between abandoned and stolen. An evaluation of the existing discrimination approaches is carried out.

3) Design and implementation of new single-feature approaches for abandoned and stolen discrimination

Novel approaches based on single features are developed to discriminate stationary regions between abandoned and stolen objects, with the aim of providing additional robustness in those cases in which existing approaches have shown weaknesses.

4) Design and implementation of new multi-feature approaches for abandoned and stolen

discrimination

Classical fusion schemes are studied and evaluated with the aim of combining information from the available approaches in the VPU-Lab system and the proposed ones for the discrimination task.

5) Generation of an evaluation framework

This framework consists on an evaluation using both manually annotated and real data. Manual data is generated by annotating video sequences from publicly available data sets. For real data, an automatic process is applied to the video-surveillance system of the VPU-Lab for generating such data from video meta-data files. Furthermore, data is grouped into different categories with varying degrees of complexity.

6) Comparative evaluation of improvement achieved with the proposed approaches

The performance of the proposed discriminators and the fusion schemes is compared against the existing approaches provided by VPU-Lab to identify their advantages and drawbacks.

1.3 Document Structure

The document is structured as follows:

- Chapter 1. This chapter presents the motivation, objectives and structure of this document.
- Chapter 2. In this chapter, the problem is presented by first identifying the information that needs to be extracted from the video sequence, followed by an overview of the most relevant approaches for each processing stage. Then, this chapter focuses on the discrimination between abandoned and stolen objects.
- Chapter 3. This chapter describes the base-system for abandoned and stolen object detection provided by the VPU-Lab. The approaches employed in the abandoned and stolen discrimination stage are described in detail.

- Chapter 4. In this chapter, we describe the single-feature proposed approaches for the abandoned and stolen discrimination task.
- Chapter 5. In this chapter, different combination schemes are described and employed for the fusion of the available approaches for discrimination.
- Chapter 6. This chapter presents the dataset used for training and testing, the evaluation metrics and the experimental results. Furthermore, a comparison against other state-of-the-art approaches is performed.
- Chapter 7. This chapter summarizes the main achievements of the work, discusses the obtained results and provides suggestions for future work.
- Appendix
 - A. Introduction to Support Vector Machines.
 - B. Extraction of real foreground masks from annotations of abandoned and stolen objects.
 - C. Publications produced within this project.

Chapter 2

Related work

2.1 Introduction

The detection of abandoned and stolen objects in video surveillance comprises different stages of analysis, ranging from low-level stages (e.g., segmentation at the pixel level) to high-level processing (e.g., event recognition by analyzing stationary objects). Prior to this detection, these two events have to be clearly defined in order to determine what kind of information needs to be extracted from the video sequence. In this chapter, we study how this problem has been formalized. Then, we overview the different stages of analysis for the detection of objects of interest: foreground segmentation, stationary region detection and object classification. This study allows to understand the limitations of existing video-surveillance systems for abandoned and stolen object detection. Finally, we concentrate this literature review on the existing approaches and available datasets for the discrimination between abandoned and stolen objects.

The rest of this chapter is organized as follows. In section 2.2, the formalization of the abandoned and stolen object detection problem is presented. In section 2.3, the different stages for stationary object detection are overviewed: foreground segmentation (subsection 2.3.1), stationary region detection (subsection 2.3.1) and object classification (subsections 2.3.3). Finally, section 2.4 presents a comprehensive study of the different approaches for the discrimination task and section 2.5 describes the data sets employed in the work presented in this document.

2.2 Problem statement

In an automated video analysis system in which the goal is the detection of abandoned and stolen objects, different considerations have to be made regarding the definition of the two events. We can distinguish between three types of information that allow us to determine whether any of these two events have occurred in a video sequence. They are:

- **Contextual information:** regards the nature of the object of interest (people, luggage, ...) as well as its interaction with the environment and other objects in the scene.
- **Spatial information:** describes the location of the object of interest and its distance to other objects or people. **Temporal information:** considers the motion parameters (such as trajectory and speed) that allow to determine if an object has remained stationary for a certain amount of time.

By combining these three sources of information, we can then define specific rules for the abandoned and stolen object events. In PETS 2006 ¹, these events were specified as follows:

- **Abandoned object:** an object belongs to a person that enters the scene until they become separated (contextual rule), and it is considered abandoned when it remains at a certain distance from its owner (spatial rule) for a certain period (temporal rule). Figure 2.1 (first row) shows an example of this definition.
- **Stolen object:** all objects belonging to the background of the scene are susceptible of being removed (contextual rule). An object is considered stolen if it is away from its location (spatial rule) for a certain amount of time (temporal rule), or when the person that takes the object is no longer in the scene (contextual rule). Figure 2.1 (second row) shows an example of this definition.

The main implication of these definitions is that in order to perform abandoned and stolen object detection, a video analysis system has to be able to detect stationary foreground regions

¹Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (<http://www.cvg.rdg.ac.uk/PETS2006/index.html>)

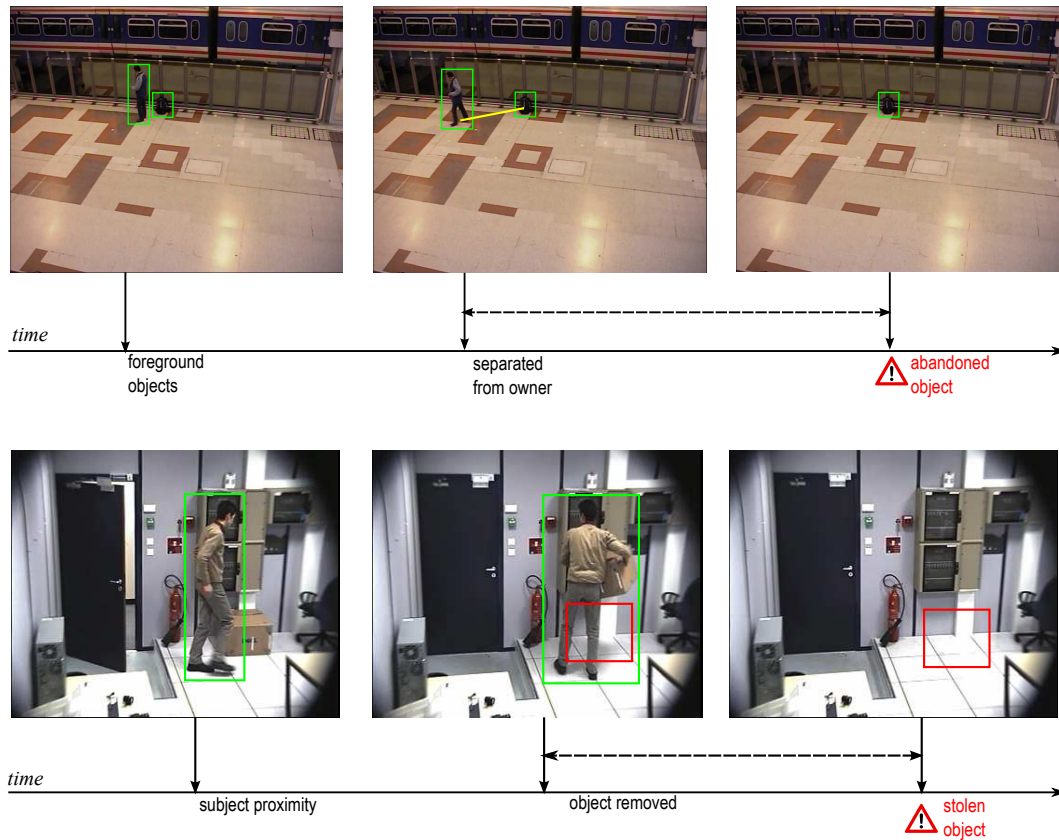


Figure 2.1: Example of abandoned (first row) and stolen (second row) object event definitions.

in the scene and classify them according to the possible object categories that may appear in the video sequence (e.g., luggage and people). Once candidate objects are detected, features are extracted to discriminate between the abandoned and stolen events.

2.3 Detection of objects of interest

As previously stated, only stationary objects in the scene are considered for the abandoned and stolen event discrimination. These objects of interest share common characteristics:

- They belong to the foreground of the scene (*foreground segmentation*)
- They remain stationary for a certain period of time (*stationary object detection*)
- They are generic objects (*object classification*: people, group of people, luggage...)

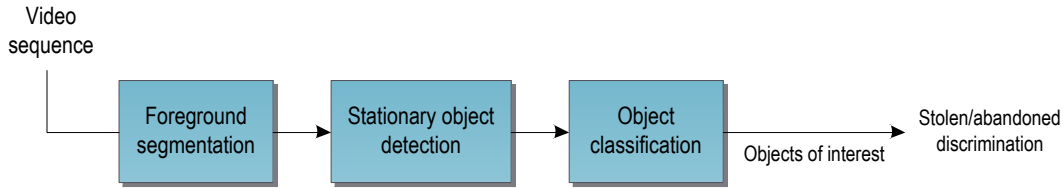


Figure 2.2: Common stages of analysis for the detection of objects of interest.

In a typical visual analysis system, these characteristics are independently analyzed by different processing modules, as depicted in figure 2.2. In the following subsections, we survey the most relevant approaches found in the literature for each stage of analysis.

2.3.1 Foreground segmentation

The first step in many computer vision applications involves the localization of objects of interest within a scene by distinguishing between the pixels that belong to the foreground (stationary or moving objects) from those pixels that compose the background of the scene. The most widely employed approach for this task is background subtraction (BGS), where each incoming video frame is compared against a model of the scene’s background. This model must be an accurate representation of the background scene in the absence of foreground objects, and should be able to take into account changes of the environment, such as illumination changes. The main differences between most BGS methods found in the literature lie on how the background model is obtained and updated, and the distance measure employed to compare incoming frames against the model.

BGS techniques have been classified in the literature according to different criteria. In [9], a distinction is made between recursive and non-recursive methods. Recursive methods maintain a single model of the background that is updated with each new frame, whereas non-recursive methods estimate the background from a buffer of the N previous frames in the sequence. A different classification is proposed in [10], distinguishing between predictive and non-predictive. Predictive approaches model the input as a time series and recover the current input based on past observations, while non-predictive methods estimate the background by building a probabilistic model neglecting the order of the input observations. More recently, the

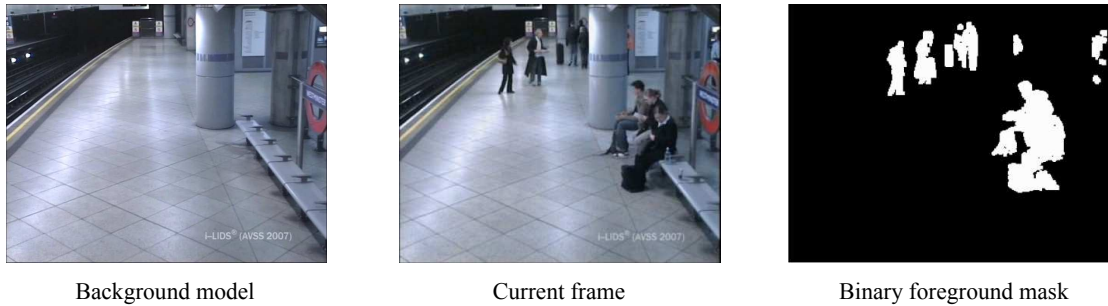


Figure 2.3: Foreground segmentation example

authors of [11] have proposed a more comprehensive classification, in which the spatial scope of the background model is taken into account at pixel, region, or frame level.

The most widely used approaches found in the literature work at the pixel level. Some simple approaches model the background as the average of past frames; or by computing the median at each pixel location in a buffer of past N previous frames. Many authors model each pixel with a probability density function. In [12], background pixels are modeled with a Gaussian distribution. Pixels with low probabilities are more likely to belong to moving foreground objects. Mean and variance for each pixel are typically updated in a running average fashion. More robust approaches model pixels with multimodal distributions. Currently, the Gaussian Mixture Model (GMM) proposed by [13] is widely employed because of its ability to handle background variations, such as gradual lighting changes and repetitive motion (e.g. swaying trees). In [14], a non-parametric method is proposed where pixels are modeled with using Kernel Density Estimators constructed from the past N frames. A popular method that maintains a single model for the entire frame rather than for each pixel is Eigenbackgrounds [15], based on eigenvalue decomposition. Principal component analysis is performed on a set of training frames. The best principal components are selected to build the eigenspace. Incoming frames are then back-projected from the eigenspace, which serves a model for the background for that particular frame.

Two important applications of background modeling are background initialization and background maintenance [Wallflower]. In most cases, the initialization is assumed to be performed when the scene is clear of foreground objects. This requirement, however, is often impossible

to satisfy. For analysis over long periods of time, BGS methods must be able to cope with changes in the environment in order to avoid incorrect segmentation. Different key issues have been identified in the literature [16, 11], that affect the performance of BGS and therefore, they should be taken into account:

Changes in illumination: They can alter the appearance of the background significantly. If the update scheme fails to reflect these changes in the background model, it will most likely result in background pixels being incorrectly labeled as foreground.

Moving objects in the background: When an object that belongs to the background is removed, the position occupied by it will be classified as foreground, as well as the object itself as it moves in the scene. The model will have to be adapted to this background change. However, if this is done too quickly, the removed object may not be correctly identified as a stationary region in a posterior analysis. Therefore, there exists a trade-off between the rate at which the background is updated, and the ability to detect objects of interest (and events).

Stationary objects in the foreground: This is similar to the previous case, as motionless objects and people may be incorporated into the model as it is updated. It is often difficult to avoid stationary people from being incorporated into the model, and this is especially problematic when there are stationary people in the initial background. In addition, if the presence of stationary objects is reflected too quickly in the background model, it will be more difficult to identify an abandoned object.

Multimodal backgrounds: In some cases, backgrounds are not completely stationary and have specific types of motion that should not be classified as foreground, such as swaying trees and waves in water's surface. Robust BGS methods have to take these situations into account, and correctly classify those instances as background.

Shadows and reflections: While these two situations are different in nature, they both may produce problems in posterior stages if they are identified as part of the foreground moving objects. For example, shadows adjacent to moving objects can interfere with object

classifiers or detectors that rely on different geometrical features. Most approaches in the literature deal with shadow removal by performing post-processing operations on the foreground masks obtained in the segmentation stage.

Incorrect foreground segmentation has a negative impact in the posterior analysis leading to the detection of abandoned and stolen objects. Illumination changes or reflections often result in false positives, which subsequently results in background regions being detected as either abandoned or stolen objects. If cast shadows are detected as part of stationary objects, a discriminator that relies on color features may give a wrong detection, as the detected area will share color features with the background. If foreground objects remain stationary, they eventually become part of the background model. Therefore, the update rate of the model should be taken into account when deciding the minimum time a region has to be stationary to be detected as abandoned or stolen. As a conclusion, the detection of abandoned and stolen objects is heavily related with the foreground segmentation stage. Thus, the update rate of the background model, the analysis of multimodal backgrounds and the post-processing operations should be taken into account for developing effective abandoned and stolen object detection systems.

2.3.2 Stationary object detection

After the foreground segmentation process, the following step is to determine which objects in the scene have remained stationary. Most approaches found in the literature rely on *object tracking* to perform the stationary object detection using the previously computed foreground maps (known as blob-based tracking). In this case, tracking consists on establishing a correspondence between blobs² in consecutive frames. By accurately determining the position of the same object in the image sequence, motion parameters such as speed and trajectory are obtained. This information can then be used to determine which foreground regions have not moved, for example, by analyzing their speed. Object tracking approaches, however, present many challenges in crowded environments and therefore, their expected accuracy is low. In these

²In this document, we consider a blob as a connected region extracted from a binary mask that represents the foreground pixels of the scene.

scenarios, high populated sequences make difficult to extract isolated blobs from the foreground mask. Additionally, the computational cost increases as the system has to keep track of a high number of objects. For these reasons, tracking-based approaches are generally not suitable for its application in crowded environments.

According to a recent survey [2], non-tracking based stationary object detection approaches can be classified depending on the type of analysis performed: based on the accumulation of foreground masks; and based on the properties of the background subtraction model. In [17], an approach based on the accumulation of foreground masks is proposed. In this algorithm, a confidence map that indicates the presence of stationary foreground objects is computed. For each pixel, a counter is maintained. The counter is updated with every new foreground mask, increasing its value if the pixel is highlighted as foreground, and decreased if it belongs to the background. When the counter hits a predefined threshold, the pixel is considered to belong to a stationary object. In [1], an approach based on sub-sampling the sequence of foreground masks is proposed. In this approach, the sequence of foreground masks for the past 30 seconds is sub-sampled; taking advantage of the fact that, at lower frame rates, stationary objects are more likely to be found in the same position. Once the samples have been taken, a simple binary AND operation will produce a binary mask that depicts stationary objects. A method based on the properties of the BGS model is proposed in [18]. In this method, GMM is employed to model the background. This approach performs the detection by observing the transition states between the different Gaussian modes for each pixel. When a new object enters the scene, new modes are created for its pixels. As the object remains stationary, these newly created modes are assigned a higher weight. The detection of stationary objects is performed by combining this information with a set of spatio-temporal rules.

2.3.3 Object classification

The goal of this stage is to classify the detected stationary regions between objects and people. Hence, this stage is considered as a two-class discrimination problem (people vs. objects), as regions not detected as people can simply be classified as generic objects. This stage is

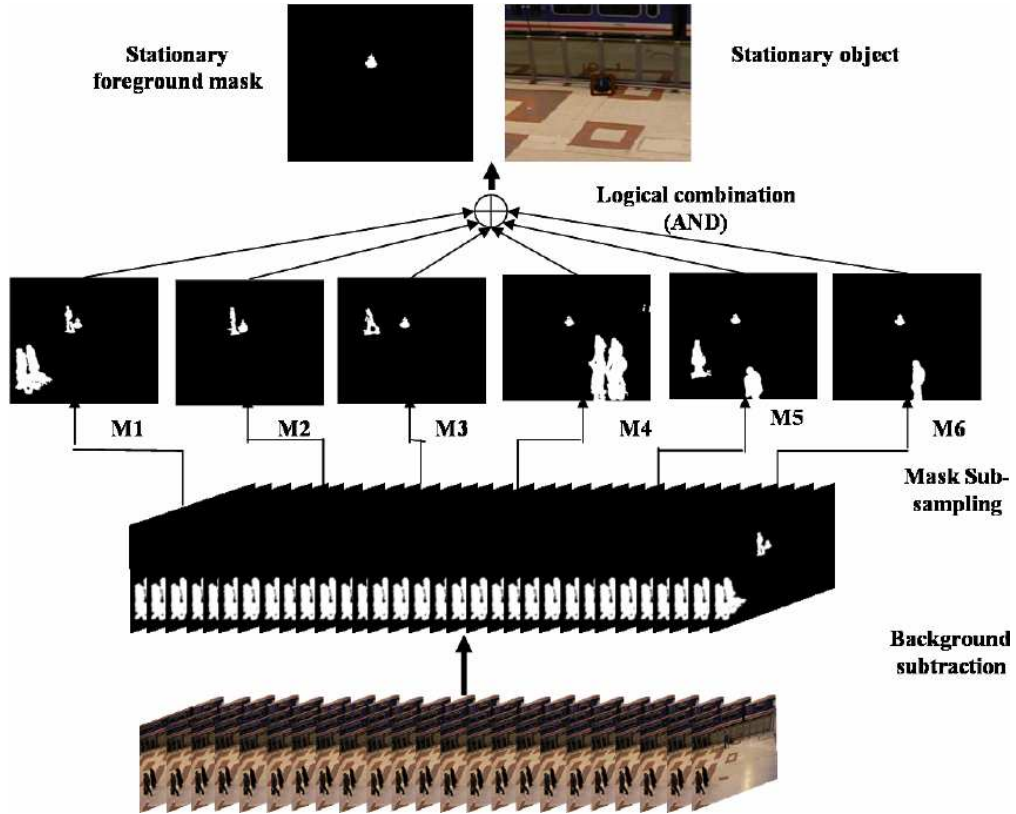


Figure 2.4: Example of the stationary object detection method proposed in [1], figure taken from [2]

important, as it allows us to exclude stationary people from further analysis; while at the same time obtaining information from the performers of the actions.

People detection presents several challenges. The large variability in appearance makes difficult to accurately characterize the entire class by using a single feature. In addition, the variety of poses and articulations that a person may adopt results in complex silhouettes that are difficult to model. Occlusions and interactions between people should also be taken into account, as a single blob may contain more than one person or nearby objects.

According to [19], two broad types of people detection approaches are predominant, depending on whether they rely on contours (silhouettes) or regions. In most cases, a model of people features is first trained prior to the classification. In the literature, a variety of human modeling schemes can be found: silhouettes [20], articulations [21], volumetric models [22]. In some cases, detecting body parts [23], as opposed to full body models, may be enough to perform the


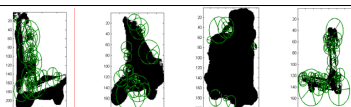

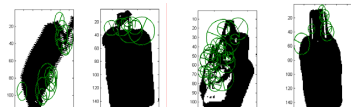

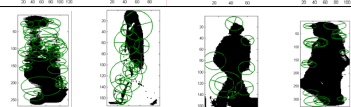

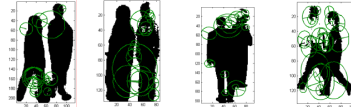
Classes	Statistics of geometric primitives features	SIFT features
Trolley		
Bag		
Person		
Group of People		

Figure 2.5: Examples of features of object classification [3]

classification. For example, in [24], skin and face detection are used to detect people.

Furthermore, specific object classifiers are also used to perform an accurate categorization of the stationary regions. For example, [3] proposed a four-class classification problem: trolley, bag, person and group of people. For the classification, primitive geometric features are employed, such as corners, lines and circles. Additionally, other features such as area, compactness (ratio between the area of the bounding box and the area of the region), aspect ratio and SIFT features are considered. Features are modeled statistically and machine learning classifiers are employed for solving the classification problem. Figure 2.5 shows an example of the proposed features. However, one of the main limitations of these approaches is the need of prior information about the appearance of the scene objects as it is not be available in real world conditions.

2.4 Discrimination of abandoned and stolen objects

In this section, we provide a comprehensive survey of different discrimination approaches found in the literature to distinguish between abandoned and stolen objects. As explained in the preceding sections, the objects of interest are those foreground regions that have been classified as both stationary and non-people by the previous stages of analysis.

Some approaches in the literature simplify the problem by assuming that only object in-

sertions are allowed in the scene [6, 25, 26]. While this may be a valid assumption in certain constrained scenarios, such as the detection of unattended luggage in airports, it does not take into account possible foreground artifacts generated by the background subtraction technique such as ghosts produced by the motion of stationary parts of the scenario (or moving objects that have remained stationary for a long period of time). Therefore, these approaches are expected to fail in the real world conditions.

Few techniques have been proposed that deal with the discrimination problem. Among these, we can classify them according to the features employed between *edge-based*, *color-based*, and *hybrid*.

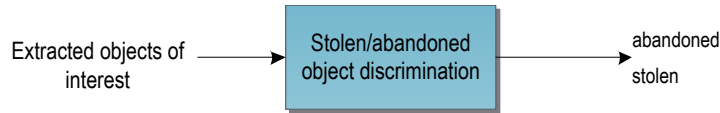


Figure 2.6: Stolen/abandoned discrimination analysis stage

2.4.1 Edge-based approaches

Edge-based approaches rely on inspecting the edge energies around the stationary region boundaries. This energy is assumed to be high when an object has been added to the scene, and low when an object has been removed. For example, in [27], the change in edge energy is analyzed. For abandoned objects, a higher average edge energy is expected, suggesting that the object has been inserted. Conversely, this energy is expected to be lower when the object has been removed from the scene (stolen objects). Similar approaches are described in [28, 29]. They propose the use of the canny edge detector inside the bounding box of the detected stationary object in both the background and the current frame. If edge presence is stronger in the current frame, the object is classified as abandoned, otherwise, the object is classified as stolen. In [30], the SUSAN edge operator is applied on the current image and the foreground binary mask, and are then compared by a proposed matching technique. Figure 2.7 shows an example of edge-based approaches: in the current frame, strong edges indicate that an object has been abandoned. Weak edges in the current frame indicate that the background has been uncovered due to a

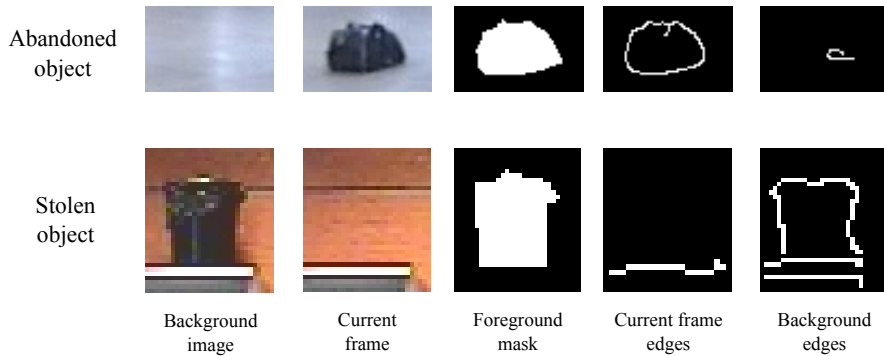


Figure 2.7: Examples of extracted edges for abandoned and stolen objects

stolen object. However, these assumptions only hold true for simple backgrounds with weak edges.

2.4.2 Color-based approaches

Color-based approaches employ the color information in the regions delimited by the boundaries of the stationary region and its bounding box. These approaches exploit the assumption that the color features of the object are different enough from those of its surrounding. When the object is removed, the portion of the background uncovered is expected to have similar color properties than its surroundings. In [31], two Bhattacharya distances are computed between the color histograms of the internal (in the current frame) and the external (in the background frame) regions. The difference between these distances is then thresholded to perform the classification. In a similar fashion, a color-richness measure is proposed in [32] to count the number of colors (histogram bins above a threshold), and then the same comparison as in [31] is performed.

Moreover, [33] proposed the use of image inpainting to reconstruct the hidden background and compare it against the external region using color histograms. In [34], the color information within and outside the stationary region is compared using segmentation techniques.

2.4.3 Hybrid approaches

Hybrid approaches combine information from both edge and color. In [8], using two detectors (color and edge-based), a probabilistic model is built for each algorithm in each class (abandoned

and stolen). The discrimination is performed by computing the average probability of belonging to each class; the object is classified as the class for which it obtained the highest average probability. Another approach proposed in [35], combined the information of different features related to edge energy, color contrast and shape to build a generative model to perform the classification.

In conclusion, the different techniques found in the recent literature use either edge or color information to perform the abandoned/stolen discrimination. Although these approaches work well for simple scenarios, they have difficulties in complex scenarios as they do not consider the possibility of occlusions or complex backgrounds (e.g., high textured backgrounds). In addition, these approaches rely on the precision of foreground object detection, and they may perform poorly in complex scenarios.

2.5 Existing datasets

Several public datasets are available for abandoned and stolen object detection in video. Additionally, they are widely used in the field of video surveillance to assess the performance of different processing modules (e.g., foreground segmentation, tracking). Figure 2.8 shows sample frames from the most representative datasets. They are:

- PETS 2006

URL: <http://www.cvg.rdg.ac.uk/PETS2006/data.html>

This dataset consists on different examples of left-luggage events, with increasing scene complexity in terms of nearby people. A total of 6 left-luggage events in a railway station are recorded by four cameras positioned at different angles (28 videos in total). Videos from this data set are between 1 and 2 minutes long, with standard PAL resolution (768x576 pixels, 25fps).

- PETS 2007

URL: <http://www.cvg.rdg.ac.uk/PETS2007/data.html>

This dataset contains 8 examples of abandoned luggage at an airport. Each event is

recorded by four different cameras. Additionally, a background training sequence is provided. Complexity is defined with the following criteria: loitering, stolen luggage and abandoned luggage. Video sequences have been recorded in a dense, crowded scenario. Videos are between 2 and 3 minutes long, with standard PAL resolution (768x576 pixels, 25fps).

- AVSS 2007 (iLids dataset)

URL: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

This dataset has 3 sequences containing abandoned object events at an underground station, with 3 complexity levels: easy, medium, and hard, defined in terms of the density of the crowd. Each sequence is about 3.5 minutes long, with PAL resolution.

