An investigation into common challenges of 3D scene understanding in visual surveillance

Katy Tarrit

A thesis in partial fulfilment of the requirements of the University of the West of England, Bristol for the degree of Doctor of Philosophy

Faculty of the Environment and Technology
University of the West of England, Bristol

July 2014

Abstract

Nowadays, video surveillance systems are ubiquitous. Most installations simply consist of CCTV cameras connected to a central control room and rely on human operators to interpret what they see on the screen in order to, for example, detect a crime (either during or after an event). Some modern computer vision systems aim to automate the process, at least to some degree, and various algorithms have been somewhat successful in certain limited areas. However, such systems remain inefficient in general circumstances and present real challenges yet to be solved. These challenges include the ability to recognise and ultimately predict and prevent abnormal behaviour or even reliably recognise objects, for example in order to detect left luggage or suspicious objects. This thesis first aims to study the state-of-the-art and identify the major challenges and possible requirements of future automated and semi-automated CCTV technology in the field. This thesis presents the application of a suite of 2D and highly novel 3D methodologies that go some way to overcome current limitations.

The methods presented here are based on the analysis of object features directly extracted from the geometry of the scene and start with a consideration of mainly existing techniques, such as the use of lines, vanishing points (VPs) and planes, applied to real scenes. Then, an investigation is presented into the use of richer 2.5D/3D surface normal data. In all cases the aim is to combine both 2D and 3D data to obtain a better understanding of the scene, aimed ultimately at capturing what is happening within the scene in order to be able to move towards automated scene analysis. Although this thesis focuses on the widespread application of video surveillance, an example case of the railway station environment is used to represent typical real-world challenges, where the principles can be readily extended elsewhere, such as to airports, motorways, the households, shopping malls etc. The context of this research work, together with an overall presentation of existing methods used in video surveillance and their challenges are described in chapter 1.

Common computer vision techniques such as VP detection, camera calibration, 3D reconstruction, segmentation etc., can be applied in an effort to extract meaning to video surveillance applications. According to the literature, these methods have been well researched and their use will be assessed in the context of current surveillance requirements in chapter 2. While existing techniques can perform well in some contexts, such as an architectural environment composed of simple geometrical elements, their robustness and performance in feature extraction and object recognition tasks is not sufficient to solve the key challenges encountered in general video surveillance context. This is largely due to issues such as variable lighting, weather conditions, and shadows and in general complexity of the real-world environment. Chapter 3 presents the research and contribution on those topics – methods to extract optimal features for a specific CCTV application – as well as their strengths and weaknesses to highlight that the proposed algorithm obtains better results than most due to its specific design.

The comparison of current surveillance systems and methods from the literature has shown that 2D data are however almost constantly used for many applications. Indeed, industrial systems as well as the research community have been improving intensively 2D feature extraction methods since image analysis and Scene understanding has been of interest. The constant progress on 2D feature extraction methods throughout the years makes it almost effortless nowadays due to a large variety

of techniques. Moreover, even if 2D data do not allow solving all challenges in video surveillance or other applications, they are still used as starting stages towards scene understanding and image analysis. Chapter 4 will then explore 2D feature extraction via vanishing point detection and segmentation methods. A combination of most common techniques and a novel approach will be then proposed to extract vanishing points from video surveillance environments. Moreover, segmentation techniques will be explored in the aim to determine how they can be used to complement vanishing point detection and lead towards 3D data extraction and analysis.

In spite of the contribution above, 2D data is insufficient for all but the simplest applications aimed at obtaining an understanding of a scene, where the aim is for a robust detection of, say, left luggage or abnormal behaviour; without significant a priori information about the scene geometry. Therefore, more information is required in order to be able to design a more automated and intelligent algorithm to obtain richer information from the scene geometry and so a better understanding of what is happening within. This can be overcome by the use of 3D data (in addition to 2D data) allowing opportunity for object "classification" and from this to infer a map of functionality, describing feasible and unfeasible object functionality in a given environment. Chapter 5 presents how 3D data can be beneficial for this task and the various solutions investigated to recover 3D data, as well as some preliminary work towards plane extraction.

