FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Image Processing for Event Detection in Retail Environments

Pedro Daniel Nunes Querido

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Dr. Luís Corte-Real

Co-supervisor: Dr. Pedro Carvalho

February 24, 2017

Resumo

A deteção de eventos através da utilização de técnicas de Visão Computacional é um problema estudado e com aplicações em várias áreas da sociedade, mas sem uma solução global que se aplique a todos os casos. A sua aplicação em ambientes de retalho pode fornecer informações valiosas para as empresas desse setor como, por exemplo, qual a área de uma loja com mais movimento ou onde e quando os clientes interagem com os objetos lá dispostos. Estas informações podem ser utilizadas para melhorar as técnicas de marketing e expandir os modelos de negócio existentes.

A presente dissertação foca-se no estudo da deteção de eventos como ocultações e alterações que acontecem regularmente em determinadas regiões de interesse de um ambiente de retalho, através da análise de vídeo capturado por câmeras RGB-D. Para cumprir o objetivo proposto, vários descritores visuais e métodos de segmentação de imagem foram estudados e testados para propor a melhor solução possível, nas condições apresentadas.

Os resultados obtidos indicam que a deteção dos eventos propostos é possível de alcançar com níveis de precisão promissores. No entanto, mais trabalho de investigação é necessário para tornar a solução proposta mais robusta a ruído e passível de ser aplicada em situações de tempo-real.

Este documento descreve toda a metodologia proposta para atacar o problema definido, bem como todos os resultados experimentais obtidos e conclusões sobre os mesmos. São também referidas diferentes propostas de trabalho futuro, com vista a melhorar os resultados obtidos.

Abstract

The detection of events through the use of Computer Vision techniques is a well studied problem with application in various areas of society, but without a global solution that applies to all cases. Its application in retail environments can provide valuable information for companies in this sector, such as what is the most crowded area of a shop or where and when customers interact with the objects placed. This information can be used to improve marketing techniques and expand existing business models.

The present dissertation focuses on detecting events such as occlusions and alterations that occur regularly in certain regions of interest of retail environments, through the analysis of video captured by RGB-D cameras. To accomplish the objective set, several visual descriptors and image segmentation methods were studied and tested in order to propose the best possible solution under the conditions presented.

The obtained results indicate that the detection of the proposed events is possible to reach with promising precision levels. However, more research work is needed to make the proposed solution more robust to noise and possible to be applied in real-time situations.

This document describes all the methodology proposed to attack the defined problem, as well as all the experimental results obtained and conclusions extracted. Different proposals for future work are also mentioned in order to improve the results achieved.

"Be grateful with everything you have and you will be successful in everything you do."

Conor McGregor

Contents

List of Figures

Glossary

2D	Two-Dimensions
3D	Three-Dimensions
AUC	Area Under the Curve
BRIEF	Binary Robust Independent Elementary Features
C-colour-SIFT	C-colour Scale Invariant Feature Transform
CENTRIST	Census Transform Histogram
CIE	International Commission on Illumination
CRT	Cathode Ray Tube
CSIFT	Colour Scale Invariant Feature Transform
CS-LBP	Center Symmetric Local Binary Pattern
CV	Computer Vision
DoG	Difference-of-Gaussian
FAST	Features from Accelerated Segment Test
FPR	False Positive Rate
GLOH	Gradient Location and Orientation Histogram
HOG	Histogram of Oriented Gradients
HSV	Hue Saturation Value
HVS	Human Visual System
IR	Infrared
LBP	Local Binary Pattern
LLC	Locality-constrained Linear Coding
LTP	Local Ternary Pattern
MOG	Mixture of Gaussians
MRF	Markov Random Fields
OCLBP	Opponent Color Local Binary Pattern
ORB	Oriented Fast and Rotated BRIEF
PBAS	Pixel-Based Adaptive Segmenter
PCA	Principal Component Analysis
RGB	Red Green Blue
RGB-D	Red Green Blue - Depth
ROC	Receiver Operating Characteristics
ROI	Region of Interest
ScSPM	Sparse Coding Spatial Pyramid Matching
SIFT	Scale Invariant Feature Transform
SIFT-LLC	Scale Invariant Feature Transform - Locality-constrained Linear Coding
SIFT-ScSPM	Scale Invariant Feature Transform - Sparse Coding Spatial Pyramid Matching

SILTP	Scale Invariant Local Ternary Pattern
SPM	Spatial Pyramid Matching
sRGB	standard Red Green Blue
SURF	Speeded Up Robust Features
SVM	Support Vector Machines
TPR	True Positive Rate
ViBE	Visual Background Extractor
VLBP	Volume Local Binary Pattern

Chapter 1

Introduction

1.1 Context

The retail sector has a vast influence in modern society, with some of its most notable brands accounting for a large part of the world's most valuable companies. For instance, some of the biggest clothing brands record annual profits in the order of the billions of dollars [1]. So, to maintain and even increase those numbers, companies continuously look for ways to expand and improve their existing business models and marketing techniques. However, retail shops are already scattered around worldwide and fashion shows or commercials are regularly broadcast on television and other multimedia platforms, which may difficult a simple expansion process. Therefore, fields of study like Computer Vision (CV) and Data Mining may then prove to be very useful, as they allow to process data and provide valuable information, so that improvements may be implemented and expansions made viable and structured.

Since its inception, CV has always been a very interesting field of study. The process of capturing, processing and analyzing images or videos to understand real life problems and model three dimensional (3D) human-like perception is not trivial, but has many applications [2]. Furthermore, with computational power constantly growing, more and more developments can be made in CV related problems thus making it a rapidly expanding field. Event detection or, more specifically, the need to detect and predict the occurrence of certain events is a very common CV problem. The performance of a spatiotemporal analysis of an image or video may provide a description of the scene through the search and recognition of homogeneous characteristics like color [3], texture [4] or other features contained in certain objects or beings [5, 6]. The variations on those same characteristics may then indicate the occurrence of an event, depending on the definition of what an event really is in that specific problem (events may be more than alterations in the scene and not all alterations are events). In other words, it is possible to detect the occurrence of a desired event by observing critical changes in a scene or in part of it.

Event detection has a real application in the retail industry. For instance, examining alterations in a specific region of a shop can lead to information about how and when the clients interact with the objects placed there or even give information about what is the most crowded section of the

store, which is valuable data for marketing purposes. Furthermore, the real-time detection of specific events may allow the prevention of future unwanted situations as the appropriate authorities or persons of interest can be alerted whenever necessary.

1.2 Motivation

Although work has already been done regarding event detection problems in various distinct contexts like sporting events [7], infrastructure security [8] and human gestures [9], a definitive solution that applies to all problems doesn't exist. This is mainly due to the gigantic heterogeneity of conditions (and possible variations) presented in these problems. For example, slight variations in the light or shading of a given object can lead to a very different digital representation and description of that object [2].

The application of event detection in retail environments (shops) provides a solid amount of different possible events to consider and detect. With shops exposed to constant changes due to people moving and interacting with each other and with the objects in the scene, it is mandatory to understand and define what a significant event is and what it's just noise or other temporal perturbations. Therefore, to be able to present the best and most robust scene description possible (and thus be able to perform an appropriate event detection that provides valuable information), it is very important to define and comprehend the regions of interest inside that scene and the specific group of events happening in those regions that are relevant to the problem.

1.3 Objectives

The main purpose of this dissertation is to accomplish a viable detection of a predetermined set of events happening in a retail environment, through the spatiotemporal analysis and description of video captured by generic RGB and RGB-D cameras. To achieve that objective, it is crucial to study, understand and apply a group of existing CV methods and techniques.

1.4 Document Structure

This document is organized in six chapters. Chapter 2 presents a review of related work, with a focus on image segmentation and processing methods used to extract spatiotemporal information for scene decomposition, analysis and description. Chapter 3 defines the workspace, providing a description of the test video sequences, the set of events to detect and the evaluation metrics to evaluate the tests results. Chapter 4 covers the description of the different approaches taken to accomplish the defined objective. Chapter 5 presents a guideline of the experimental procedures, an explanation on the decisions made throughout, and a discussion of the results obtained. Chapter 6 presents a final discussion of the present work, followed by a suggestion of possible future work and improvements.

Chapter 2

Literature Review

This chapter contains a review of related work regarding image segmentation and processing methods and techniques to extract spatiotemporal information for scene decomposition, analysis and description. First, in section 2.1, a very brief analysis on modern capture devices and the image information they provide is presented. Then, in section 2.2 a review on some image segmentation tecnhiques is provided, with a focus on the foreground detection and background modelling methods. In section 2.3 a study is made on the most well-known image (or visual) descriptors and the image properties in which they are based. Finally, a small discussion of all the reviewed methods and their application in retail environments is presented in section 2.4.

2.1 Image Information and Capture Devices

From a visual standpoint, a video is simply a temporal sequence of images. Therefore, most (if not all) image segmentation and processing methods and techniques can be applied to videos, by doing it on a frame-by-frame basis. Furthermore, there are also some methods that take advantage of a video's temporal frame, requiring several different images for processing. These two notions make video more interesting to use in image processing problems as it contains more information (spatial and also temporal) to be processed.

Most modern video and image capture devices are based on the trichromatic theory [10], which states that there are three types of photoreceptors that are approximately sensitive to the red, green and blue (RGB) regions of the electromagnetic spectrum - RGB cameras. However, some devices expand on that subject, providing image per-pixel depth (D) information along with the visual data - the RGB-D cameras. In fact, in recent times, the number of these devices available to the general public has increased, mainly due to the rise of modern gaming hardware such as the Microsoft Kinect [11]. For instance, the latter uses a Infrared (IR) Projector to project IR dots in the scene and an IR Camera to observe them. Making use of the known relative geometry between the IR devices and the projected dots pattern, the Kinect tries to match them in order to reconstruct the image in three-dimensions (3D) using triangulation methods. In figure 2.1, it is possible to observe that IR dot matching from the Kinect, on the top image, and the correspondent depth information

on the bottom. The depth value is encoded in grey-scale, meaning that the darker a pixel is, the closest it is to the camera, with the black pixels being the exception - no depth values possible to attain for those pixels or regions.

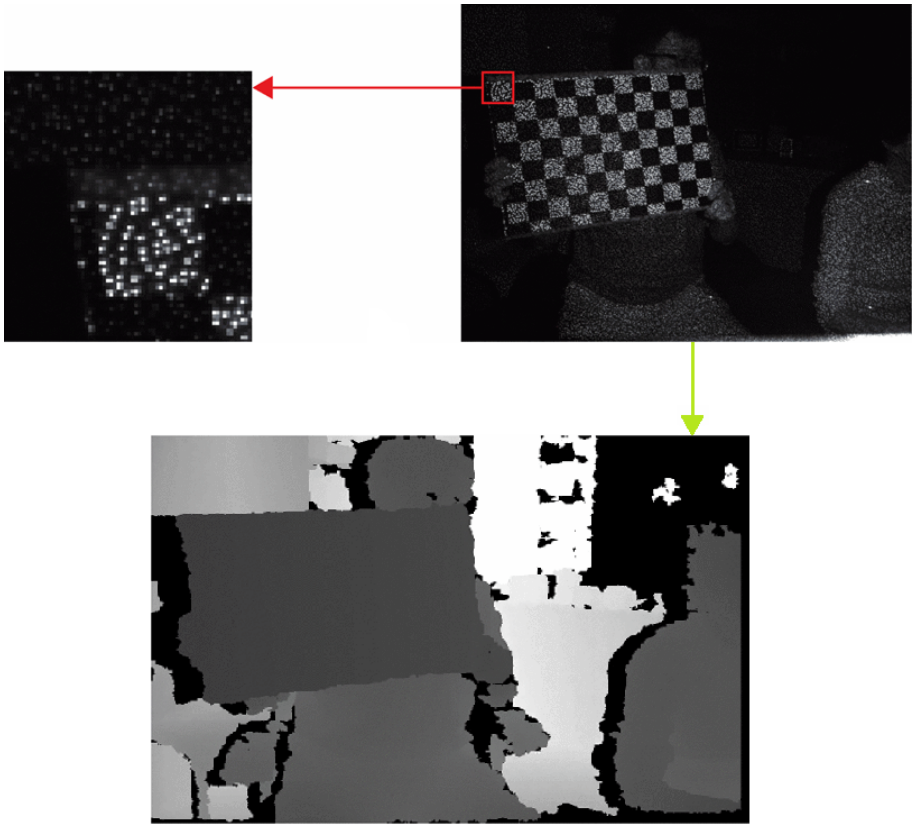


Figure 2.1: Microsoft Kinect depth information. On top, its IR mechanism and on the bottom, the corresponding image depth data. Adapted from [11].

While the depth information provided by these RGB-D devices is not ideal, due to the general low of range of capture and sometimes noisy estimate, the good cost-results ratio has opened a lot of doors to new CV-related researches and developments, specifically in problems like 3D modelling [12, 13] and people tracking [14, 15, 16].

2.2 Image Segmentation

Image segmentation is generally the first step in image or video analysis as its main objective is to divide an image (or a video frame) into multiple segments (which can contain similar features or attributes) for further process. It contains a wide variety of methods that go from the most basic, like the Otsu threshold segmentation and its newer and improved variations [17, 18], to edge detection methods like Canny [19, 20] and Gabor Filters [21], to the most recent foreground extraction and background modelling techniques [22].

Edge detection, or the visual separation of the region boundaries in an image, has a longstanding relevance in the CV community. The aforementioned Canny Edge Detector is a fine example of this sort of image segmentation due to its simplicity, fairly good results and possibility to easily apply in conjunction with other image processing methods. It is usually comprised of five steps to achieve the final result: first, the image is smoothed to remove noise; then it considers the edges as the spots of the image where the gradients have large magnitudes; filters are finally applied to remove all the values that are not local maximums, that don't meet the defined threshold or that are not connect to a very strong edge (hysteresis approach).

Figure 2.2: Foreground detection and background modelling example. The rows represent different frames from the same video. From left to right: The original frame, the background, the ground-truth and the foreground mask. Extracted from [22].

Typically, it is very useful and relevant to separate the foreground from the background of an image for better processing. An example of this background and foreground separation can be observed in the figure 2.2. Foreground detection has many applications in real-life and CV related problems like video surveillance of human activities [23, 24], optical motion capture [25] or gaming activities with devices such as the aforementioned Microsoft Kinect [11]. Therefore, there are currently a vast number of methods with different approaches and most of them are used to subtract/model the stationary parts of a scene (the background) to be able to analyze only the foreground. Bouwmans [22] published a study where a review of most of the known methods was performed, including both traditional and more modern approaches. To help on that analysis, the author also defined a general block model for how background modelling and foreground detection usually works. Like shown in figure 2.3, that model consists of three main steps: the first, consists of the background initialization or detection, where N video frames are utilized to obtain and define the first background image; the other two steps are executed in a loop and comprehend the foreground detection, which consists in comparing the current frame with the known background and then classifying the elements (pixels, blocks or clusters) of that frame as either part of the foreground or the background, and the background maintenance that simply updates the known background image over time. This whole process can be performed based on different image/video features as texture, edges and motion. Also, the choice of the element of comparison influences

the precision and the robustness to noise, as the smaller elements generally obtain better precision (pixels obtain pixel-based precision) but also are much more subject to noise.