- CVSG: A Chroma-based Video Segmentation Ground-truth

URL: <http://www-vpu.eps.uam.es/CVSG/>

In this dataset, different sequences have been recorded using chroma based techniques for simple extraction of foreground masks. Then, these masks are composed with different backgrounds. Provided sequences have varying degrees of difficulty in terms of foreground segmentation complexity. Sequences contain examples of abandoned objects and objects removed from the scene.

- ViSOR: Video surveillance online repository

URL: <http://www.openvisor.org/>

This dataset is classified in different categories including outdoor and indoor events (human actions, traffic monitoring, cast shadows. . .). A total of 9 abandoned-object sequences are included, recorded in an indoor setting. These are low-complexity sequences. Videos are around 10 seconds long and are provided at 320x256@25fps resolution.

- CANDELA project

URL: <http://www.multitel.be/~va/candela/abandon.html>

This dataset contains 16 examples of abandoned objects inside a building lobby, with different interactions between object owners. Videos are around 30 seconds long, provided



Figure 2.8: Frame samples from selected datasets

at 352x288 resolution. Despite the simplicity of the scenario, the low resolution and the relatively small size of objects present challenges for foreground segmentation.

- CANTATA Left-objects dataset

URL: <http://www.multitel.be/~va/cantata/LeftObject/>

Videos from these dataset contain examples of left objects. A total of 31 sequences of 2 minutes have been recorded with two different cameras. Some videos include a number of people leaving objects in the scene (abandoned objects) and other videos show people removing the same objects from the scene (stolen objects). Videos are provided at standard PAL resolution, compressed using MPEG-4.

Chapter 3

Base system

3.1 Introduction

The work presented in this document starts from a video analysis system for abandoned and stolen object detection provided by the Video Processing and Understanding Lab [8]. This system is designed to work as part of a video-surveillance framework capable of triggering alarms for detected events in real time. This requirement imposes limits on the time complexity of the algorithms used in each of the analysis modules. The system's block diagram is depicted in Figure 3.1.

After the initial frame acquisition stage, a foreground mask is generated for each incoming frame at the *Foreground Segmentation Module*. This foreground mask consists on a binary image that identifies the pixels that belong to moving or stationary blobs. Then, post-processing techniques are applied to this foreground mask in order to remove noisy artifacts and shadows. After that, the *Blob Extraction Module* determines the connected components of the foreground mask. In the following stage, *Blob Tracking Module* tries to associate an unique ID for each extracted blob across the frame sequence. This information is analyzed by the *Static Region Detection Module*, in order to determine which blobs have become stationary (i.e., their velocity is zero). These blobs are then classified as objects or people by the *Object Classification Module*. Finally, blobs that have been classified as stationary objects are analyzed to discriminate whether

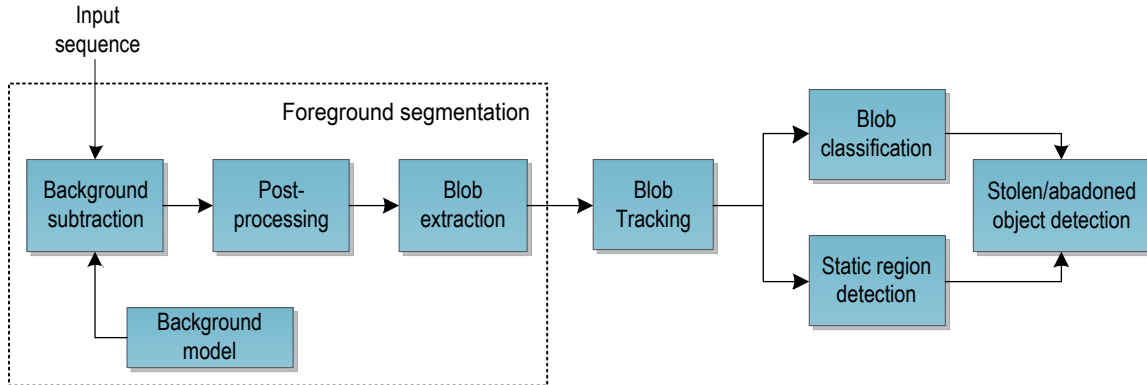


Figure 3.1: Block diagram of the existing video analysis system for abandoned and stolen object detection.

they correspond to abandoned or stolen objects.

In the remaining sections of this chapter, we firstly present, in section 3.2, the *Foreground Segmentation Module*. Then, we overview the *Blob Tracking Module* and the *Stationary Object Detection Module* in sections 3.3 and 3.4, respectively. In section 3.5, the framework for the discrimination of stationary objects between abandoned and stolen is described in detail.

3.2 Foreground segmentation module

The purpose of the *Foreground Segmentation Module* is the generation of binary masks that represent whether pixels belong to the background or foreground (moving or stationary blobs). Based on the Background Subtraction (BGS) segmentation technique, a background model is created and then updated with the incoming frames. This initial mask then undergoes noise and shadow removal operations in order to obtain the final foreground mask for the current frame and perform connected component analysis for blob extraction. Figure 3.2 depicts the block diagram of the *Foreground Segmentation Module*.

3.2.1 Background subtraction

The background subtraction technique employed in this system is based on the one proposed in [36]. The background model is initialized with the average value of a short sequence of

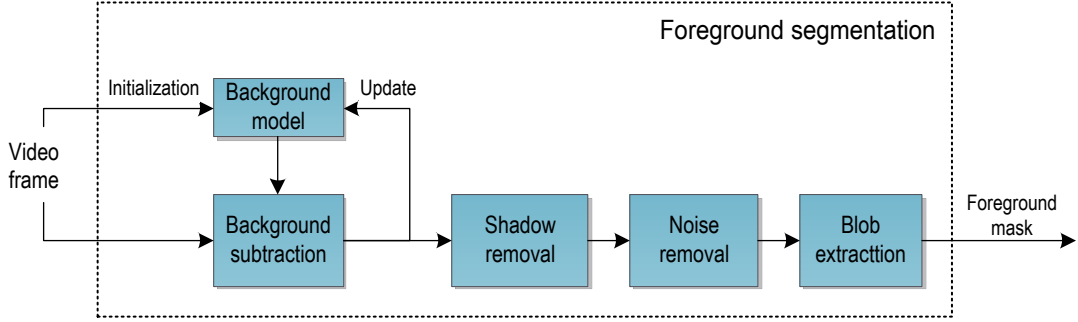


Figure 3.2: Block diagram of the *Foreground Segmentation Module*.

training frames that do not contain foreground objects. This model is adaptively updated to consider slow changes in global illumination conditions using a running average method [37]. Then, the distance to the background model is calculated for each incoming frame. It consists on the squared difference between the two images (background and current), calculated around a square window for each pixel. Finally, foreground segmentation is computed by thresholding this distance according to the following equation.:

$$F(I[x, y]) \iff \sum_{i=-W}^W \sum_{j=-W}^W (|I[x + i, y + j] - B[x + i, y + j]|)^2 > \beta \quad (3.1)$$

where W is a square window centered in each pixel, I is the current frame, B is the background model and β is the threshold for foreground segmentation.

3.2.2 Shadow removal

Shadows cast by objects and people are often misclassified as being part of the foreground due to their significant difference with the background model. Hence, high-level stages of analysis, that take as valid the data from the foreground masks (e.g., blob contour), will also be affected when adjacent shadows are wrongly considered as part of the object and therefore, their performance is decreased.

A shadow removal technique is applied to the foreground mask for removing those pixels that belong to shadows produced by moving or stationary entities (e.g., objects and people). For this purpose, the Hue-Saturation-Value (HSV) color space is used, as it allows us to explicitly

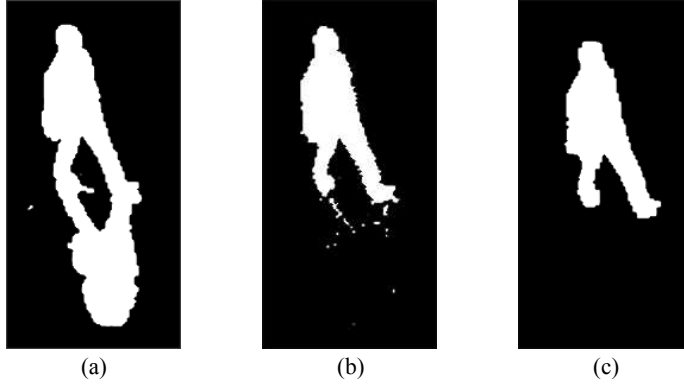


Figure 3.3: Foreground mask at different stages: initial foreground mask (a), after shadow removal (b) and after noise removal (c)

separate between chromaticity and intensity, as suggested in [38].

To perform shadow removal, the system employs the technique proposed in [39]. This approach takes advantage of the fact that for cast shadows, the change in chromaticity (hue and saturation) between the current and background image is not significant. The ratio intensity between both images is also computed, to detect intensity changes that are likely due to the presence of shadows. To classify a pixel as part of a shadow, the following decision function is used:

$$SP(x, y) = \begin{cases} 1 & \text{if } \alpha \leq \frac{I_V(x,y)}{B_V(x,y)} \leq \beta \wedge D_S \leq \tau_S \wedge D_H \leq \tau_S \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

where $SP(x, y)$ is the foreground mask that highlights pixels that belong to cast shadows at coordinates (x, y) ; I and B are the current frame and the reference background respectively; subindexes H, S and V indicate the channel in the HSV color space; and D_S and D_H denote the chromatic difference between the current frame and background for both channels.

The final foreground mask with removed shadows is obtained by performing logical XOR operation on SP and the mask generated by the preceding module. An example of shadow removal is shown in Figure 3.3b.

Algorithm 3.1 Opening by reconstruction of erosion

- 1) Start with image X
 - 2) Obtain marker image Y , by eroding X with structuring element B .
 - $Y = X \ominus B$ (erosion)
 - 3) Initialize H_1 to be the marker image Y and proceed iteratively:
 - $D = H_k \oplus B$ (dilation)
 - $H_{k+1} = D \cap X$
 - Stop condition: $H_{k+1} = H_k$
-

3.2.3 Noise removal

Additionally, morphological operations are performed on the resulting foreground mask for removing noisy artifacts. In particular, a combination of erosion and reconstruction operations known as “*Opening by Reconstruction of Erosion*” is applied as described in [40]. Its purpose is to remove small objects (in our case blobs due to noise), while retaining the shape and size of all other blobs in the foreground mask.

Morphological reconstruction involves an image X (the foreground mask to be processed), a marker Y and a structuring element B . In the selected approach, the marker Y is first calculated by performing the erosion operation on X , and the final mask is then obtained by performing dilation iteratively. The procedure is described in Algorithm 3.1.

The size and shape of the structuring element will determine the artifacts that will be removed from the foreground mask. In Figure 3.3c we can see an example of the described operation applied to a noisy foreground mask with a 3x3 squared structuring element.

3.2.4 Blob Extraction

After applying background subtraction and post-processing the obtained foreground mask, the *Blob Extraction Module* labels each isolated groups of pixels in the mask using Connected Component Analysis. The implemented algorithm uses *8-connectivity* as the criteria to determine if pixels belong to the same connected region. It works as described in Algorithm 3.2.

Algorithm 3.2 Connected component labeling using *8-connectivity* condition

For each pixel p of the foreground mask, the values of neighboring pixels in the west, north-west, north and north-east positions are analyzed; and is labeled according to the following rules:

- If all four neighbors have a value of 0, assign a new a label to p , *else*
- If there is only one neighbor with a value of 1, assign its label to the p ; else
- If more than one neighbor have a value of 1, assign one of the labels to p , and store the equivalence between neighboring labels in a table.

On a second pass, the table is used to merge neighboring labels.

After the labeling process, some very small regions may be detected. These may be due to noise that was not correctly eliminated by the *Noise Removal Module*, or due to residual artifacts as a result of incorrect segmentation. In order prevent higher level modules from analyzing these regions, the ones below a certain area (in pixels) are discarded.

3.3 Blob Tracking

This module performs tracking of the blobs extracted by the previous module. This is done by estimating the correspondence of blobs between consecutive frames (current and previous frames). A match-matrix MM is used to quantify the likelihood of correspondence between blobs in the previous and current frame. For each pair of blobs, the values of this matrix are computed using the normalized Euclidean Distance between their blob centroids and their color. It is calculated as follows:

$$MM_{nm} = \sqrt{\left(\frac{\Delta X}{Xdim}\right)^2 + \left(\frac{\Delta Y}{Ydim}\right)^2} + \sqrt{\left(\frac{\Delta R}{255}\right)^2 + \left(\frac{\Delta G}{255}\right)^2 + \left(\frac{\Delta B}{255}\right)^2} \quad (3.3)$$

where each row n corresponds to blobs in the previous frame and each column m to the number of blobs in the current frame. ΔX and ΔY are the differences in the X and Y directions between the centroids in the past and previous frame, normalized to their maximums values, $Xdim$ and $Ydim$ (the frame dimensions). ΔR , ΔG and ΔB are the differences between mean R , G and B color values, also normalized to the maximum value (255 for 8-bit RGB images).

Then, the correspondence for each blob m is defined as the index i ($i = 1 \dots n$) that presents minimum value MM_{im} .

3.4 Stationary object detection

In order to perform the detection of abandoned and stolen objects, only stationary blobs that correspond to objects, as opposed to people, are considered. Stationary blobs are first detected using the results of tracking analysis. These blobs are then classified using simple geometric features to discriminate between people and generic objects.

3.4.1 Stationary blob detection

Motion data generated by the *Blob Tracking Module* is used to determine which blobs have remained stationary for a certain amount of time. The detection is performed by analyzing blobs's speeds as they move around the scene. Firstly, the speed of the blob is calculated as follows:

$$BlobSpeed = \sqrt{v_x^2 + v_y^2} \quad (3.4)$$

where v_x^2 and v_y^2 are, respectively, the difference between the current and previous blob centroids in x and y directions.

Then, if a blob's speed remains close to zero for a period of time, it is detected as a stationary region. In this system, this period was chosen to be 2 seconds, equivalent to 50 consecutive frames for 25fps video data.

3.4.2 Object classification

This task is performed by combining the information from two simple geometric features, in order to discriminate between people and generic objects.

1) Bounding box aspect ratio

The ratio between the dimensions of the blob's minimum bounding rectangle, or bounding

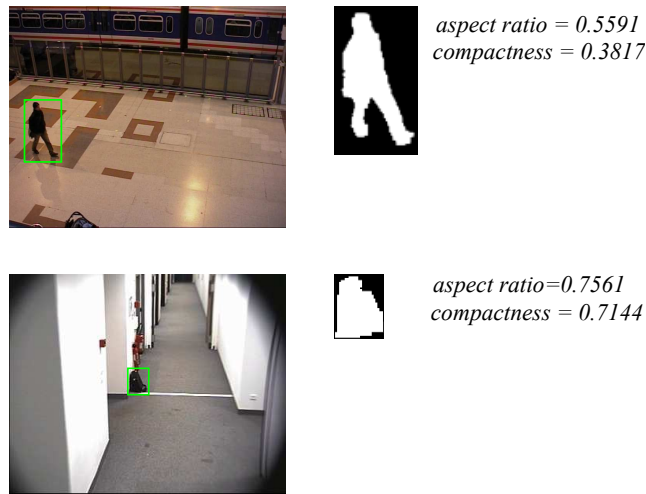


Figure 3.4: Object classification examples

box, is used to determine if it belongs to a person. This ratio (*width/height*) is modeled with a Gaussian distribution with mean μ and standard deviation σ . For people, these values were determined to be $\mu = 0.3$ and $\sigma = 0.2$ from a training sequence.

2) Compactness

Compactness is defined as the percentage of pixels inside the blob's bounding box that correspond to foreground pixels (value 1 in the binary mask). It has been observed that blobs showing a compactness value lower than 70 – 75% can be classified as people.

These two measures are then combined using Bayesian inference to perform the classification. Examples of these measures for people and objects are shown in figure 3.4.

3.5 Abandoned and stolen object discrimination

Stationary foreground objects extracted by the preceding modules will be analyzed in order to determine the nature of the event that has occurred (removed or abandoned object). As depicted in Figure 3.5, we can distinguish two distinct processing stages. First, desired features are extracted from the foreground mask, the reference background, the current frame and the location of the static object; as detected by lower level analysis modules. Based on the extracted features, a likelihood measure (*score*) is then generated for each object. Based on the score or

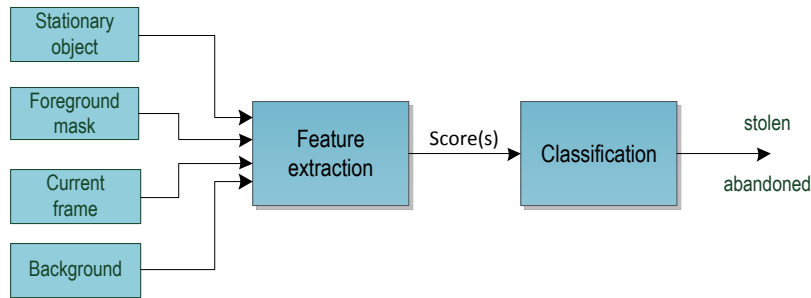


Figure 3.5: Stolen/abandoned object discriminator

set of scores, a classifier will then assign each object to a class (stolen or abandoned).

As described in section 2.4, we can distinguish between color-based and edge-based methods for the abandoned and stolen discrimination task. In this system, detectors of both classes are available. In subsection 3.5.1, the feature extraction is described for color-based and edge-based methods. Later on, the subsections 3.5.2 and 3.5.3 describe, respectively, the model for computing the likelihood of each method and the hybrid discrimination scheme.

3.5.1 Features

3.5.1.1 Color histogram detector

The color histogram detector is a variation of the work proposed in [31]. This approach is based on measuring the color similarity between the regions delimited by the foreground mask (internal and external regions of the blob bounding box) in both the background and the current frame. The assumption is that in the current frame, stolen objects show a higher color similarity (in the current frame) between these two regions than abandoned objects. Analogous reasoning is applied to the background frame and therefore, abandoned objects present high color similarity between these regions in the background frame. By combining the similarity measure on both images (current and background), the robustness of the detector is increased.

For each candidate object, the bounding box dimensions are increased by a factor of 1.5, centered in the object. The region inside the bounding box is extracted from the foreground mask, the current frame and the background, as shown in the first row of Figure 3.6. We can further consider that two regions are delimited by the foreground mask inside the bounding box:

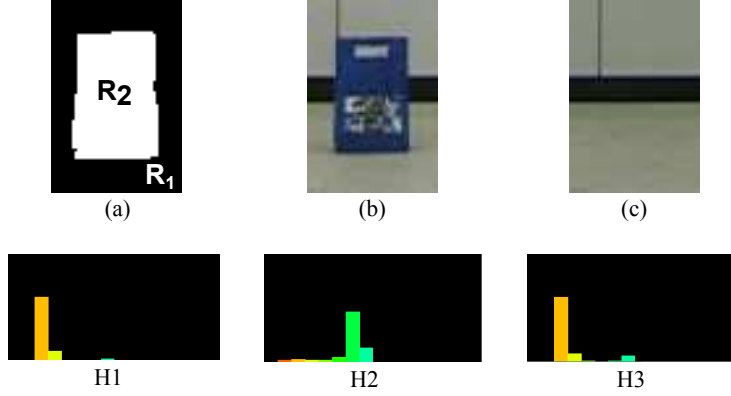


Figure 3.6: First row: Bounding-box in foreground mask (a), current frame (b) and background (c). Second row: Color histograms: R1 in background frame (H1), R2 in current frame (H2), R2 in background frame (H3).

R1 (foreground pixels with value zero) and R2 (foreground pixels with value 1).

To obtain the color similarity measures, the color histogram of the hue channel in the HSV color space is computed for pixels belonging to the R1 and R2 regions in the background and the current frame. Thus, three histograms are computed and normalized:

- H1: histogram of R1 in the background frame.
- H2: histogram of R2 in the current frame.
- H3: histogram of R2 in the background frame.

A fourth histogram (R1 in the current frame) could be used, as described in [31]. In this implementation, however, H1 is used for both comparisons, as it was seen that it provides robustness in those cases in which nearby moving objects and shadows are present in R1.

Histogram similarity (between H1 and H3, and H1 and H2) is then computed employing the Bhattacharyya distance:

$$D_{bat}(H_i, H_j) = -\ln \left(\sqrt{\sum_{x \in X} H_i(x) H_j(x)} \right) \quad (3.5)$$

where H_i and H_j are the histograms being compared, and x is the number of histogram bin. Since the histograms are first normalized, D_{bat} is a value between 0 and 1. The color histogram detector score (S_{CH}) is then computed as follows:

$$S_{CH} = D_{bat}(H_1, H_3) - D_{bat}(H_1, H_2) \quad (3.6)$$

3.5.1.2 Gradient detectors

Two detectors are available based on comparing the values of the image gradient along the contour of the object (as obtained from the foreground mask) in the background and current frame images. The implemented approaches are similar to the one described in [30]. The main differences lie on the inclusion of a contour adjustment stage, the gradient operator employed (*Sobel* instead of *SUSAN* edge detector), and a scheme to remove redundant information the analyzed gradient image.

The first step is to extract the contour of the object from the foreground mask. The *Sobel* operator is applied to the current image I and the background B to obtain an estimation of their gradient images, G_I and G_B . The initial contour is then adjusted in the current image so that it correctly fits the actual object boundaries. This is done in order to correct defects in the contour's shape that may have been caused by an imprecise segmentation process. Then, redundant information is removed from the current image gradient by subtracting the background gradient image. The gradient difference image is computed as follows:

$$G_{diff} = \begin{cases} G_I - G_B & \text{if } (G_I - G_B) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

This operation eliminates those gradient points that are present on both images (redundant information), as we assume that they belong to the background and they generally fall outside of the object boundaries. Only positive values in the gradient difference are taken into account. For abandoned objects, G_{diff} will highlight the object's edges; for stolen object, edges uncovered by the removal of the objects will be highlighted.

The *high gradient detector* analyzes pixel values in the gradient difference image, along the adjusted contour, that are above a certain threshold (τ_{GH}). For each pixel, this condition is checked inside a small neighborhood. Conversely, the *low gradient detector* accounts for pixel

values that are below a threshold (τ_{GL}). A small window around each contour pixel is used to analyze these conditions.

The adjusted contour is a set of N points $C = \{p_1, \dots, p_N\}$, specified by their coordinates (x, y) . For each pixel in the contour, we consider a small window of neighboring pixels $W(x, y)$ of size $K \times K$ centered in position (x, y) . The test conditions of both detectors for contour pixels are defined as follows:

$$f_{GH}(x, y) = \begin{cases} 1 & \text{if } \{p \in W(x, y) \mid G_{diff}(p) \geq \tau_{GH}\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

$$f_{LH}(x, y) = \begin{cases} 1 & \text{if } \sum_{i=1}^{K^2} G_{diff}(W_i) \leq \tau_{LH} \cdot K^2 \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

The score generated by each detector is then defined by the following equation:

$$S_{GH} = \frac{1}{N} \sum_C f_{GH}(x, y) \quad (3.10)$$

$$S_{GL} = \frac{1}{N} \sum_C f_{GL}(x, y) \quad (3.11)$$

The selected thresholds are $\tau_{GH} = 220$ and $\tau_{GL} = 30$. For abandoned objects, a high number of contour pixels are expected to meet the high-gradient test condition, obtaining scores close to 1.0. Conversely, low scores (close to 0.0) will be generated by the low-gradient detector. Analogous reasoning can be applied to stolen objects. To take advantage of this duality, a combination of both detectors is proposed by generating the following score:

$$S_{GRD} = S_{GH} - S_{GL} \quad (3.12)$$

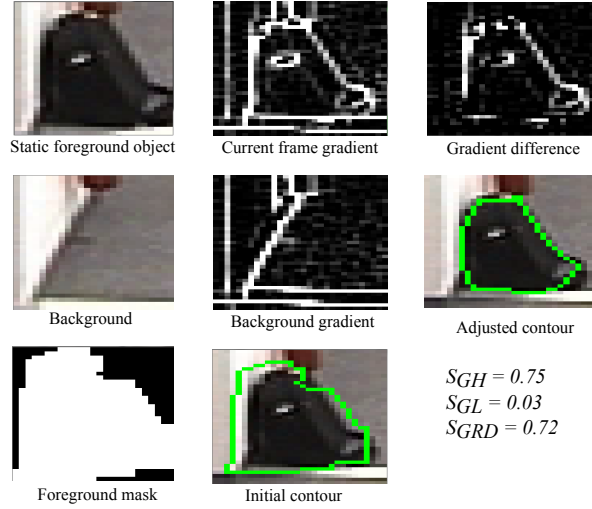


Figure 3.7: Gradient detectors example for an abandoned object

3.5.2 Evidence model

In order to obtain a likelihood for performing the discrimination into abandoned or stolen objects, two evidence measures (one for each class) are derived from the detector's score. The score is assumed to follow a Gaussian distribution of mean μ and standard deviation σ . These two parameters are obtained from a training set consisting of abandoned and stolen objects in different scenarios. Given a detector score x , the evidence measures are defined as values between 0.0 and 1.0, the latter when the score is equal to the mean, as shown in the following equation:

$$E_{\mu,\sigma}(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.13)$$

Two evidence measures E_U and E_S (for unattended and stolen, respectively) are obtained for each detector, computed from the scores S_{CH} (color histogram detector), S_{GH} , S_{GL} (gradient detectors).

3.5.3 Hybrid abandoned and stolen object discrimination

The final evidences E_U and E_S are obtained by combining the evidences provided by the three detectors using a fusion scheme, as described in [8]. E_U and E_S are computed as shown in Eq. 3.14 and 3.15. Only evidence measures with a value above a certain significance threshold ρ are

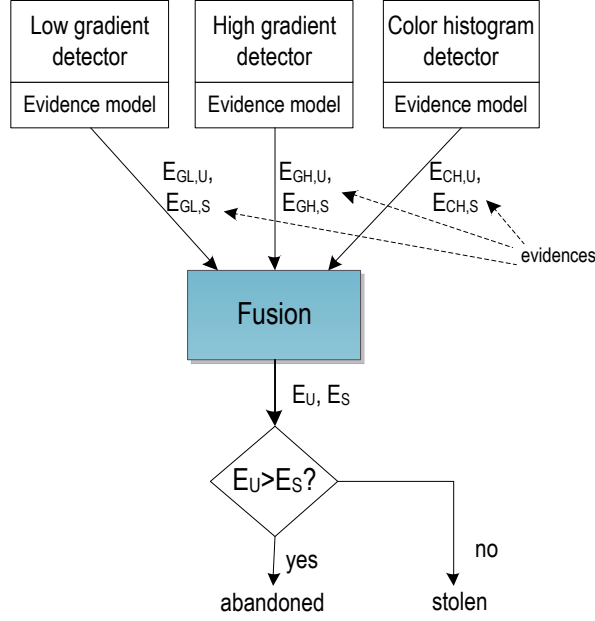


Figure 3.8: Classification by fusing evidence data

considered.

$$E_U = \frac{H(E_{GH,U} - \rho)E_{GH,U} + H(E_{GL,U} - \rho)E_{GL,U} + H(E_{CH,U} - \rho)E_{CH,U}}{H(E_{GH,U} - \rho) + H(E_{GL,U} - \rho) + H(E_{CH,U} - \rho)} \quad (3.14)$$

$$E_S = \frac{H(E_{GH,S} - \rho)E_{GH,S} + H(E_{GL,S} - \rho)E_{GL,S} + H(E_{CH,S} - \rho)E_{CH,S}}{H(E_{GH,S} - \rho) + H(E_{GL,S} - \rho) + H(E_{CH,S} - \rho)} \quad (3.15)$$

where H is the Heaviside step function. In this system, a significant threshold $\rho = 0.7$ was selected. To avoid indetermination, in case all evidence measures fall below the significance threshold, the arithmetic mean of the three measures is taken as the final evidence value.

Finally, the candidate object is classified as stolen if $E_S > E_U$, or abandoned if $E_U > E_S$. This discrimination scheme is depicted in figure 3.8.