It is apparent that VPs and planes give useful information about a scene's perspective and can assist in 3D data recovery within a scene. However, neither VPs nor plane detection techniques alone allow the recovery of more complex generic object shapes - for example composed of spheres, cylinders etc - and any simple model will suffer in the presence of non-Manhattan features, e.g. introduced by the presence of an escalator. For this reason, a novel photometric stereo-based surface normal retrieval methodology is introduced to capture the 3D geometry of the whole scene or part of it. Chapter 6 describes how photometric stereo allows recovery of 3D information in order to obtain a better understanding of a scene, as well as also partially overcoming some current surveillance challenges, such as difficulty in resolving fine detail, particularly at large standoff distances, and in isolating and recognising more complex objects in real scenes. Here items of interest may be obscured by complex environmental factors that are subject to rapid change, making, for example, the detection of suspicious objects and behaviour highly problematic. Here innovative use is made of an untapped latent capability offered within modern surveillance environments to introduce a form of environmental structuring to good advantage in order to achieve a richer form of data acquisition. This chapter also goes on to explore the novel application of photometric stereo in such diverse applications, how our algorithm can be incorporated into an existing surveillance system and considers a typical real commercial application.

One of the most important aspects of this research work is its application. Indeed, while most of the research literature has been based on relatively simple structured environments, the approach here has been designed to be applied to real surveillance environments, such as railway stations, airports, waiting rooms, etc, and where surveillance cameras may be fixed or in the future form part of a mobile robotic free roaming surveillance device, that must continually reinterpret its changing environment. So, as mentioned previously, while the main focus has been to apply this algorithm to railway station environments, the work has been approached in a way that allows adaptation to many other applications, such as autonomous robotics, and in motorway, shopping centre, street and home environments. All of these applications require a better understanding of the scene for security or safety purposes. Finally, chapter 7 presents a global conclusion and what will be achieved in the future.

Contents

List of figures

List of tables

Glossary

APHT: Adaptive **P**robabilistic **H**ough **T**ransform.

CMW: Continuous **M**odulation **W**ave.

DSaM: Direct **S**plit-**a**nd-**M**erge.

EM: Expectation-**M**aximisation.

FCM: Fuzzy **C-M**eans clustering method.

GIST: Holistic shape feature based on the spatial envelope.

GT: Ground **T**ruth.

HOG: Histogram of **O**riented **G**radient.

HT: Hough **T**ransform.

ICP: Iterative **C**losest **P**oint.

LCD: Liquid **C**rystal **D**isplay.

LCOS : Liquid **C**rystal **O**n **S**ilicon.

LiDAR: Light **D**etection **A**nd **R**anging.

LSR: Line Support Regions.

MLE: Maximum **L**ikelihood **E**stimate.

MSE: Mean-**S**quare **E**rror.

PCA: Principal Component Analysis.

PHT: Probabilistic **H**ough **T**ransform.

PM: Pulse **M**odulation.

PNA: **P**roposed and **N**ovel **A**pproach.

PPHT: Progressive **P**robabilistic **H**ough **T**ransform.

PS: Photometric-**S**tereo.

RANSAC: RANdom **S**ample **C**oncensus.

RG: Region **G**rowing.

RHT: Randomized **H**ough **T**ransform.

SHT: Standard **H**ough **T**ransform.

SIFT: **S**cale **I**nvariant **F**eature **T**ransformation.

SFS: **S**hape **F**rom **S**hading.

SURF: **S**peeded **U**p **R**obust **F**eatures.

SVD: **S**ingular **V**alue **D**ecomposition.

SVM: **S**upport **V**ector **M**achine.

VL(s): **V**anishing **L**ine(s).

VP(s): **V**anishing **P**oint(s).

VPD: **V**anishing **P**oint **D**etection.