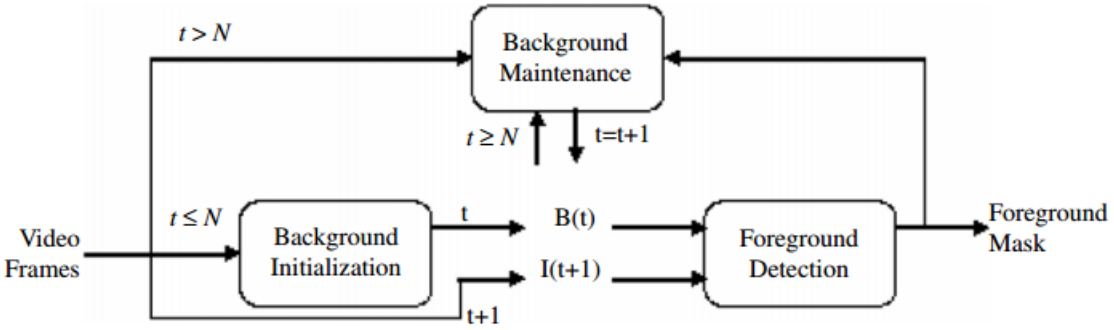


Figure 2.3: Generic background modelling algorithm. N represents the frame number, B(t) the background and I(t) the current image. From [22].

There are also several challenges that foreground detection algorithms must address [22, 26]. The main difficulties are usually concerned with noisy images, camera jitter, illumination changes (a light source turning off can completely change the description of a scene and thus lead to a bad foreground detection) and dynamic backgrounds. For example, a waving tree or water rippling must be detected as part of the background, but their constant motion constantly leads to false foreground positives.

A simple approach to foreground detection may be through the use of the image depth information mentioned on section 2.1 - by constantly subtracting the data of the current frame to the known background model, it is possible to obtain a mask representing the foreground due to the different depth values. However, there are many more possible approaches to foreground detection and background modelling problems, and probably the most discussed and utilized are the statistical models and specifically the traditional Gaussian-based ones. The basic idea behind this approach is that the history over time of the pixel's intensities values can be modeled through a Gaussian. Wren et al. [27] proposed the use a single Gaussian to model the background. This idea, while having its use, severely struggled with dynamic backgrounds. To address this issue, Stauffer et al. [28] introduced the Mixture of Gaussians (MOG) which uses a mixture of K Gaussians to model the history of the color features (in the RGB color space) of each pixel. Although any value of K can theoretically be utilized, the authors, based on the computational power available at the time, proposed the use of a K between 3 and 5.

Since the introduction of the MOG model, there have been several proposals to build on top of its properties and correct its flaws. Bouwmans et al. [29] made a full review and tested the proposals with more recognition by the CV community. The one whose results were generally better was the MOG adaptation via Markov Random Fields (MRF) proposed by Schindler et al. [30]. This method greatly decreases the number of false foreground detections by incorporating the smoothness assumption, which states that the world the consists of spatially consistent entities.

To accomplish this, a continuous background probability value is retained for each pixel and the foreground segmentation becomes a simple labeling problem on a first-order MRF.

Other, more modern, statistical-based models are the Visual Background Extractor (ViBE) [31] and the Pixel-Based Adaptive Segmenter (PBAS) [32], both nonparametric. The first, developed by Barnich et al., builds the background model by putting together the values previously observed for each pixel location. Using a random selection policy, ViBE constantly updates those observed values, which assures that the older ones have an exponentially decaying lifespan and thus will not be used when they're not supposed to. Furthermore, the method also randomly diffuses those pixel values across the neighbouring pixels to guarantee spatial consistency. The latter, proposed by Hofmann et al., works on the concept of a decision block, which decides if a pixel is or is not part of the foreground through a comparison of the current image with the known background, that itself is progressively updated via a pixel-level learning parameter. The essential idea here is that the PBAS works based on a foreground decision which depends on another decision, with them both being pixel-level.

2.3 Scene Description

The general approach towards describing a scene is to first set the interest regions (which can simply be the whole image), whether by applying some image segmentation method or by simply dividing the image into several geometrical parts and then, for each region, build a descriptor based on the local properties or features. Therefore, there are several proposed image (or visual) descriptors built from very diverse properties like color, texture or pixel intensity values of mathematical transformations applied to the image. Most of them are distribution-based, which means that they use histograms to represent the different characteristics of shape and appearance. In fact, several comparative studies have been conducted on this subject and it is widely proved and accepted that the best results for scene description are obtained through the use of distribution-based descriptors [33, 34].

2.3.1 Color

Color is one of the main properties of an image and may be defined as the way the Human Visual System (HVS) measures or perceives the visible part of the electromagnetic spectrum. A color space is then a notation by which humans can group and specify different colors [35].

As mentioned on section 2.1, most of the modern video and image capture devices are based on the trichromatic theory [10]. Then, a color can simply be specified as the sum of the three different components of light (red, green and blue) captured from those devices. That idea gave way to the most basic and well known color spaces: the RGB. Furthermore, as the same theory may also be applied to video displays (and specifically computer monitors), the RGB color spaces are very device dependant as they depend heavily on the specific sensitivity function of the capturing or displaying device [35]. To help combat this RGB dependency and create a standard (and at that

time to allow their use on older cathode ray tube (CRT) screens), Microsoft and Hewlett-Packard developed the standard RGB (sRGB) color space [36].

Although RGB color spaces are the most widely recognized color spaces, they contain some notable flaws, like the high correlation presented between its components [37] and the psychological non-intuitivity of their concept, as it may be hard for humans to visualize a color as the sum of three others. These issues usually restrain their use to simple applications and more often then not a conversion to another color space is required.

Based on linear transformations from the RGB, the Hue Saturation Value (HSV) color space represents the concepts of hue as the attribute which categorizes the color (blue, yellow, etc.), saturation as the level of intensity of the color (non-whiteness) and value as the maximum between the red, green and blue components [38]. This color space and the ones similar to it (like Hue Saturation Lightness (HSL), Hue Saturation Intensity, etc.), also possess some of the RGB color space's shortcomings, as they are directly dependent from it and its capture devices and thus inherit most of its flaws. However, they do provide a different, more interesting (and much more similar to the HVS) view on color.

The high correlation between the RGB components creates a large amount of redundant information that makes RGB signals inefficient for transmission [39]. Furthermore, the HVS is much more sensitive to luminance changes than to chrominance [38]. Those notions allowed the creation of the luma and chrominance color spaces used for television signal transmission - YUV for the European PAL and SECAM coded and YIQ for the American NTSC - that transmit the luma signal (Y') separately from the two chroma components. Y'CbCr (with Cb and Cr corresponding roughly to the blue and red color components) is a scaled, digital version of YUV that is commonly used on image and video compression schemes like JPEG.

The International Commission on Illumination (CIE) defined a system that classifies color according to the HVS and allows the representation of any visible color in terms of its CIE-coordinates [35]. As such, all the color spaces based on this system are device independent. The CIELab (commonly represented as L*a*b*) is one of those color spaces and was introduced with the main objective of being as linear as possible with human visual perception. The L* represents the lightness of the color, and a* and b* the color coordinates (with a* being the position between magenta and green and b* between yellow and blue). CIELab also has a much larger color gamut (subset of colors) than the RGB color spaces (and the ones based on linear transformations of that model) but as the capture devices are usually based on the trichromatic teory, that advantage is somewhat mitigated[38].

Mathematical and computationally speaking, a grey-scale image is a matrix (width by height) of pixel intensity values, with the values ranging from least intense (black) to the most intense (white). This way, as expected, a colored image is composed by three separate matrices (one for each component, whichever the color space). These notions can be used to compute histograms from the images in order to describe the scene using the color values, as it is shown in figure 2.4. On the left, two colored images from the same source with some moderate differences, and on the right a 3D representation of their respective RGB histogram.

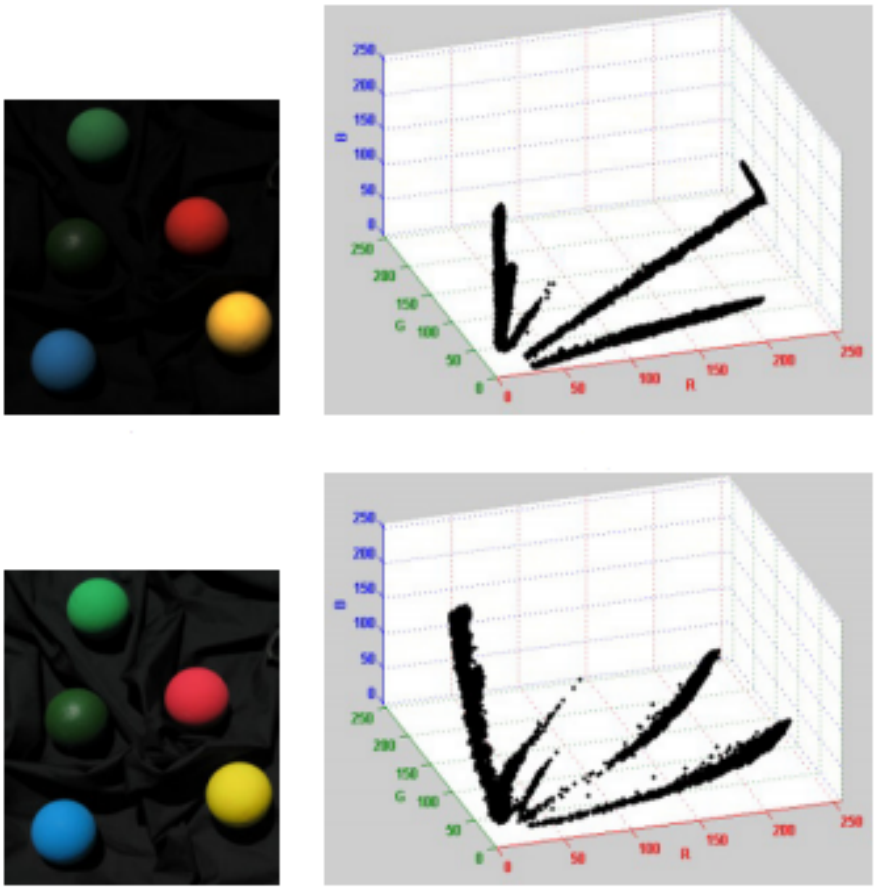


Figure 2.4: Images (left) and a 3D representation of their corresponding RGB histogram (right). Adapted from [37].

While color histograms are, in fact, simple concepts, they can provide valuable information in several different image processing problems. However, they are not the only type of visual descriptors that try to describe color, or rely on it in order to perform a description, as it may be seen in the sections below.

2.3.2 Texture

Very much like color, the texture of an image and its use for scene description problems (amongst others) is a constantly researched topic in the CV community and several texture operators have been developed and proposed throughout the years. The Local Binary Pattern (LBP) [40] is probably the most widely used as it has a very clear and easy concept and thus is fast to compute, which allows its use in more complex real-time applications. The idea behind it is to simply form labels/textons (from which a feature vector and subsequently an histogram can be built) for the image pixels, by thresholding the values in a 3x3 neighbourhood of each pixel with the center

value (1 if bigger, 0 if lower) and consider the result as a binary number. Its main quality is the tolerance against illumination changes, with its main drawbacks being its poor robustness on flat image areas and its restriction to grey-scale images.

Throughout the years, several expansions have been made on the original LBP. For instance, Ojala et al. [41] expanded on the original concept by using circular neighbourhood and by bilinearly interpolating values at non-integer pixel coordinates which opened the possibility for the use of neighbourhoods of any radius and thus of any pixel dimensions. The authors also introduced the notion of uniform patterns which can be used to implement rotation-invariance and at the same time reduce the size of the feature vectors. They defined that a local binary pattern is considered uniform if it contains 2 or less binary transitions (from 0 to 1 or vice-verse) and proved that such patterns occur way more often (around 90% for a circular neighbourhood of 8 pixels) than the others with more transitions.

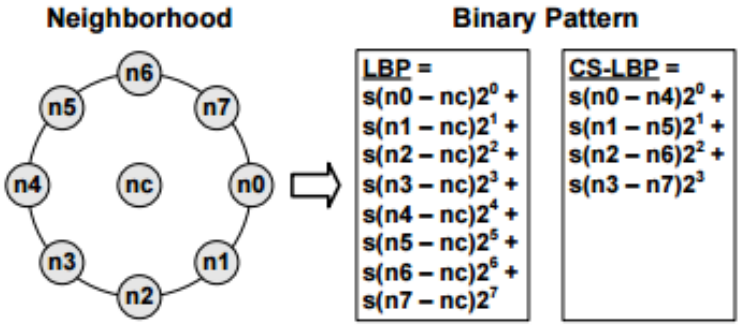


Figure 2.5: LBP and CS-LBP Algorithm applied to an 8 pixel neighbourhood. Extracted from [4].

Heikkilä et al. [4] introduced the Center Symmetric Local Binary Pattern (CS-LBP), which instead of comparing each pixel with the center pixel, compares center-symmetric pairs of pixels, thus reducing the number of total comparisons to half. This idea and its comparison to the original LBP can be observed in figure 2.5. Furthermore, the tests conducted by the authors show that the CS-LBP operator has equal or better performance than the original LBP in terms of object description and classification problems.

Tan et al. [42] showed that when neighboring pixels are similar (uniform areas) which is common in face recognition problems, the LBP is not very robust to local noises. The authors then proposed the Local Ternary Pattern (LTP) which combats that issue by adding a tolerative range to the LBP operator. Liao et al. [43] expanded on that concept by adding scale invariance - the Scale Invariant Local Ternary Pattern (SILTP). By introducing the intensity value of the central pixel to the texture operator's calculation, the authors showed that the SILTP is much more robust than either the LBP or the LTP as the scale invariance property makes it much more tolerant to illumination changes while only adding one more comparison (three instead of two). Silva et al. [44] applied a similar approach to the CS-LBP texture operator by also considering the value of

the intensity of the central pixel. This process may be observed in figure 2.6 and compared with the CS-LBP algorithm represented in figure 2.5. The eXtended Center Symmetric Local Binary Pattern (XCS-LBP) maintains the short histogram of the CS-LBP, while making it more robust overall.

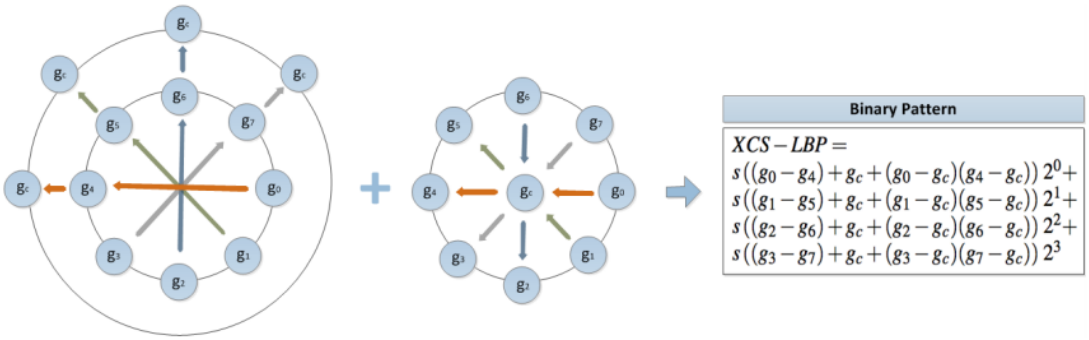


Figure 2.6: The XCS-LBP algorithm applied to an 8 pixel neighbourhood. Extracted from [44].

Although there are variations that consider color information like the Opponent Color Local Binary Pattern (OCLBP) [45, 46], regarding the use of the LBP in colored images, maybe the most common practice is to apply it in a color channel-wise base and then combine the results. While this use of color and texture in parallel seems promising, studies have showed that the increase in performance is rather minimal and in several cases even non-existant or negative [45]. Mäenpää et al. proved that generally, color texture analysis only outperforms the grey-level analysis for static illumination conditions, due to the sensitivity of color to illumination changes. Therefore, the authors claim that a parallel color-texture study should only be used in applications were that minimal increase in performance is guaranteed and critical to the final result, recommending instead a sequential use of both information.

A popular texture analysis, other than using the LBP and its extensions and variations, is the one performed through the use of Gabor filters [47]. The typical approach is to convolve the input image with a Two-Dimensional (2D) Gabor function to obtain a Gabor feature image. That image can then be used in the same manner as the LBP feature image to obtain histograms. The extension of the original bandpass Gabor filters [48] to 2D functions and the fact that, under certain conditions, the phase response of those filters is approximately linear allowed that through the use of Gabor filters with different frequencies and orientations it is possible to detect and describe certain recognized patterns or textures from complex images, both color and grey-scale. As expected from the nature of their concept, they can also be used to detected edges, although it is not their main application (as mentioned on section 2.2).