Chapter 4

Single-feature discrimination for abandoned and stolen objects

4.1 Introduction

As described in chapter 3, the discrimination between abandoned and stolen objects is performed by using gradient and color information, respectively, at pixel and region level. The results reported in [8] have shown that these approaches present limitations for the analysis of real sequences with intermediate or high complexity. In this chapter, we propose two novel approaches for this discrimination task using the contour of the blob. Firstly, a discrimination approach based on active contours is defined to compare the contour adjustments performed on the current frame and the background image. Then, a color-based discrimination approach is also presented, based on measuring the average pixel color contrast along the initial contour, between points that are inside and outside the object boundaries (as opposed to the color-based analysis at region level of the base system).

The rest of the chapter is organized as follows: the approach based on active contours is described in section 4.2. Subsection 4.2.1 details the proposed scheme, while the remaining subsections describe the different active contour techniques that have been tested. The color-based approach is described in section 4.3. Subsection 4.3.1 overviews the discrimination scheme

whilst subsection 4.3.2 describes the pixel color contrast feature employed.

4.2 Based on active contours

Active contour models consist on a deformable spline that is driven towards object boundaries through the minimization of an energy measure. This measure consists on boundary (edge) or region energy terms, or a combination of both. When a contour adjustment is performed around the boundaries of abandoned or stolen objects, it has been observed that the obtained contours in the background image and the current frame differ significantly. By exploiting this fact, we compute a similarity measure by comparing the adjusted contours with the initial contour extracted from the foreground mask.

According to [41], active contours models can be classified as either parametric or geometric, considering whether their contour representation is explicit or implicit (using level sets). Moreover, we can further differentiate between approaches based on boundaries (edge information) or regions (color, texture). In the presented work, we have tested the most representative approaches of each category.

4.2.1 Overview of the discrimination scheme

The block diagram of the proposed approach for discriminating abandoned and stolen objects based on active contours is depicted in Figure 4.1. It starts from the initial contour of the static object, at time t , defined as the set of points $C_t^I = \{p_1, \dots, p_i, \dots, p_N\}$, where p_i represents the x, y coordinates of the i th contour, and N is the total number of contour points. It is obtained as follows:

$$C_t^I = h(F_t, M_t) \tag{4.1}$$

where F_t and M_t are the current frame and foreground mask at time t , and $h(\cdot)$ denotes the contour extraction technique. In our approach, $h(\cdot)$ consists on point-scanning the result of applying the Canny edge detector to the M_t mask. The contour indicates the boundaries of the inserted (i.e., abandoned) or removed (i.e., stolen) object, as detected by the *Foreground*

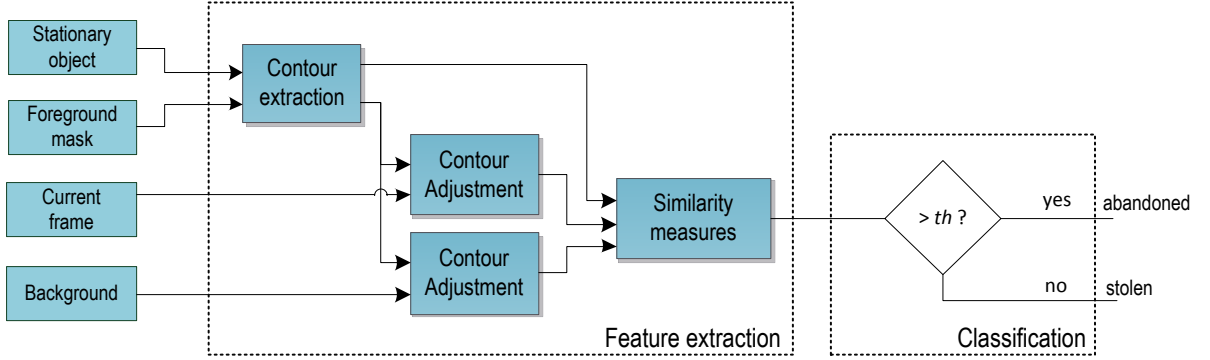


Figure 4.1: Scheme for abandoned and stolen object detection by active contours adjustment.

Segmentation Module.

Then, the regions enclosed by the stationary object's bounding box are extracted from the current frame and the background image, followed by a fitting process is performed on the initial contour C_t^I , by applying an active contours algorithm. Thus, two adjusted contours are obtained:

$$C_t^{EF} = f(F_t, C_t^I) \quad (4.2)$$

$$C_t^{EB} = f(B_t, C_t^I) \quad (4.3)$$

where $f(\cdot)$ denotes the contours adjustment technique; F_t and B_t are the current and background frames; and C_t^{EF} and C_t^{EB} are the adjusted contours in those frames, respectively. For abandoned objects, the adjusted contour will be attracted to object boundaries in the current frame, and thus it is expected to be largely similar to the initial contour. Conversely, the contour is expected to undergo significant deformations when adjusted using the background frame, due to the absence of object boundaries. In most cases, this uncovered area does not have strong edges and the contour tends to shrink or disappear. For stolen objects, the results are the opposite.

After that, a similarity measure is defined to quantify the deformation against the initial contour. For this purpose, we have decided to use the Dice's coefficient [42], which is defined as

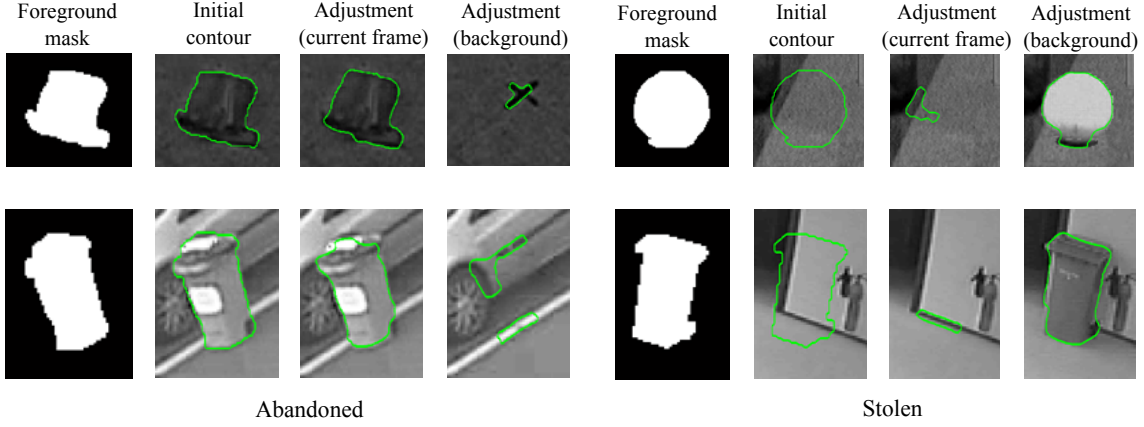


Figure 4.2: Examples of contour adjustments for abandoned objects (left) and stolen objects (right).

follows:

$$d(C_1, C_2) = \frac{2|A_1 \cap A_2|}{|A_1| + |A_2|} \quad (4.4)$$

where C_1 and C_2 represent two contours and A_1 and A_2 are the regions enclosed by them; $|A_1 \cap A_2|$ is their spatial overlap (in number of pixels) and $|A_1|$ and $|A_2|$ represent their areas (in number of pixels). We then obtain two similarity measures (d_t^F and d_t^B), by comparing the two adjusted contours C_t^{EF} and C_t^{EB} with the initial contour C_t^I . The values of d_t^B are expected to be close to 0.0 for stolen objects, and 1.0 for abandoned objects; with d_t^F getting opposite values. Afterwards, a score is obtained by combining both distances as follows:

$$S_t = d_t^F - d_t^B \quad (4.5)$$

Finally, the discrimination is performed by thresholding the final score S_t :

$$D = \begin{cases} \textit{abandoned} & \textit{if } S_t > th \\ \textit{stolen} & \textit{if } S_t \leq th \end{cases} \quad (4.6)$$

The applied threshold, th , is obtained from training data. Figure 4.2 shows examples of the contour adjustment for abandoned and stolen cases.

4.2.2 Parametric, edge-based active contours

The first algorithm we have considered is the classic active contours model [43], or snakes. Starting from the initial blob contour $C_t^I = \{p_1, \dots, p_N\}$, the adjustment is performed by iteratively minimizing an energy function E that is composed of the following terms:

$$E = \sum_{i=1}^N \alpha_i E_{cont} + \beta_i E_{curv} + \lambda_i E_{imag} \quad (4.7)$$

where N is the number of contour points, E_{cont} is the continuity energy, E_{curv} is the curvature (i.e., smoothness) energy, E_{imag} is the external energy (e.g., image edges), and $\alpha_i, \beta_i, \lambda_i \geq 0$ are the weights given to each energy. These energies are defined as:

$$\begin{aligned} E_{cont} &= \|p_i - p_{i-1}\|^2 \\ E_{curv} &= \|p_{i-1} - 2p_i + 2p_{i+1}\|^2 \\ E_{imag} &= \text{gradient}(I_t) \end{aligned} \quad (4.8)$$

With each iteration, points that minimize the defined energy functional are search inside a window of size $W \times W$ around each contour pixel.

4.2.2.1 Edge map

One of the main disadvantages of this approach is the fact that the initial contour has to be located close to the desired boundaries in order to achieve an accurate adjustment. However, this condition only holds true in adjustments performed on the current frame (for abandoned objects), or the background frame (for the stolen case). In the opposite scenario, it was observed that performing a contour adjustment in the absence of nearby strong boundaries generally resulted in adjusted contours that were largely similar to the initial one, reducing the discriminative power of the computed detector score. Additionally, we found the algorithm to be highly sensitive to local maxima in the gradient image (E_{imag} in Eq. 4.8), stopping the curve evolution at those points. These local maxima points can be due to noise in the gradient image, as well as

redundant edge information that is present in both the background image and the current frame.

For these reasons, we need to build an edge map that highlights the edges of the object in the analyzed image, removing other irrelevant information to prevent the contour from stopping its evolution. For this purpose, different schemes have been considered for computing the edge map. First, the gradient of the image is computed using a gradient or edge operator, followed by an operation to remove redundant information:

1) *Sobel* operator

In this case, the *Sobel* operator [44] is employed to obtain an approximation of the image gradients (background and current frame). Since the *Sobel* operator is only defined for single-channel (gray scale) images, we have also considered an scheme to consider gradient data from all three channels.

- Gray-scale image: The 3-channel image is first converted to gray scale before applying the gradient operator.
- RGB image: *Sobel* operator is applied separately to the three RGB image channels. The image gradient is then computed by taking the maximum of the single-channel gradients for each position of the three maps.

2) *Canny* edge detector

The *Canny* edge detector [44] produces a binary map that highlights the edges in the image, by thresholding the values of the image gradient. In this case, the image gradient is obtained by applying the *Sobel* operator. First, we apply the *Canny* edge detector to each RGB channel. These are then combined using the OR logical operation.

Once the gradient of both images are obtained, redundant information is removed by subtracting the two gradients. From the gradient of the image under analysis (the one in which the contour adjustment is performed), we subtract the gradient of the other image (current frame or background) to remove edge data that is present in both images. This is the same scheme applied in the gradient detectors (section 3.5.1.2):

$$E_{imag} = G_{diff} = \begin{cases} G_1 - G_2 & \text{if } (G_1 - G_2) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

where G_1 is the gradient of the image under analysis, and G_2 the gradient of the other image and G_{diff} is their thresholded difference. This difference is then used as the external energy (E_{imag}) in Eq. 4.8.

In our approach, we have found the *Canny* edge detector produced the best results. The detector described in subsection 3.5.1.2 employs the gradient approximated with the *Sobel* operator. Since our goal is to get a contour that should be considerably deformed when the object is not present, a binary edge map provides a more suitable alternative as it eliminates the effects due to the possible local minima (opposed to the map with 256 levels obtained by the *Sobel* operator). A comparison between contour adjustments on an abandoned object is shown in Figure 4.3. If we generate G_{diff} with Sobel operator, the adjusted contour on the background image is still too similar to the initial one. The deformations obtained employing a binary edge map ultimately lead to a better decision score.

4.2.3 Geometric active contours

Geometric active contour models have been proposed [45] to solve some of the limitations present in the parametric approaches. These models are based on curve evolution and geometric flows (Partial Difference Equations, PDE), and evolve the curve towards the desired boundaries by means of average curvature motion. The contour is represented as the zero level set $\phi^{-1}(x, y) = \{(x, y) \mid \phi(x, y) = 0\}$ of a scalar function $\phi(x, y)$, commonly referred as the level set function. In *variational level sets approaches*, the curve evolution is by driven by the minimization of an energy functional defined in the level set domain [41]. These allow the incorporation of additional information into the energy functions, such as region or shape-prior information.

The main advantage of geometric models is the ability to adapt to topology changes, allowing contours to split and merge. Additionally, these approaches eliminate the necessity to initialize the contour very close to the desired object boundaries.

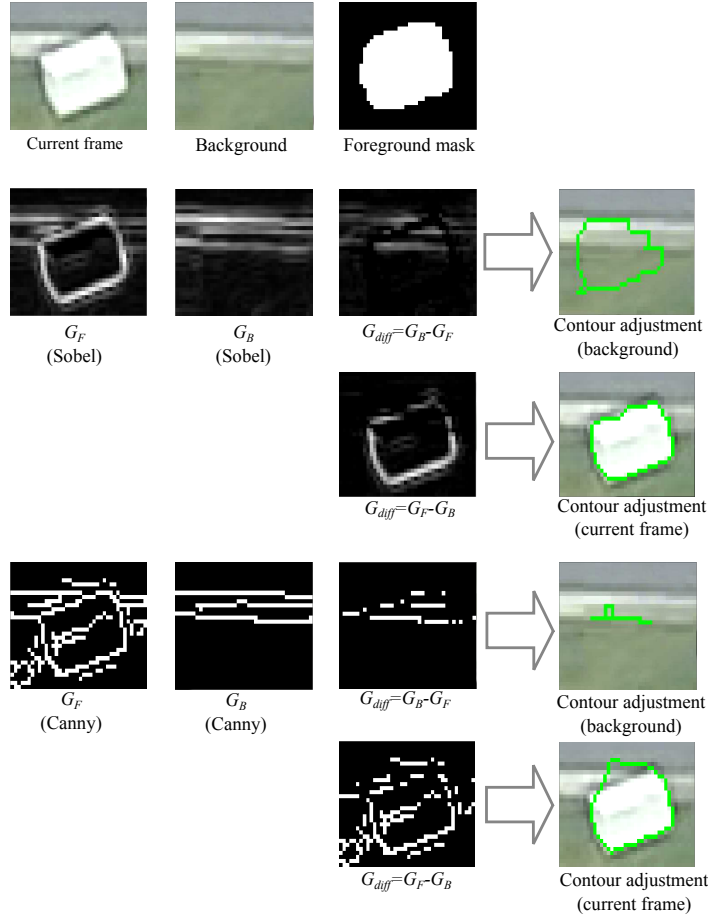


Figure 4.3: Contour adjustment comparison using the *Sobel* gradient operator (rows 2&3), and *Canny* edge detector (rows 4&5). Both operators combine information from all color channels.

4.2.3.1 Geometric, region-based active contours

Among the existing region-based active contour models, we have selected the widely referenced work described in [46]. Derived from the Mumford-Shah energy functional for segmentation [47], piecewise constant functions are defined considering the intensity means of the different regions delimited by the contour. The energy functional is defined as follows:

$$E = \lambda_1 \int_{in(C)} |I(x, y) - m_{in}(C)|^2 dx dy + \lambda_2 \int_{out(C)} |I(x, y) - m_{out}(C)|^2 dx dy + \mu L(C) + \alpha A(c) \quad (4.10)$$

where $m_{in}(C)$ and $m_{out}(C)$ are the mean intensity values of the internal and external regions

delimited by the contour; $L(C)$ is the length of the contour; $A(C)$ is the area enclosed by the contour; and λ_1 , λ_2 , α and μ are fixed positive parameters. Then, a minimization problem is considered:

$$\min_{m_{in}, m_{out}, C} E(m_{in}, m_{out}, C) \quad (4.11)$$

To compute this minimization, level set optimization is jointly performed with the estimation of mean intensity values attempting to recover two regions such that $|m_{in}(C) - m_{out}(C)|$ is maximum, assuring regularity properties for these measures. This model overcomes certain limitations of traditional parametric approaches, as it can detect objects with smooth boundaries (weak gradients) and it is more robust to noise in the image. Moreover, the initial contour can be placed at a higher distance from the object boundaries than the parametric approaches, while still attaining correct results.

4.2.3.2 Geometric, edge-based active contours

Extending the geometric approaches based on level sets, an edge-based approach is proposed in [41] that eliminates the need to re-initialize the level set, overcoming the limitations of previous re-initialization schemes that involve moving the zero level set away from its original location, resulting in inaccurate extracted contours. A new energy term is included to force the level set function to be close to a signed distance function. Thus, the proposed energy functional is defined as follows:

$$\mathcal{E}(\phi) = \mathcal{E}_m(\phi) + \mu\mathcal{P}(\phi) \quad (4.12)$$

where $\mathcal{E}_m(\phi)$ is the external energy that adjusts the zero level set to the image boundaries; $\mathcal{P}(\phi)$ is the internal energy that penalizes the deviation of the level set function away ϕ away from a signed distance function; and μ is a fixed positive parameter to control the influence of the internal energy. The external energy, $\mathcal{E}_m(\phi)$, is composed of two terms:

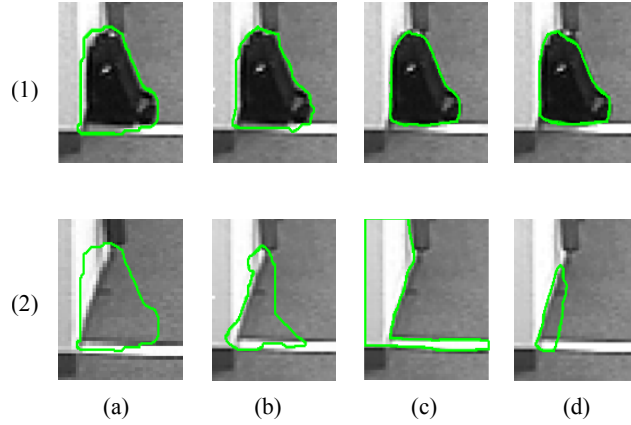


Figure 4.4: Contour adjustments for an abandoned object, performed on the current frame (row 1) and the background image (row 2). (a) Initial contour and adjustments using approaches (b) PE, (c) GR and (d) GE.

$$\mathcal{E}_m(\phi) = \lambda \mathcal{L}_g(\phi) + \nu \mathcal{F}_g(\phi) \quad (4.13)$$

where $\mathcal{L}_g(\phi)$ is the length of the zero level curve of ϕ ; $\mathcal{F}_g(\phi)$ is the speed of curve evolution; g is an edge indicator function (obtained from the image); and $\lambda > 0$ and ν are the parameters that weight the energy contributions. Parameter ν is of particular interest, as it can be used to control the curve evolution, causing its expansion ($\nu > 0$) or shrinking ($\nu < 0$), based on the position of the initial contour relative to the object boundaries (outside or inside the object). We take advantage of this behavior in the current frame (for stolen objects) and the background frame (for abandoned objects), driving the evolution of the contour and causing it to shrink, due to the lack of nearby edges. In the opposite situation, where the object is present, the initial contour is already close to the object boundaries. An example of contour adjustment using the three described approaches is shown in Figure 4.4.

4.2.3.3 Edge indicator

The edge indicator function highlights the edges in the image, and is obtained by first applying a Gaussian filter to the image, and then applying the gradient operator to the result. It is defined as follows:

$$g(x, y) = \frac{1}{1 + |\nabla G_\sigma * I|^2} \quad (4.14)$$

where G is a Gaussian kernel with standard deviation σ ; and I is a single channel image; and x, y are pixel coordinates. To accommodate information from the color channels, different approaches have been tested:

- 1) Gray-scale image conversion: I is obtained by converting the RGB image to gray-scale.
- 2) Single channel extraction: I is obtained from one of the three image channels.
- 3) A method to compute the gradient from all three color channels is described in [48], as proposed in [49]. The edge indicator is defined as:

$$g(x, y) = \frac{1}{1 + \Lambda^2} \quad (4.15)$$

where x, y are pixel coordinates, and Λ is the largest eigenvalue of the following matrix g_{ij} :

$$(g_{ij}) = \begin{pmatrix} 1 + R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & 1 + R_y^2 + G_y^2 + B_y^2 \end{pmatrix} \quad (4.16)$$

where R, G and B represent the pixel values of the respective channels after convolution with a Gaussian kernel, and C_u denotes the first order partial derivative of the C channel with respect to the u variable.

In our tests, we have observed that some edge information is lost when the color image is first converted to gray scale to compute the edge indicator (method 1), obtaining the best results using only the Red channel (method 2). While method 3 combines information from all color channels to calculate the edge indicator, no significant improvements were seen during the tests performed.

4.3 Based on pixel color contrast

4.3.1 Overview of the discrimination scheme

This detector is based on measuring the average color contrast along the boundaries (contour) of the detected objects, as proposed in [50]. First, the contour C_t^I is extracted from the foreground mask as described in Eq. 4.1. The average contrast between points inside and outside the detected region is computed, on the current frame and the background images. These contrast measures are computed as follows:

$$A_{PCC}^F = z(F_t, C_t^I) \quad (4.17)$$

$$A_{PCC}^B = z(B_t, C_t^I) \quad (4.18)$$

where $z(\cdot)$ denotes the technique for contrast analysis; F_t and B_t are the current and background frames; and A_{PCC}^B and A_{PCC}^F , are the contrast results in those frames, respectively.

For increasing robustness, color information from all channels is employed to compute the averages. Finally, the two average measures are subtracted to generate a final score S_{PCC} , as follows:

$$S_{PCC} = A_{PCC}^B - A_{PCC}^F \quad (4.19)$$

The proposed scheme for this detector is depicted in Figure 4.5.

4.3.2 Boundary Spatial Color Contrast

First, the contour is extracted from the foreground mask. For each pixel in the contour, segments of length $2L + 1$, normal to the contour's curve, are defined. The values of the pixels on both ends of the segment, points P_I and P_O , are then compared. This comparison is performed by defining a small window of size $M \times M$ centered in those pixels. The scheme is depicted in Figure 4.6. The distance measure between the two endpoints, *Boundary Spatial Color Contrast*,

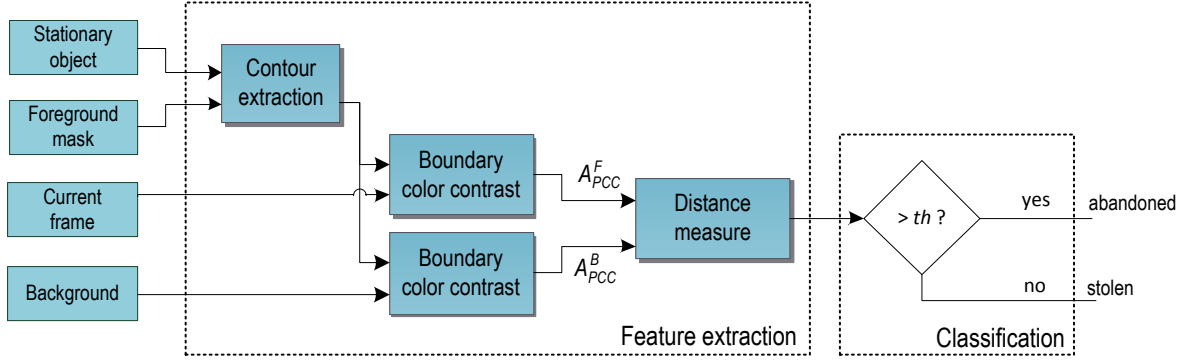


Figure 4.5: Scheme for abandoned and stolen object detection using color contrast features.

is defined for each boundary pixel as follows:

$$BSCC(F; C_t^I; i) = \frac{\|W_O^i(t) - W_I^i(t)\|}{\sqrt{3 * 255^2}} \quad (4.20)$$

where W_O and W_I are the average color values computed in the $M \times M$ neighborhood of points P_I and P_O (in the RGB color space) for the i th contour pixel from the C_t^I contour in the F frame (that could be either the current or background frame). This measure is only defined for those boundaries pixels for which P_I , P_O and the pixels inside their neighborhoods fall inside the image boundaries (considering as non valid those pixels that fall outside image boundaries).

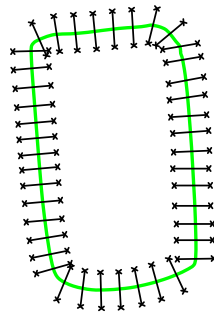
The average $BSCC$ value along the analyzed contour pixels is then expressed as follows:

$$z(F, C_t^I) = \frac{1}{K_t} \sum_{i=1}^{K_t} BSCC(F; C_t^I; i) \quad (4.21)$$

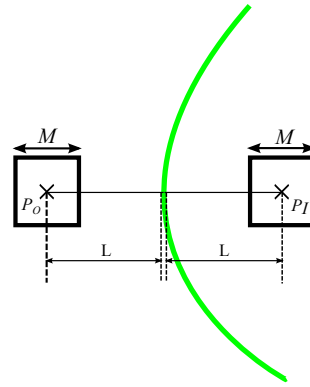
where K_t is the total number of analyzed pixels with valid values, and $BSCC$ is the spatial color contrast measure for the i th pixel. This function computed on both the current frame and the background (as shown in Eqs. 4.17 and 4.18), and combined to obtain the detector's score (as shown in Eq. 4.19). A_{PCC}^B is expected to have a value close to 0.0 for abandoned objects, and a higher value for stolen objects, due to the contrast between the object and its surroundings; with A_{PCC}^F getting opposite values in the same situations.



(a)



(b)



(c)

Figure 4.6: Pixel color contrast detector: (a) static foreground object, (b) analyzed points along the boundary and (c) analyzed contour point.

Chapter 5

Multi-feature discrimination for abandoned and stolen objects

5.1 Introduction

As explained in section 2.4, the problem of discriminating stationary objects between abandoned and stolen objects has been modeled as a classification problem. So far, we have employed simplistic approaches to perform classification based on extracted features obtained from the foreground mask, current frame, and background image. In section 3.5.3, the fusion scheme employed in the base system is presented, combining information from different features using an evidence (Gaussian) model. For the proposed detectors, single-feature discrimination is performed by thresholding the obtained scores (feature values).

In the presented work, we have tested different *Machine Learning* (ML) techniques to combine information from different detectors to perform the discrimination between abandoned and stolen objects. ML techniques assign a set of observed features to a category, by previously training a method from a set of data instances for which correct classification is known.

Firstly, we describe the motivations for multi-feature fusion in section 5.2. In section 5.3 we describe the general structure of a multi-feature classifier. In the following sections, the three ML techniques that have been tested are described: Naïve Bayes (section 5.4.1), Support Vector

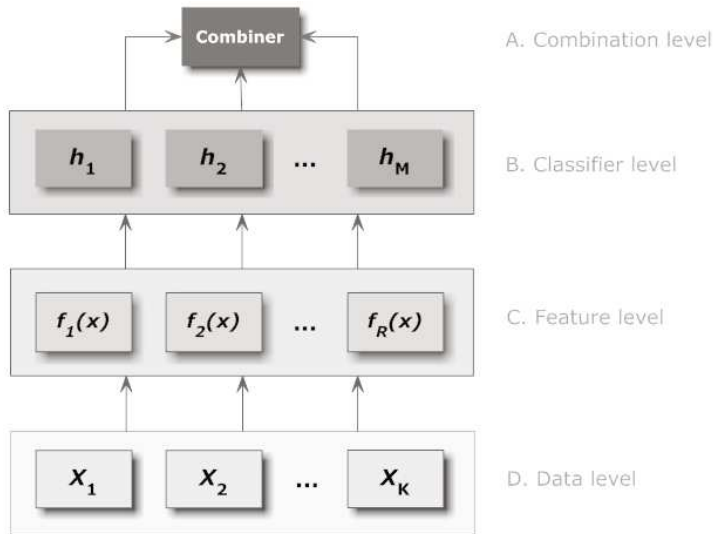


Figure 5.1: Levels of information fusion in a visual system [4]

Machines (section 5.4.2), and K-nearest neighbor (section 5.4.3).