TAR: **Tar**dif's method.

ToF: **T**ime-**o**f-**F**light.

Symbols

$\{F_O^i\}$: Set of original frames with $i = 0\ to\ 5$ number of the current frame firstly

I_O^e: Better quality image obtained after image enhancement.

$\{\mathcal{E}_{t_c}\}$: Set of edges detected with the Canny edge detection filter.

$\{\mathcal{L}_i\}$: Set of lines extracted using the Hough Transform.

$\{\mathcal{L}_i^{\mathcal{H}}\}$: Set of horizontal lines.

$\{\mathcal{L}_i^{\mathcal{V}}\}$: Set of vertical lines.

$\{\mathcal{L}_i^{\mathcal{T}}\}$: Set of lines in the same direction of the tracks.

$\{\mathcal{L}_f^{\mathcal{H}}\}$: Final set of horizontal lines.

$\{\mathcal{L}_f^{\mathcal{V}}\}$: Final set of vertical lines.

$\{\mathcal{L}_f^{\mathcal{T}}\}$: Final set of lines in the same direction of the tracks.

$\{\mathcal{I}_i^{\mathcal{H}}\}$: Set of intersection points originating from the set of horizontal lines.

$\{\mathcal{I}_i^{\mathcal{V}}\}$: Set of intersection points originating from the set of vertical lines.

$\{\mathcal{I}_i^{\mathcal{T}}\}$: Set of intersection points originating from the set of lines in the same direction of the tracks.

$\{\mathcal{I}_f^{\mathcal{H}}\}$: Final set of intersection points originating from the set of horizontal lines.

$\{\mathcal{I}_f^{\mathcal{V}}\}$: Final set of intersection points originating from the set of vertical lines.

$\{\mathcal{I}_f^{\mathcal{T}}\}$: Final set of intersection points originating from the set of lines in the same direction of the tracks.

$\{\mathcal{C}_f^{\mathcal{D}}\}$: Final set containing the coordinates of VPs from the three orthogonal directions.

t_1 and t_2 : Thresholds used for the hysteresis procedure of the Canny edge detection filter.

t_H: Threshold used for the Hough transform.

$t_{\mathcal{L}}$: Threshold used to highlight the relevant lines to extract the vanishing points.

I_G^e : Enhance image converted in grey-scale.

I_{GS}^e: Grey-scale enhance image smoothed.

$\{\mathcal{L}_i = (\rho_i, \theta_i)\}$: Set of lines detected with i the index of the line considered.

ρ_i: the distance between the line and the origin of the ith line.

θ_i: The angle of the vector from the origin to the closest point of the i^{th} line.

N: The total number of lines in a temporary cluster.

$\mathcal{M}(\rho_{\mathcal{C}})$: The mean value of ρ parameters in the temporary considered cluster $\mathcal{C}$.

$\mathcal{M}(\theta_{\mathcal{C}})$: The mean value of θ parameters in the temporary considered cluster $\mathcal{C}$.

$\mathcal{M}(\mathcal{L}_{\mathcal{C}})$: the mean set for the temporary line cluster associated to a flag here defined as $\mathcal{C}$ to distinguish one temporary cluster $\mathcal{C}$ from another.

T: threshold used to measure the success of VPs detected.

$\mathrm{x}_{\mathrm{PNA}}$ and $\mathrm{y}_{\mathrm{PNA}}$: The coordinates of the VP detected with the proposed and novel approach.

x_{GT} and y_{GT}: the ground truth VP coordinates.

d_{GT} : The Euclidean distance between the estimated VP and GT.

$\mathrm{x}_{\mathrm{TAR}}$ and $\mathrm{y}_{\mathrm{TAR}}$: The coordinates of the VP detected with Tardif's method.

d_{ref}: The Euclidean distance between the estimated VP and the image centre.

t_b: Threshold used for the Binary thresholding method.

t_o: Threshold used for the Otsu thresholding method.