The original LBP (and its already aforementioned extensions) was designed to process only spacial information (static texture). However, since the original concept, attempts have been made to expand it to the spatiotemporal domain to handle dynamic textures. The Volume Local Binary Pattern (VLBP) [49] is able to process video texture (dynamic) by looking at it the same way the

LBP does in the spatial domain (X and Y axis), while adding a new temporal axis (a frame index). As the dynamic texture is viewed as sets of volumes (because of the 3D approach) and their features are extracted on the basis of those volume textons, the VLBP combines both appearence and motion in order to give a description. One of the main issues with Zhao's et al. proposed method is that in the VLBP, a parameter P (the number of local neighboring points around the central pixel in one frame) is what determines the total number of features and as such, a large P will produce a very long histogram and a small one will lead to losing a lot of information. To address that problem, the same authors developed the LBP-TOP [50] which differs from the VLBP in two major points: first, while the VLBP used three parallel planes, of which only the middle one contains the center pixel, the LBP-TOP uses three orthogonal planes (XY, XT and YT) which intersect in that center pixel, and then, while the VLBP considers the co-occurrences of all neighboring points from the three parallel planes (which makes the feature vector very long when the number of neighbouring points is big), the LBP-TOP separates each feature distribution in its orthogonal planes and then concatenates the result (keeping the feature vector always much shorter). The LBP-TOP principles are possible to observe in figure 2.7. While the image represented in the figure is in color for a better representation and perception, both the VLBP and LBP-TOP are restricted to grey-level images.

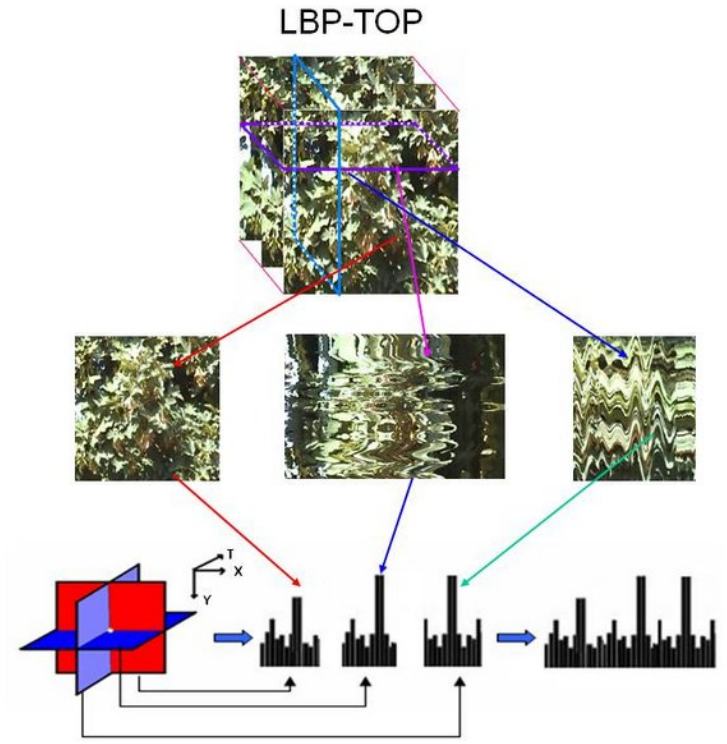


Figure 2.7: LBP-TOP principles. On the top, an image (a video frame) and its division in the XY, XT and YT axis. On the bottom, a representation of those axis, the histograms extracted from the application of the method in them and the final histogram (the concatenation of the three). Extracted from [50].

2.3.3 Local Features

The detection of the so called "local features" and its use to build descriptors is a very well studied and documented subject in the CV community. Probably the most largely used feature descriptor is the Scale Invariant Feature Transform (SIFT) developed by David Lowe [5]. In its original design, Lowe provided a method to obtain interest points from the image by constructing a Difference-of-Gaussian (DoG) pyramid and then use those points to give a description of the local image structures in the neighbourhood around each point. A DoG pyramid can be defined as the differences between adjacent levels in a Gaussian Pyramid which is constructed by consecutively subsampling and smoothing the image, with the base level being the original input image [51]. With the DoG pyramid computed, the interest points are then the points where its values are either maximums or minimums with respect to both the spatial coordinates and the scale level in the pyramid. To achieve scale and rotational invariance, the SIFT descriptor both normalizes the size of the local neighbourhoods in a scale-invariant manner and determines the local dominant orientation from the orientations of the gradient vectors. Thus, based on its concept and behaviour, the SIFT descriptor can simply be viewed as a histogram of gradient locations and orientation. Its application on a object matching problem is obvious as it can save the description of the objects (a process called training) and look for them in a scene through simple comparisons. An example of this can be observed in figure 2.8.

Figure 2.8: SIFT descriptor applied to an object matching problem. The objects are represented on the left images and the scene containing them on the right. The big squares represent the object match and the small squares the interest points detected. Extracted from [5].

As the SIFT descriptor is also grey-scale restricted, there have been, since its original concept, several proposals to work this issue. As is the case for the LBP, the SIFT can simply be applied to the different color channels, but that doesn't guarantee color invariance. Probably the most well respected methods with regards to that were developed by Abdel-Hakim et al. [52], the Colour Scale Invariant Feature Transform (CSIFT), and by Burghouts et al. [53], the C-colour Scale Invariant Feature Transform (C-colour-SIFT), and they both apply the SIFT model to the Gaussian-based color invariance concept of Geusebroek et al. [54]. Furthermore, the tests made

by Burghouts et al. prove in fact that this approach provides the best results when compared with several other methods and even the original SIFT.

Other widely used distribution-based local feature descriptors are the Gradient Location and Orientation Histogram (GLOH) descriptor [33] and the Speeded Up Robust Features (SURF) descriptor [55]. The first works in a very similar way to the SIFT, simply replacing the Cartesian location grid by a log-polar one and applying Principal Component Analysis (PCA) to reduce the size of the descriptor. The second, relies on the concept of integral images for image convolutions and thus replaces the DoG pyramid model by a simpler Hessian matrix-based detector and Haar wavelet responses descriptor. With this, the SURF descriptor is commonly faster than the SIFT and more often than not obtains similar results. However, studies like the one performed by Panchal et al. [56] show that when computation time is not a factor, the SIFT descriptor is still the better option of the two.

With the growing expansion of local feature descriptor proposals, Winder et al. [57] published a study were the process of creating such descriptors was broken down in modules to allow that new combinations become easier to make and test. The generic descriptor proposed by the authors is composed of 6 modules/blocks: the image patch, the Gaussian smoothing block, the Transformation (T-Block) which comprises the linear or non-linear transformations or classifiers, the Spatial Pooling (S-Block) to incorporate the distribution-based concept by including some form of histogramming, the Post-Normalization (N-Block) and finally the Descriptor. The idea behind the aforementioned CS-LBP (section 2.3.2) was, in fact, to include it in a SIFT-like local features descriptor. The tests conducted by Heikkilä et al. [4] even show that the CS-LBP descriptor generally provides better results than the SIFT descriptor in addition to being computationally simpler, thus making it very viable.

Other, more recent, descriptor is the Oriented Fast and Rotated BRIEF (ORB) [6], which is based on the Binary Robust Independent Elementary Features (BRIEF) descriptor [58] and the Features from Accelerated Segment Test (FAST) keypoint detector [59]. Rublee et al. combined the two by adding an orientation component to the FAST detector and a rotation invariance to the BRIEF descriptor, which were, respectively their main drawbacks. In the tests conducted by the authors, the ORB descriptor outperformed the SIFT (and also SURF) while outpacing it by more than two orders of magnitude. However, a key shortcoming in ORB might be the possible lack of scale invariance [6].

2.3.4 Global Structures

Local feature descriptors are not really designed to handle scene categorization problems as they tend to focus on local image points. However, that class of descriptors can also be applied to said problems (although with some changes) with interesting results, as they can generally identify the same objects appearing under different conditions (which explains their application in object recognition and matching issues). For instance, Bosch et al. [60] showed that when applying the SIFT to scene description and classification problems, it is better to compute the descriptor in

dense grids instead of sparse interest points (hence the name Dense SIFT). This is a logical conclusion, as information from much more points is computed which almost every time guarantees better results (at the cost of requiring much more computational power). Furthermore, Carvalho et al. [61] showed that the use of a dense scan in alternative to sparse key points vastly improves SURF in relation to SIFT. A similar approach was also taken to develop the Histogram of Oriented Gradients (HOG) descriptor [62], widely used in human detection.

An alternative approach with local feature descriptors would be the use of the visual bag-of-words model [63], as several methods have already been proposed and with fairly good results [64, 65]. The biggest downside with this is that the visual bag-of-words thoroughly ignores the spatial arrangement information. To address this issue, Lazebnik et al. proposed the Spatial Pyramid Matching (SPM) algorithm [66] that systematically incorporates the spatial information into the visual bag-of-words-based descriptors. Yang et al. [67] and Wang et al. [68] later expanded on the SPM algorithm proposing the Sparse Coding Spatial Pyramid Matching (ScSPM) and the Locality-constrained Linear Coding (LLC) respectively.

Oliva et al. [69] suggested that the process of recognizing a scene can be accomplished by only analyzing the global spatial structure of the scene, without much object information, thus creating the concept of global descriptors. Furthermore, the authors also proposed the Gist descriptor, a global descriptor, to analyze and represent said structures. By computing the spectral information in an image through the Discrete Fourier Transform and then compressing the obtained spectral signals with the Karhunen-Loeve Transform, the Gist descriptor is able to obtain fairly good results in scene recognition for outdoor and "natural" categories, struggling a bit more in indoor scenes.

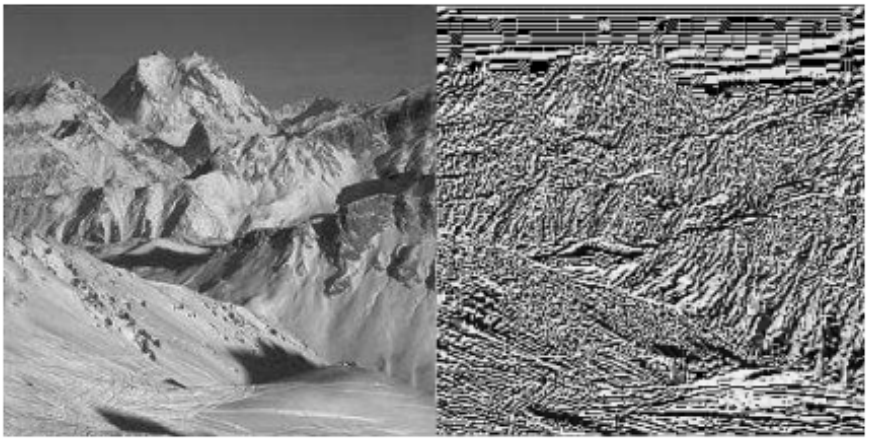


Figure 2.9: Application of the Census transform on a image. On the left, the original image and on the right, the transformed image. Extracted from [70].

To address the Gist and visual bag-of-words-based descriptors shortcomings, Wu et al. [70] proposed the Census Transform Histogram (CENTRIST) descriptor. Based on the Census transform (which is very similar to the LBP operator) CENTRIST captures the structural properties of a scene by modeling the distribution of the local structures. As shown in figure 2.9 , the Census

transform not only retains the global structures of the image, but also gathers information about the local ones. Moreover, the CENTRIST descriptor shows very strong dependency between its components (which allows the application of PCA to reduce the size of the feature vectors) and also encodes the image structures which opens the way for establishing correlations between its feature vectors and the images that they describe. However, it also has some limitations as it only works for grey-scale images and it is not invariant to rotation or scale. While the first is certainly a problem, the latter only means that CENTRIST is not suitable for other applications, because for scene recognition problems, scale invariance is not very relevant and rotation invariance can usually be suppressed. Furthermore, to prove the accuracy of CENTRIST in scene recognizing, the authors realized several tests comparing their proposed descriptor with others like the SIFT and Gist descriptors. CENTRIST showed that not only it is very reliable in outdoor scene categories as it even outperformed Gist, but also far outclasses the other descriptors regarding indoor locations.

Wei et al. [71] recently published a study on the effectiveness of some of the state-of-the-art visual descriptors on scene categorization by using Support Vector Machines (SVM) to train and then classify the different scenes. The authors showed that in those conditions, the SIFT-ScSPM and the SIFT-LLC generally outperform other descriptors.

2.4 Discussion

As far as it is known from the literature review, there is no application of any CV-related approach in a scenario similar to the proposed in this dissertation. However, the problem of event detection may be closely related to scene description and, throughout the last decades, there have been several visual descriptors developed (with different approaches and objectives) in order to describe the scenes observed in digital images or videos. Obtaining a robust scene description is a difficult, time consuming task, but if accomplished, can lead to the detection of changes happening in a region throughout a time period which may indicate the occurrence of events.

To ensure better processing of the interest regions of the scene, the use of image segmentation methods and techniques might prove to be crucial. Although older and simpler techniques like Canny may have their use, the main image segmentation in event detection problems is, without a doubt, foreground and background detection/separation. This is especially true in retail environments as they usually contain a lot of people moving and interacting with each other and with the objects of the shops. So, being able to process the data from the foreground separately from the background might be the only way to properly classify events, instead of just labelling them as changes. While there exist a great amount of proposed approaches about this topic, the most common are generally statistical-based - from the MOG model and its adaptations, to the newer (and far more complex) algorithms like the ViBE or the PBAS. Furthermore, based on its concept, the use of the image depth information for this process should also provide very decent results if the noise level is kept low and negligible.

Color-based description is normally performed through the calculations of color histograms in one (or more) of the various known color spaces. As they are all represent different color

properties, all can have their utility in specific cases and can even be used in succession. However, color histograms are not the only way of describing color (or using it as a part of a description), but are generally accepted as good simple solutions that may be combined with the other descriptors.

Regarding texture (or texture-based descriptors), the general consent seems to be that the best texture operator is the LBP and its variants/extensions. From the latter, the use of uniform patterns may lead to a severe reduction of the method computation time, while maintaining good results. A similar reduction might be achieved with CS-LBP which only calculates center symmetric pairs of pixels, thus halving the number of total calculations. The LTP and its more robust expansion SILTP, were introduced to combat the LBP's problems with local noises. A further expansion to improve the CS-LBP robustness in a resembling fashion to the LTP was proposed as the XCS-LBP. All of them, in addition to CENTRIST (which is based on a transform similar to the LBP) may or may not provide good results in retail environments as not always an event implies alterations in texture and should therefore be studied.

A very different approach was taken by the spatiotemporal LBP (specifically the more recent LBP-TOP) which uses a both spatial and temporal information, instead of just spatial. However, this concept might not be very applicable in retail environments due to the aforementioned constant movement of people and the occlusions of the interest regions caused by that.

Based on their concept, the local feature descriptors should behave well in this specific problem as they are mostly designed for object matching and recognition. However, the constant movement of people may cause slight physical differences in the scenes (for example, affine transformations of objects) for which this type of descriptors are very sensitive, causing a lot a false detections of events. Their application with SPM, ScSPM, LLC and SVMs seems unpractical for this problem as it is quite different from scene classification.

Chapter 3

Workspace Definition

This chapter contains an overview of the dataset utilized in the experiments in section 3.1, a description of the events to consider in section 3.2, and an analysis of the test video sequences and the events they contain in section 3.3. It also contains some considerations taken for the preparation of the ground-truth in section 3.4, and finally, a brief explanation of the evaluation metrics utilized to evaluate the obtained results is presented in section 3.5.