5.2 Motivation for multi-feature fusion

As explained in section 2.4, we have described the discrimination of objects of interest between abandoned and stolen as a classification problem. Based on the information extracted at the *Stationary Object Detection* stage, features are extracted from the background and the current frame, and combined into a single likelihood measure (score). The final decision is then provided by a classifier based on the observed score, or set of scores from different extracted features.

According to [4], we can distinguish between four levels of data fusion, as shown in Figure 5.1. At the data level, different data sources are fused, such as the three color channels to provide color information, or an additional infrared channel. The feature level merges the output of different extracted features, by combining features into a single measure, or by adaptively selecting the best discriminating features. At the classifier level, decisions are made based on one or more of these combined features. Finally, in the combination level we include techniques that combine the decision from different classifiers.

This analysis can be applied to our problem, as fusion of information is being performed at

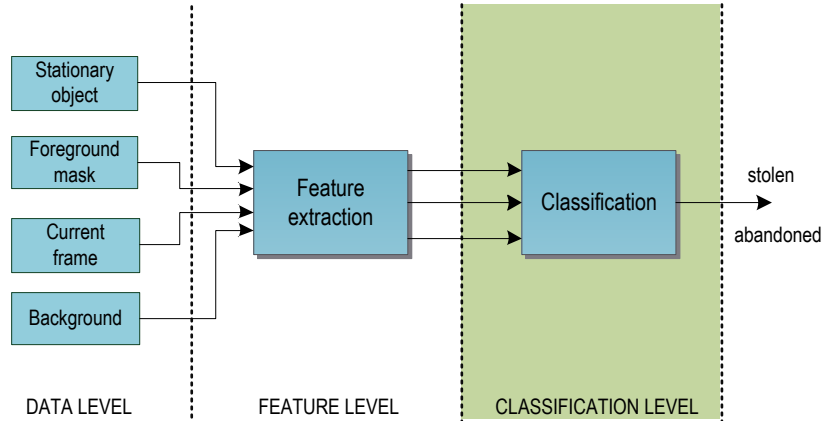


Figure 5.2: Fusion levels in the stolen/abandoned discrimination problem

all stages. At the data level, the information from the three color sensors is combined to obtain a single image. At the feature level, we extract relevant edge or color information from the background image and the current frame. The extracted features (color histogram distances, average edge energy, contour similarity measures and average color contrast) from these two images are then combined onto a single value by subtracting them, obtaining a robust discriminating value (score). At the classification level, we have proposed a simple scheme based on applying a decision threshold to a single score. In this chapter, we explore different classification schemes to overcome certain limitations observed in the single-feature approaches.

5.2.1 Observed limitations of single feature discrimination

In our tests, we have found that in non-ideal conditions, the discriminators not always generate a score with enough discriminative power, resulting in some cases in wrong classification.

- *Imprecision.* All detectors of the detectors rely on accurate background-foreground segmentation. Inaccurate foreground masks can be obtained due to one of the following reasons: camera noise, sudden changes in illumination, insufficient background model initialization and update or inadequate parameters. While the employed techniques (Chapter 3) take these issues into account, there still may be inaccuracies in the foreground masks.
- *Uncertainty.* As opposed to imprecision, it depends on the stationary object being ob-

served, rather than on the mechanism employed to detect stationary objects. In some cases, color data may not be sufficient enough to obtain a good measure, due to the object sharing similar color properties than its surroundings (camouflage effect). For edge-based detectors, there may be problems due the presence of strong edges near the object boundaries that belong to the background (and the uncovered background in absence of the object). Objects that present these problems generally obtain scores near the decision threshold.

The described discriminators (base system and proposed discriminators) deal with imprecision in a variety of ways. For example, the gradient-based approaches perform a contour adjustment prior to the analysis to try to match the contour to the actual object boundaries. The similarity measure used in the proposed active contours detectors has shown robustness as long as the two adjusted contours are sufficiently different. The proposed color contrast detector analyzes the difference between points distant from the contour, allowing for some margin of error in the foreground mask. Uncertainty, however, cannot be solved using single-feature approaches, and there is a need for fusing the data from multiple features. For example, there may be instances in which gradient-based discriminators fail to produce a good discriminative score due to strong edges in the background. In those cases, a color-based approach may produce a better measure if the color properties of the object and the background are different.

5.2.2 Advantages of multi-feature fusion

For these reasons we explore different classification schemes based on multiple inputs, to overcome the limitations observed in single-feature approaches. This way, the system is capable of compensating errors when one of the detectors produces erroneous measurements. According to [4] and [51], we can expect the following advantages from the fusion of multiple data (features).

- 1) *Robustness and reliability.* The system is able to provide a correct decision even if one of the detectors fails.
- 2) *Increased confidence.* The decision of one detector can be confirmed by the others, therefore

therefor increasing the confidence of the classifier’s decision.

- 3) *Dimensionality expansion.* If we consider single features as dimensions in a space, the combination of different features results in a higher dimensional space, better discrimination results may be obtained.
- 4) *Adaptability.* As we have explained, the detectors provide better results under certain conditions. If these conditions can be evaluated online, a selection mechanism can be put in place to employ the detectors that are known to produce the best results under those conditions, increasing the overall robustness of the system.

5.3 Structure of the multi-feature discriminator

Under the aforementioned considerations, we have studied different ML classification techniques for the multi-feature discrimination task. In ML, the classification task is commonly known as *supervised learning*. The goal of such classifier is to produce (“*learn*”) a decision rule from training data, so that future observations can be attributed to a group or class. The classifier typically maintains a model, and the parameters of this model are determined from the training data.

For each stationary object detected by the preceding modules, we first obtain the score values from the available edge-based and color-based detectors. Then, a previously trained classifier performs the decision between abandoned and stolen based on a combination of the obtained scores. The proposed scheme is shown in Figure 5.3.

More formally, we can characterize a classifier as function f that maps a vector of K input features (*feature vector*) $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iK})$ to a class label $c_i \in \{i = 1, \dots, C\}$. In our particular case, the extracted features are the scores obtained by the available detectors, and $c_i = \{abandoned, stolen\}$. Prior to the classification, the classifier’s parameters are initialized based on training data. The training data consists on a set of N samples for which correct classification is known. They are denoted as:

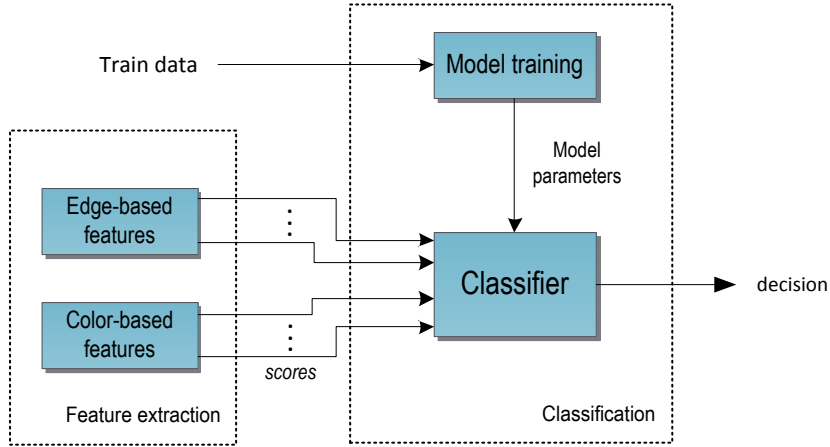


Figure 5.3: Scheme of the proposed multi-feature discriminator

$$\mathcal{D} = \{(\vec{x}_i, c_i) \mid \vec{x}_i \in \mathbb{R}^K, c_i \in \{abandoned, stolen\}\}_{i=1}^N \quad (5.1)$$

Then, the function f to predict the classes for the test data is defined as follows:

$$c_i = f(\vec{x}_i, M) \quad (5.2)$$

where \vec{x}_i is the feature vector under test and M represents the parameters of the selected combination (or classification) scheme. For a more readable notation, we have omitted M and the subindex i and, therefore, used $f(\vec{x})$ instead of $f(\vec{x}_i, M)$ in the following sections.

5.4 Selected combination techniques

In this section, we describe the selected techniques for multi-feature discrimination of abandoned and stolen objects. They are: Naïve Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). These are widely studied ML techniques for the classification task.

NB [52] is a popular technique that performs classification that predicts the output classes based on the probabilistic properties observed in the training data (*probabilistic classifier*). This approach assumes that input features are conditionally independent, resulting in a classifier with low computational complexity that scales very well for a high number of attributes. The NB

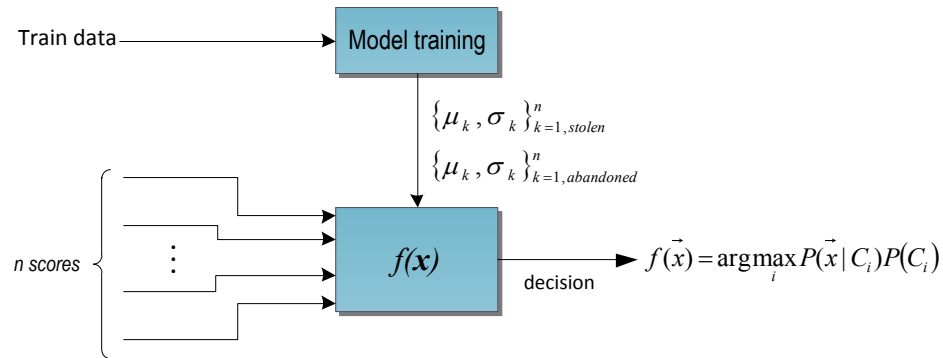


Figure 5.4: Scheme of Naïve Bayes classifier using normal distributions

has proven effective in a variety of applications, competing with more sophisticated classification schemes.

In the SVM [53] classification scheme, the training data is represented as points in space. It is a very popular technique due to its ability to map input features into an arbitrarily large space in which linear separability is achievable. The algorithm tries to find a hyperplane that separates the two classes by solving an optimization problem. The solution is obtained with independence of attributes dimensionality, and it is completely characterized based on a subset of the input data (called support vectors).

The KNN [54] algorithm is a simple technique that requires no specific training phase. In spite of its simplicity, it has shown to provide good results with negligible computational cost. However, the entire training dataset is required for the classification, as opposed to a reduced set of parameters employed by other approaches.

5.4.1 Naïve Bayes

In the NB technique, the Bayes rule is used to calculate the conditional probabilities of a class label c_i given an instance of features $\vec{x} = \{x_1, \dots, x_K\}$:

$$P(c_i | \vec{x}) = \frac{P(\vec{x} | c_i) P(c_i)}{P(\vec{x})} \quad (5.3)$$

where $P(c_i | \vec{x})$ is referred to as the *a posteriori* probability (or *posterior*) that an object belong to class c_i given a feature vector \vec{x} . The classification is performed by assigning a

data instance \vec{x} to the class for which it has the highest posterior probability conditioned on the values of the input features (maximum *a posteriori* hypothesis). We can then express the classifier function as follows:

$$f(\vec{x}) = \underset{i}{\operatorname{argmax}} P(c_i | \vec{x}) \quad (5.4)$$

This equation involves maximizing Eq. 5.3. Class prior probabilities $P(c_i)$ and class conditional probabilities $P(\vec{x} | c_i)$ are estimated from training data. The estimation of class priors is straightforward. For abandoned and stolen discrimination, class priors are assumed to be equal (0.5). Estimating the class conditionals, however, entails high computational costs. In a NB classifier, it is assumed that the attributes x_1, \dots, x_n are statistically independent (*class conditional independence*). While this assumption does not generally hold true, it dramatically simplifies the computation of the class conditionals $P(\vec{x} | c_i)$, as they can be decomposed into the product of class conditional densities $P(x_k | c_i)$, which can be easily estimated from the training data. Under the independence assumption, we can obtain the class conditionals as follows:

$$P(\vec{x} | c_i) = \prod_{k=1}^K P(x_k | c_i) \quad (5.5)$$

Under the independence assumption, we can write Eq. 5.4 as follows:

$$f(\vec{x}) = f(x_1, \dots, x_K) = \underset{i}{\operatorname{argmax}} P(c_i) \prod_{k=1}^K P(X_k = x_k | c_i) \quad (5.6)$$

The denominator $P(\vec{x})$ can be ignored, as it is constant for both classes and does not influence the decision. The class conditionals $P(\vec{x} | c_i)$ have to be separately estimated for each class. A common approach is to assume that these densities follow a normal distribution, with mean μ_k and standard deviation σ_k . Other approaches, such as Kernel Density Estimation, may be used to model the class conditional densities if the features do not follow a normal distribution.

5.4.2 Support Vector Machines

The Support Vector Machine technique (SVM) provides a way to classify instances of data inputs based on a model obtained from a set of training samples. Each training sample \vec{x} (with K attributes) is represented as a point in the \mathbb{R}^K space (*data space*), and belongs to one of two categories $y_i \in \{+1, -1\}$. The goal is to build a decision function from the training data.

In the simplest scenario, we consider the samples in the data space to be linearly separable, i.e., we can draw decision boundaries that separate the training samples in the data space. Formally, we say that there exist infinite hyperplanes that separate the training samples. Given an hyperplane that linearly separates the two categories, characterized by its equation:

$$\vec{w} \cdot \vec{x} + b = 0 \quad (5.7)$$

where \vec{w} is the normal vector to the hyperplane and $\frac{b}{\|\vec{w}\|}$ determines the offset from the origin. We can define a classifier function f as follows:

$$f(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \quad (5.8)$$

To minimize the classification error, the goal is to find an hyperplane such that its distance to the nearest samples from both classes is maximized (*maximum margin classifier*). The goal is to determine \vec{w} and b so that the margin between the two classes is maximized. Figure 5.5 shows an example of a separating hyperplane and the margin ρ . Geometrically, we can express the margin ρ as:

$$\rho = \frac{2}{\|\vec{w}\|} \quad (5.9)$$

The problem can be expressed as an optimization problem, that consists on maximizing $\frac{1}{\|\vec{w}\|}$, or equivalently, minimizing $\|\vec{w}\|$, subject to the following constraint condition:

$$y_i (\vec{w} \cdot \vec{x} + b) \geq 1 \quad (5.10)$$

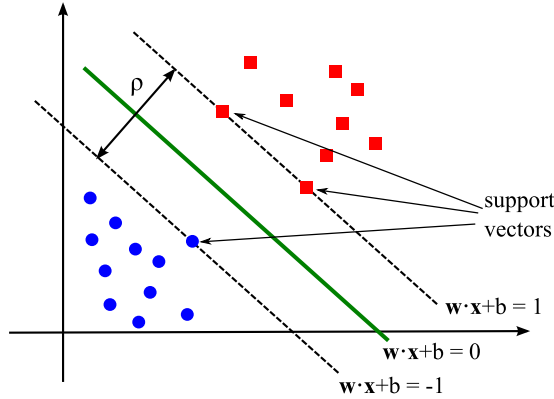


Figure 5.5: Maximum margin hyperplane and support vectors

This formulation is equivalent to a quadratic optimization problem, and the solution for \vec{w} is a linear combination of all training points, weighed by the α_i solutions (Lagrange coefficients) of the optimization problem:

$$\vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad (5.11)$$

where n is the total number of training points; b is expressed as $b = y_k - \vec{w} \cdot \vec{x}_k$, for any \vec{x}_k such that $\alpha_k = 0$. For points other than the support vectors, the α_i coefficients equal zero. Therefore, the solution is uniquely identified by a linear combination of a subset of the training points, called the *support vectors* (the ones that lie on the margin). The discriminant function f is finally expressed as:

$$f(\vec{x}) = \text{sign} \left(b + \sum_{i=1}^n \alpha_i y_i (\vec{x}_i \cdot \vec{x}) \right) \quad (5.12)$$

where \vec{x}_i are the training points. As can be seen in this equation, the solution depends only on the coefficients and the inner products between the support vectors and the input data. This implies that the solution can be obtained irrespective of the number of dimensions, as long as we can compute the inner product between vectors. Taking advantage of this fact, if the data in the input space is non linearly separable, a non-linear transformation $\phi : \mathbb{R}^n \rightarrow \mathcal{F}$ can be applied to map the input data to a higher dimensional space \mathcal{F} (*feature space*) in which separability is

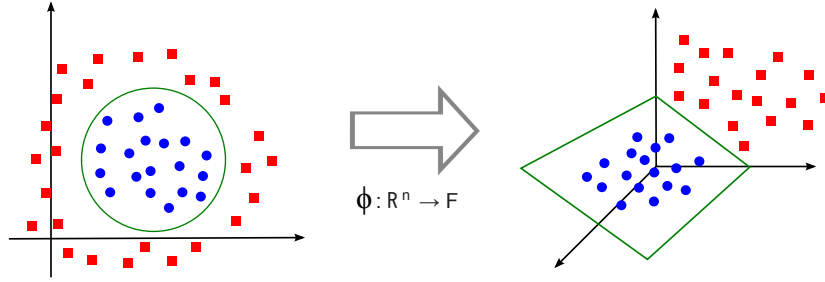


Figure 5.6: Non-linear transformation example

achieved. Figure 5.6 shows an example of such a transformation.

After applying the non-linear transformation, the solution can be found by solving the quadratic optimization problem previously described. The resulting discriminant function is then expressed in terms of a kernel function K that characterizes the inner product in the feature space:

$$f(\vec{x}) = \text{sign} \left(b + \sum_{i=1}^n \alpha_i y_i K(\vec{x}_i, \vec{x}) \right) \quad (5.13)$$

Additionally, to account for noisy data (misclassified samples) in the training set, a slack variable $\xi_i \geq 0$ can be added to the constraint condition (Eq. 5.10), making the inequality easier to satisfy:

$$y_i (\vec{w} \cdot \vec{x} + b) \geq 1 - \xi_i \quad (5.14)$$

The optimization problem becomes:

$$\min_{\vec{w}, \xi_i} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (5.15)$$

where C is a cost parameter that penalizes the sum of all ξ_i .

A more detailed explanation of the SVM technique can be found in Appendix A.

5.4.3 K-Nearest Neighbor

The K-Nearest Neighbor technique (KNN) is a simple technique that allows classification without any statistical knowledge of the data distribution, and can work with an arbitrary number of classes. In this technique, no specific training phase is required, as the decision is based only on the distance between the input vector and the training samples.

In the simplest case, the KNN technique will assign an input vector \vec{x} the class to which its closest neighbor belongs to (*nearest neighbor rule*). This is the particular case of $k = 1$. For larger values of k , the most common class among the closest k training points will be assigned. For continuous-valued attributes, Euclidean distance is generally employed. It is defined as follows:

$$d(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{p=1}^P (x_{ip} - x_{jp})^2} \quad (5.16)$$

where \vec{x}_i and \vec{x}_j are two feature vectors. Then, k is usually determined heuristically (e.g., with cross-validation). Larger values of k help to reduce the effect of noise in the classification. When dealing only with two possible classes, as in the case for abandoned and stolen objects, it is more convenient to select an odd value for k . For each feature vector to be classified \vec{x} , this decision is determined as follows:

$$f(\vec{x}) = \underset{c_i \in C}{\operatorname{argmax}} \sum_{k=1}^K \delta(c_i, \hat{f}(\vec{x}_k)) \quad (5.17)$$

where c_i represents the classes to identify ($C = \text{abandoned, stolen}$), $\hat{f}(\vec{x}_k)$ defines the classes of the k -nearest neighbors of \vec{x} and $\delta(a, b)$ is a function that is equal to 1 if $a = b$ and equal to 0 in other case (i.e., if the class under test and the classes of the neighbors are the same or different). Since we are only dealing with two possible classes (abandoned and stolen), it is more convenient to select an odd value for k , in order to avoid indetermination. A classification example is shown in Figure 5.7. The test case (green triangle) is assigned to the blue class if $k = 3$ (inner circle), but it is assigned to the red class if $k = 7$ (outer circle).

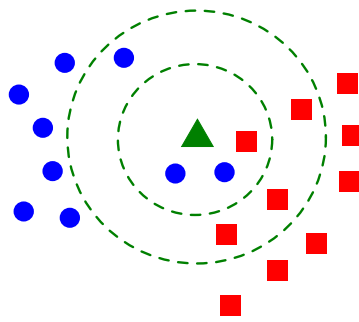


Figure 5.7: KNN classification example

Chapter 6

Experimental validation

6.1 Introduction

In this chapter, we present the results of the proposed approaches (chapters 4 and 5) and compare them against the base system (chapter 3). This evaluation is performed on the discrimination module without considering if the system was able to detect the stationary object (see definition of abandoned and stolen object in section 2.2). For this purpose, we have manually annotated the sequences of the dataset. Moreover, we have performed the evaluation on real data by automatically generating the foreground data to consider the effect of non-accurate masks.

The rest of this chapter is organized as follows. In section 6.2, we describe the experimental setup in terms of the selected sequences (subsection 6.2.1), performance measures (subsection 6.2.2), implementation issues (subsection 6.2.3) and parameter selection (subsection 6.2.4). Then, sections 6.3 and 6.4 present the results of, respectively, the proposed single-feature and multi-feature discriminators.

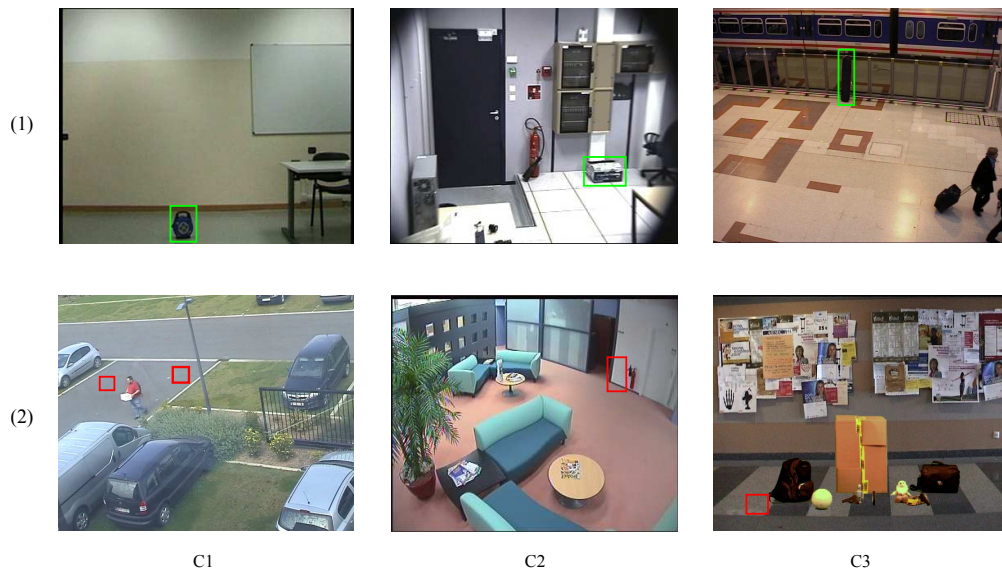


Figure 6.1: Examples of video sequences with abandoned (1) and stolen (2) objects, from all three categories (C1, C2, C3).

6.2 Setup

6.2.1 Dataset

We have selected a content set with video sequences from the PETS2006¹, PETS2007², AVSS2007³, CVSG⁴, VISOR⁵, CANDELA⁶, CANTATA⁷ and WCAM⁸ public datasets. For performance evaluation, we have divided these sequences into three categories according to the background complexity in terms of the presence of edges, multiple textures and objects belonging to the background near the objects of interest. Examples of video sequences from the three categories are shown in Figure 6.1. Category 1 includes low complexity situations, such as well defined objects placed over a solid homogenous background. Medium-complexity situations have included in category 2. High-complexity scenarios include situations such as poorly-contrasted objects placed in non-homogenous backgrounds with strong edges near the objects of interest.

¹<http://www.cvg.rdg.ac.uk/PETS2006/>

²<http://www.cvg.rdg.ac.uk/PETS2007/>

³<http://www.avss2007.org/>

⁴<http://www-vpu.eps.uam.es/CVSG>

⁵<http://www.openvisor.org/>

⁶<http://www.multitel.be/~va/candela/abandon.html>

⁷<http://www.multitel.be/~va/cantata/LeftObject/>

⁸<http://wcam.epfl.ch/>

For each sequence, we extract a number of frames containing stationary foreground regions that correspond to abandoned or stolen objects. For determining these stationary regions, we have followed the criteria described in the definitions of section 2.2. In particular, we have considered an object as abandoned or stolen if its foreground blob remains in the same position for 15 seconds. The duration of these events (the framespan for which we extract video frames) is considered as 150 frames, and one of every 10 frames is extracted for the evaluation. For some sequences in which it not possible to extract such quantity of data (i.e., not enough frames for stationary detection or short sequences), we only extract the foreground mask of the object of interest. Since we are only evaluating the last step of the system, this extraction has no influence in the final results as we are only interested in the foreground mask of such object for the discrimination task.

For each extracted frame, we elaborate the foreground masks that show the regions to be analyzed by the discriminators. Two datasets have been constructed using the same content set: one with manually annotated foreground masks (*Annotated data*), and the other with automatically generated foreground masks (*Real data*). The rationale behind this decision is the fact that in realistic settings, the foreground masks are not accurate in most of the cases. In some scenes, the background model may not correctly represent the actual background, resulting in imprecise object segmentation. This has an impact on the performance of all detectors, as they all rely on the boundaries obtained from the foreground mask. For this reason, we evaluate the discriminators on these two datasets. Table 6.1 summarizes the number of annotations (blobs) for each category, for manually annotated and real data. The higher number of real data annotations is explained by incorrect segmentation in some cases, where the mask of a foreground object splits into one or more connected components. The generation of these two datasets is explained as follows.

Annotated data A total of 89 videos have been annotated, extracting frames according to the specified criteria. For each frame, a foreground segmentation mask is manually produced, highlighting *only* those regions in which the event has occurred. The background frame is also

Table 6.1: Dataset description.

Category	Number of annotations (blobs)				Complexity
	Annotated sequences		Real Sequences		
	Abandoned	Stolen	Abandoned	Stolen	
C1	771	442	756	863	Low
C2	666	316	794	397	Medium
C3	595	174	852	660	High
All	2032	932	2402	1920	

extracted. Since the analysis is limited to the regions indicated by the foreground mask (and their surroundings), for the background frame we simply take a frame that shows the analyzed area before the event occurs. An example of a manually annotated foreground mask is shown in Figure 6.2 (left) for an abandoned object.

Real data For each video sequence, a video metadata file has been elaborated, employing the ViPER Toolkit Ground truth authoring tool⁹. These metadata files are stored in XML format, and contain event information such as event nature (stolen, abandoned, ...), coordinates and bounding box. Employing this information, foreground masks are automatically generated by the base analysis system (chapter 3), only around the regions of interest as specified in the file metadata files. The detailed procedure is explained in Appendix B. An example of a foreground mask obtained by this procedure is shown in Figure 6.2 (right), for an abandoned object.

As discussed in chapter 5, we can distinguish between two types of problems that affect a correct detection: uncertainty and imprecision. By evaluating the performance of all detectors on the *annotated dataset*, we artificially reduce the imprecision, making us able to determine the scenarios in which detectors are unable to provide a discriminating measure. In contrast, the tests performed on the *real dataset* allow us to evaluate the ability of the detectors to cope with imprecision, as well as providing performance measurements on a more realistic scenario. Both datasets are available at <http://www-vpu.eps.uam.es/ASODds>.

⁹<http://viper-toolkit.sourceforge.net/docs/gt/>

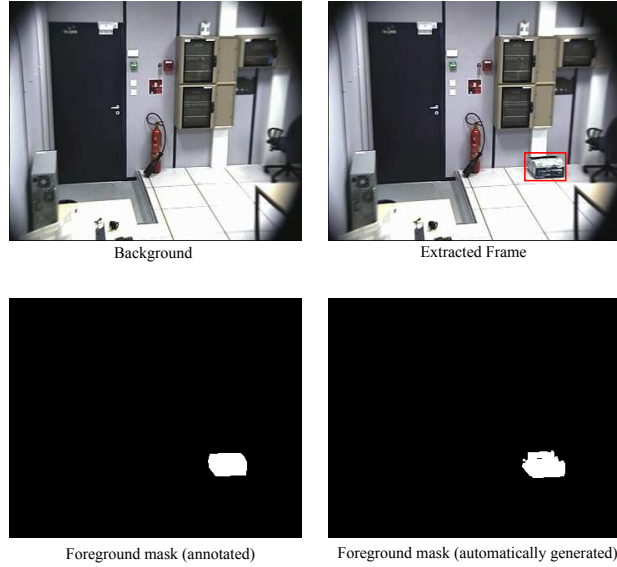


Figure 6.2: Examples of extracted frames and foreground masks.