$\{\mathcal{J}\}$: Input set for the plane extraction algorithm.

$P_k = (a_k, b_k, c_k, d_k)$: the kth plane estimated from RANSAC.

$r_{ki} = a_k x_i + b_k y_i + c_k z_i + d_k$ for a set of points $\{i\}$ and represents a measure of closeness of a point from this set to the k^{th} estimated plane.

r_t: Similarity metric used to decide within which tolerance a point belong to the k^{th} estimated plane.

$\{\mathcal{C}_k\}$: represents the cluster of point that belongs to the k^{th} estimated plane.

$\{\overline{\mathcal{J}_k}\}$: is the new input set used by the proposed plane extraction approach once a set of point $\{j\}$ has been removed from the original input set $\{\mathcal{J}\}$.

$\{\mathrm{N}_i; i = 1,2, \dots M\}$: The set of unit surface normals for the M-pixel image.

$\{I_{k,i}; i = 1,2, \dots M\}$: The set of pixel intensities for image$k$,

L_l : the l^{th} light source vector.

ρ_i : The albedo of the i^{th}pixel.

$N_{x,i}$, $N_{y,i}$ and $N_{z,i}$: the$x$, y and z components of the i^{th} surface normal.

$L_{l,x}$, $L_{l,y}$ and $L_{l,z}$: the$x$, y and z components of the l^{th} light source vector.

$N_x^{GT\_shape}$ and $N_z^{GT\_shape}$: the x and z component of the surface normals recovered for a GT shape.

$N_x^{Exp\_shape}$ and $N_z^{Exp\_shape}$: the x and z component of the surface normals recovered for an experimental shape.

$d_{GT\_Exp}^{shape\_shape}$: the Euclidean distance which is used a metric to compare the x and z component of the surface normals recovered for a GT shape to those for an experimental shape.

T : The total number of data.

$\mathcal{M}ean_{GT\_Exp}^{shape\_shape}$: The mean value determined for Euclidean distance corresponding to a specific GT and experimental shape.

1, 2 and 3 :represent the shape considered (i.e. here: half-cylinder, prism and cuboid respectively).

$\mathcal{M}ean(i,j)$: The mean value calculated from the Euclidean distance between shape i and shape j
$\mathcal{C}_{GT/Exp}^{i}$: the cluster composed of the mean value for the Euclidean distance calculated from the same GT shape and various experimental shapes.

Declaration

I declare that the work in this thesis is solely my own except where attributed and cited to another author. Most of the material in this thesis has been previously published by the author. A complete list of publications can be found on page 26.

Acknowledgements

I would like to thank all the people who directly or indirectly contributed to this research work and helped me go through the PhD, in good as well as difficult times.

First of all, I would like to thank my director of studies Dr. Gary A. Atkinson and my two supervisors Prof. Melvyn L. Smith and Dr. Lyndon N. Smith to give me the opportunity to pursue my studies with a PhD and for all the advice they provided me along the way even if it was not always easy. I would like to thank especially Dr. Glynn C. Wright and his family who have been really supportive, as well as Aralia’s employees who provided useful information and insights of common challenges encountered by video surveillance systems in real-world environments.

I also would like to thank all my colleagues from the Centre for Machine Vision and in particular Dr. Abdul R. Farooq and Dr. Khemraj Emrith, for their precious advice.

I am also thankful to Dr. Julio Molleda, who worked on a paper with me, taught me that our only limitations comes from within and always took some time to give me advice on how to conduct research, while he was always busy.

A special thanks to my dear parents who have always been there to support me even if they had health problems and were in another country. I also would like to thank the uncle of my father Prof. Erick Dufourc, for his support and advice as he knows how challenging a PhD can be.

Finally, I am really grateful to my boyfriend Antoine, who always tried to be there to provide me support and comfort even while he was working really hard and living in another country.

List of publications

Conference paper

- K. Tarrit, G. Atkinson, M. Smith, J. Molleda, G. Wright, P. Gaal, “Vanishing point detection for CCTV in railway stations”, 5th International Conference on Imaging for Crime Detection and Prevention (ICDP 2013).