3.1 Dataset Overview

The dataset utilized for this dissertation consists in over 75 minutes of video footage captured from an RGB-D camera (the Microsoft Kinect) from a clothing store in China (Shanghai) with an average framerate of 30fps, a resolution of 640x480p and a fixed position. It was originally intended for application in people tracking problems [14, 15], but due to its real-life scenario, retail environment, crowded scene and various degrees of noise, it can be utilized for this specific problem.

As it is possible to observe in figure 3.1, there are 4 possible well defined different zones or regions of interest (ROIs) to consider. However, the low resolution and quality of the RGB video difficults a proper view and perception of the events happening in the regions 3 and 4 (represented in the yellowish boxes). Moreover, the distance at which those regions are to the camera (mostly outside of the image depth capture range of the Kinect), makes them almost impossible to consider and so the experiments were confined to the regions 1 and 2 (represented in shades of green).

The considered ROIs, despite their closeness in the scene, are themselves very distinct from each other, both morphologically (as region 1 is a stand of shirts and region 2 an underwear rack) and at color level (as region 2 is much more "white"). They are also situated at different distances from the camera which in total makes them almost completely dissimilar and independent at the processing level.

(a) (b) (c) (d) (e)

Figure 3.1: Screenshots from the dataset in different frames. The different colored boxes in 3.1e represent the different ROIs.

3.2 Events Description

The process of defining and characterizing the types of events to detect is crucial because all posterior work depends on this and therefore any unnecessary subjectivity must be removed. Therefore,

two main types of events (with some subtypes) were defined: alteration and occlusion. This may be observed in figure 3.2, a diagram representation of the event tree.

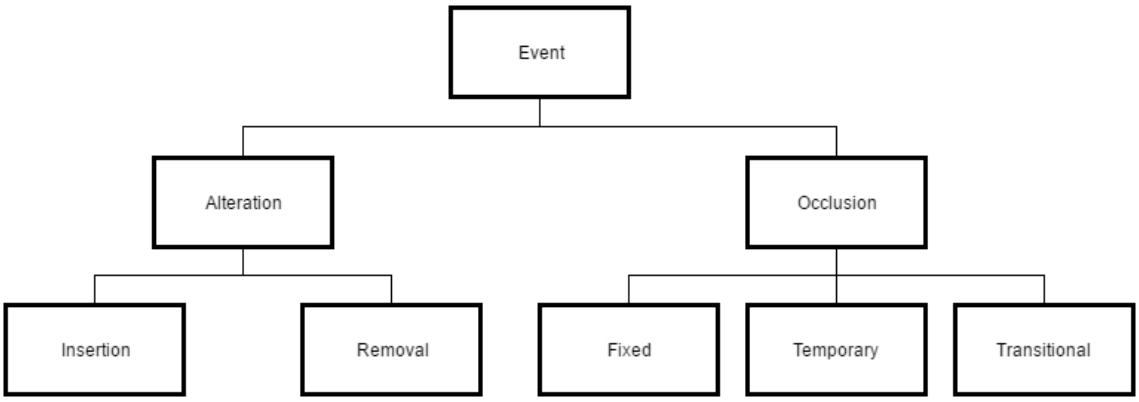


Figure 3.2: Types of Events diagram.

Occlusion has three subtypes. It is defined as the concealment of a region of interest by something or someone for a period of time. The distinction between fixed, temporary or transitional occlusions is made based on the duration of the event: it is considered a transitional occlusion when a person/object only passes by the region in a mere couple of seconds (less then 5) and it is considered fixed when the time period is larger than half a minute (30 seconds). Therefore, temporary is the middle stage. Alteration differs from occlusion as it implies that the state of the region was changed from a previously known state in a specific time frame. In other words, when comparing two separate time frames (before and after the event) the region is different, it suffered changes. Insertion and removal then refer to whether an object was inserted or removed from the region.

3.3 Video Sequences

(a) (b)

Figure 3.3: Screenshots from different video sequences to illustrate differences between them. 3.3a represents sequence 1 and 3.3b represents sequence 5.

From studying and analyzing the dataset and based on the types of events defined, 8 sequences were extracted focusing on events happening in the two ROIs. They vary on total number of events, expected level of difficulty and contain events that differ a lot from each other. For instance, as represented in figure 3.3, sequence 1 has a low amount of people moving and only an insertion of a shirt in the stand (ROI 1), but sequence 5 has a person standing in front of that same stand and occluding it for a long period of time in addition to many other smaller occlusions. It is also worth mentioning that all of the sequences were selected and cut from the overall video in order to ensure that do not contain events in the first few seconds (at least 10).

Table 3.1 represents the general information of the video sequences and the events they contain by ROI. This includes the total number of frames, the total number of events and their types (both occlusions and alterations).

Table 3.1: Video Sequence Information

			Occlusions			Alterations	
Sequence	Total Frames	ROI	Fixed	Temporary	Transitory	Insertion	Removal
1	1945	1	-	-	1	1	-
		2	-	-	1	-	-
2	2170	1	-	-	2	-	1
		2	-	-	1	-	-
3	2022	1	-	-	4	1	-
		2	-	1	-	-	-
4	2612	1	-	-	6	-	1
		2	-	-	6	-	-
5	2816	1	1	1	5	-	-
		2	-	-	9	-	-
6	2971	1	-	1	6	-	-
		2	-	1	4	-	-
7	3965	1	-	1	1	-	-
		2	-	2	4	-	-
8	3850	1	-	3	2	-	-
		2	1	1	3	-	-

A couple of notes may be taken from table 3.1: first, the number of occlusions is way higher than the number of alterations; and second, it's clear that the most common event are the transitional occlusions. This makes sense in a way that people walk by stands or racks in shops much more than they interact with them. Furthermore, it is also important to refer that simultaneous occlusions occurring in the same ROI (for example, two people passing by) were only counted as 1 event, encompassing the total period in which the region was occluded. Finally, an alteration is only considered as such if there was a change from a previous known state and the ROI is not occluded at that instant. That also includes a case like sequence 4, in which there is an alteration in ROI 1 which is later reversed to the original state - only one event is considered as there was technically only one alteration.

3.4 Preparation of the Ground-Truth

The preparation of the Ground-Truth involved the annotation of the events described in sections 3.2 and 3.3 in .xml files (one for each sequence). Along with the event type, sub-type and region mentioned and the usual frame numbers and/or intervals, all the events annotated in the ground-truth have a "level of trust" that varies in the interval [0, 10], with 0 being absolutely unsure and 10 absolutely certain about the annotation. This information is not so much an attribute of the specific events, as it is an attribute of the annotation as a whole, due to it being a very subjective procedure. Moreover, it may provide further information to a latter study of the tests results. For instance, it is expected that the algorithm generates better results when the confidence in an event is 8, comparing to when it is 1.

Figure 3.4 may provide a better understanding of when an occlusion was considered and annotated in the Ground-Truth and when it was not, as they are a much more subjective type of event than alterations.

(a) (b) (c)

Figure 3.4: Screenshots from ROI 1 in different frames. 3.4a is considered an occlusion as is 3.4b. 3.4c is not.

3.5 Evaluation Metrics

In order to proper evaluate the results obtained by each test method with the highest degree of objectivity possible, concrete evaluation metrics must be defined. As is common in CV related problems, the utilized procedure was the computation and analysis of Receiver Operating Characteristics (ROC) curves and Area Under the Curve (AUC) and F-Measure (also known as F-1) scores as described by Fawcett [72]. All of these evaluation metrics are based on the confusion matrix, represented in figure 3.5 (even AUC, which is calculated of the ROC curve, that itself is obtained from the confusion matrix). The matrix is constructed by considering the data from the Ground-Truth as the true class (the instances) and the values obtained from the experiences as the hypothesis class (the classifications). In a binary situation (it is or it isn't), as in figure 3.5, there are only four possible outcomes: the true positives represent the cases when the instance is positive

and the classification is also positive; the false positives when it the instance is positive, but the classification is negative; the true negatives when both are negative; and the false negatives, when the instance is positive, but the classification negative [72].

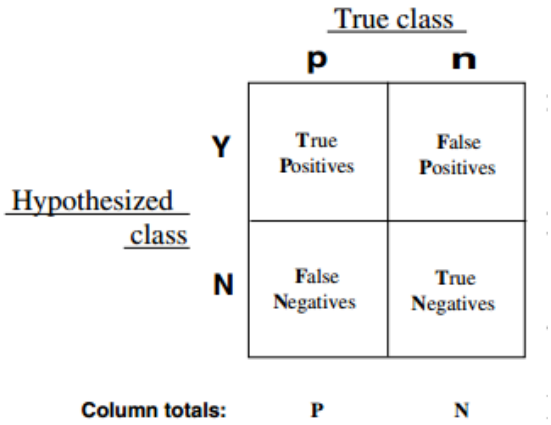


Figure 3.5: Confusion Matrix representation. Extracted from [72].

After the construction of the confusion matrix, the true positive rate (TPR), also known as recall, or probability of detection is calculated as

$$TPR = \frac{True_{P}ositives}{TotalPositives} = \frac{True_{P}ositives}{True_{P}ositives + False_{N}egatives} \tag{3.1}$$

The false positive rate (FPR), or probability of false alarm is calculated as

$$FPR = \frac{False_{P}ositives}{TotalNegatives} = \frac{False_{P}ositives}{True_{N}egatives + False_{P}ositives} \tag{3.2}$$

The precision, or the positive predictive value (a measure on the relevance of the obtained values) is obtained as

$$Precision = \frac{True_{Positives}}{True_{Positives} + False_{Positives}} \tag{3.3}$$

A ROC curve can be obtained by varying the thresholds utilized on the classification values, in order to obtain the best possible combination of TPR and FPR (and consequently the best threshold(s) value(s)), from equations 3.1 and 3.2. The AUC is then estimated as

$$AUC = \int_{+\infty}^{-\infty} TPR(T) \times FPR'(T)dT \tag{3.4}$$

The F-Measure is simply an harmonic mean of the precision and recall scores, obtained as

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.5}$$

Chapter 4

Methodology

As mentioned in section 1.3, the main objective of this dissertation was to achieve a viable detection of a predetermined set of events happening in a retail environment through the spatiotemporal processing of video captured by generic RGB and RGB-D cameras. Chapter 3 set the workspace by covering the definition and description of the test video sequences, the set of events to detect and the evaluation metrics to evaluate the tests results. This chapter then focus on the description of the actual processing utilized to accomplish the defined objective.

The first and very high-level approach taken was that in order to accomplish the event detection, some type of processing algorithm had to be applied to the video sequences. From there, based on the data available from the dataset and the information retrieved from the literature review from chapter 2, three separate strategies were defined to survey the problem. As it may be observed in figure 4.1, the processing was divided in: using only RGB video, only depth information and using the sum of the two. Thus, sections 4.1, 4.2 and 4.3 cover each one respectively, providing a detailed description of the implemented algorithm, its purpose and known (or expected) limitations.

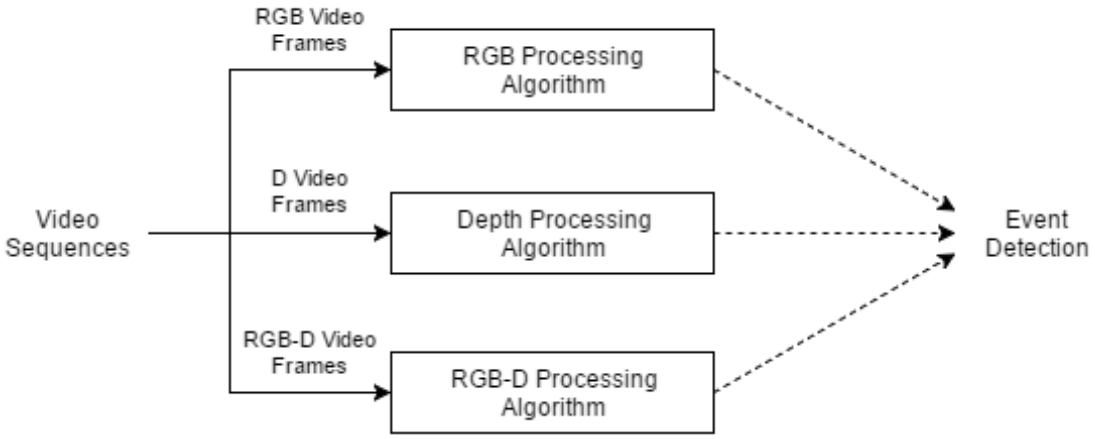


Figure 4.1: Block diagram of the first approach.

4.1 RGB Processing

As explained in section 2.1, most of the modern video capturing devices are RGB cameras. Therefore, a natural approach to the event detection problem, would be to first develop a solution using only that information, not only to test what type of results may be achieved, but also to understand the limitations behind it.

As reviewed in chapter 2, the construction of visual descriptors from the local properties or features of RGB data allows the representation and analysis of the different characteristics of shape and appearance of an image. Thus, the comparison of descriptions obtained from distinct images (for instance, different video frames) may allow a measure of the differences between said images. However, without resorting to any tracking mechanisms or image segmentation methods (namely foreground/background separation), no actual recognition of distinct types of events is possible to achieve. Nevertheless, as differences between images can be detected, several of the reviewed visual descriptors were tested in order to understand how each one performed in the detection of changes happening in the defined ROIs. As all of the events from the ground-truth theoretically represent changes in the scene (when compared with an initial state), they were considered as such for this specific processing stage - an event either exists in a ROI, or it doesn't.

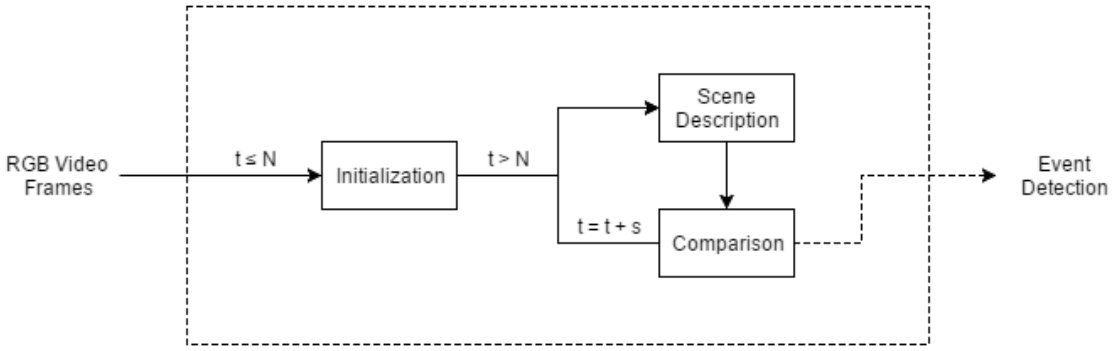


Figure 4.2: Block diagram of the RGB Processing Algorithm.

Figure 4.2 contains the block diagram of the RGB processing algorithm referred in figure 4.1. The algorithm is divided in two steps: the initialization, in which a model for the visual descriptor under test is created (representing the initial state of each ROI), and the comparison loop, where that model is compared with the description obtained from specific frames to detect the existence (or not) of events in the ROIs. For the initialization, only the first N frames are considered, leaving the remaining to the loop. The parameter s then represents the amount of frames skipped in each iteration of the loop.

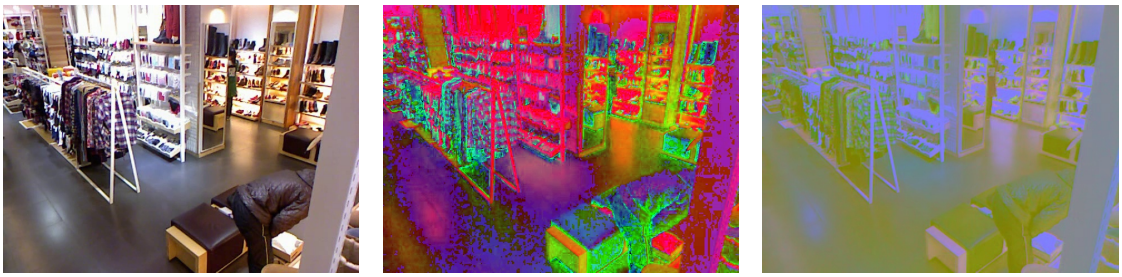
The following section 4.1.1 contains an overview of the tested visual descriptors, along with some considerations regarding their choice and some figures representing their application. Section 4.1.2 then contains information about the methods used to compare the descriptors with their created model.