6.2.2 Performance metrics

Two metrics have been selected for evaluating and comparing the performance of the proposed approaches: predictive accuracy and ROC analysis. Furthermore, we have performed the training and test phases following a *K-fold cross validation* method. They are described as follows:

Predictive accuracy Given a trained classifier and an appropriately labeled data set, the most intuitive way to assess the classifier’s performance is its predictive *accuracy*, i.e. the proportion of samples that are correctly classified. Conversely, the accuracy error is the proportion of misclassified samples. When there are only two possible classes (usually denoted as *positive* and *negative*), the confusion matrix depicts the four possible classification outcomes (Table 6.2). In our experiments, we compare all discriminators by using the *accuracy* measure. It can be computed as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \approx \frac{\#\text{correctly classified samples}}{\#\text{total samples}} \quad (6.1)$$

Table 6.2: Layout of the confusion matrix.

Observed classification	Predicted classification	
	abandoned(+)	stolen(-)
abandoned(+)	True positive (TP)	False Negative (FN)
stolen(-)	False positive (FP)	True negative (TN)

ROC curves Another common way of assessing and comparing the performance of classifiers is the *Receiver Operator Characteristic* (ROC) graph. In a ROC graph, the performance of a classifier is represented as a point inside the unit square. The point (0,1) corresponds to the case of perfect classification. For classifiers that base their decision by applying a threshold to a probability or confidence value (such as the score measures produced by the described detectors), a different point in the ROC space can be computed for each possible threshold. This way, we obtain a curve (*ROC curve*) rather than a single point to evaluate the performance of the classifier. To reduce the comparison to a single value, the area under the ROC curve (AUC) is generally used. It measures how discriminatory the classifier is, that is, its ability to correctly classify unseen instances. A point in the ROC graph depicts the true positive rate (TPR) against the false positive rate (FPR). These are computed as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \approx \frac{\#\text{positives correctly classified}}{\#\text{total positives}} \quad (6.2)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \approx \frac{\#\text{negatives incorrectly classified}}{\#\text{total negatives}} \quad (6.3)$$

Cross-validation Given that training sets are incomplete representations of the real world, classifiers generally perform well when tested against the training set, but may perform poorly for unseen data. This phenomenon is known as over-fitting. Once the performance metrics have been selected, it is important to correctly determine how well they perform when classifying unseen data. Among the different training schemes found in the literature, we have selected the *K-fold cross validation* [55] method. It splits the entire dataset into K equally sized, disjoint partitions (or *folds*). Only one fold is used for testing in each iteration while the remaining folds

are used for training. Then, K iterations of training and testing are performed by choosing a different fold for testing. In our experiments, we have used a value of $K = 10$ and provided the mean and standard error results of the 10 evaluations.

6.2.3 Implementation

For the parametric-edge based active contours approach, we have used the implementation of the OpenCV 2.2 C++ API library¹⁰ (released Dec. 2010). Given a set of points (initial contour) and an edge-map, the included method `cvSnakeImage()`¹¹ iteratively performs the adjustment. For the geometric region based approach, we employed an existing Matlab implementation by S. Lankton¹²; adapting the code to account for those cases in which the contour completely disappears. The geometric edge-based algorithm has been entirely implemented in OpenCV by porting the author's Matlab code¹³.

6.2.4 Parameter selection

We have selected specific parameters for the approaches under evaluation. They are:

6.2.4.1 Single-feature approaches

Discriminators of the base system The color histogram discriminator does not require parameters. For the gradient detectors, thresholds $\tau_{GH} = 220$ (high gradient) and $\tau_{GH} = 30$ (low gradient) are used; and a neighboring window of size $K = 5$ is employed. For the contour adjustment based on snakes, we use the parameter values $\alpha = 1.0$, $\beta = 1.0$, $\gamma = 2.0$.

Parametric, edge-based active contours As detailed in subsection 3.5.1.2, the three energy terms are weighed by parameters α , β and γ (in a window size W). The optimal parameters have been determined by subjectively comparing their effect on abandoned and stolen samples. The criteria was to find a set of parameters such that the adjusted contour

¹⁰<http://opencv.willowgarage.com/>

¹¹http://opencv.willowgarage.com/documentation/motion_analysis_and_object_tracking.html

¹²<http://www.shawnlankton.com/2008/04/active-contour-matlab-code-demo/>

¹³<http://www.engr.uconn.edu/~cml/>

in absence of the object of interest shrinks considerably. We found the algorithm to be more sensitive to the parameters as the window size increased. For small values of W , the adjusted contours were found to be largely similar to the initial ones in all situations. For large values of W (≥ 7), contours are considerably deformed regardless of the presence of nearby objects. These two situations would cause the detector's scores to lack discriminative power. Therefore, a value of $W = 5$ has been selected. By visually inspecting the adjustments for values of α , β and γ between 0.0 and 3.0, the optimal values have been determined to be $\alpha = 0.97$, $\beta = 1.30$, $\gamma = 0.52$. The stop condition (number of iterations) is set to 1000.

Geometric, region-based active contours We have opted to use the default values proposed by the authors. The parameters that weigh the different energy terms are λ_1 , λ_2 , μ and α . Additionally, we have parameters h (step space) and Δt (time step) from the numeric approximation to the PDE equation. The default values are: $\lambda_1 = 1$, $\lambda_2 = 1$, $\mu = 1.0$, $\alpha = 0.0$, $h = 1.0$ and $\Delta t = 0.1$. The number of maximum iterations has been set to 200.

Geometric, edge-based region contours For this discriminator, the energy terms are weighed by the parameters λ , μ and ν . Additionally, we have the time step parameter Δt for the PDE equation. Parameter ν is of special importance because it controls whether the either shrinks or expands towards nearby edges. For these parameters, we have used the proposed default values [41], $\lambda = 5.0$, $\mu = 0.04$, $\nu = 1.5$, $\Delta t = 5$; with a maximum of 200 iterations.

Boundary pixel color contrast detector This detector has two parameters: length of the segment L and size of the neighboring window M , as detailed in section 4.3. In these experiments, we have used $L = 5$, $M = 3$.

6.2.4.2 Multi-feature approaches

Naive Bayes (NB) The fusion scheme of the base system, and the NB classifier are probabilistic classifiers, that is, their decision is based on the probabilistic models of the input data. In the base system, a Gaussian distribution is employed. For the NB classifier, we have deter-

mined that modeling the class conditionals with normal distributions may result in a reduction in accuracy, as the we have observed the input features for all classifier do not follow a normal distribution. This has been confirmed by performing the Lilliefors test [56] on the scores obtained from all detectors, for each class. For this reason, we have decided to employ Kernel Density Estimators (KDE) for the probabilistic modeling of the NB classifier.

Support Vector Machine (SVM) For the SVM classifier we employ the publicly available `libsvm` library [57]. A radial basis kernel function is used to map input data into the feature space, and the soft-margin optimization problem is solved.. Default values for all parameters are used, as detailed in the documentation.

K-Nearest Neighbor (KNN) For the KNN classifier, we have determined the k parameter heuristically, obtaining best results when $k=9$.

6.3 Single-feature evaluation

In this section, we evaluate the performance of the discriminators described in chapter 4: parametric edge-based active contours (PE), geometric region-based active contours (GR), geometric edge-based active contours (GE) and pixel-color contrast (PCC). Moreover, we compare them against the discriminators of the base system: Color Histogram (CHIST), High-Gradient (GH), Low-Gradient (GL) and Combined Gradient (GRD). Results are presented for the annotated and real data. Additionally, a computational cost comparison is presented.

6.3.1 Annotated data

A summary of the results is shown in Table 6.3 and Figure 6.3. The observed results of each discriminator are discussed as follows.

Color histogram discriminator (CHIST) Given accurate foreground masks, it performs generally well on all categories. As mentioned in chapter 3, CHIST computes the color histogram

Table 6.3: Single-feature discrimination results for annotated data. (Key. ACC:accuracy, AUC:Area Under Curve).

(a) Base system single-feature discriminators

Discrim.	C1		C2		C3		ALL	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
CHIST	.864±.031	.933±.020	.819±.031	.959±.016	.947±.035	1.0±.0	.870±.018	.941±.013
GH	.892±.026	.973±.009	.837±.040	.925±.033	.804±.058	.932±.038	.852±.018	.940±.015
GL	.965±.016	.997±.002	.911±.013	.961±.012	.872±.044	.990±.009	.923±.010	.982±.004
GRD	.955±.012	.995±.002	.883±.040	.960±.015	.871±.046	.985±.013	.910±.019	.977±.006

(b) Proposed single-feature discriminators

Discrim.	C1		C2		C3		ALL	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
PE	.999±.002	1.0±.0	.979±.014	.993±.006	.978±.018	.995±.007	.988±.008	.996±.002
GR	.962±.014	.996±.002	.920±.037	.971±.016	.942±.025	.987±.010	.943±.012	.985±.006
GE	1.0±.0	1.0±.0	.999±.003	1.0±.0	1.0±.0	1.0±.0	1.0±.001	1.0±.0
PCC	1.0±.0	1.0±.0	.991±.008	.999±.001	1.0±.0	1.0±.0	.997±.002	1.0±.0

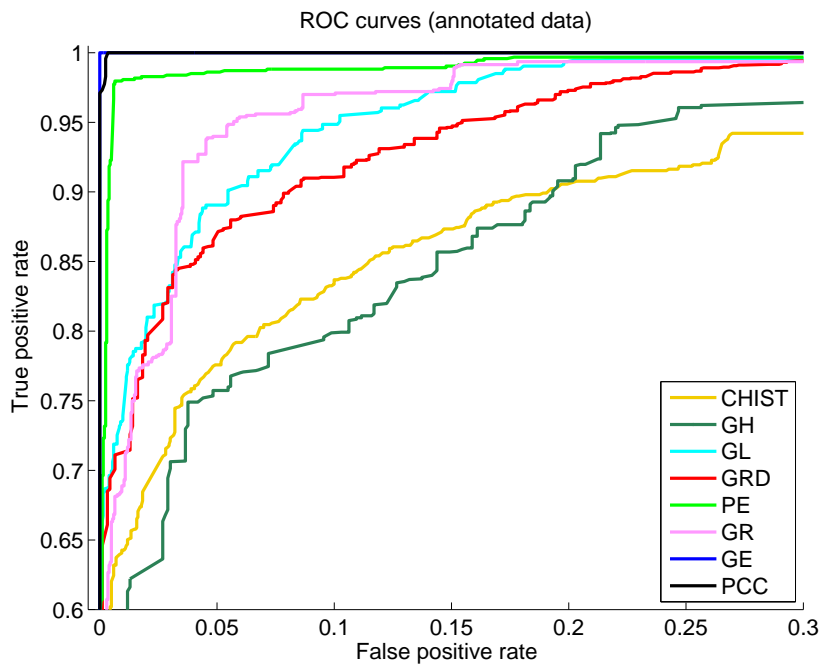


Figure 6.3: ROC analysis for single-feature discrimination on annotated data.

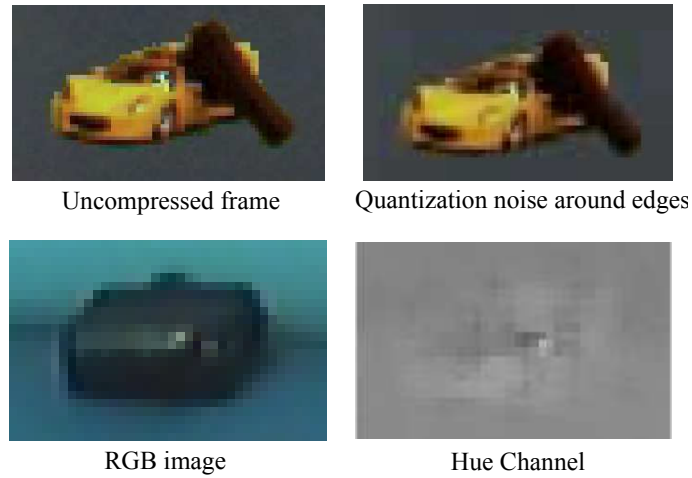


Figure 6.4: Examples of problematic scenarios for the color histogram discriminator of the base system (CHIST).

distances between the two areas delimited by the foreground mask inside the bounding box. Hence, its discrimination relies completely on the ability of the foreground mask to separate the object from its surrounding background. Some problems have been observed due to quantization noise introduced by the video compression scheme. In some cases, it causes color information to “leak” beyond object boundaries. We have seen that this problem affects smaller objects, which explains why it performs poorly on category 1 (low complexity); as this category includes mostly small objects. Additional problems have been observed due to the fact that the color histograms are only computed on the Hue channel of the HSV color space. The effectiveness is reduced when there is not enough contrast between the object and the background in this channel. Figure 6.4 shows examples of objects affected by noise around the edges after compression (first row), and low contrast in the Hue channel (second row).

Gradient discriminator For GH, a contour pixel satisfies the condition if *any* pixel in its neighborhood is above a predefined threshold. For GL, the condition is satisfied if the *average gradient value* inside the neighborhood is below a threshold. As observed in Table 6.3, GL shows very good results for blobs of both classes. By using a window-based approach instead pixel-based one, GL has shown to provide better results than GH. Due to the imbalance of the obtained scores for GH (Figure 6.5), the combined score (GRD) does not improve the results

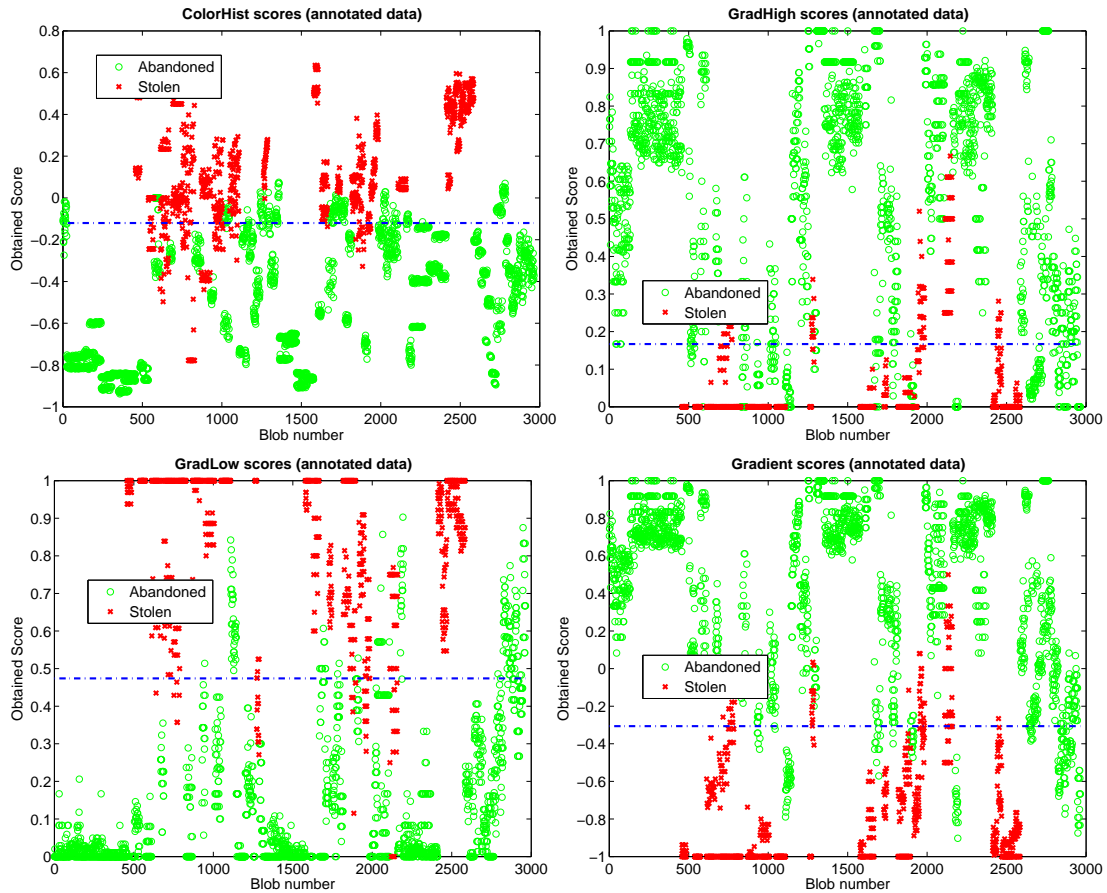


Figure 6.5: Scores of the single-feature discriminators of the base system for the annotated data of GL. Both discriminators, however, are affected by the presence of strong edges near the object boundaries that can be attributed to the background, as this causes the discriminators to produce a score that would correspond to objects from the opposite class. Figure 6.6 shows an example of a wrong detection due to a background with strong edges.

Active contours discriminators Tested on annotated sequences, the three proposed active contour discriminators outperform the base-system ones. PE has been effective in all cases, attaining perfect classification for all blobs in the annotated data set. As can be seen in Figure 6.7, the decision thresholds used on the obtained scores stay close to zero for the three discriminators. We have observed that for the same blob across different frames (same foreground mask), GR and GE produce similar scores. In the scores of Figure 6.7, we notice how scores for

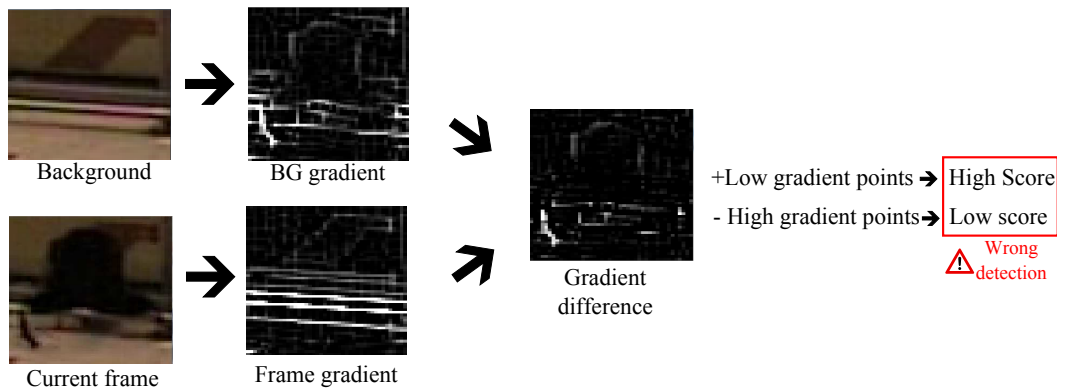


Figure 6.6: Examples of problematic scenarios for the gradient-based discriminators of the base system (GH, GL and GRD).

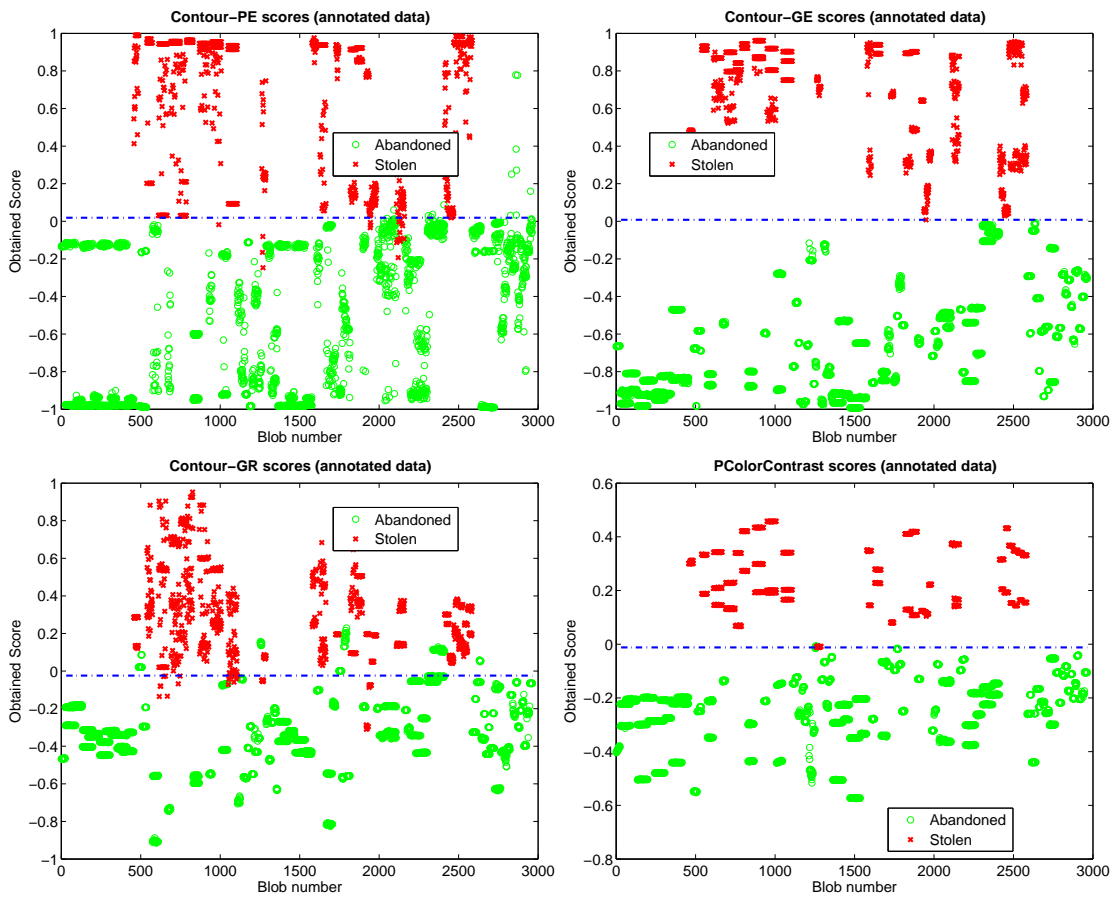


Figure 6.7: Scores of the proposed single-feature discriminators for the annotated data

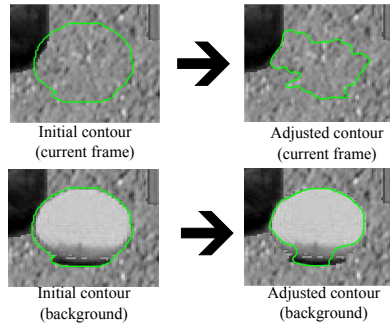


Figure 6.8: Example of contour problematic adjustments for a stolen object using the GE discriminator.

the same blob in different frames tend to group around the same score value. For static objects, small changes between frames are attributed to noise (camera noise, compression noise...). This has not been observed for the PE discriminator, which seems to be more vulnerable to noise. The same can be inferred about the base-system discriminators, as scores for the same blob at different times tend to scatter across the entire range. Like with the gradient discriminators, the edge-based discriminators PE and GR produce uncertain scores (near the 0.0 value) in those cases with strong edges in the background. An example of adjustments on a highly-textured background is shown in Figure 6.8.

Pixel color contrast discriminator PCC has proven very robust in situations in which the other discriminators have shown weaknesses. The grouping observed in the scores graph (Figure 6.7) suggests that all effects due to noise are mitigated. Noise is eliminated from the final measure by averaging the color values inside the $M \times M$ neighborhood. The most important drawback, however, is that the obtained score is not very reliable on very small objects (of the order of parameter L).

6.3.2 Real data

A summary of the results is shown in Table 6.4 and Figure 6.9. The observed results for each discriminator are discussed as follows.

Table 6.4: Single-feature discrimination results for real data. (Key. ACC:accuracy, AUC:Area Under Curve).

(a) Base system discriminators

Discrim.	C1		C2		C3		ALL	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
CHIST	.755±.035	.845±.026	.712±.055	.858±.036	.848±.025	.910±.019	.777±.024	.856±.019
GH	.918±.024	.964±.017	.791±.039	.867±.025	.743±.041	.792±.030	.821±.022	.879±.014
GL	.970±.020	.995±.004	.817±.029	.902±.032	.825±.030	.919±.026	.877±.016	.947±.013
GRD	.963±.015	.995±.003	.815±.025	.907±.027	.841±.025	.915±.023	.879±.014	.944±.011

(b) Proposed discriminators

Discrim.	C1		C2		C3		ALL	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
PE	.882±.026	.949±.015	.768±.031	.821±.030	.806±.027	.905±.022	.824±.020	.90±.016
GR	.857±.018	.945±.010	.80±.031	.839±.037	.748±.042	.845±.027	.803±.020	.882±.015
GE	.960±.010	.996±.002	.952±.027	.984±.012	.929±.016	.954±.009	.947±.011	.981±.004
PCC	.967±.014	1.0±.0	.943±.013	.987±.006	.951±.013	.996±.002	.954±.009	.994±.001

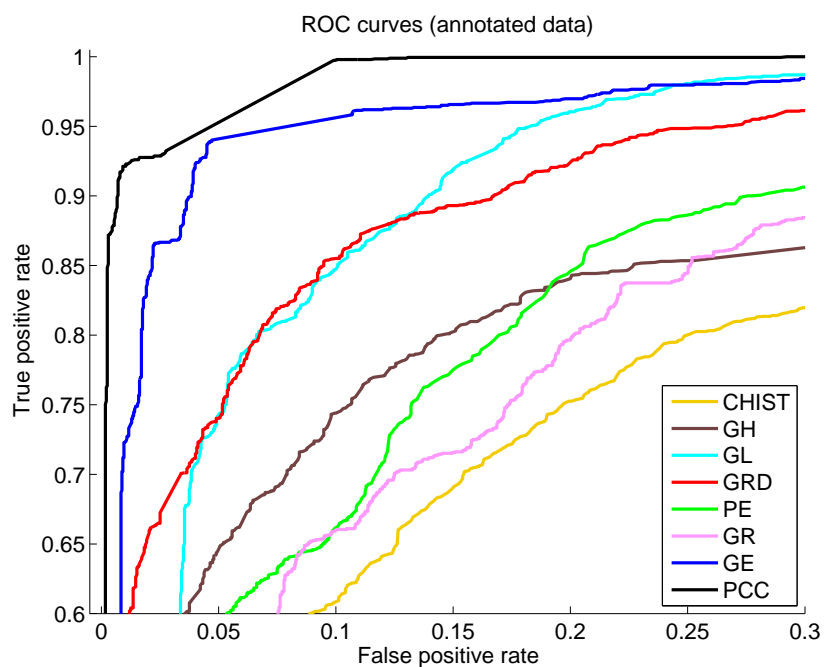


Figure 6.9: ROC analysis for single-feature discrimination on real data.