Chapter 1: Introduction

1-1- Context and aim of the proposed approach

A scene [1] can be defined as a view of a real-world environment composed of a multitude of surfaces and objects organized in a meaningful way. The understanding of what is happening within a given scene is then subject to interpretation as well as to its geometry.

A common characteristic of most man-made environments is the prevalence of simple geometrical cues such as lines, planes and 3D volumes. So, the use of such elements allows scene understanding depending on how they are manipulated. Moreover, these elements not only represent the scene geometry but also provide perspective information. Interpretation does not only depend on how a scene is perceived but also on its perspective and geometry.

It is then essential to differentiate static from dynamic context of a scene when trying to understand it. Indeed, the static context of a scene can be defined as the scene background whereas the dynamic context can be defined as the scene foreground where humans play the role of agents. As mentioned in [1], objects are dense and act upon the scene whereas the latter is sparse and act within; they can then be distinguished depending on human actions.

Most current intelligent CCTV systems deal with scene understanding mainly by analysing the scene foreground. They then proceed by detecting and tracking dynamic objects that allow for differentiation of static objects such as bags, chairs etc., from dynamic ones such as humans, vehicles etc. However, human actions do not only depend on humans but also on objects surrounding them. It is then essential to look at the scene background in addition to its foreground to completely understand what is happening within a given scene.

Objects have many different properties including size, colour, texture, function as well as functionality. Whereas their function provides information on how to use them, their functionality corresponds to behaviours their function implies. This concept can be illustrated by taking the example of a chair. The function of a chair is a seat whereas its functionality, which corresponds to the behaviour implied by its function, is that people can sit on it. It is indeed possible to stand or lie down etc. on a chair but in some contexts, these behaviours can be considered as dangerous and can then be implied by the "functionality" of the chair. In other words, the "recognition" of static objects can significantly improve scene understanding by assisting human detection and behavioural analysis.

However, it is not that simple as there are many different objects with many different properties. So, how is it possible to "recognise" an object and differentiate it from another?

First of all, it is essential to set up a specific context. This research work is not trying to recognise any object but only relevant objects related to the application. The proposed approach is meant to be applied to video surveillance railway and underground stations. So, this thesis shows how in such context, it is possible to "recognise" objects and differentiate one from another with distinct geometry- i.e. by recognising the geometry, predicting possible behaviour – e.g. sharp edge can stab, blunt edge more benign. Furthermore, this thesis also shows how it is possible to apply this approach to other applications such as airports, shopping malls, city centres and many more.

There exist many very different methods for scene understanding, object recognition etc. However, none from our knowledge has yet been applied to the context of video surveillance. Moreover, as mentioned earlier on, most methods or current systems focus mainly on the dynamic aspect of a scene and only use the static one as a reference for comparison or subtraction etc.

As mentioned previously, the proposed method depends on the static aspect of a scene and its geometry. For this purpose, the focus of this research work is only based on the extraction of 2D data as well as 3D data to be able to infer objects from their geometry by analysing features extracted from each dataset. Geometry being the basis of perspective and omnipresent in man-made environments, it makes the extraction of geometrical features easy. Indeed, most objects or surfaces in man-made environments are composed of lines, vanishing points (VPs), planes, volumes etc. The problem has then been divided into two categories:

- 2D data analysis :
    - Feature extraction;
    - Feature Analysis;
- 3D data analysis :
    - Feature extraction;
    - Feature Analysis;
    - Interpretation of the scene from 2D and 3D features analysed.