4.1.1 Visual Descriptors

The objects contained in retail environments (such as the clothes from the clothing store of the test dataset) are typically very colorful and with distinct visual patterns and features. So, it seems coherent to try to detect the occurrence of events based on variations of those properties in the scene. Therefore, from the literature review in chapter 2, a selection of a test group of visual descriptors was made. To ensure that that selection was as comprehensive as possible (as the number of descriptors available in literature, built from the diverse image properties, is quite large), the test group includes: Color Histograms, Local Binary Patterns (LBPs), the CENTRIST descriptor and Local Feature Descriptors - sub-sections 4.1.1.1 to 4.1.1.4.

4.1.1.1 Color Histograms

As mentioned on section 2.3.1, histograms may be computed from color images to achieve a scene description based on that property. Even though this is a simple concept, it allows a factual representation of color quantities (whichever the color space), which can lead to a proper analysis of its variations during the occurrence of events. From the reviewed color spaces, two were selected for testing purposes: the HSV and the CIELab color spaces. Their selection was due to the fact that although they are conceptually different, both represent a perception on color similar to the one from the HVS [35] and allow a separate measure of variations of lightness and chromacity. Moreover, given that the video sequences are represented in the RGB color space, this color space was also tested. Figure 4.3 represents an example of a video frame (specifically, 320 of sequence 4) in the different considered color spaces.

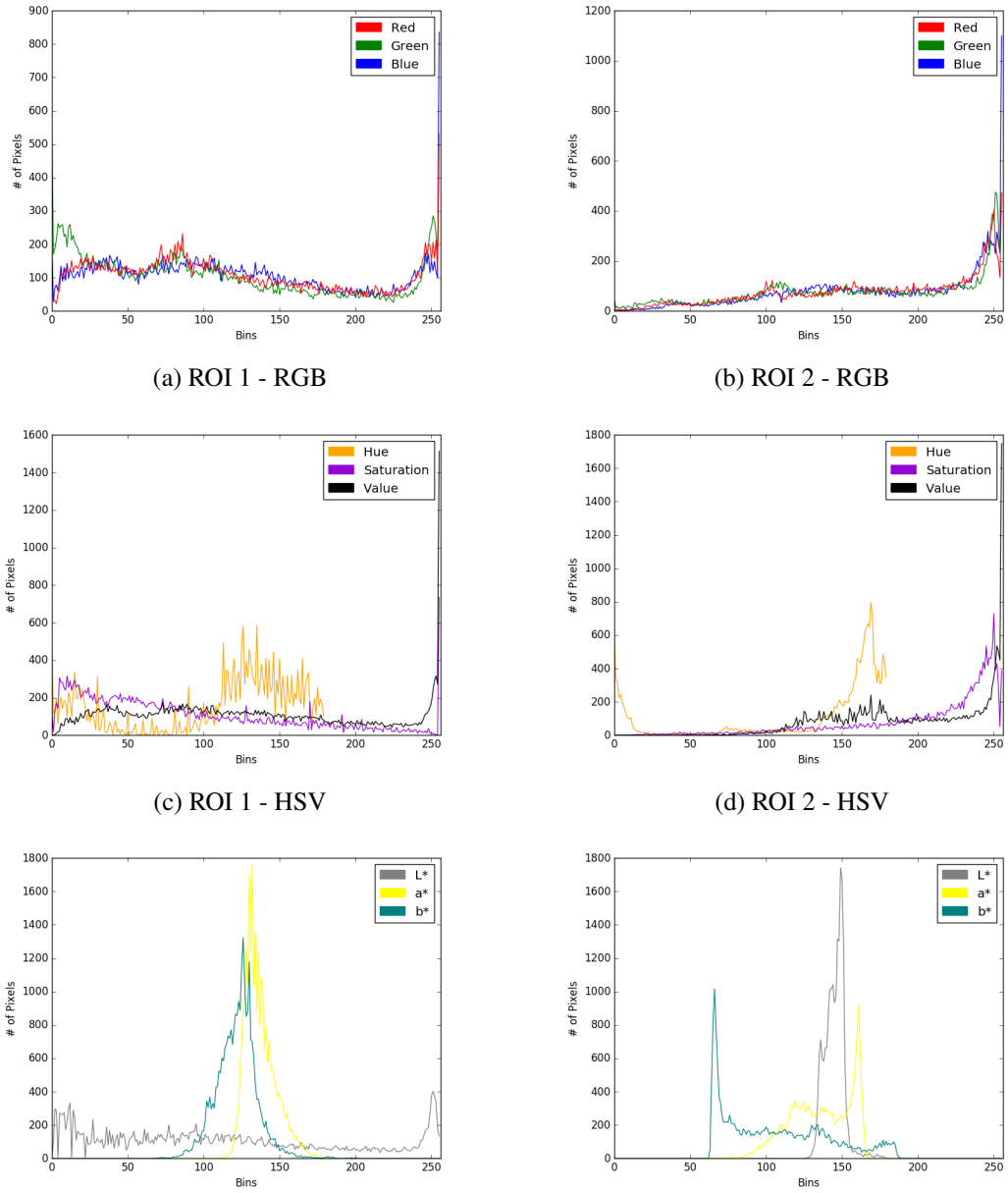


(a) RGB (b) HSV (c) CIELab

Figure 4.3: Representation of an example video frame in the different considered color spaces.

The initialization process for color histograms (the creation a model for each ROI) is pretty straightforward: the RGB video sequence is converted to the desired color space (if needed) and histograms are computed (one for each ROI) for the frame N. This process is then repeated for the remaining frames, in the loop, so that a comparison between histograms (the models and the current ones) may be made.

Figure 4.4 represents the color histograms (from the different color spaces) obtained for each of the ROIS from the same video frame (320, sequence 5) of figure 4.3. From observing the figure and as expected from the analysis of the dataset in chapter 3, it is possible to notice that the ROIs



(a) ROI 1 - RGB

(b) ROI 2 - RGB

(c) ROI 1 - HSV

(d) ROI 2 - HSV

(e) ROI 1 - CIELab

(f) ROI 2 - CIELab

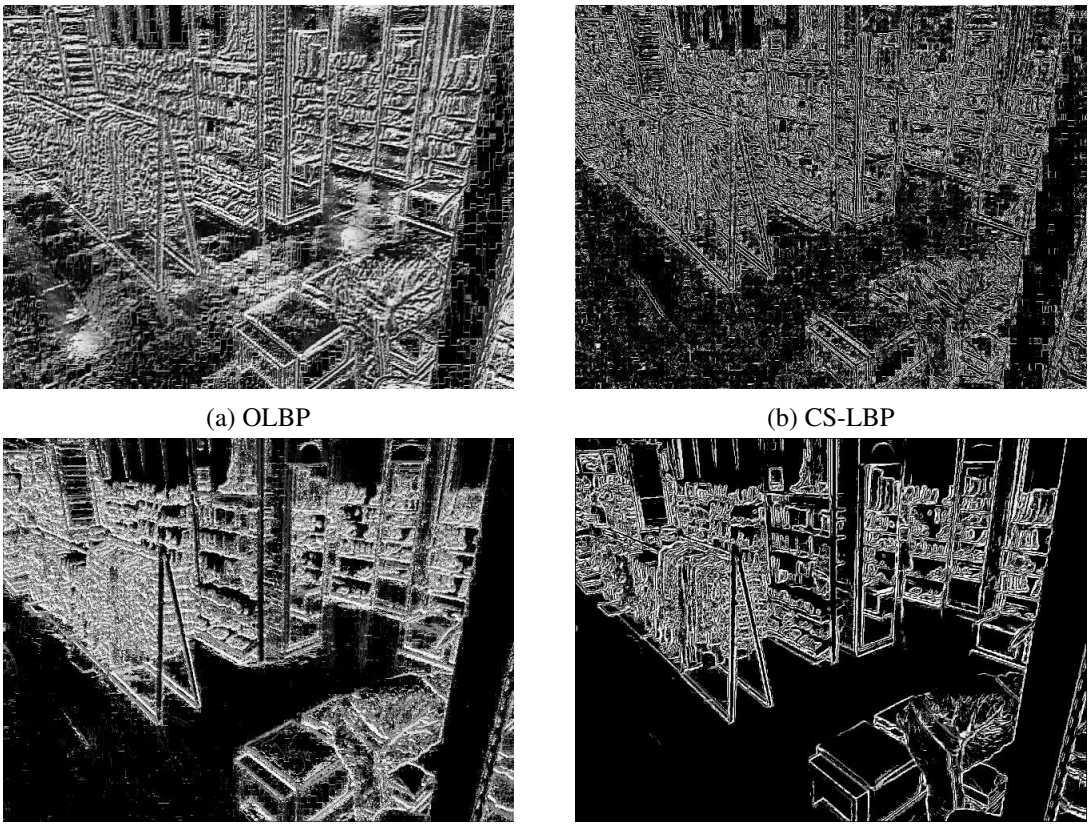
Figure 4.4: Computed color histograms for each ROI from an example frame. 4.4a, 4.4c and 4.4e represent ROI 1 and 4.4b, 4.4d and 4.4f, ROI 2.

are very distinct at a color level. Specifically, ROI 2 has most of it pixels around the white region of the histograms - around 255 for the all curves of the RGB histogram, around 0 and 180 for the hue and 255 for the saturation and value of the HSV, and around the peak value of L* for CIELab (from the conversion of RGB to CIELab, L is obtained in the range $[0, 100]$ and then normalized to cover the range $[0, 255]$). On the other hand, ROI 1 has its pixels much more distributed around the each curve, with the exceptions being the values of around 255 for saturation of HSV (representing the "pure colors" observed in the region) and the peaks of a* and b* for CIELab (representing that

most of the colors in the region are around those values, for that specific color space).

4.1.1.2 Local Binary Patterns

As reviewed in chapter 2, the resort to the LBP and its variations is probably the most popular approach to the use of texture for the description of scenes, due to their clear and easy concept that enables the encoding of local primitives from images (flat areas, spots, edges, etc.) [44]. From all the variations surveyed, four were selected for testing purposes: the original LBP (from now referred to as OLBP) [41], the CS-LBP [4], the SILTP [43] and the XCS-LBP [44]. An example of the output image of each of them is represented in figure 4.5.



(a) OLBP (b) CS-LBP

(c) SILTP (d) XCS-LBP

Figure 4.5: Application of the selected LBPs to an example video frame.

The OLBP labels the pixels of image blocks by thresholding the grey-scale values of each pixel in a circular neighbourhood with its center value and considering the result as a binary number. In turn, the CS-LBP works in similar fashion, but reduces the number of total calculations in half as it only does the thresholding process to center-symmetric pairs of pixels. The XCS-LBP further expands that concept by also considering the intensity value of the center pixel to the calculations. Finally, the SILTP represents an expansion to the LTP [42] (which simply adds a tolerative range to the OLBP) analogous to that of the XCS-LBP in relation to the CS-LBP. The observation of figure 4.5 may allow a better perception of the differences between the described methods. For

instance, it is clear that each of them retains gradually less information (from OLBP to XCS-LBP) from the scene, which is even more noticeable in flat areas (like the floor) what may help to filter some of the visual noise. Furthermore, the SILTP and particularly the XCS-LBP (figures 4.5c and 4.5d) seem to mostly retain the edges of the scene, while the other two methods generate outputs much harder to visualize.

The application of the selected methods in the processing algorithm (initialization and comparison loop) is identical to the described above for the color histograms - the texture operator is computed for the scene and then, for the resulting image, the histograms for each ROI are calculated.

4.1.1.3 CENTRIST Descriptor

Figure 4.6: Application of the CENTRIST descriptor to an example video frame.

Global descriptors, as reviewed in chapter 2, were designed for image classification problems, and thus focus on the global structural properties of scenes without regarding much object information. So, for a better perception if that type of approach is viable in the detection of changes in a scene, the CENTRIST descriptor [70] was included in the visual descriptors test group for this specific problem. The specific choice of CENTRIST was made based on the fact that it captures the global properties of a scene by modeling the distribution of the local structures and thus doesn't completely ignore them.

Figure 4.6 represents the application of the CENTRIST descriptor in the same given example video frame (320, sequence 4). Furthermore, as CENTRIST has a working principal similar to the aforementioned OLBP [41] - the Census Transform is essentially the LBP texture operator with a different bit ordering in the binary result - and so their application in the designed processing algorithm (the model creation and the comparison loop) was analogous.

4.1.1.4 Local Feature Descriptors

From the local feature descriptors reviewed in chapter 2, three were selected for testing purposes: the SIFT [5], SURF [55] and ORB [6] descriptors. The decision to include the first two was made based on their longstanding relevance in the CV community and CV-related problems. On the other hand, ORB was selected exactly for the opposite reason, as it is a much more recent, state-of-the-art, descriptor. Moreover, all of them follow an identical procedure - the retrieval of local interest points (also known as keypoints) from images to build feature vector descriptors - but are conceptually very different, which offers a better surveying on the performance of local feature descriptors in the event detection problem.

As these type of descriptors differ quite a lot from the others considered above (sub-sections 4.1.1.1 to 4.1.1.3), their application in this problem is also distinct. The model creation was achieved through successively matching the description vectors obtained for each of the ROIs (and for each of the local feature descriptors) from frame $N-s$ to frame N (which, as known, represent a "clean", uneventful scene and should then be a total match). The resulting number of total keypoints was then divided by s to obtain an average value to be used as comparison. The same exact principal is applied to the comparison loop, where that matching process is made between the ROIs in frame N and the ROIs in the current frame. The resulting values of each iteration are then compared with a thresholded percentage of known model values which allows a decision of the occurrence or not of an event in that specific video frame. In other words, if the number of keypoints obtained from the matching process drops below a certain determined number, the algorithm considers that that ROI is different from what is expected and thus contains an event. The way the matching was performed for each descriptor is explained in the following section 4.1.2.

4.1.2 Comparison Methods

The comparison of the histograms obtained from each iteration of the loop with the created models, for the descriptors mentioned from sub-section 4.1.1.1 to 4.1.1.3, was made by measuring the Hellinger distance between them. For two discrete probability functions (two histograms) $H_1 = (h_{1,1}, h_{1,2}, ..., h_{1,k})$ and $H_2 = (h_{2,1}, h_{2,2}, ..., h_{2,k})$, the Hellinger distance represents a numerical measure of the overlap between them and is given by

$$(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} \tag{4.1}$$

If the measured overlap between H_1 and H_2 is high, it indicates that the distributions are similar and thus, their Hellinger distance is small. Furthermore, it is worth noting that it is a normalized function in the interval $[0, 1]$. In this specific case, as the images and their histograms are more dissimilar, the bigger the value of Hellinger distance will be. Then, if a threshold value is defined and the distance calculation surpasses that value, an event has been possibly detected.

As explained above, in sub-section 4.1.1.4, the same process of comparing the histograms cannot be applied to the Local Feature descriptors, due to their different concept. The described model for each ROI (the average of total of keypoints matched between the s identical images) and the comparison value were then obtained by applying a brute-force matcher with ratio test, as described in [5]. That matcher takes the descriptor of one feature (one obtained keypoint) in the first set (first image) and tries to match it with all the others in the second set (second image) using a given distance calculation and returns the two best matches. The ratio test guarantees that if those matches are too far apart, that keypoint is not good and thus should be ignored. While for the SIFT and SURF descriptors, the distance calculation was made using the Euclidean distance, which for two points a and b represents the line segment connecting them ($\overline{ab}$), for ORB, due to its binary descriptor status, the used distance was the Hamming distance, which measures the minimum number of "substitutions" required to change one feature vector into the other. On figure 2.8, an example of that matching process may be observed for the SURF descriptor in ROI 1. For a better perspective, only the 10 best matches between images were drawn.