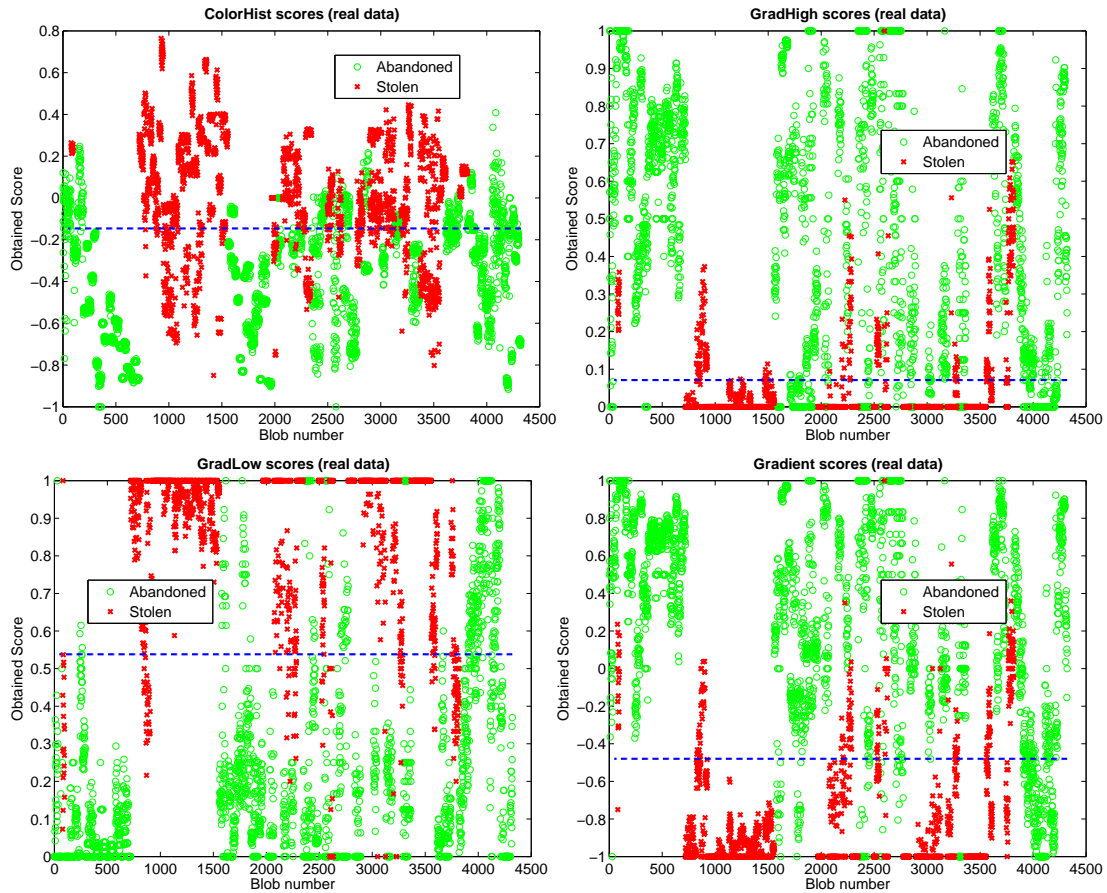


Figure 6.10: Scores of the single-feature discriminators of the base system for the real data

Color histogram discriminator For the CHIST, we can see a decrease in accuracy of roughly 10% as compared to Table 6.3. This is explained by the fact that the CHIST completely relies on correct segmentation to obtain the correct histograms. If region R1 covers part of the background, or if R2 contains part of the object (in the current frame), the three extracted histograms may be too similar, and the discriminator is incapable of producing a discriminative measure.

Gradient discriminators The reduction in accuracy for the gradient discriminators is not as significant as with other discriminators. This can be primarily attributed to the contour adjustment operation applied to the initial extracted contour, as it drives the contour to match the actual boundaries, as well as the small neighboring window in which the analysis is performed.

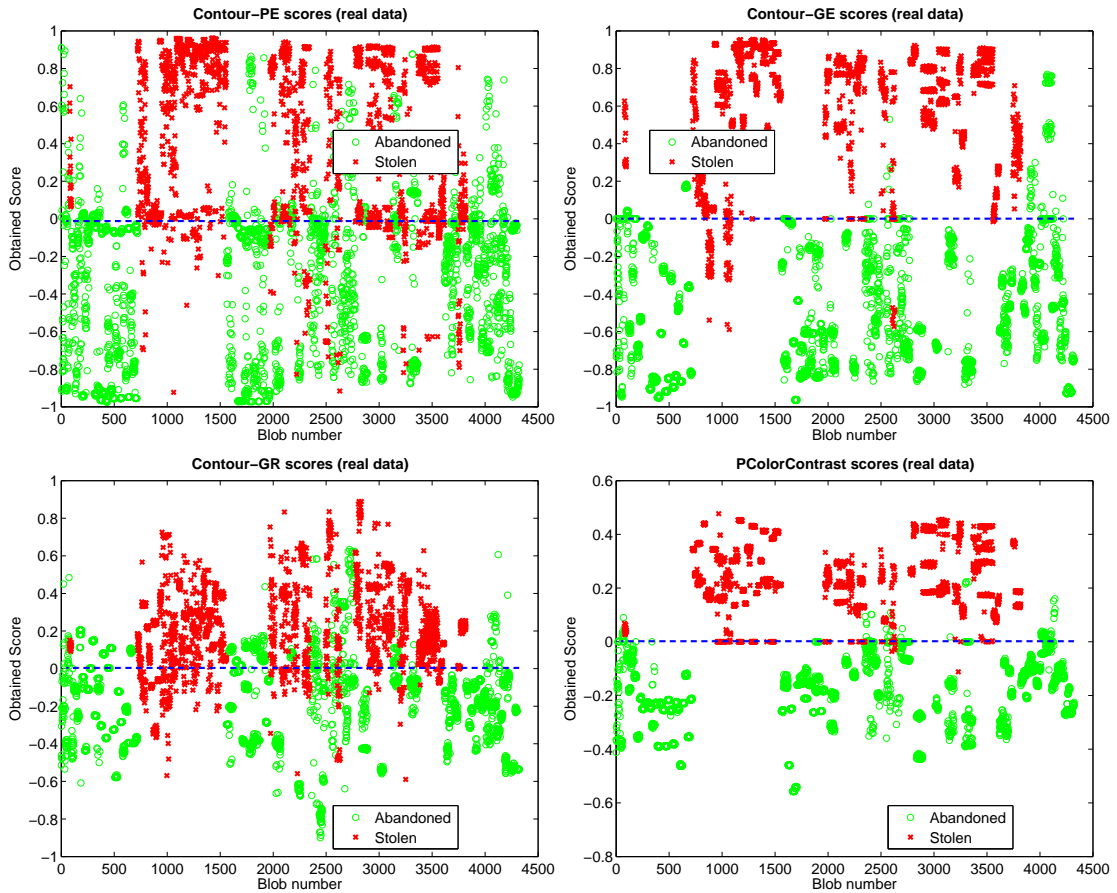


Figure 6.11: Scores of the proposed single-feature discriminators for the real data

We can conclude that these discriminators are less affected by imprecise segmentation with a $\sim 5\%$ decrease.

Active contours discriminators For real data, we can observe that the accuracy of the PE and GR discriminators is greatly reduced due to incorrect segmentation. As explained in section 4.2.2, the final contour adjustment obtained by the PE algorithm greatly depends on where the contour is initialized. This is further confirmed by analyzing the score results (Figure 6.11), noting the scatter across the entire range. A similar problem can be observed for the GR discriminator. Among all evaluated discriminators, the GR shows the steepest decrease in accuracy when evaluated on the *real dataset*. In contrast, the GE has shown robustness against incorrect foreground segmentation. We can attribute this to the fact that the adjusted contour

generally tends to shrink unless the object boundaries are nearby. This is usually the case, even for inaccurate foreground masks. We have observed, however, that if the foreground mask is initialized inside a uniformly colored object, the contour tends to shrink or even disappear. When this happens in both the current frame and the background the resulting score is very close to 0.0. This problem is more evident for smaller blobs, as the contours tend to quickly disappear in both images. This explains the presence of scores with value 0.0 (Figure 6.11), that in the vast majority of cases correspond with very small foreground masks due to over-segmentation.

Pixel color contrast discriminator This discriminator is the least affected incorrect segmentation. We could explain this by taking into consideration that the measures are taken at a distance from the corresponding contour pixel (parameter L), and averaged inside a small window (parameter M), which leave the discriminator a margin to overcome segmentation inaccuracies. As mentioned before, this discriminator is unable to produce a score for very small blobs, which explains the number of blobs with score 0.0 shown in Figure 6.11.

6.3.3 Computational cost comparison

Table 6.5 shows the obtained computational costs of all the evaluated discriminators. Maximum and minimum values correspond to large and small objects, respectively. As it is shown, base system discriminators have a lower computational cost than the proposed active contours ones. This is due to the complexity of the employed active contours algorithms, and the fact that the adjustments are performed on both the current and background frame images, as opposed to a single adjustment in the gradient difference image for the gradient discriminators. Among all evaluated discriminators, PCC has shown the lowest computational cost, improving existing approaches, due to the simplicity of the performed analysis (average color contrast).

6.4 Multi-feature evaluation

In this section, we evaluate the performance of the discriminators described in chapter 5: Naive Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Moreover, we

Table 6.5: Comparative computational cost

	Time (ms)							
	CHIST	GH	GL	GRD	PE	GR *	GE	PCC
Min.	5.67	0.15	0.14	0.29	2.30	96.89	20.78	0.19
Max.	44.57	133.80	255.78	354.33	1401.80	8207.50	1187.10	8.66
Avg.	23.23	28.35	57.14	84.61	234.47	901.40	246.39	1.78

Note: times for the GR discriminator are given for the Matlab implementation

compare their performance against the fusion method of the base system (FUS).

6.4.1 Feature selection

For the multi-feature classification stage, we must select a subset of scores from discriminators to train all classifiers. Feature selection is performed mainly for two reasons: to increase the classification’s accuracy and to reduce the computational cost of the training and classification phases. This is achieved by selecting only relevant features (with low uncertainty) to increase accuracy, and removing the redundant ones (to decrease the dimensionality). Dimensionality reduction based on techniques such as *Principal Component Analysis* (PCA) that analyze the correlation between features may reduce redundant information, but they do not take into account the relation between attributes and classes, and fail to produce discriminative features [58]. The obtained correlation coefficients for the scores produced by all discriminators are shown in Table 6.6. As it can be seen, the GRD discriminator has a high-valued correlation with the GH and GL detectors. This is explained by the fact that the GRD score is merely a combination of the other two scores, and is therefore highly redundant.

We have decided to train the the described classification schemes with the scores produced by the following discriminators: CHIST, GH, GL, PE and GE and PCC. The GRD discriminator is discarded for redundancy reasons, as it is merely a combination of other two discriminators that produce more relevant scores. The active contours GR discriminator is discarded as it does not provide a sufficiently discriminative measure (low AUC) as the other discriminators.

Due to the reduced amount of selected features, we have deemed it unnecessary to perform further reductions of dimensionality. The study and application of different feature selection algorithms is beyond the scope of this work.

Table 6.6: Correlation coefficients between discriminators scores

	CHIST	GH	GL	GRD	PE	GR	GE	PCC
CHIST	1.00	-0.37	0.46	-0.44	0.49	0.42	0.58	0.61
GH	-0.37	1.00	-0.81	0.94	-0.49	-0.33	-0.59	-0.63
GL	0.46	-0.81	1.00	-0.96	0.63	0.49	0.80	0.81
GRD	-0.44	0.94	-0.96	1.00	-0.59	-0.44	-0.74	-0.77
PE	0.49	-0.49	0.63	-0.59	1.00	0.52	0.71	0.66
GR	0.42	-0.33	0.49	-0.44	0.52	1.00	0.64	0.60
GE	0.58	-0.59	0.80	-0.74	0.71	0.64	1.00	0.83
PCC	0.61	-0.63	0.81	-0.77	0.66	0.60	0.83	1.00

Table 6.7: Multifeature-feature discrimination results. (Key. ACC:accuracy).

(a) Results for annotated data

	C1	C2	C3	ALL
Discrim.	ACC	ACC	ACC	ACC
NB	1.0±.0	1.0±.0	1.0±.0	1.0±.0
SVM	1.0±.0	1.0±.0	1.0±.0	1.0±.0
KNN	1.0±.0	1.0±.0	1.0±.0	1.0±.0
FUS	.902±.030	.818±.051	.805±.037	.850±.022

(b) Results for real data

	C1	C2	C3	ALL
Discrim.	ACC	ACC	ACC	ACC
NB	.997±.005	.961±.011	.952±.018	.972±.008
SVM	.994±.007	.982±.011	.946±.019	.974±.006
KNN	.997±.004	.978±.010	.983±.008	.987±.003
FUS	.737±.052	.556±.034	.665±.052	.662±.025

6.4.2 Annotated data

As can be seen in Table 6.7a, the described classification techniques are able to correctly classify 100% of samples in this data set. This is to be expected, as the GE discriminator already achieves perfect classification on its own. The FUS classifier, however, fails to improve the accuracy of any of the other discriminators.

6.4.3 Real data

Table 6.7b shows the accuracy measures obtained on the *real dataset*. The performance of NB and SVM is similar, improving the accuracy of the best discriminator (PCC) by about 3%. KNN, in spite its simplicity, provides the better results, correctly classifying over 98% of samples.

Chapter 7

Conclusions and future work

7.1 Summary of work

In this project, a comprehensive study has been carried out for abandoned and stolen objects detection; focusing our attention in the discrimination of stationary regions between these two events. This task aims to determine whether stationary foreground objects are due to abandoned or stolen objects. Few approaches in the literature deal with the discrimination problem directly; among them, we can distinguish between color-based and edge-based detectors depending on the type of extracted information.

The contributions of this work can be summarized as follows:

- **Design and implementation of novel single-feature discrimination techniques.**

A generic approach based on active contours have been defined for the discrimination task. It measures the difference between the adjustments performed on the background and current images. For this analysis, three relevant active contours techniques have been studied and employed. Hence, three approaches have been proposed based on the active contour technique. Later on, a color-based approach has been presented, that computes the average pixel value contrast (along the contour of the object under analysis) between the detected stationary region and its surroundings, in all color channels.

- **Study of different classification methods for multi-feature discrimination.** Three

widely employed machine learning techniques have been considered for multi-feature discrimination task, with the goal of improving efficiency by using information from different discriminators. In particular, Naive Bayes, Support Vector Machine and K-Nearest Neighbor techniques have been selected for the multi-feature discriminator.

- **Elaboration of two datasets for abandoned and stolen object detection.** Two datasets have been created for the evaluation of the discrimination task using the same content set. It can be also used for the assessment of complete systems for abandoned and stolen object detection. The content set is composed of selected sequences from publicly available video datasets. The first dataset contains manual annotations of these sequences. For the second dataset, a procedure to automatically generate foreground masks using the video analysis system available at the VPU-Lab has been developed for acquiring real data. This procedure extracts information from metadata files and automatically obtains the desired information (masks, location,...) required for discrimination evaluation.
- **Evaluation of existing and proposed discrimination approaches on the two datasets.** We evaluate the performance of all discriminators on annotated data (ideal segmentation) and real data (coarse segmentation). The *annotated dataset* allows us to evaluate the discriminative power of all approaches, enabling us to determine the scenarios in which the discriminators show weaknesses. In realistic settings, we expect the foreground masks to be imprecise as many challenges are present for the background segmentation stage. With the *real dataset*, we are able to assess the performance of the different approaches in real scenarios, while observing how well they cope with imprecise foreground segmentation.

7.2 Conclusions

By evaluating the performance of existing and proposed approaches on annotated sequences, we have been able to identify key issues affecting the final discrimination. For the existing color-based approach of the base system, we have concluded that it is particularly affected by

noise, as the use of only one color channel has proven to be insufficient. The edge-based gradient discriminators have shown poor performance on scenarios with highly textured backgrounds that present high gradient information. In contrast, the four proposed approaches (three based on active contours and the one based on color contrast at pixel level) have shown robustness in these situations showing an accuracy close to 100% in most of the categories.

For the real dataset, we have observed a decrease in performance in all cases (as expected), as all discriminators depend on the accuracy of the foreground mask to extract the desired features. In particular, color-based approach of the base system has shown a noticeable reduction of the accuracy when dealing with real data. On the other hand, two of the proposed methods (*Geometric edge-based active contours* and *Pixel Color Contrast*) demonstrated high robustness against non-accurate foreground segmentation, achieving excellent accuracy results on the *real dataset*. The use of the active contour adjustment introduces an inherent ability to deal with imperfect data. Furthermore, the combination of features from different discriminators increased the final accuracy results. In our results, over 98% of all samples in the *real dataset* have been correctly classified.

In conclusion, we can affirm that the proposed discriminators are suitable for their integration into a video-analysis system for detecting abandoned and stolen objects, as they have shown to be less dependent on the accuracy of the foreground mask produced by preceding processing stages.

7.3 Future Work

In this project, we have identified key issues that affect the the discrimination problem. The following lines of research can be considered:

- Multi-feature schemes have demonstrated an efficient way to combine information for increasing the overall accuracy. However, these schemes require the use of multiple discrimination techniques. This introduces a high computational cost in the system that reduces its ability to operate in real time. For this reason, more efforts should be put towards

developing procedures for determining when the discriminators are operating as expected. In order words, we suggest to infer which discriminators would provide the most discriminative measure and avoid to use (and execute) the worst ones. For example, the gradient detectors have shown weaknesses when the region outside the foreground mask contains strong edges in both the background frame and the image. In this situation, a color based detector would be able to provide better results. Within the multi-feature framework, can be used for two objectives: to reduce the overall processing time and to increase the confidence in the discrimination results. If further fusion schemes are used, better results can be obtained by selecting *only* those features we know are relevant for a particular object.

- A more exhaustive evaluation should be carried out in environments with dense crowds, where the presence of nearby objects may have an impact on the extracted features. For extending the current dataset, new challenging situations should be considered such as high complex backgrounds, multimodal backgrounds, high textured objects and different compression levels of the video sequence.
- As observed in the results of the performed experiments, the discrimination task obtained very high accuracy within the employed datasets. This fact suggest that the complexity of the abandoned and stolen object detection depends on the modules devoted to the extraction of the object of interest. In particular, we have noticed that the main issue to address is the *Stationary Object Detection* task as it presents many challenges that remain unsolved.

Bibliography

- [1] 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*, pages 132–139, sept. 2008. [xiii](#), [14](#), [15](#)
- [2] Alvaro Bayona, Juan Carlos SanMiguel, and Jose M. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:25–30, 2009. [xiii](#), [14](#), [15](#)
- [3] A.F. Otoom, H. Gunes, and M. Piccardi. Feature extraction techniques for abandoned object classification in video surveillance. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1368–1371, oct. 2008. [xiii](#), [2](#), [16](#), [117](#)
- [4] Lauro Snidaro, Ingrid Visentini, and Gian Foresti. Data fusion in modern surveillance. In Paolo Remagnino, Dorothy Monekosso, and Lakhmi Jain, editors, *Innovations in Defence Support Systems 3*, volume 336 of *Studies in Computational Intelligence*, pages 1–21. Springer Berlin - Heidelberg, 2011. [xiv](#), [52](#), [54](#)
- [5] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings -*, 152(2):192–204, april 2005. [1](#)
- [6] Hui Kong, J.-Y. Audibert, and J. Ponce. Detecting abandoned objects with a moving camera. *Image Processing, IEEE Transactions on*, 19(8):2201–2210, aug. 2010. [2](#), [17](#), [116](#)
- [7] A. Bayona, J.C. SanMiguel, and J.M. Martinez. Stationary foreground detection using background subtraction and temporal difference in video surveillance. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4657–4660, sept. 2010. [2](#), [117](#)

- [8] J.C. San Miguel and J.M. Martinez. Robust unattended and stolen object detection by fusing simple algorithms. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pages 18 –25, sept. 2008. [2](#), [18](#), [23](#), [35](#), [37](#), [117](#)
- [9] Sen ching S. Cheung and Rika Kamath. Robust background subtraction with foreground validation for urban traffic video. In *Proc. Video Communications and Image Processing, SPIE Electronic Imaging*, January 2004. [10](#)
- [10] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II-302 – II-309 Vol.2, june-2 july 2004. [10](#)
- [11] Marco Cristani, Michela Farenzena, Domenico Bloisi, and Vittorio Murino. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP J. Adv. Signal Process*, 2010:43:1–43:24, February 2010. [11](#), [12](#)
- [12] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780 –785, jul 1997. [11](#)
- [13] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 2 vol. (xxiii+637+663), 1999. [11](#)
- [14] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In David Vernon, editor, *Computer Vision ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767. Springer Berlin / Heidelberg, 2000. [11](#)
- [15] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831 –843, aug 2000. [11](#)
- [16] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. *Computer Vision, IEEE International Conference on*, 1:255, 1999. [12](#)
- [17] S. Guler and M. K. Farrow. Abandoned object detection in crowded places. In *IEEE CVPR*, pages 18–23, New York City, NY, June 2006. [14](#)

- [18] R. Mathew, Zhenghua Yu, and Jian Zhang. Detecting new stable objects in surveillance video. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, pages 1 –4, 30 2005–nov. 2 2005. 14
- [19] V. Fernandez-Carbajales, M.A. Garcia, and J.M. Martinez. Robust people detection by fusion of evidence from multiple methods. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, pages 55 –58, may 2008. 15
- [20] Mohamed Hussein, Wael Abd-Almageed, Yang Ran, and Larry Davis. Real-time human detection, tracking, and verification in uncontrolled camera motion environments. *Computer Vision Systems, International Conference on*, 0:41, 2006. 15
- [21] Yu Huang and T.S. Huang. Model-based human body tracking. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 552 – 555 vol.1, 2002. 15
- [22] Roland Kehl and Luc Van Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 104(2-3):190 – 209, 2006. Special Issue on Modeling People: Vision-based understanding of a person’s shape, appearance, movement and behaviour. 15
- [23] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008. 15
- [24] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37:175–185, 2000. 10.1023/A:1008103604354. 16
- [25] Jing Ying Chang, Huei-Hung Liao, , and Liang-Gee Chen. Localized detection of abandoned luggage. In *EURASIP Journal on Advances in Signal Processing, vol. 2010*, 2010. 17
- [26] Medha Bhargava, Chia-Chih Chen, M. Ryoo, and J. Aggarwal. Detection of object abandonment using temporal logic. *Machine Vision and Applications*, 20:271–281, 2009. 10.1007/s00138-008-0181-8. 17
- [27] J. Connell, A.W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, and S. Pankanti. Detection and tracking in the ibm peoplevision system. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 2, pages 1403 –1406 Vol.2, june 2004. 17
- [28] P.L. Venetianer, Z. Zhang, W. Yin, and A.J. Lipton. Stationary target detection using the objectvideo surveillance system. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 242 –247, sept. 2007. 17

- [29] Deng-Yuan Huang Wu-Chih Hu and Wei-Hao Chen. Adaptive wide field-of-view surveillance based on an ip camera on a rotational platform for automatic detection of abandoned and removed objects. In *ICIC Express Letters*, volume 1, pages 45–50, 2010. 17
- [30] P. Spagnolo, A. Caroppo, M. Leo, T. Martirrigiano, and T. D’Orazio. An abandoned/removed objects detection algorithm and its evaluation on pets datasets. In *Video and Signal Based Surveillance, 2006. AVSS ’06. IEEE International Conference on*, page 17, nov. 2006. 17, 33
- [31] S. Ferrando, G. Gera, and C. Regazzoni. Classification of unattended and stolen objects in video-surveillance system. In *Video and Signal Based Surveillance, 2006. AVSS ’06. IEEE International Conference on*, page 21, nov. 2006. 18, 31, 32
- [32] Qiuji Li, Yaobin Mao, Zhiquan Wang, and Wenbo Xiang. Robust real-time detection of abandoned and removed objects. *Image and Graphics, International Conference on*, 0:156–161, 2009. 18
- [33] Jian Zhang Sijun Lu and David Dagan Feng. Detecting ghost and left objects in suverillance video. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 23, pages 1503–1525, 2009. 18
- [34] Y. Tian, R. Feris, H. Liu, A. Hampapur, and M. Sun. Robust detection of abandoned and removed objects in complex surveillance videos. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, PP(99):1–12, 2010. 18
- [35] Jianting Wen, Haifeng Gong, Xia Zhang, and Wenze Hu. Generative model for abandoned object detection. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 853–856, nov. 2009. 19
- [36] A. Cavallaro, O. Steiger, and T. Ebrahimi. Semantic video analysis for adaptive content delivery and automatic description. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(10):1200–1209, oct. 2005. 24
- [37] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4, oct. 2004. 25
- [38] Yong Shan, Fan Yang, and Runsheng Wang. Color space selection for moving shadow elimination. *Image and Graphics, International Conference on*, 0:496–501, 2007. 26
- [39] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1337–1342, 2003. 26

- [40] P. Salembier and J. Ruiz. On filters by reconstruction for size and motion simplification. In *Proc. of Int. Symposium in Mathematical Morphology*, pages 425–434, 2002. 27
- [41] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox. Level set evolution without re-initialization: A new variational formulation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:430–436, 2005. 38, 43, 45, 72
- [42] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):pp. 297–302, 1945. 39
- [43] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988. 10.1007/BF00133570. 41
- [44] E.R. Davies. *Machine Vision*. Morgan Kaufmann, 2005. 42
- [45] V. Caselles. Geometric models for active contours. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 9–12 vol.3, oct 1995. 43
- [46] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, feb 2001. 44
- [47] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math*, 42:577–685, 1989. 44
- [48] Ling Pi, Chaomin Shen, Fang Li, and Jinsong Fan. A variational formulation for segmenting desired objects in color images. *Image and Vision Computing*, 25(9):1414–1421, 2007. 47
- [49] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *Image Processing, IEEE Transactions on*, 10(10):1467–1475, oct 2001. 47
- [50] C.E. Erdem, B. Sankur, and A.M. Tekalp. Performance measures for video object segmentation and tracking. *Image Processing, IEEE Transactions on*, 13(7):937–951, july 2004. 48
- [51] Paolo Lombardi. A study on data fusion techniques for visual modules. Technical report, Computer Vision Laboratory, University of Pavia, 2002. 54
- [52] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI-01 workshop on Empirical Methods in AI*, pages 41–46, Sicily, Italy, 2001. 56
- [53] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, jul/aug 1998. 57

- [54] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21 – 27, jan 1967. **57**
- [55] Ludmila I. Kunchev. *Combining Pattern Classifiers: Methods and Algorithms*. WileyInterscience, 2004. **70**
- [56] H. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Assoc.*, 62:399–402, 1967. **73**
- [57] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. **73**
- [58] Jianzhong Fang and Guoping Qiu. Learning sample subspace with application to face detection. *Pattern Recognition, International Conference on*, 2:423–426, 2004. **83**

Appendix A

Support vector machines

The Support Vector Machine technique (SVM) provides a way to classify instances of data inputs based on a model obtained from a set of training samples. Each training sample \vec{x} (with n attributes) is represented as a point in the \mathbb{R}^n space (*data space*), and belongs to one of two categories $y_i \in \{+1, -1\}$. The goal is to build a decision function from the training data.