Figure 4.7: Representation of the 10 best matching keypoints obtained for an example frame for ROI 1, using SURF.

4.2 Depth Processing

The use and processing of image per-pixel depth information opens the possibility of segmenting the scene into background and foreground, which, contrary to the pure RGB approach described in section 4.1, allows a proper acknowledgement and classification of occlusions. As mentioned in section 2.2, the constant subtraction of data from the current frame to a known background model, enables the achievement of a mask representing the foreground due to the different depth values. For instance, if there is an object, in one of the ROI(s), in a specific frame that wasn't there in the background model, it is expected that that object is represented in the foreground mask, due to its different distance to the camera in relation to the model (and consequently different depth values).

However, the same can not be said to the detection of alterations, as their concept represents that the change was within the same distance range as the previous known values (the depth variation is too small to be properly detected). Therefore, the events considered for testing purposes of depth-only processing were restricted to the occlusions.

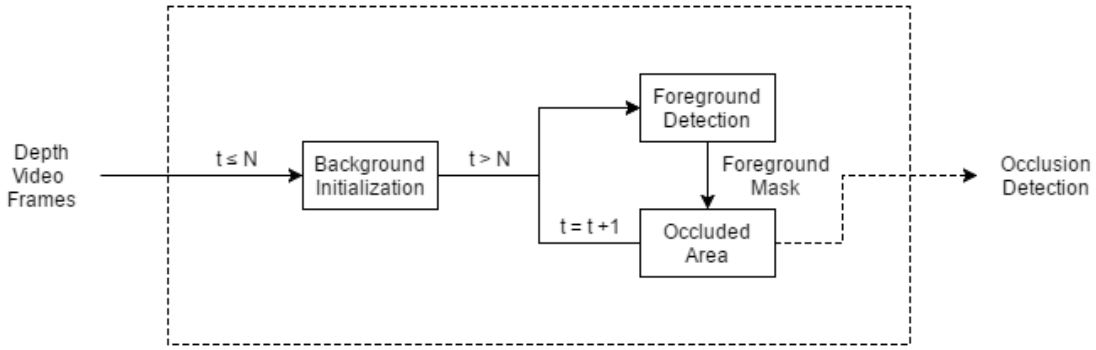


Figure 4.8: Block diagram of the Depth Processing Algorithm.

Figure 4.8 represents the block diagram of the depth processing algorithm from figure 4.1. The algorithm consists in the institution of a background model through the first N frames - Background Initialization, the creation of a foreground mask through the comparison of the remaining frames (in a loop) with the known model - Foreground Detection, and the calculation of the percentage of area occluded for each ROI - Occluded Area. A full explanation of these blocks is presented in sections 4.2.1 (Background Initialization), 4.2.2 (Foreground Detection) and 4.2.3 (Occluded Area).

4.2.1 Background Initialization

The concept of creating a background model requires the test ROIs to be without events for a defined N frames to ensure that the model actually represents "clean", unnocluded ROIs so that a proper detection of occlusions may be achieved, as explained above. This is similar to the procedure defined in section 4.1.

Figure 4.9 represents the frame 150 (from now used as representative example value for N) of the depth video sequence 2 of the dataset. Based on the review from section 2.1, it is possible to observe that it contains large zones without depth information (black) and various points with noisy estimates (white points). To cover those issues, three major steps were defined: first, instead of taking the sample from just one frame, the arithmetic mean of the last s was computed; second, a low-pass filter was applied to filter the noisy white points; and third, a median filter to add blur to the image to further smooth the information. As figure 4.10 can attest, by utilizing more than just one frame, some of the black zones (at least in the ROIs) were covered with actual depth measures. However, it also shows that without the application of the filters, the usage of more frames would induce even more noisy points.

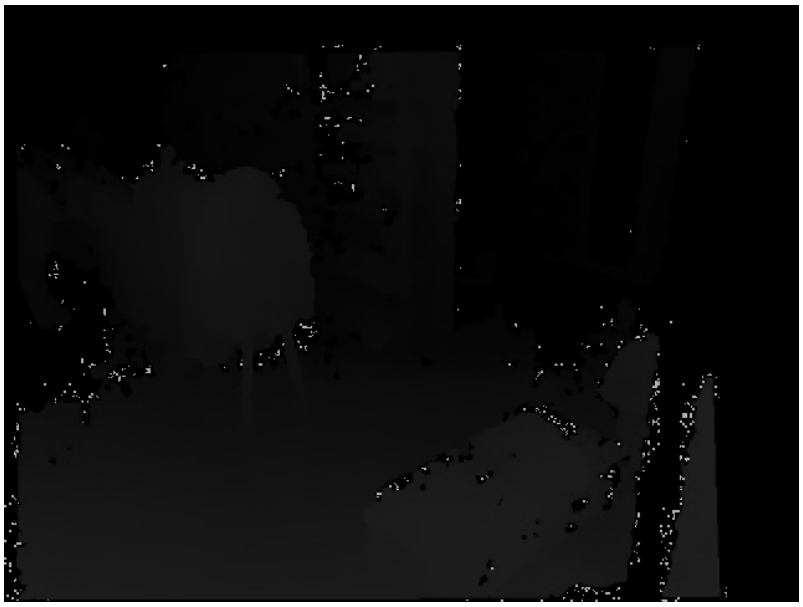


Figure 4.9: Depth image of an example video frame.

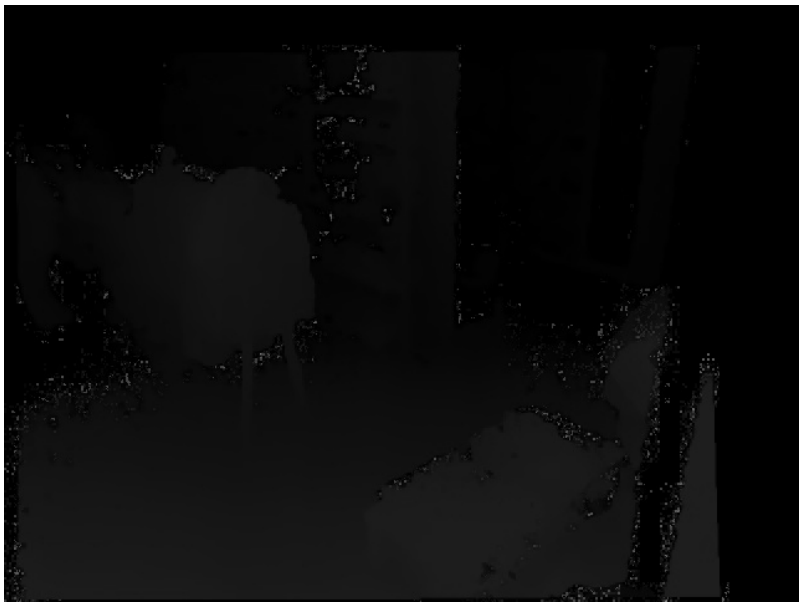


Figure 4.10: Representation of averaging s depth video frames.

The concept of a low-pass filter is simple: only the values below a defined value a are kept, the rest are removed. In this case, the range of values corresponds to the interval of a grey-scale image ($[0,255]$), so a was empirically defined as 50. As for the median blur filter, it replaces the value of the pixels in a $k \times k$ neighbourhood with the median of said values [73]. To ensure that the image was as smooth as possible, without losing too much information, k was empirically set as 9. The result of the application of these filters may be observed in figure 4.11.

Figure 4.11: Resulting depth image of the filtering process.

The final step of the background modelling process consisted in an equalization of the histogram of the image represented in figure 4.11. This was done to normalize the brightness and increase the contrast of the image. It was achieved by calculating the histogram of the image, normalizing it so that the sum of all the bins equals 255 (the range of a grey-scale image) and then transform it using $Image_{destination}(x,y) = H'(Image_{source}(x,y))$, where

$$H'_i = \sum_{0 \leq j < i} H(j) \tag{4.2}$$

The resulting image was considered the background model and may be observed (for the example given for sequence 2) in figure 4.12.

4.2.2 Foreground Detection

After obtaining the background model and while the video sequence is not over, the foreground mask is achieved through the absolute subtraction of a similar processing of the next s frames to the known model, in a loop. In other words, for the duration of the video sequence, the successive next s frames are processed the exact same way as the s frames that defined the background model and then the absolute difference of the two is calculated to obtain a foreground mask. An example of a mask obtained for sequence 2, from the background model shown in figure 4.12 and the processing of the frames in the interval $[371, 380]$ (using $s = 10$), is represented in figure 4.13.

In order to be able to calculate the occlusion (or not) of one of the ROIs in the next step, the obtained mask needs to be thresholded so that it becomes a binary image (where pixels only take values of pure white or pure black). The threshold value that separates if a pixel is white

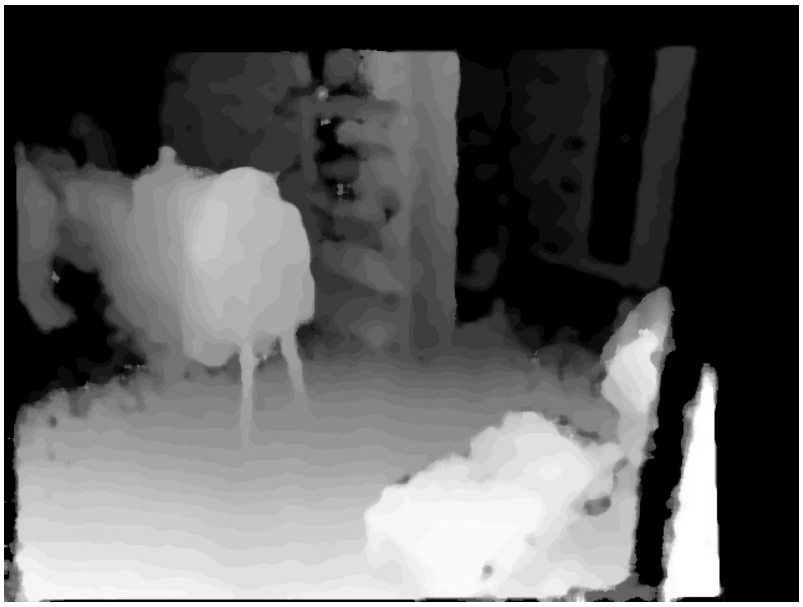


Figure 4.12: Resulting depth image of the histogram equalization.

Figure 4.13: Example of an obtained foreground mask.

(foreground) or black (background) was then empirically set as 30: if a pixel intensity is over 30 it is set to black (255), otherwise, to white (0).

To remove the white noise associated with the calculation of the mask and the thresholding process, two last filters were applied: erosion, followed by dilation, which is also known as an opening filter. Both run through the image performing successive convolutions of a kernel of size $k \times k$ with a region of the same size. However, they do opposite things: as the first (erosion) sets

to zero (0) all the white blobs smaller (or equal) than the kernel, and the second sets to one (255) all the pixels in a region the size of the kernel, if there is at least one non-zero. This is very useful as it allows the removal of some of the smaller white blobs (usually noise) while not affecting (of course, depending on the size of the kernel) the bigger ones. The kernel utilized in this procedure was a block of 5×5 ones which means that the convolutions were performed with the whole region of the kernel. The final foreground detection output (the final foreground mask) for the example given in figure 4.13 may be observed in figure 4.14

Figure 4.14: Thresholded and filtered version of the foreground mask from figure 4.13

4.2.3 Occluded Area Calculation

From the foreground mask, it is possible to obtain a measure of the occluded area in each of the ROIs by counting the number of non-zero pixels (which in this case, is equal to counting the number of white pixels) in each ROI and dividing it by its total area. In other words, for both ROIs, the percentage of occluded area is given by equation 4.3. For a better perception, figure 4.15 represents the mask from figure 4.14 in each of the ROIs.

$$O_{ROI}(\%) = \frac{\sum NonZero}{Area_{ROI}} \times 100 \tag{4.3}$$

Although it is clear that the ROIs still contain a noticeable degree of noise (which is expected due to the use of image depth information, as mentioned above), the application of equation 4.3 may provide a useable measure of the area occluded. For instance, for the example images in figures 4.15a and 4.15b, which are both annotated as occlusions in the ground-truth, values of 23.4% and 25.5% were obtained, respectively. Thus, the definition of an optimized threshold value may to help to achieve a viable detection of these events.

(a) ROI 1

(b) ROI 2

Figure 4.15: Separation of figure 4.14 to both ROIs.

4.3 RGB-D Processing

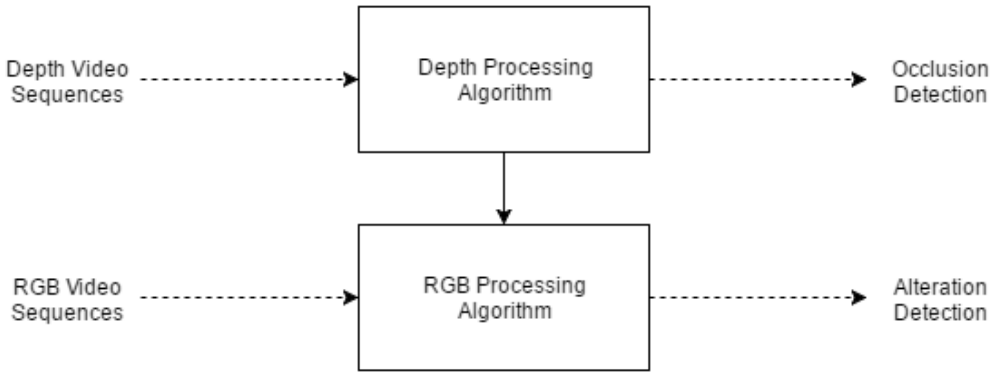


Figure 4.16: Block diagram of the RGB-D processing algorithm.

The RGB video processing algorithm from section 4.1 allowed for no more than the detection of changes happening in the scene. However, as explained in section 4.2, the processing of the image per-pixel depth information for the segmentation of the scene into background and foreground permits the detection of occlusions. Thus, if used in cascade, the combination of both algorithms may accomplish a viable detection and classification of all the considered events through a "divide-and-conquer" strategy. That is, if the processing of the depth information detects the occurrence of an occlusion and transmits that information to the RGB processing algorithm, it will know that it doesn't need to process that specific frame. If on the other hand, the depth processing algorithm detects no current occlusion, it means that if the RGB detects changes in the scene, they represent alterations. The figure above, figure 4.16, represents this final approach to the problem, with the

blocks represented denoting the algorithms described in previous sections.

Chapter 5

Results and Discussion

In this chapter, the tests results from the application of the methodologies proposed in chapter 5 in the workspace defined in chapter 3 are evaluated, analyzed and discussed. Furthermore, the whole testing procedure is described, providing some insight on the decisions that were made.

As already stated, three separate, but complementary, strategies were developed in order to accomplish the objective defined in this dissertation: the processing of RGB video information, of image per-pixel depth data, and the sum of both. The following sections (5.1 to 5.3), then contain the experimental results of each.

The application of the proposed methodology implied that some parameters needed to be set. As mentioned in section 3.3, the video sequences do not contain any event in the first 10 seconds and so, for testing purposes, the parameter N, referred in chapter 4, was set as 150, which is equivalent to 5 seconds. In turn, the parameter s (which represents the amount of frames skipped in each iteration of the loop, was set as 10 ($\frac{1}{3}$ of a second). Although technically s may assume any value in the interval $[0, T-N-1]$, where T represents the total duration of the video sequence in test, it was set to 10, as testing every single frame would be typically unpractical in a real-life application, where 3 tests for each second seems more logical. Also, and more importantly, each of the test visual descriptors (particularly the local feature descriptors) also contains certain internal parameters, which were all set exactly as defined by the authors to avoid the induction of external errors and thus misleading results. Finally, it is important to mention that all the testing procedure was implemented using the Open Source Computer Vision Library (OpenCV) library [74, 75, 76].

5.1 RGB Processing

As explained in section 4.1, the RGB processing algorithm was designed to obtain a perception of how the reviewed visual descriptors performed when trying to detect visual changes happening in the test dataset (specifically in the two ROIs). The evaluation of that performance was accomplished through the construction of ROC curves and the calculation of their AUC score, as previously mentioned. The table below, table 5.1, represents the obtained AUC score for each of

the descriptors tested (HSV depicts the analysis of the three channels in the HSV colorspace, H-HSV the analysis of solely the Hue channel, SV-HSV the analysis of just the saturation and value, and so on).

Table 5.1: Visual Descriptors and their AUC scores.

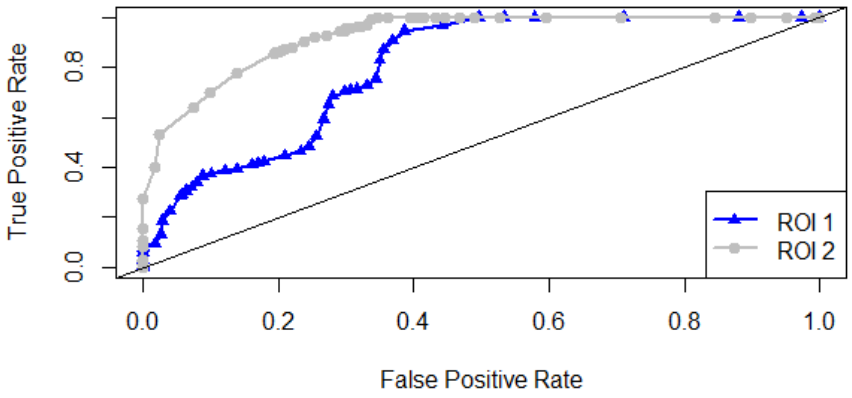
		AUC	
Visual Descriptors		**ROI 1**	**ROI 2**
Color Histograms	RGB	0.7987597	0.929002
	HSV	0.796746	0.927043
	H-HSV	0.762366	0.886007
	SV-HSV	0.819632	0.927579
	CIELab	0.799224	0.924422
	L-CIELab	0.794264	0.909139
	AB-CIELab	0.789040	0.886609
LBPs	OLBP	0.710723	0.912651
	CSLBP	0.658013	0.775411
	SILTP	0.645383	0.782582
	XCSLBP	0.769438	0.651480
CENTRIST		0.563244	0.613861
Local Features	SIFT	0.876096	0.977278
	SURF	0.912470	0.977373
	ORB	0.857218	0.968225

Promptly, a brief overview of the table above enables the perception that the local feature descriptors obtain the best results (for ROI 2, even very close to a perfect score of 1) and CENTRIST the worst (very close to the score of 0.5 that represents chance). Furthermore, it also allows to observe that the general detection of visual changes is worse in ROI 1, than it is in ROI2, which coincides with the preliminary analysis of the difference between the ROIs, back in chapter 3 - ROI 1 is very homogeneous, both at color and texture level, while ROI 2 is much more sharp, with clear edges, and its color values center around white. By analyzing the table with more detail, it is possible to understand that a color-based analysis provides mixed results. While all the considered color spaces obtain similar AUC scores, it is clear that looking for alterations in chromacity is generally inferior than looking for variations of lightness. This was expected particularly in ROI 1, as the region has a very distributed color representation (as depicted in the color histograms presented in section 4.1.1.1) and so, slight variations of chromacity should be much harder to detect than variations of light.

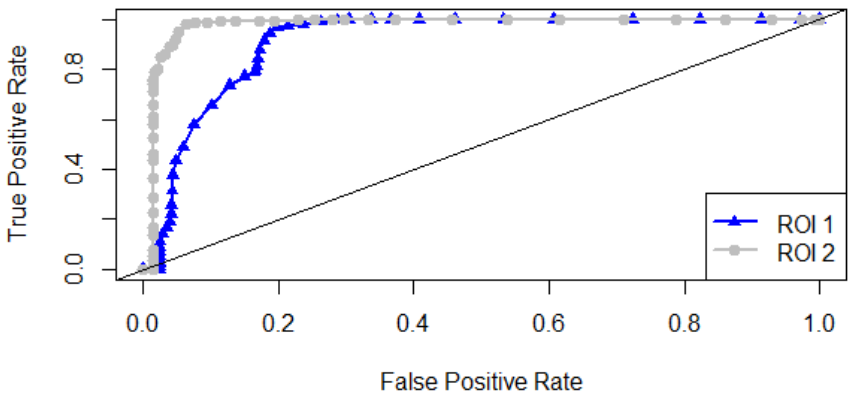
The analysis of texture (LBPs) follows a similar path to that of color (although much worse). Again, ROI 1 is very homogeneous in terms of texture (figure 4.5) and the events are mostly caused by people either inserting or removing a piece of clothing from the ROI or walking/standing by it, dressed in similar fashion, which doesn't alter much the texture of the region as a whole. This may be comproved by the much better AUC score of the OLBP for ROI 2. The better performance of the OLBP in comparison to the other LBPs may be attributed to the fact that it doesn't discard much of the data that the others do (none of the ROIs represent "flat" areas). This whole idea also

applies to CENTRIST, which tends to neglect local information to focus on global structures of the scenes (which, in this case, do not suffer very noticeable changes).

Lastly, the performance of the Local feature descriptors is clearly the best (for both ROIs). This was expected as due to their concept designed for application in object detection and matching problems, they are able to detect most of the small changes (that still represent events) that other descriptors are not. Furthermore, the better performance of SURF in relation to the other two (SIFT and ORB) proves the points made above about color, as SURF is typically more robust to illumination variations [77].



(a) CIELab



(b) SURF

Figure 5.1: ROC curves obtained for two of the tested visual descriptors.

To allow a better perception of the presented AUC scores, figure 5.1 represents two of the obtained ROC curves, one for the CIELab color histogram, the other for the SURF descriptor. It is clear from the observation of the figure that a TPR close to 1 is achieved much later (for a higher

FPR) for CIELab than for SURF, which validates their respective AUC scores. The straight line in both images symbolizes the line of chance - the straigth $y = x$.

From the results of the tests conducted in this part of the methodology, it is possible to conclude that the better perfomance to the task of detect changes in a scene (at least in this scenario) is accomplished by the local feature descriptors (and more specifically SURF), which made them the descriptors to use in the RGB-D experimental tests. Furthermore, the ROC curves and AUC scores obtained suggest that different threshold values are needed for each of the ROIs, in order to maximize the results obtained from them.

5.2 Depth Processing

As mentioned in section 4.2, the usage of image per-pixel depth information allows the classification of some of the detected events as occlusions. Then, to attest that idea and to analyze the proposed methodology for the depth processing algorithm, a similar testing procedure to the one made to test the visual descriptors was conducted, reducing the types of event in study to solely occlusions.

The obtained ROC curve (for each of the ROIs) in this experimental step is represented in figure 5.2. It is clear by observing the figure that this specific ROC is superior to each of the ones represented in figure 5.1. The respective AUC scores also prove this notion: 0.948873 for ROI 1 and 0.996427 for ROI 2.

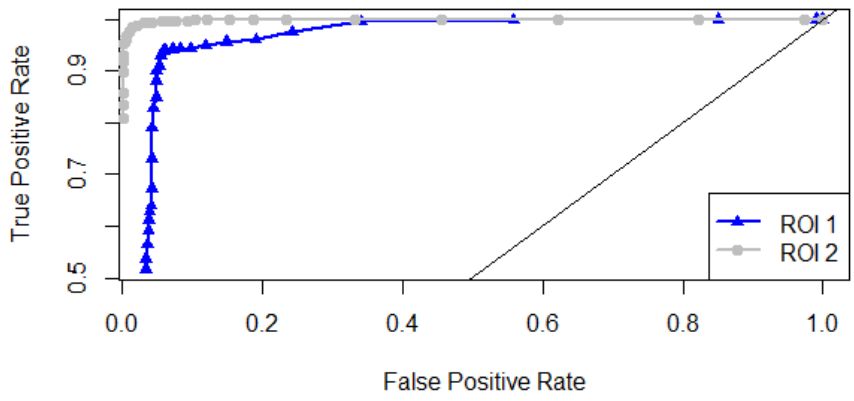


Figure 5.2: ROC curve for the depth processing algorithm.

The worst result obtained for ROI 1 is caused by two specific video sequences - 5 and 6 (and two specific events), represented below in figures 5.3 and 5.4, respectively. For sequence 5, the problem lies that the noise caused by the red jacket, as it stays immediately closer (from a depth perspective) to the camera and is impossible to filter out without causing a severe decay in the

performance for other sequences. As for sequence 6, the problem refers to the event represented in figure 5.4, which was considered as an occlusion for the preparation of the ground-truth, but such event is hard to be detected using depth information, as the arms that cause the occlusion of the ROI are technically "inside" the region and therefore in the same depth range. Again, to be able to include that specific event, the performance in others would suffer vastly.

(a) 830 (b) 950 (c) 1080

Figure 5.3: Problematic event from video sequence 5. The captions represent the frame numbers.

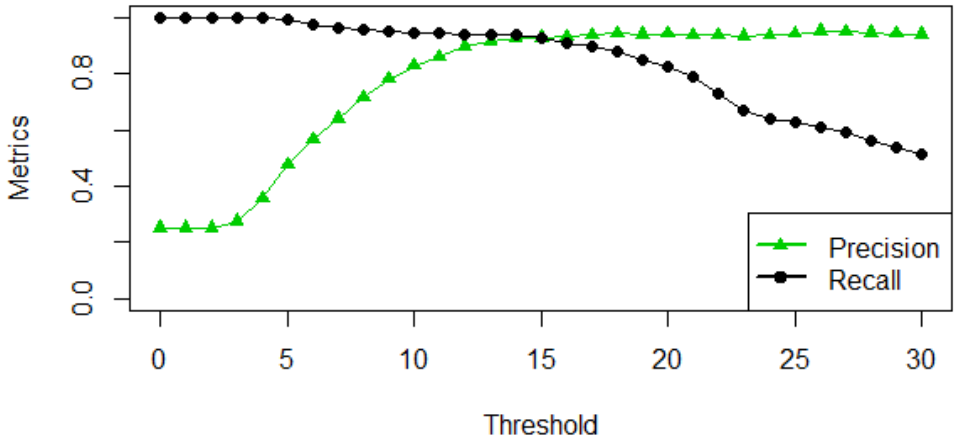
(a) 1212 (b) 1330

Figure 5.4: Problematic event from video sequence 6. The captions represent the frame numbers.

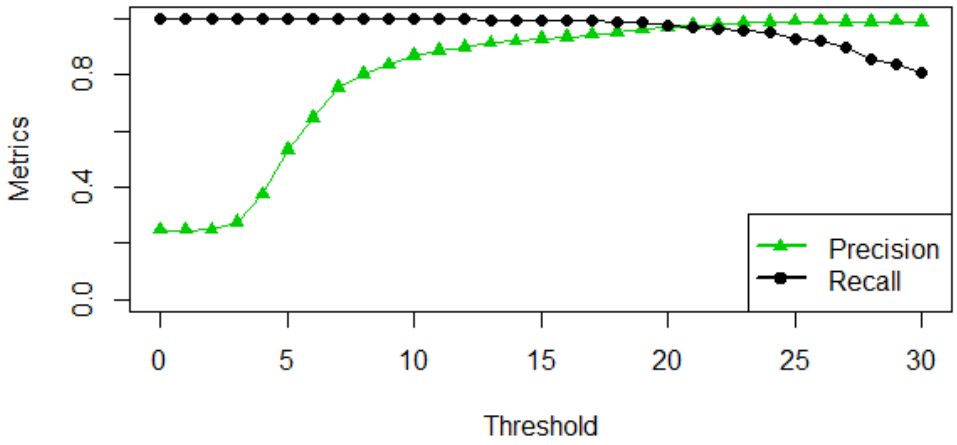
Other than for the aforementioned problematic occlusions, the use of depth data may provide very good results in occlusion detection, as its ROC curve and AUC score show. As is the case for the visual descriptors from section 5.1, the use of different thresholds for each ROI is needed to maximize the final output results.

5.3 RGB-D Processing

As proposed in section 4.3, the final methodology uses both depth and RGB video, in cascade, to detect the events from the ground-truth. From the experimental results described in sections 5.1 and 5.2, it was possible to, respectively, take away that the local feature descriptors are the best (from the test group) at detecting the occurrence of changes in a scene, and that the use of depth information for occlusion detection provides solid results. Then, the final experiments were conducted using the depth information to detect occlusions and the local feature descriptors to detect alterations.



(a) ROI 1



(b) ROI 2

Figure 5.5: Precision and recall curves, in function of the threshold values, for the proposed occlusion detection methodology.

The choice of the threshold values (for each visual descriptor and for the depth processing) has an impact on the final results, as demonstrated by the ROC curves represented above. A blind approach would be to simply choose the values which obtain the best precision, the best recall (TPR) or the best overall f-measure in each specific case. However, in a retail environment, it seems coherent to try to maximize the precision with which events are detected, but without totally disregarding the recall, as sometimes two threshold values may have a very small difference in one evaluation metric, but a large difference in the other. For instance, in a real-world situation of a

security guard obtaining feedback from this type of application, it is plausible that he is only alerted with the most accurate detections possible, but a small drop in that accuracy is accepted if it means that a much larger detection range is achieved.

Figure 5.5 represents the constructed curve of precision and recall in function of the threshold value for the occlusion detection algorithm (the percentage of area occluded), with 5.5a representing ROI 1 and 5.5b ROI 2. Similarly, 5.6 represents the same idea applied to the SURF visual descriptor for the conditions set in 5.1. However, for the visual descriptors, as no type of event "alteration" occurs in ROI2 and their use was now limited to these events, this process and the following experimental evaluations were limited to ROI 1 and video sequences 1 to 4. Furthermore, as mentioned in section 4.1.1.4, the threshold values for the local feature descriptors portray the percentage of keypoints matched, in comparison to the set model. The defined thresholds values, obtained from all the considered precision and recall curves, are then presented in table 5.2.

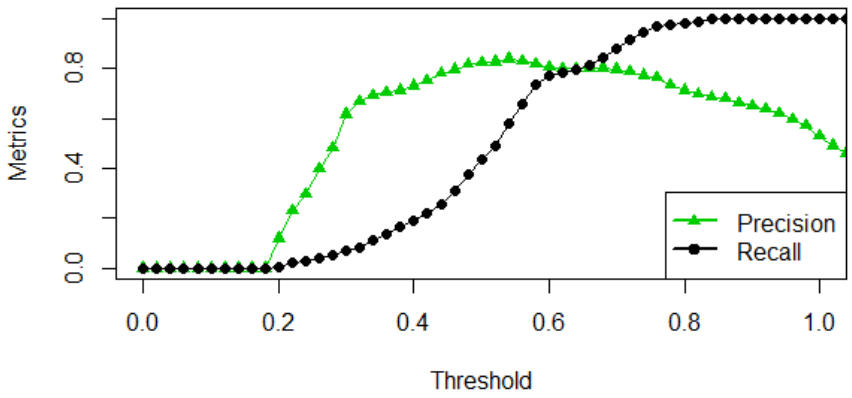


Figure 5.6: Precision and recall curves, in function of the threshold values, for the application of SURF in the conditions set in 5.1.

Table 5.2: Threshold values obtained from the precision and recall curves in the considered conditions from 5.1 and 5.2.