Linearly Separable case In the simplest scenario, we consider the samples in the data space to be *linearly separable*; that is, we can draw decision boundaries that separate the training samples in the data space. More formally, there exist infinite hyperplanes that can separate the training samples, as shown in figure [A.1](#) for the 2-dimensional case. An hyperplane is a set of points that satisfies the following equation:

$$\vec{w} \cdot \vec{x} + b = 0 \tag{A.1}$$

where \vec{w} is the normal vector to the hyperplane and $\frac{b}{\|\vec{w}\|}$ determines the offset from the origin. If an hyperplane separates the two categories, we can define the discriminant function f as follows:

$$f(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \tag{A.2}$$

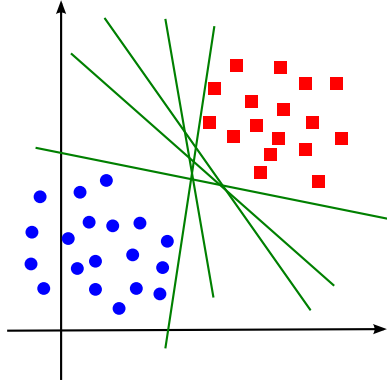


Figure A.1: Possible decision boundaries for linearly separable data in the 2-dimensional space

The goal is to find an hyperplane such that its distance to the nearest samples from both classes is maximized, in order to minimize the classification error. This is done by maximizing the margin around the separating hyperplane (*maximum margin hyperplane*). Assuming that all training data has a distance of at least 1 unit (along the \vec{w} direction) from the hyperplane, we define the margin as the distance between the two parallel hyperplanes:

$$\begin{aligned}\vec{w} \cdot \vec{x} + b &= +1 \\ \vec{w} \cdot \vec{x} + b &= -1\end{aligned}\tag{A.3}$$

The samples that lie on the margin hyperplanes are called *support vectors*. Geometrically, the distance between these two hyperplanes ρ is expressed as:

$$\rho = \frac{2}{\|\vec{w}\|}\tag{A.4}$$

An example of a maximum margin hyperplane is shown in figure A.2. The goal is to determine \vec{w} and b so that the distance ρ is maximized. Therefore, the problem consists on maximizing $\frac{1}{\|\vec{w}\|}$, or equivalently, minimizing $\|\vec{w}\|$, subject to the following constraint:

$$y_i (\vec{w} \cdot \vec{x} + b) \geq 1\tag{A.5}$$

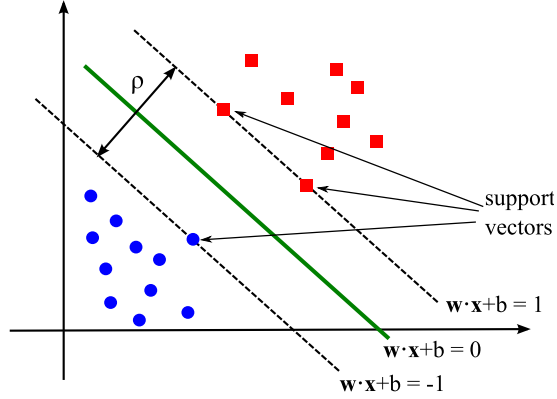


Figure A.2: Maximum margin hyperplane and support vectors

This formulation is equivalent to a quadratic optimization problem, and the solution for \vec{w} is a linear combination of all training points, weighed by the α_i solutions of the optimization problem:

$$\vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \quad (\text{A.6})$$

where n is the total number of training points; b is expressed as $b = y_k - \vec{w} \cdot \vec{x}_k$, for any \vec{x}_k such that $\alpha_k = 0$. For points other than the support vectors, the α_i coefficients equal zero. Therefore, the solution is uniquely identified by a linear combination of the support vectors. The discriminant function f is finally expressed as:

$$f(\vec{x}) = \text{sign} \left(b + \sum_{i=1}^n \alpha_i y_i (\vec{x}_i \cdot \vec{x}) \right) \quad (\text{A.7})$$

Non separable data: soft-margin classifier A linear SVM cannot produce a discriminant function if the training data is not linearly separable. When the data contains noise due to misclassified samples, the optimization problem constraints can be relaxed to account for noisy data. This can be done by adding a slack variables $\xi_i \geq 0$ to the constraint condition (eq. A.5), making the inequality easier to satisfy:

$$y_i (\vec{w} \cdot \vec{x} + b) \geq 1 - \xi_i \quad (\text{A.8})$$

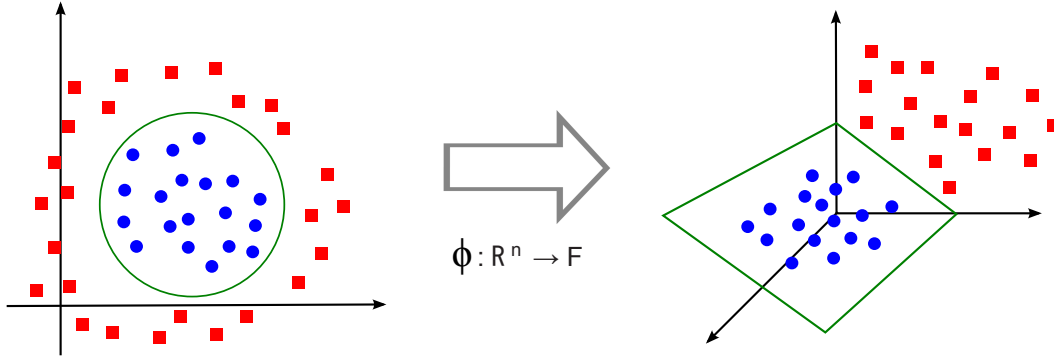


Figure A.3: Non-linear transformation example

This allows a point that is on the wrong side of the hyperplane (or too close to it) to satisfy the constraint, if the value of ξ_i is sufficiently large. The slack variable can be interpreted as the degree of misclassification for that sample. To avoid all points from satisfying the constraint, a cost parameter C is added to the formulation to penalize the sum of all ξ_i . The optimization problem becomes:

$$\min_{\vec{w}, \xi_i} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n \xi_i \quad (\text{A.9})$$

subject to the constraint expressed in eq. A.8. The cost parameter C allows us to control the constraints enforcement. The discriminant function is again:

$$f(\vec{x}) = \text{sign} \left(b + \sum_{i=1}^n \alpha_i y_i (\vec{x}_i \cdot \vec{x}) \right) \quad (\text{A.10})$$

where coefficients α_i are the solutions to the quadratic optimization problem, and b can be computed from the support vectors x_i such that $0 < \alpha_i < C$.

Non-linear case: Kernel trick In those cases in which a linear boundary is not enough to separate the data, a non-linear transformation $\phi : \mathbb{R}^n \rightarrow \mathcal{F}$ can be applied to map the input data to a high-dimensional space \mathcal{F} (*feature space*) in which the data is linearly separable, as depicted in figure A.3.

The discrimination can then be performed with a linear discriminant function (eq A.2), by

considering input vectors \vec{x} in the feature space $\phi(\vec{x})$. However, this operation can be computationally costly when the feature space is high dimensional. Since the discriminant function relies only on the inner products of vectors, it is not strictly necessary to represent each input vector in the feature space, as long as the inner product can easily be computed. A kernel function is a function that computes the inner product in some high-dimensional space. This is referred to as the *kernel trick*, which allows us to compute the inner products in the feature space without explicitly mapping input points.

The kernel-trick allows us to formalize the non-linear problem as a quadratic optimization problem. For a given kernel function K , we would obtain a non-linear discriminant function of the form:

$$f(\vec{x}) = \text{sign} \left(b + \sum_{i=1}^n \alpha_i y_i K(\vec{x}_i, \vec{x}) \right) \quad (\text{A.11})$$

Appendix B

Automatic generation of foreground masks from video annotations

B.1 Introduction

Employing previously annotated data available at VPU-Lab, we have designed a module that extracts foreground masks from frames that contain events of interest. Annotated data was generated employing a tool from ViPER Toolkit¹. Analyzing this data, foreground masks are extracted from relevant frames around regions of interest.

Section [B.2](#) details the structure of the meta-data annotations, and the process employed to extract the foreground masks is explained in section [B.3](#).

B.2 ViPER-GT

ViPER-GT² is a tool for annotating videos with meta-data. Annotations are stored in separate XML files that follow the ViPER file format specification. The file contains a header in which a set of descriptors are predefined by the user. Descriptors contain information about specific events or objects in the scene, and are characterized by a set of attributes. In VPU-Lab's pre-

¹<http://viper-toolkit.sourceforge.net/>

²<http://viper-toolkit.sourceforge.net/products/gt/>

existing framework, four relevant event descriptors have been defined: `GetObject`, `PutObject`, `AbandonedObject`, `StolenObject`. The first two descriptors correspond to people placing and removing objects from the scene, respectively. Objects are considered abandoned or stolen when a certain time has passed after the `GetObject` and `PutObject` events. Each descriptor is defined by a unique ID and a time span (*frame span*), and four attributes as specified in table B.1. Attributes `Point` and `BoundingBox` allow us to completely locate the event in the corresponding frames.

Table B.1: Descriptor attributes

Attribute	Description
<code>Point</code>	x,y coordinates of the object's centroid
<code>BoundingBox</code>	bounding box coordinates (x,y) and dimensions (w,h)
<code>DetectionScore</code>	double value
<code>DetectionDecision</code>	Boolean value

The following is an XML sample code snippet for a stolen object:

```
<object framespan="266:328" id="0" name="StolenObject">
  <attribute name="Point">
    <data:point x="251" y="87"/>
  </attribute>
  <attribute name="BoundingBox">
    <data:bbbox height="24" width="22" x="240" y="74"/>
  </attribute>
  <attribute name="DetectionScore">
    <data:fvalue value="1.0"/>
  </attribute>
  <attribute name="DetectionDecision">
    <data:bvalue value="true"/>
  </attribute>
</object>
```

B.3 Foreground mask extraction

A meta file interpreter module has been developed in C++ to be integrated with the visual analysis system (Chapter 3), using the Xerces-C++ XML parser library³. The goal is to extract

³<http://xerces.apache.org/xerces-c/>

frames and their foreground based on the information provided by the meta-data file. The procedure is as follows: first, all instances of the `AbandonedObject` and `StolenObject` event descriptors are extracted from the XML meta file, along with their `BoundingBox` and `frame-span` attributes. For each extracted event, a C++ object is initialized with these attributes which is then stored in an array structure. For each of these objects, a processing mask is generated containing the bounding box rectangle. If more than one event occur in the same time span, the masks are combined. The processing masks, along with their associated frame spans, are then passed on to the Video Analysis system described in Chapter 3. The system will process the video sequence, and the *Foreground Segmentation module* (section 3.2) will only analyze those areas specified by the processing masks. If a frame contains an event of interest (i.e., its frame number is associated with a processing mask), the frame is extracted along with the foreground mask and the current background model. This procedure is detailed in figure B.1.

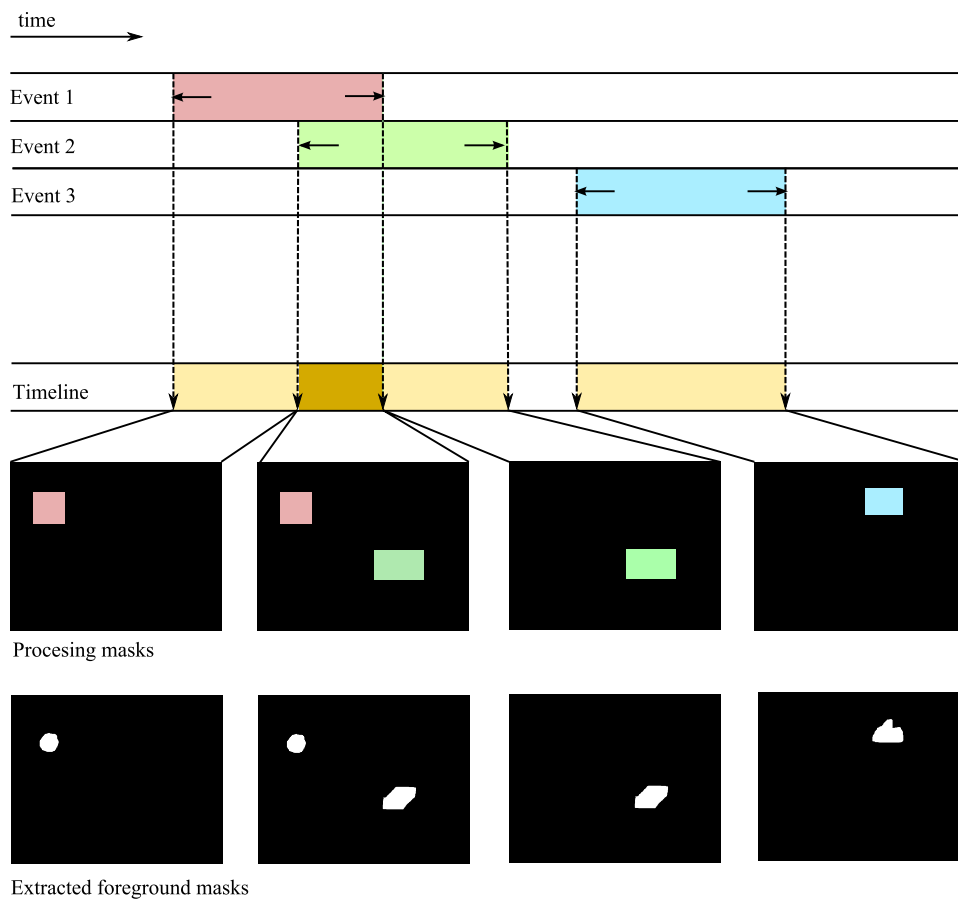


Figure B.1: Foreground masks extraction process

Appendix C

Publications

Part of this work has produced the following publication:

- Luis Caro, Juan Carlos San Miguel, José M. Martínez, “*Discrimination of abandoned and stolen object based on active contours*”, in Proceedings of the 2011 IEEE International Conference on Advanced Video and Signal based Surveillance, AVSS’2011, Klagenfurt, Austria, September 2011 (in press).

Conferencia IEEE AVSS: Clasificación ERA B

Discrimination of abandoned and stolen object based on active contours

Luis Caro Campos, Juan Carlos SanMiguel, José M. Martínez
Video Processing and Understanding Lab
Escuela Politécnica Superior, Universidad Autónoma de Madrid, SPAIN
E-mail: {Luis.Caro, Juancarlos.SanMiguel, JoseM.Martinez}@uam.es

Abstract

In this paper we propose an approach based on active contours to discriminate previously detected static foreground regions between abandoned and stolen. Firstly, the static foreground object contour is extracted. Then, an active contour adjustment is performed on the current and the background frames. Finally, similarities between the initial contour and the two adjustments are studied to decide whether the object is abandoned or stolen. Three different methods have been tested for this adjustment. Experimental results over a heterogeneous dataset show that the proposed method outperforms state-of-art approaches and provides a robust solution against non-accurate data (i.e., foreground static objects wrongly segmented) that is common in complex scenarios.

1. Introduction

Nowadays, there is a growing video surveillance demand as a consequence of the increasing global security concern which turned into a massive system deployment [1]. Traditionally, the monitoring task is performed by a human operator who has to simultaneously process a huge amount of visual data. Therefore, a significant efficiency reduction is expected. Automatic video interpretation was proposed as a solution to overcome this limitation.

In this situation, the detection of abandoned and stolen objects has become one of the most promising research topics especially in crowded environments such as train stations and shopping malls. It presents several challenges related to lighting conditions, object occlusions and object classification. Moreover, since these potential abandoned or stolen objects may have arbitrary shape and color, specific object recognition methods can not be applied.

Many methods have been proposed for abandoned and stolen object detection focusing on the stabilization of the image sequence from a moving camera [2], based on the static foreground region detection [3], based on blob classification (e.g., between people and object) [4] or dealing with the discrimination of the static regions between abandoned or stolen [5]. They yield acceptable

results in simple scenarios where high analysis accuracy is expected. However, this is not always valid for complex situations where a performance decrease may occur.

In this paper, we propose an approach based on active contours for the discrimination of static foreground objects between abandoned and stolen. It provides a robust solution against non-accurate segmentations of stationary objects in the analyzed video sequence. Starting from an initially extracted contour, active contour technique is applied to check whether the object contour is present in the current or in the background image and thus, decide if the object has been abandoned or stolen. Three different active contour methods have been tested based on edge and region information. Finally, this proposal is evaluated over a heterogeneous dataset with sequences with varying complexity and compared against state-of-art approaches.

This paper is structured as follows: section 2 discusses the related work; in sections 3 and 4 we overview the proposed discriminator and the selected active contours methods, experimental results are given in section 5, whilst section 6 ends the paper with the conclusions.

2. Related work

Abandoned and stolen object detection comprehends several tasks such as foreground extraction [2], static region analysis [3], blob tracking [6], blob classification (such as people [7] or baggage [8] recognition) and discrimination between abandoned or stolen objects.

Focusing on the discrimination of static foreground regions between abandoned and stolen, some of the existing approaches simplify this problem by assuming that only object insertions are allowed (i.e., detection of stolen objects is forbidden) and, therefore, all static objects are abandoned objects [2][6][8]. However, this assumption does not provide solutions for common artifacts generated by the background subtraction technique (e.g., ghosts) limiting the potential application of these proposals. On the other hand, few approaches have been proposed for this discrimination. Among existing literature, we can classify them according to the nature of the employed features into *edge-based*, *color-based* and *hybrid*.

Edge-based methods rely on inspecting the energy of the static region boundaries. It assumes that this energy is

high in the current frame for abandoned objects and low for stolen objects. For example, [9] proposed a system that analyzes the change in edge energy, and determines that an object has been added to the scene if the energy in the current frame is significantly higher or lower. Similarly, [10][11] proposed to use a Canny edge detector inside the bounding box of the static region in both the background and the current frame, and then they are compared to determine whether the object has been removed from or added to the scene. Moreover, [12] described a matching method to compare the results of the SUSAN edge detection in the current frame and foreground mask.

Color-based methods use the color information extracted from the internal and external regions delimited by the bounding box and the contour of the static region. In [7], two Bhattacharya distances are computed between the color histograms of the internal (in the current and background frames) and the external (in the background frame) regions. Discrimination is determined as the lowest distance. Similarly, a color-richness measure is proposed in [13] to count the number of colors (i.e., histogram bins above a threshold) and perform the same comparison as [7]. Moreover, [14] proposed to use image inpainting to reconstruct the hidden background and compare it against the external region using color histograms. Additionally, [15] compares color information within and outside the candidate static region by using segmentation techniques.

Hybrid discriminative methods combine the previous approaches. For example, [5] fused two algorithms based on edge and color by building probabilistic models for each algorithm in both cases (abandoned or stolen). Then, the decision is given by the maximum average probability of each case. Furthermore, [16] combined several features related with the edge energy, color contrast and shape into a classifier by using generative models for them.

In conclusion, the different techniques found in the recent literature use either edge or color information to perform the abandoned/stolen discrimination. Although these methods work well for simple scenarios, they have difficulties in complex scenarios as they do not consider the possibility of occlusions or complex backgrounds (e.g., high textured backgrounds). In addition, these methods rely on the precision of foreground object detection, and they may perform poorly in complex scenarios.

3. Discrimination based on active contours

A new approach based on active contours is proposed for discriminating static objects between abandoned and stolen. Let the initial contour of the static object, at time t , be defined as $C_t^I = \{p_1 \dots p_i \dots p_N\}$ and obtained as follows:

$$C_t^I = h(F_t, M_t), \quad (1)$$

where $h(\cdot)$ represents the contour extraction algorithm; F_t and M_t the current frame and foreground mask of the static object; p_i is the x,y coordinates of the i th contour

point and N is the number of contour points. In our approach, $h(\cdot)$ is a simple point-scanning of the result after applying the Canny edge detector to the M_t mask. This contour indicates the boundaries of the inserted (i.e., abandoned) or removed (i.e., stolen) of the scene object.

Then, a fitting process of the contour C_t^I is performed on the current and the background frame by using active contours. Thus, two adjusted contours are obtained.

$$C_t^{EF} = f(F_t, C_t^I), \quad (2)$$

$$C_t^{EB} = f(B_t, C_t^I), \quad (3)$$

where $f(\cdot)$ represents the contour adjustment method; F_t and B_t are the current and background frames; C_t^I is the initial contour; C_t^{EF} and C_t^{EB} are the adjusted contours in the current and background frames. For abandoned objects, the adjusted contour will be attracted to object boundaries in the current frame, and thus it will be similar to the initial contour. Conversely, the contour is expected to be deformed in the adjustment using the background frame as there are no object boundaries. In most cases, this uncovered area does not have strong edges and the contour may shrink or disappear. For stolen objects, the adjustment result will be the opposite; it will be attracted in the background frame and deformed in the current frame.

After that, a similarity measure is defined to quantify the deformation of the initial contour. We have decided to use the Dice coefficient [17], which is defined as follows:

$$d(C_1, C_2) = \frac{2|A_1 \cap A_2|}{|A_1| + |A_2|} \quad (4)$$

where $C_{1,2}$ represent two contours, $|A_1 \cap A_2|$ is their spatial overlap (in pixels); $|A_1|$ and $|A_2|$ represent the area (in pixels) of each contour. Thus, we obtain two distances (d_t^F and d_t^B) from the comparison of the initial contour C_t^I with the adjusted contours C_t^{EF} and C_t^{EB} . The values of d_t^B will be close to 0.0 and 1.0 in case of, respectively, abandoned and stolen objects. Distance d_t^F will get opposite values. Afterwards, a score is obtained by combining both distances as follows:

$$s_t = d_t^F - d_t^B \quad (5)$$

Finally, the discrimination is performed by thresholding the final score s_t as follows:

$$D = \begin{cases} \text{abandoned} & \text{if } s_t > th \\ \text{stolen} & \text{if } s_t \leq th \end{cases} \quad (6)$$

where th is the threshold applied for taking the abandoned or stolen decision, and is obtained from a training sequence. Figure 1 shows examples of the contour adjustments for abandoned and stolen cases.

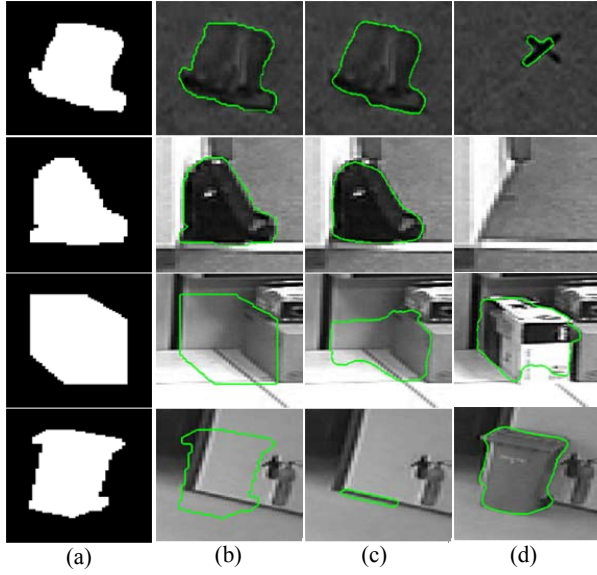


Figure 1: Examples of the proposed discrimination approach for abandoned (rows 1&2) and stolen (rows 3&4) objects. (a) Foreground mask, (b) initial contour and contour adjustment for the (c) current and (d) background frames.

4. Selected active contour algorithms

Up to this point, an approach for discriminating static foreground objects has been defined depending on a contour adjustment function $f(\cdot)$. According to [18], active contours methods can be either parametric or geometric considering whether their contour representation is explicit [19] or implicit (using level sets [18][20]). Moreover, we can further differentiate between methods based on boundaries or regions. We have tested the most representative methods in our approach.

4.1. Parametric active contours

We first consider the classic active contour model [19]. Starting from an initial contour $C = \{p_1 \dots p_N\}$, it iteratively minimizes a global energy function defined as:

$$E = \sum_{i=1}^N \alpha_i E_{cont} + \beta_i E_{curv} + \lambda_i E_{imag} \quad (7)$$

where N is the number of contour points, E_{cont} is the continuity energy, E_{curv} is the curvature (i.e., smoothness) energy, E_{imag} is the external energy (e.g., image edges) and $\alpha_i, \beta_i, \lambda_i \geq 0$ are the weights of each energy. These energies are defined as:

$$\begin{aligned} E_{cont} &= \|p_i - p_{i-1}\|^2 \\ E_{curv} &= \|p_{i-1} - 2p_i + p_{i+1}\|^2 \\ E_{imag} &= \text{gradient}(I_t) \end{aligned} \quad (8)$$

In our approach, we remove redundant edge data to compute the E_{imag} energy (i.e., edges that are present in current and background images). First, we have applied the *Canny* edge operator to each RGB channel of the current and the background frame. Then, the channel edge maps are combined using the logical operation OR. Finally, edges that appear in the background and current frame are removed to obtain the relevant edge data.

To achieve best results, parametric active contours algorithms such as this one require the initial contour to be initialized close to the true boundary. This holds true for abandoned objects, since the final contour is expected to be close to the initial contour. However, this limitation may be problematic when there are stolen objects. Although very simple to develop, traditional active contour models depend on the correct initialization.

4.2. Geometric active contours

Geometric active contour methods are proposed to solve the limitations of the parametric approaches by assuming an energy formulation invariant with respect to the curve parameterization. The contour is represented as the zero level set $\phi^{-1}(0) = \{(x, y) | \phi(x, y) = 0\}$ of a scalar function $\phi(x, y)$ usually referred as the level set function. The evolution of this function is guided by an energy minimizing process.

4.2.1 Geometric region-based active contours

A natural extension to overcome limitations of boundary analysis is the processing of regions. Among existing region-based approaches, we have selected the widely referenced work described in [20]. Derived from the Mumford-Shah energy functional for segmentation [20], piecewise constant functions are defined considering the intensity means of the different regions delimited by the contour. These cost functions are defined as follows:

$$E = \lambda_1 \int_{in(C)} |I(x, y) - m_{in}(C)|^2 dx dy \quad (9)$$

$$+ \lambda_2 \int_{out(C)} |I(x, y) - m_{out}(C)|^2 dx dy + \mu L(C) + \alpha A(C)$$

where $m_{in}(C)$ and $m_{out}(C)$ are the mean intensity value of the internal and external regions delimited by the contour; $L(C)$ is the length of the contour; $A(C)$ is the area of the contour; $\lambda_1, \lambda_2, \alpha$ and μ are fixed positive parameters. Then, a minimization problem is considered:

$$\min_{m_{in}, m_{out}, C} E(m_{in}, m_{out}, C) \quad (10)$$

To compute this optimization, level set optimization is jointly performed with the estimation of mean intensity values attempting to recover two regions such that

$|m_{in}(C) - m_{out}(C)|$ is maximum whilst assuring regularity properties for these regions. This model overcomes certain limitations of traditional parametric methods. It can detect objects with smooth boundaries (weak gradient) and it is more robust to noise. Moreover, contour initialization can be done at a higher distance from the real contour than the parametric approaches.

4.2.2 Geometric edge-based active contours

Extending the geometric methods based on level sets, [18] proposed an edge-based method to eliminate the re-initialization of the level set method that moves the zero level set from its original location extracting wrong contours. A new energy term is included to force the level set function to be close to a signed distance function. Thus, the proposed cost function to be minimized is:

$$\mathcal{E}(\phi) = \mathcal{E}_m(\phi) + \mu\mathcal{P}(\phi) \quad (11)$$

where $\mathcal{E}_m(\phi)$ is the external energy that adjusts the zero level set (i.e. contour) to the image boundaries; $\mathcal{P}(\phi)$ is the internal energy that penalizes the deviation of the level set function from a signed distance function; ϕ is the level set function and μ is a fixed positive parameter to control the influence of the internal energy. The external energy, $\mathcal{E}_m(\phi)$, is composed of two terms:

$$\mathcal{E}_m(\phi) = \lambda\mathcal{L}_g(\phi) + \nu\mathcal{F}_g(\phi) \quad (12)$$

where $\mathcal{L}_g(\phi)$ is the length of the zero level curve of ϕ ; $\mathcal{F}_g(\phi)$ is the speed of the curve evolution; g is a weight indicator function based on edges; $\lambda > 0$ and ν are the parameters to weight the energy contributions. Particularly, the parameter ν can be used to expand ($\nu > 0$) or shrink ($\nu < 0$) the evolution of the contour depending on whether the initial contour is placed outside or inside the object. We can take advantage of this behavior in the case of stolen objects, driving the motion of the curve and causing it to shrink. Abandoned objects will not be affected since the initial contour is already close to the object boundaries.

5. Experimental validation

5.1. Setup

We have carried out experiments using annotated and real data. The proposal has been implemented in C++ using the OpenCV image processing library¹. Tests were executed on a P-IV (3.0GHz) with 2GB RAM. Moreover, we compare the versions of our proposal (PE[19], GR[20] and GE[18],) against the most popular methods based on edge energy (ED[12]) and color-histogram (CH[7]).

¹<http://sourceforge.net/projects/opencvlibrary/>

Table 1: Test sequences categorization.

Category	Number of annotations		Background complexity
	Abandoned	Stolen	
C1	841	252	Low
C2	520	200	Medium
C3	671	480	High
Total	2032	932	-

For the experiments with annotated data, we have selected several sequences from the PETS2006², PETS2007³, AVSS2007⁴, CVSG⁵, VISOR⁶, CANDELA⁷ and WCAM⁸ public datasets. The annotations⁹ consist of the foreground binary masks of the abandoned or stolen objects. For performance evaluation, we have divided all the annotations into three categories according to the background complexity in terms of the presence of edges, multiple textures and objects belonging to the background. Table 1 summarizes the annotated content of the dataset. Finally, ROC curves are employed for the evaluation.

To find the optimum parameters of the active contour algorithms, we have proceeded as follows. For the PE algorithm, different combinations for parameters α , β and λ were considered ranging from 0.0 to 3.0 (with a step size of 0.01). Optimal achieved configuration was $\alpha = 0.97, \beta = 1.30, \lambda = 0.52$. In addition, the optimal size of the search window was determined to be 5. For the GR algorithm, we have used the proposed default values for the following variables: $\lambda_1 = 1, \lambda_2 = 1, \nu = 0, h = 1$, and $\Delta t = 0.1$. The α and μ were empirically defined ($\alpha = 1.0$ and $\mu = 0.05 \cdot 255^2$). For the GE algorithm, we have used a slightly higher time step to reduce the iterations needed ($\Delta t = 15$). For parameters λ , μ and ν ; best results were obtained with $\lambda = 5, \mu = 0.0133, \nu = 1.8$.

5.2. Evaluation with annotated data

A summary of the experiments is shown in Figure 2 and Table 2. As it can be observed, the proposed approach outperforms the state-of-art methods having higher AUC (Area under curve) values. The existence of complex backgrounds reduces the accuracy of the state-of-art methods as they assume low-textured background with little edge information. Our proposal is robust against this kind of complexity. Among the selected active contour methods, GE obtained the best results. For all methods, the contour is accurately adjusted when the object boundaries are present (i.e., current and background frames for, respectively, the abandoned and stolen cases). However,

²<http://www.cvg.rdg.ac.uk/PETS2006/>

³<http://www.cvg.rdg.ac.uk/PETS2007/>

⁴<http://www.avss2007.org/>

⁵<http://www-vpu.eps.uam.es/CVSG/>

⁶<http://www.openvisor.org/>

⁷<http://www.multitel.be/~va/candela/>

⁸<http://wcam.epfl.ch/>

⁹Available at <http://www-vpu.eps.uam.es/ASODds>

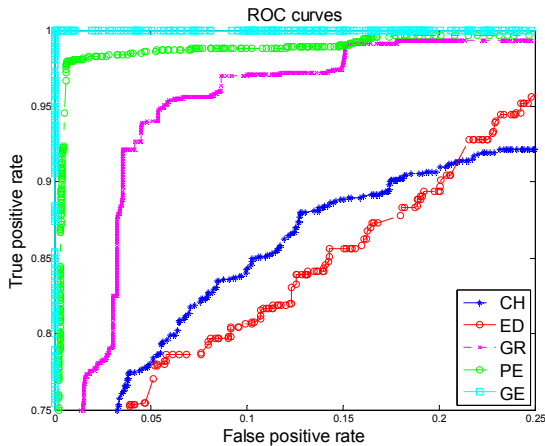


Figure 2: ROC curves for the discrimination of static objects between abandoned and stolen using the evaluation dataset.

their results differ for the adjustment without boundaries (i.e., background and current frames for, respectively, the abandoned and stolen cases). For PE, the deformations are not substantial, unless the initial mask belongs to a small object. For GR, sometimes the adjustment turns into a contour expansion limited by its bounding box size. Thus, the similarity measure (Eq. 4) exhibits a lower bound for expansion cases that depends on the contour area and its bounding box. To decrease this bound, the bounding box size can be extended. However, the computational cost of the adjustment is increased. In our experiments, the bounding box was increased by a factor of 1.5 as a trade-off between accuracy and time. GE overcomes this problem by allowing us to control the contour adjustment (expansion or shrinkage) with the selected value for parameter v . Thus, GE showed better results, as the contours shrink considerably or completely disappear.

Table 3 describes the computational cost results. Maximum and minimum values correspond to, respectively, large and small objects. As it is shown, state-of-art approaches have lower cost (as they perform simple operations) than the active contour ones. Among them, edge-based methods are faster than region-based methods as they consider local data (e.g., edges in a neighborhood) instead of global data (e.g., region statistics). Despite the higher cost of our approach, it should be noted that this analysis is not typically performed on a frame-by-frame basis and a slightly higher cost can be affordable.

5.3. Evaluation with real data

For the experiments with real data, we have selected some sequences of the above-mentioned datasets. A state-of-art abandoned/stolen object detection system has been implemented to get real data (i.e., masks of static foreground objects) [5]. Figure 3 shows the obtained contour adjustments using real data and Table 4 their corresponding scores (incorrect scores are marked using

Table 2: Comparative results for the ROC analysis

Category	Area Under Curve				
	PE	GR	GE	CH	ED
C1	1.0	0.99714	1.0	0.99754	0.97721
C2	0.99539	0.97788	1.0	0.95049	0.89105
C3	0.99240	0.97029	1.0	0.88167	0.94099
Total	0.99617	0.98475	1.0	0.94396	0.94

Table 3: Comparative computational cost (ms)

Computational Cost	Time (ms)				
	PE	GR	GE	CH	ED
Minimum	2.30	69.21	20.78	5.67	0.14
Maximum	1401.80	865.70	1187.10	44.57	130.87
Average	234.47	273.01	246.39	23.23	28.36

bold font). While PE and GR are able to perform correct detection in most cases, GE still produces more accurate adjustments, leading to higher class separability (i.e., difference between the scores of the abandoned and stolen cases). In addition, it can be seen that the Dice coefficient distance comparison between shapes produces satisfactory results even when the contour is attracted to nearby objects instead of shrinking or disappearing in those frames in which the object is not present (background frame for the abandoned case, and current frame for the stolen one), thus allowing the detectors to perform better in more complex scenes.

6. Conclusions

We have proposed a new approach for discriminating abandoned and stolen objects in video surveillance. It is based on adjusting the contour of candidate static foreground region to the current image and the reference background. Three different active contour strategies have been tested. Experiments on annotated and real data show that the proposed approach is significantly better than the state-of-art approaches. Geometric active contours based on edge information obtained the best results due to the more accurate adjustments obtained on images where the object is not present, making the contour disappear in many cases and adding robustness to the detector.

7. Acknowledgments

This work has been partially supported by the Consejería de Educación of the Comunidad de Madrid and by The European Social Fund.

References

- [1] Valera, M.; Velastin, S.; "Intelligent distributed surveillance systems: a review". IEE Proceedings - Vision, Image and Signal Processing, 152(2):192-204, April 2005.

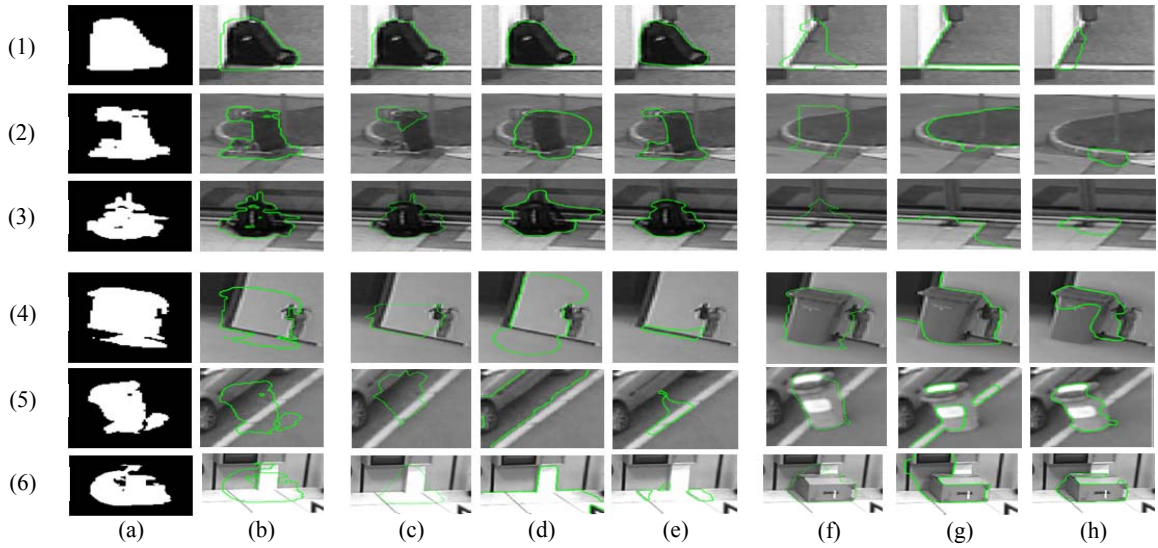


Figure 3: Real examples of abandoned (rows 1, 2 & 3) and stolen (rows 4, 5 & 6) objects. (a) Foreground mask, (b) initial contour and its adjustment in current frame ((c) PE, (d) GR and (e) GE) and background frame ((f) PE, (g) GR and (h) GE).

Table 4: Scores obtained for real examples

	PE			GR			GE			ED	CH
	d^F	d^B	s	d^F	d^B	s	d^F	d^B	s	s	s
(1)	0.90	0.66	0.24	0.84	0.31	0.52	0.84	0.33	0.50	0.82	-0.15
(2)	0.37	0.70	-0.33	0.62	0.37	0.24	0.85	0.31	0.54	0.52	-0.18
(3)	0.88	0.68	0.20	0.79	0.42	0.37	0.85	0.55	0.29	0.22	-0.34
(4)	0.65	0.88	-0.22	0.68	0.74	-0.05	0.17	0.63	-0.46	0.29	0.15
(5)	0.70	0.82	-0.12	0.41	0.46	-0.05	0.36	0.80	-0.43	0	-0.06
(6)	0.75	0.90	-0.14	0.49	0.67	-0.17	0.27	0.80	-0.53	0.10	0.12

[2] Kong, H.; Audibert, J.-Y.; Ponce, J., "Detecting Abandoned Objects With a Moving Camera," IEEE Trans. on Image Processing, 19(8):2201-2210, Aug. 2010.

[3] Bayona, A.; SanMiguel, J.; Martínez, J. "Stationary foreground detection using background subtraction and temporal difference in video surveillance", in Proc. of IEEE ICIP, p. 4657-4660, Sept. 2010.

[4] Otoom, A.F.; Gunes, H.; Piccardi, M., "Feature extraction techniques for abandoned object classification in video surveillance", Proc of IEEE ICIP, pp.1368-1371, Oct. 2008.

[5] San Miguel J.; Martinez, J. "Robust unattended and stolen object detection by fusing simple algorithms", in Proc. of IEEE AVSS, pp. 18-25, Sept. 2008.

[6] Chang, J.; H Liao, H.; Chen, L.; "Localized Detection of Abandoned Luggage", EURASIP Journal on Advances in Signal Processing, Article ID 675784, 9 pages, 2010.

[7] Ferrando, S.; Gera, G.; Regazzoni. C.; "Classification of Unattended and Stolen Objects in Video-Surveillance System", in Proc. of IEEE AVSS, pp. 21-27, Nov. 2006.

[8] Bhargava, M.; Chen, Ch.; Ryoo, M. and Aggarwal, J. "Detection of Object Abandonment using Temporal Logic", Mach. Vision and Applications, 20(5):271-281, Jan. 2009.

[9] Connell, J.; Senior, A.; Hampapur, A.; Tian, Y.; Brown, L.; "Detection and tracking in the IBM People Vision system", in Proc of IEEE ICME, vol.2, pp.1403-1406, June 2004.

[10] Venetianer, P.; Zhang, Z.; Yin, W.; Lipton, A.; "Stationary target detection using the objectvideo surveillance system," in Proc. of IEEE AVSS, pp.242-247, Sept. 2007.

[11] Hu, W.; Huang, D.; Chen, W.; "Adaptive Wide Field-of-View Surveillance Based on an IP Camera on a Rotational Platform for Automatic Detection of Abandoned and Removed Objects", ICIC Express Letters, pp. 45-50, 2010.

[12] Spagnolo, P.; Caroppo, A.; Leo, M.; Martiriggiano, T.; D'Orazio, T; "An Abandoned/Removed Objects Detection Algorithm and Its Evaluation on PETS Datasets", in Proc. of IEEE AVSS, pp. 17-21, Nov. 2006.

[13] Li, Q.; Mao, Y.; Wang, Z.; Xiang, W.; "Robust Real-Time Detection of Abandoned and Removed Objects", in Proc. of ICIG, pp. 156-161, 2009.

[14] Lu, S.; Zhang, J.; Feng, D. "Detecting Ghost and Left Objects in Surveillance Video". Int. Journal of Pattern recognition and Artificial Intelligence, 23(7):1503-1525, 2009.

[15] Tian, Y.; Feris, R. Lui, H.; Humpapur, A.; Sun, M.; "Robust detection of abandoned and removed objects in complex surveillance videos", IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2010.

[16] Wen, J.; Gong, H.; Zhang, X.; Hu, W.; "Generative Model for Abandoned Object Detection", in Proc. of IEEE ICIP, pp. 853-856, Oct. 2009.

[17] Nghiem, A.; Bremond, F.; Thonnat, M.; Valentin, F.; "Etiseo, performance evaluation for video surveillance systems", in Proc. of IEEE AVSS, pp. 476-481, Sept. 2007.

[18] Li, C.; Xu, C.; Gui, C.; Fox M.; "Level set evolution without re-initialization: A new variational formulation", in Proc. of IEEE CVPR, pp. 430-436, June 2005.

[19] Kass, M.; Witkin, A.; Terzopoulos. D; "Snakes: Active contour models". Int. Journal of Computer Vision, 1(4): 321-331, 1988.

[20] Chan, T.; Vese, L.; "Active contours without edges". IEEE Trans. On Image Processing, 10(2):266-277, Feb. 2001.

Appendix D

Introducción

D.1 Motivación

En la actualidad, existe un creciente interés en sistemas automáticos de videovigilancia como consecuencia de la mayor preocupación que generan las cuestiones relacionadas con la seguridad global. Tradicionalmente, la tarea de monitorización es llevada a cabo por personal que se encarga de analizar de manera simultánea la información proveniente de múltiples cámaras. Este hecho conlleva una reducida eficacia, debido a la gran cantidad de información generada por estas cámaras. Es por este motivo que la detección automática de eventos en tiempo real surge como una solución encaminada a facilitar que los operadores de monitorización puedan encentrar su atención en determinados eventos de interés.

En este contexto, la detección de objetos robados y abandonados se ha convertido en uno de los temas de investigación más prometedores, en especial para su aplicación en entornos altamente concurridos, como estaciones de transporte y centros comerciales. Entre sus posibles aplicaciones, podemos destacar la detección de paquetes sospechosos en estaciones de tren, o la detección de objetos robados en oficinas, salas de exposición o museos. El objetivo de esta detección es realizar una supervisión continua de la información capturada por la cámara, con el fin de poder tomar las medidas oportunas. En la figura D.1, se muestran ejemplos de posibles escenarios de aplicación.



Figura D.1: Ejemplos de objetos abandonado (izda.) y robado (dcha.)

En general, para un sistema cuyo objetivo es la detección de objetos robados y abandonados podemos destacar las siguientes etapas de análisis: segmentación de regiones de interés (primer plano), detección de regiones estáticas, clasificación de objetos y discriminación entre robo y abandono. En la primera etapa, se separan del fondo aquellas regiones de la imagen que pertenecen al primer plano de la escena. Posteriormente, se determina cuáles de estas regiones han permanecido estáticas temporalmente (seguimiento). Después, dichas regiones son clasificadas por tipo (persona, grupo de personas, equipaje...). Para aquellas regiones clasificadas como objetos estacionarios, se realiza una etapa de análisis adicional para determinar si el objeto ha sido robado o abandonado.

En cada etapa de análisis, existen varios factores que limitan su rendimiento. Así por ejemplo, los cambios en la iluminación y fondos no estacionarios pueden desembocar en una segmentación inadecuada de regiones del primer plano, dificultando la detección de objetos de interés. En escenarios altamente concurridos donde son más frecuentes las oclusiones, se dificulta la detección de regiones estáticas. En este tipo de escenarios, la elevada cantidad de objetos supone incrementar el coste computacional de la etapa de seguimiento. La etapa de clasificación se ve afectada por una alta variabilidad en la apariencia de los objetos y personas, que complica la aplicación de métodos específicos de reconocimiento de objetos. Finalmente, el requisito de análisis en tiempo real implica utilizar algoritmos de baja complejidad.

Varias técnicas han sido propuestas para la detección de objetos robados y abandonados. Entre los ejemplos, podemos destacar aquellas técnicas que se basan en la estabilización de la imagen proveniente de una cámara en movimiento [6]; las basadas en la detección de regiones

estáticas [7], en la clasificación de blobs (personas u objetos) [3] o en la discriminación de regiones estáticas entre objetos robados y abandonados [8]. La aplicación de estas técnicas resulta adecuada en escenarios en los que los objetos de interés pueden ser detectados con exactitud. Sin embargo, esta suposición no es válida para escenarios más complejos; en particular, la discriminación de regiones estáticas entre robo y abandono no ha sido lo suficientemente estudiada para escenarios de diversa complejidad.

D.2 Objetivos

El principal objetivo de este Proyecto es el estudio de la última etapa de análisis de un sistema de videovigilancia capaz de detectar objetos robados y abandonados, con la finalidad de introducir mejoras a un sistema existente actualmente en desarrollo en el grupo de investigación Video Processing and Understanding Lab (VPU-Lab) en la Universidad Autónoma de Madrid. Para ello, se proponen los siguientes subobjetivos:

1) Estudio del estado del arte

Para las etapas de análisis anteriormente descritas, se realizará un estudio de las técnicas más representativas en el estado del arte, haciendo especial hincapié en técnicas existentes para la discriminación de regiones estáticas entre objetos robados y abandonados.

2) Estudio del sistema de detección de objetos robados y abandonados disponible en VPU-Lab

Se realizará un estudio exhaustivo del sistema existente en VPU-Lab con el fin de identificar los factores clave en la detección de objetos de interés y la discriminación entre robo y abandono.

3) Diseño e implementación de nuevos discriminadores basados en la extracción de una única característica

Se implementarán nuevos clasificadores entre robo y abandono basados en la extracción de una única característica, con el fin de dotar de robustez al sistema para aquellos escenarios en los que los métodos existentes presentan problemas.

4) Diseño a implementación de esquemas de fusión de varias características

Se estudiarán y evaluarán esquemas clásicos de fusión con el fin de combinar la información producida por los distintos discriminadores disponibles en el sistema de VPU-Lab, así como los nuevos métodos propuestos.

5) Definición del entorno de evaluación

Para evaluar los distintos discriminadores, se han elaborado, a partir de secuencias de uso público, dos conjuntos de datos de prueba: conjunto de datos *anotados* y conjunto de datos *reales*. Para los datos anotados, la información necesaria se ha extraído manualmente. Para los datos reales, se ha desarrollado un proceso para extraer la información automáticamente a partir de archivos con meta-datos, utilizando el sistema disponible en VPU-Lab.

6) Evaluación comparativa de la mejoras introducidas por los métodos propuestos

Se evaluará el rendimiento de los métodos de clasificación propuestos, así como de los esquemas de fusión, y se compararán frente a los métodos existentes provistos por VPU-Lab, con el fin de identificar sus ventajas e inconvenientes.

D.3 Organización de la memoria

La memoria de este Proyecto de Fin de Carrera consta de los siguientes capítulos:

- Capítulo 1: En este capítulo se presentan la motivación y los objetivos de este Proyecto, así como la estructura de la memoria.
- Capítulo 2: En este capítulo, se presenta primero una definición del problema, identificando el tipo de información que es necesario extraer para realizar la detección y posteriormente se describen las técnicas más representativas para cada etapa de análisis, poniendo énfasis en la etapa de discriminación.
- Capítulo 3: En este capítulo se describe el sistema base para la detección de objetos robados y abandonados, provisto por VPU-Lab. Los métodos utilizados para la etapa de discriminación son descritos en detalle.

- Capítulo 4: En este capítulo se describen los métodos propuestos para la discriminación, basados en la extracción de una única característica.
- Capítulo 5: En este capítulo se describen los distintos esquemas de fusión para múltiples características.
- Capítulo 6: En este capítulo se describen los conjuntos de datos de prueba, las medidas de rendimiento y los resultados experimentales. Se realiza también una comparación de los métodos propuestos frente a los más significativos del estado del arte.
- Capítulo 7: En este capítulo se resumen las principales contribuciones de este Proyecto, a partir de los resultados obtenidos. Adicionalmente, se presentan sugerencias para posible trabajo futuro.
- Anexos:
 - A. Introducción a Máquinas de Soporte Vectorial
 - B. Extracción automática de máscaras de *foreground* para objetos robados y abandonados a partir de anotaciones
 - C. Publicaciones

Appendix E

Conclusiones y trabajo futuro

E.1 Resumen del trabajo realizado

En este Proyecto, se ha llevado a cabo un estudio exhaustivo del problema de detección de objetos robados y abandonados, centrandó nuestra atención en la discriminación de regiones estáticas entre uno de los dos eventos. El objetivo de esta tarea es determinar si la región estática detectada se debe a un objeto robado o abandonado. En el estado del arte, encontramos pocos ejemplos que se centren en el problema de discriminación. Entre ellos, podemos distinguir entre los basados en color y los basados en contorno, dependiendo del tipo de información extraída.

Las contribuciones de este Proyecto pueden resumirse en los siguientes puntos:

- **Diseño e implementación de nuevos métodos de discriminación basados en la extracción de una única característica.** Un método genérico basado en contornos activos ha sido definido para etapa de discriminación. Este método mide la diferencia entre los ajustes realizados tanto en el fondo de la escena como en la imagen actual. Para este análisis, se han seleccionado y estudiado tres diferentes técnicas de contornos activos, resultando en tres discriminadores distintos. Posteriormente, se presenta un discriminador basado en color, que calcula el valor medio del contraste a nivel de píxel (a lo largo del contorno del objeto bajo análisis) entre la región estacionaria detectada y sus alrededores, en los tres canales de color.

- **Estudio de diferentes técnicas de clasificación para la discriminación combinando múltiples características.** Se han considerado tres métodos populares de aprendizaje artificial para la etapa de discriminación combinando varias características, con el objetivo de mejorar la eficiencia combinando la información proveniente de diversos discriminadores. En particular, se han seleccionado las siguientes técnicas: Clasificador de Bayes (*Naive Bayes*), Máquinas de Soporte Vectorial (*Support Vector Machines*), y K-vecinos más cercanos (*K-nearest neighbor*).
- **Elaboración de dos conjuntos de datos de prueba para la detección de objetos robados y abandonados.** Dos conjuntos de datos han sido elaborados para la evaluación de distintos discriminadores utilizando el mismo conjunto de secuencias. También pueden utilizarse para la evaluación de sistemas completos de detección de objetos robados y abandonados. Para el conjunto de secuencias, se han seleccionado diversos vídeos de repositorios disponibles al público. El primer conjunto de datos de prueba consiste en anotaciones manuales realizadas a los vídeos. Para el segundo conjunto, se ha diseñado un procedimiento para generar máscaras de frente (*foreground masks*) utilizando el sistema de análisis de vídeo disponible en VPU-Lab. Este procedimiento extrae automáticamente la información necesaria para la evaluación de los discriminadores, a partir de ficheros de metadatos
- **Evaluación de los métodos de discriminación existentes y propuestos sobre los dos conjuntos de datos de prueba.** Hemos evaluado el rendimiento de los distintos discriminadores sobre datos anotados (segmentación ideal) y datos reales (segmentación inexacta). El *conjunto de datos anotados* nos permite evaluar la capacidad discriminadora de los distintos métodos, posibilitándonos determinar aquellos escenarios para los que presentan problemas. En condiciones más realistas, un sistema de análisis de vídeo producirá máscaras de frente imprecisas, debido a los diversos problemas que afectan a la etapa de segmentación. Con *el conjunto de datos reales*, hemos sido capaces de evaluar el rendimiento de los discriminadores en escenarios realistas, pudiendo determinar qué

impacto tiene etapa de segmentación sobre la discriminación.

E.2 Conclusiones

Al evaluar el rendimiento de los métodos de discriminación propuestos y existentes en secuencias anotadas, hemos sido capaces de identificar problemas clave que afectan a la etapa de discriminación. Para el método existente basado en color, hemos concluido que es particularmente sensible al ruido, y la utilización de un sólo canal de color ha resultado insuficiente. Los discriminadores de gradiente (basados en análisis de bordes) han mostrado un bajo rendimiento en escenarios con fondos con texturas que presentan una alta energía de gradiente. En contraste, los cuatro métodos propuestos se han mostrado robustos ante estas mismas situaciones, obteniendo un porcentaje de acierto cercano al 100% en la mayoría de las categorías.

Para el conjunto de datos reales, hemos observado, como cabía de esperar, una reducción en el rendimiento; al depender todos los discriminadores de la precisión de la máscara de foreground para extraer la información deseada. En particular, el método basado en color del sistema base ha presentado una notable reducción de su eficacia sobre el conjunto de datos reales. Por otro lado, dos de los métodos propuestos (*Contornos activos geométricos basados en la información de borde*, y *Contraste de Color a nivel de Píxel*), han demostrado ser muy robustos ante máscaras poco precisas, obteniendo excelentes resultados en este conjunto de datos. El uso de ajustes con contornos activos conlleva la posibilidad de lidiar con una segmentación inadecuada. Adicionalmente, la combinación de características de distintos clasificadores ha incrementado la tasa de detección en los resultados finales. En concreto, se ha obtenido una tasa de detección por encima del 98% en el conjunto de datos reales.

En conclusión, podemos afirmar que los métodos propuestos para la discriminación son adecuados para su integración en un sistema de análisis de vídeo para detectar objetos robados y abandonados, al haber demostrado ser menos dependientes de la precisión de la máscara de *foreground* producida por las anteriores etapas de análisis.

E.3 Trabajo futuro

En este Proyecto, hemos identificado los problemas clave que afectan a la etapa de discriminación.

En este contexto, las siguientes líneas de investigación pueden ser consideradas:

- Los esquemas de fusión de distintas características han demostrado ser una manera eficiente de combinar la información para aumentar la tasa final de detección. Sin embargo, estos esquemas requieren utilizar la información de varios discriminadores. Esto conlleva un alto coste computacional en el sistema, y reduce su capacidad para operar en tiempo real. Por este motivo, se deben concentrar más esfuerzos en el desarrollo de técnicas para determinar si un discriminador está funcionando como debe. En otras palabras, sugerimos deducir de antemano que método de discriminación proporcionará la mejor medida, para evitar utilizar los que peores resultados ofrezcan. Por ejemplo, los discriminadores de gradiente presentan problemas cuando la región que rodea al objeto de interés contiene una alta energía de gradiente (bordes) tanto en la imagen de fondo como en la imagen actual. En esta situación, un discriminador basado en color sería capaz de proporcionar mejores resultados. Para los esquemas de fusión, esto conlleva dos ventajas: reducir el coste computacional (tiempo de procesamiento) e incrementar la confianza en los resultados del discriminador final. Al utilizar esquemas de fusión, se pueden obtener mejores resultados seleccionando *sólo* aquellas medidas que sepamos son relevantes para un objeto en particular.
- Se debe llevar a cabo un análisis más exhaustivo en entornos altamente concurridos, donde la presencia de objetos cercanos puede afectar a las características extraídas. Para extender el conjunto de datos de prueba actual, situaciones más complejas han de ser tenidas en cuenta, tales como fondos de alta complejidad, fondos multimodales, objetos con texturas complejas, y distintas tasas de compresión para la secuencia de vídeo.
- Como hemos podido observar en los resultados de los experimentos llevados a cabo, para la etapa de discriminación se ha obtenido un alto rendimiento dentro de los conjuntos de datos utilizados. Este hecho sugiere que la complejidad de la detección de objetos

robados y abandonados radica principalmente en los módulos encargados de la extracción de objetos de interés. En particular, podemos decir que la etapa de *Detección de Objetos Estacionarios* presenta diversos problemas que siguen sin resolverse.

Appendix F

Presupuesto

1) Ejecución Material

• Compra de ordenador personal (Software incluido)	2.000 €
• Alquiler de impresora láser durante 6 meses	260 €
• Material de oficina	150 €
• Total de ejecución material	2.400 €

2) Gastos generales

• sobre Ejecución Material	352 €
----------------------------	-------

3) Beneficio Industrial

• sobre Ejecución Material	132 €
----------------------------	-------

4) Honorarios Proyecto

• 1800 horas a 15 €/ hora	27000 €
---------------------------	---------

5) Material fungible

- Gastos de impresión 280 €
- Encuadernación 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto 32.774 €

7) I.V.A. aplicable

- 18% Subtotal Presupuesto 5.899,3 €

8) Total presupuesto

- Total Presupuesto 38.673,8 €

Madrid, Julio 2011

El Ingeniero Jefe de Proyecto

Fdo.: Luis Alberto Caro Campos

Ingeniero Superior de Telecomunicación

Appendix G

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un sistema basado en discriminar objetos estáticos entre abandonados o robados en secuencias de vídeo-seguridad. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

- 1) La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
- 2) El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

- 3) En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
- 4) La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
- 5) Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
- 6) El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
- 7) Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
- 8) Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
- 9) Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se

dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

- 10) Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
- 11) Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
- 12) Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
- 13) El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
- 14) Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

- 15) La garantía definitiva será del 4% del presupuesto y la provisional del 2%.
- 16) La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
- 17) La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
- 18) Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
- 19) El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
- 20) Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
- 21) El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
- 22) Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción

definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

- 23) Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado “Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

- 1) La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
- 2) La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
- 3) Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
- 4) En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

- 5) En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
- 6) Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
- 7) Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
- 8) Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
- 9) Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
- 10) La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
- 11) La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
- 12) El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.