	Threshold Values	
	ROI 1	**ROI 2**
Depth	18	23
SIFT	0.66	-
SURF	0.68	-
ORB	0.68	-

For the given example of SURF, in figure 5.6, it is clear that the maximum precision value would be around the $[0.5, 0.55]$ interval of thresholds, but yet, as represented in table 5.2, the considered value was 0.68. This is due to the huge jump in recall, while not dropping precision

by much, that that difference makes, as denoted above. That exact same principal was applied for every of the represented values.

With the definition of the thresholds, the final proposed methodology was then tested. The attained results are expressed in the form of precision, recall and f-measure scores, by sequence and in total, in the following tables 5.3 (Occlusions) and 5.4 (Alterations).

Table 5.3: Occlusion Detection results.

	ROI 1			**ROI 2**		
Sequence	**Precision**	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
1	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	0.869565	0.930233	1.0	1.0	1.0
3	0.909090	0.952380	0.930232	0.967742	0.967742	0.967742
4	1.0	0.800000	0.888889	1.0	1.0	1.0
5	0.821990	1.0	0.902298	0.921569	0.796610	0.854545
6	0.888889	0.640000	0.744186	1.0	1.0	1.0
7	0.968750	0.885714	0.925373	0.991803	0.937984	0.964143
8	0.955414	0.903614	0.928792	1.0	0.945544	0.972010
Total (by ROI)	0.900990	0.890410	0.895669	0.987878	0.938579	0.962598

	Precision	**Recall**	**F-Measure**
Total	0.944000	0.914729	0.929133

Table 5.4: Alteration Detection results.

		Alterations		
Descriptor	**Sequence**	**Precision**	**Recall**	**F-Measure**
SIFT	1	1.0	1.0	1.0
	2	0.936170	0.448980	0.606897
	3	1.0	0.854369	0.921467
	4	0.316901	0.789474	0.450000
	Total	0.731182	0.770538	0.750345
SURF	1	1.0	1.0	1.0
	2	0.980000	1.0	0.989899
	3	1.0	0.854369	0.921466
	4	0.317014	0.789473	0.452261
	Total	0.767059	0.923512	0.838046
ORB	1	1.0	1.0	1.0
	2	0.980000	1.0	0.989899
	3	1.0	0.854369	0.921466
	4	0.317014	0.789473	0.452261
	Total	0.767059	0.923512	0.838046

As expected from section 5.2, the use of depth information for occlusion detection provides results with high scores of precision and recall. Also, the sequences containing the problematic events described perform worse than the others, but even then, as they are one-off cases, the final total scores aren't affected by much. Considering that the sequences containing both fixed and

transitory occlusions obtained fairly the same results as the others and those events are much longer (and thus extend through much more frames), it is possible to conclude that the duration of an event doesn't affect (directly) the final output. This makes sense as the methodology proposed for detecting occlusions was based on spatial distances and not time or movements.

When regarding the results of the detection of alterations, it is clear that all the local feature descriptors perform roughly the same (oddly, ORB and SURF even got the same exact results). It is also clear that they all struggle in sequence 4. This was somewhat expected as the alteration in that sequence, as mentioned back in chapter 3, later is reversed and the regions returns to what is visually considered the initial state. That state is, however, somewhat different as the returned shirt is not in the exact same position and contains some affine transformations, which as it is known from the literature review (2 is a limitation of the local feature descriptors.

Table 5.5: Alteration Detection results with thresholds optimized.

		Alterations		
Descriptor	**Sequence**	**Precision**	**Recall**	**F-Measure**
SIFT	1	1.0	1.0	1.0
	2	0.969697	0.979592	0.974619
	3	1.0	0.854369	0.921467
	4	0.316901	0.789474	0.45
	Total	0.764151	0.917847	0.833977
SURF	1	1.0	0.784211	0.869048
	2	0.979381	0.969388	0.974359
	3	1.0	0.854369	0.921466
	4	0.483871	0.7894730	0.600000
	Total	0.857550	0.852691	0.855114
ORB	1	1.0	0.968421052632	0.983957219251
	2	0.980000	1.0	0.989899
	3	1.0	0.854369	0.921466
	4	0.316901	0.789474	0.452261
	Total	0.765403	0.915014	0.833548

If the same process of creating the ROC curves (from section 5.1) is repeated for these descriptors, while only considering the alterations (and not the whole ground-truth), some optimization may be made in relation to the threshold values. Following the same principles from above the new threshold values were then: 0.7 for SIFT, 0.58 for SURF and 0.37 for ORB. As it is shown by table 5.5, above, this procedure improved the final results (in terms of precision), especially for SURF, which as previously mentioned was expected to provide the best results.

The combination of the usage of depth to detect occlusions and the SURF descriptor to detect alterations, in the processing model described in section 4.3, represents the final proposed solution to the problem of this dissertation. While detecting 85% of the total frames containing alterations with a roughly 85% precision (plus the results obtained for occlusions, which are even higher, as stated in table 5.3) is somewhat satisfying, it shows a flaw in the proposed methodology: the occlusion detection is not perfect and every error from there will possibly carry over to the alteration

detection. Thus, the improvement of the occlusion detection would also play a factor in improving the alteration detection. Furthermore, the implementation and testing of others strategies of computing keypoints (for instance, dense grids) and matching the feature vectors, along with possibly other local feature descriptors, may also improve the final results.

Chapter 6

Conclusions and Future Work

6.1 Final Discussion

The objective behind this dissertation was to accomplish the detection of a pre-determined group of events occurring in a retail environment, through the spatiotemporal analysis of RGB and RGB-D video. Based on a literature review regarding image segmentation and processing methods and the data available to the problem, three separate processing approaches were taken, converging to one final proposed solution: the joint usage of image depth information and RGB video, in cascade.

Regarding the preparation of the workspace, the dataset chosen for testing purposes consisted in video footage captured using the Microsoft Kinect (an RGB-D camera) in a clothing store, in China. From the analysis of the full video, 8 smaller test video sequences were extracted, two separate ROIs were defined, and two types of events to detect were specified: occlusions and alterations. Occlusions were defined as the concealment of a ROI by something or someone for a period of time, while alterations implied a physical change in a ROI (from a previously known state). The ground-truth was then prepared through the annotation of the occurrence of these events for each of the video sequences. Finally, the evaluation metrics were defined: precision, recall, f-measure, ROC curves and AUC score.

As mentioned, three separate methodologies were proposed in order to achieve the set objective: the processing of only RGB video, only depth information and a sum of the two. This separate analysis established a better perception of the advantages and limitations behind the usage of each isolated data and thus, a convergence to the usage of their sum.

The "pure" RGB processing algorithm, without the assistance of any background subtraction method, allowed no classification of the events, as all it could measure were differences in each ROI in separate video frames. However, it provided a way to to test and qualify the performance of a group of visual descriptors, chosen from the literature review, in the detection of visual alterations happening inside each ROI. From the construction and analysis of the afforementioned ROC curves and the calculations of the respective AUC scores, it was possible to attest that the

usage of local feature descriptors provided the best results (particularly SURF, due to its tolerance to illumination variations).

The processing of depth data, opened the possibility of separating the scene into foreground and background and thus detect and classify events as occlusions. However, due to its concept, it allowed no detection of alterations. Nevertheless, the mere possibility to classify certain events as occlusions, validated the joint usage of both types of video information, as in a perfect scenario, the depth processing would detect all occlusions and thus, visual changes detected by the RGB analysis would represent alterations. The tests conducted for this scenario, validated this assumption, as they showed that the processing of depth is able to detect the occurrence of almost all of the occlusions, with only a few noted exceptions.

The information retrieved from both testing procedures was then agglomerated in to a final test where the threshold values for depth and the local feature descriptors were extracted from the construction of precision and recall curves, and maximized to detect events with the best precision possible (while not totally disregarding the recall). The best results in this scenario were obtained for SURF, as expected from the previous RGB tests. The final proposed solution showed that, while it is far from being perfect, it is able to detect the vast majority of events represented in the testing sequences.

6.2 Future Work

Despite the potential of the results obtained, the present work is still preliminary and a lot of developments and improvements may be accomplished.

The usage of just the depth data to detect occlusions is not perfect, as it contains some visual noise. Furthermore, this also affects the detection of alterations due to the cascading nature of the proposed solution. The addition of a people tracking mechanism in parallel with the depth processing algorithm, may be help to filter some of that noise as it would allow for a joint analysis of complementary information - if depth detects an occlusion, but there was no people nearby in the last few seconds, it probably is just noise. A similar mechanism may also be implemented to track the objects inside a ROI in a retail environment, which would allow the classification of alterations as insertions or removals.

The implementation and testing of others strategies of computing keypoints and matching the feature vectors (as dense grids) for the local feature descriptors may also improve the final results, as the information would be retrieved and matched for much more points. Furthermore, other local feature descriptors such as BRIEF or CSIFT may be tested for a more complete analysis of this particular group.

Finally, the testing in other particular retail environments (with more of the problematic sequences of events described) could help validate the obtained results and conclusions. Moreover, testing the proposed solution for its running time may help to achieve an optimization that allows its use in a real-time situation.

References

[1] FORBES Maggie McGrath. Available in http://www.forbes.com/sites/maggiemcgrath/2016/05/26/the-worlds-largest-apparel-companies-2016-christian-dior-nike-and-inditex-top-the-list/, accessed in 16th November of 2016.

[2] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag London, London, United Kingdom, 2010.

[3] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596, 2010.

[4] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436, 2009.

[5] David G. Lowe. Distinctive image features from scale-invariant key points. *International Journal of Computer Vision 60(2)*, pages 91–110, 2004.

[6] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.

[7] David A Sadlier and Noel E O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(10):1225–1233, 2005.

[8] Phillip Curtis, Moufid Harb, Rami Abielmona, and Emil Petriu. Behavior-driven video analytics system for critical infrastructure protection. In *7th IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2014, December 14, 2014 - December 17, 2014*, Proceedings of the 2014 7th IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2014. Institute of Electrical and Electronics Engineers Inc., 2014.

[9] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[10] G Stiles Wyszecki and W Stiles. Ws,(1982), color science: Concepts and methods, quantitative data and formulae.

[11] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[13] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[14] Jun Liu, Ye Liu, Ying Cui, and Yan Qiu Chen. Real-time human detection and tracking in complex environments using single rgbd camera. In *2013 IEEE International Conference on Image Processing*, pages 3088–3092. IEEE, 2013.

[15] Jun Liu, Ye Liu, Guyue Zhang, Peiru Zhu, and Yan Qiu Chen. Detecting and tracking people in real time with rgb-d camera. *Pattern Recognition Letters*, 53:16–23, 2015.

[16] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. Human detection using depth information by kinect. In *CVPR 2011 WORKSHOPS*, pages 15–22. IEEE, 2011.

[17] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[18] Miss Hetal J Vala and Astha Baxi. A review on otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(2):387–389, 2013.

[19] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[20] Thomas Moeslund. Canny edge detection. *Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark, http://www. cvmt. dk/education/teaching/f09/VGIS8/AIP/canny_09gr820. pdf*, 2009.

[21] Rajiv Mehrotra, Kameswara Rao Namuduri, and Nagarajan Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12):1479–1494, 1992.

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

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

[24] Philipp Blauensteiner and Martin Kampel. Visual surveillance of an airport's apron-an overview of the avitrack project. *Workshop of the Austrian Association for Pattern Recognition*, page 213–220, 2004.

[25] Gutemberg Guerra-Filho. Optical motion capture: Theory and implementation. *RITA*, 12(2):61–90, 2005.

[26] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE, 1999.

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

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

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

[30] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *Computer Vision–ACCV 2006*, pages 581–590. Springer, 2006.

[31] Olivier Barnich and Marc Van Droogenbroeck. Vibe: a powerful random technique to estimate the background in video sequences. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 945–948. IEEE, 2009.

[32] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43. IEEE, 2012.

[33] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–30, 2005.

[34] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–84, 2007.

[35] Mark Tkalcic, Jurij F Tasic, et al. Colour spaces: perceptual, historical and applicational background. In *Eurocon*, 2003.

[36] Gary Starkweather. Colorspace interchange using srgb. *White Paper available at http://www. srgb. com*, 1998.

[37] I. Omer and M. Werman. Color lines: Image specific color representation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–946. IEEE, 2004.

[38] Adrian Ford and Alan Roberts. Colour space conversions. *Westminster University, London*, 1998:1–31, 1998.

[39] Henryk Palus. Representations of colour images in different colour spaces. In *The Colour image processing handbook*, pages 67–90. Springer, 1998.

[40] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[41] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[42] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6):1635–1650, 2010.

[43] Shengcai Liao, Guoying Zhao, Vili Kellokumpu, Matti Pietikäinen, and Stan Z Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1301–1306. IEEE, 2010.

[44] Thierry Bouwmans Caroline Silva and Carl Frélicot. An extended center-symmetric local binary pattern for background modeling and subtraction in videos. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications - Volume 1: VISAPP, (VISIGRAPP 2015)*, pages 395–402, 2015.

[45] Topi Mäenpää and Matti Pietikäinen. Classification with color and texture: jointly or separately? *Pattern recognition*, 37(8):1629–1640, 2004.

[46] Yongcheol Lee, Jiyoung Jung, and In-So Kweon. Hierarchical on-line boosting based background subtraction. In *Frontiers of Computer Vision (FCV), 2011 17th Korea-Japan Joint Workshop on*, pages 1–5. IEEE, 2011.

[47] Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on gabor filters. *Image Processing, IEEE Transactions on*, 11(10):1160–1167, 2002.

[48] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, 93(26):429–441, 1946.

[49] Guoying Zhao and Matti Pietikäinen. Local binary pattern descriptors for dynamic texture recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 211–214. IEEE, 2006.

[50] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, 2007.

[51] P. J. Burt. Fast algorithms for estimating local image properties. *Computer Graphics and Image Processing*, 1983.

[52] Alaa E. Abdel-Hakim and Aly A. Farag. Csift: A sift descriptor with color invariant characteristics. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, June 17, 2006 - June 22, 2006*, volume 2 of *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1978–1983. Institute of Electrical and Electronics Engineers Computer Society, 2006.

[53] G. J. Burghouts and J. M. Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62, 2009.

[54] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–50, 2001.

[55] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–59, 2008.

[56] PM Panchal, SR Panchal, and SK Shah. A comparison of sift and surf. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):323–327, 2013.

[57] Simon AJ Winder and Matthew Brown. Learning local image descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[58] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[59] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2010.

[60] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[61] Pedro Carvalho, Telmo Oliveira, Lucian Ciobanu, Filipe Gaspar, Luís F Teixeira, Rafael Bastos, Jaime S Cardoso, Miguel S Dias, and Luís Côrte-Real. Analysis of object description methods in a video object tracking environment. *Machine vision and applications*, 24(6):1149–1165, 2013.

[62] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[63] Fei-Fei Li, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories. *Tutorial at ICCV*, 2005.

[64] Anna Bosch, Andrew Zisserman, and Xavier Muoz. Scene classification using a hybrid generative/discriminative approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):712–727, 2008.

[65] Jingxin Xu, Simon Denman, Clinton Fookes, and Sridha Sridharan. Unusual event detection in crowded scenes using bag of lbps in spatio-temporal patches. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 549–554. IEEE, 2011.

[66] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[67] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[68] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[69] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[70] Jianxin Wu and James M Rehg. Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2011.

[71] Xue Wei, Son Lam Phung, and Abdesselam Bouzerdoum. Visual descriptors for scene categorization: experimental evaluation. *Artificial Intelligence Review*, 45(3):333–368, 2016.

[72] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[73] Pawan Patidar, Manoj Gupta, Sumit Srivastava, and Ashok Kumar Nagawat. Image denoising by various filters for different noise. *International Journal of Computer Applications*, 9(4), 2010.

[74] Itseez. Open source computer vision library. Available in https://github.com/itseez/opencv, 2017.

[75] Prateek Joshi. *OpenCV with Python By Example*. Packt Publishing Ltd, 2015.

[76] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. " O'Reilly Media, Inc.", 2016.

[77] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.