First, a new method combining several common existing methods and a new clustering and voting scheme to extract lines and VPs is designed. The key to this new method is that it is very specific to the scene being analysed as it depends entirely on its geometry and it can then be adapted to be applied to other type of scenes such as airports, shopping malls, city centres, shops etc. Once lines and VPs are detected, an investigation is realised to find methods to extract more meaningful information from 2D data and fully exploit their potential. Segmentation methods are investigated to divide the scene background into meaningful and relevant clusters depending on the objects it contains based on colour information. The idea was to combine these two methods to obtain a more precise clustering method defining safe and dangerous areas within the scene dictating where people should and should not be, and allow security personnel to detect events more easily. While the 2D capture and analysis methods presented in this thesis proved effective in certain specific areas, it did not allow us to fully achieve our goals.

For this reason this thesis looked at methods to acquire 2.5/3D data in addition to 2D data. Indeed, 2.5/3D data are richer than 2D data as they provide additional information and features such as depth, orientation, texture etc. As most of the objects in railway and underground stations have a simple geometry, methods on how to extract planes from 3D data have first been considered. Finally, surface normals are analysed to infer the type of an object from its geometry. The research conducted on these various topics showed that it is possible to combine them to infer an object category from its geometry. Indeed, this is possible by following the stages described below:

1) Line and VP extraction;
2) Determination of the distance between the camera and an object of interest using the relationship between VPs and camera parameters;
3) Surface normal extraction;
4) Analysis of the surface normal of the object of interest;
5) Future work: map of functionality made possible by combining the previous steps to segmentation applied to 3D data.

The proposed approach consists in investigating common CCTV challenges to propose a more robust and efficient method allowing obtaining a better understanding of a 3D scene. For this purpose, this research work has been focused on various main stages:

- Investigation of current CCTV systems;
- Literature review;
- 2D feature extraction;
- Acquisition of 2.5D/3D data;
- Analysis of 2.5D/3D data.

As mentioned in the previous section, scene understanding can benefit significantly from the analysis of the scene background allowing analysing human behaviour from human-object interactions.

On the contrary to the majority of current systems and methods used, the proposed approach will only focus on the static background and static objects within the scene. The study case explored in this thesis, concerns railway and underground stations which make the object "recognition" task simpler, as only a few objects within this type of scenes can be considered as objects of interest; furthermore, an object can easily be distinguished from another based on the knowledge of some or all of their characteristics (i.e. shape, dimensions, colour, texture, function etc.). Moreover, most objects in man-made environments composed of simple geometrical elements such as lines, VPs, planes, simple cubic volumes etc, need to be recognised. So, it became clear that one way to "recognise" objects is to extract 2D as well as 3D geometrical features from a given scene and combine them to make a more accurate analysis, i.e. closer to what is really happening within the scene.

The main goal of this research work was to develop a method enabling a level of intelligent and automated understanding of a real-world scene in order to provide useful data in the context of visual surveillance. For this purpose, the procedure below was envisaged:

1) Extraction of 2D data: lines, VPs;
2) Analysis of 2D data: segmentation;
3) Extraction of 2.5D/3D data: 3D point cloud, planes, surface normal;
4) Analysis of 3D data: shape "recognition".

In this section, the context and aim of this research work has been presented. In the next section, the motivations and the contributions of this work are presented.

1-2-Contribution and motivation of the proposed approach

This section presents what motivated this research work to lead to the proposed approach as well as how it can contribute to the research community.

The motivations for the proposed approach are described below:

- The exploration of 3D data extraction and analysis offers rich information and cues to exploit and automatically obtain a better understanding of a given scene ; 3D data provide the same information as 2D data, i.e. geometry and perspective, colour but also additional information such as depth, texture etc. This, in turn, helps to identify objects from their shape and thereby help with prevention of dangerous situations, e.g. it can "recognize/detect" a knife and then help to prevent any incident that could be related;
- The investigation of logical connections between objects and their geometry, which can assist an existing human detection system to prevent abnormal behaviour relating to object function and functionality. For example, if a bench is "recognised", one can expect humans to be sitting or maybe lying down on the bench which implies that standing up on the bench or other behaviour can be considered as unsafe or dangerous depending on the context; in this case, if the inference about the chair is correct, and combining this with a human detection system, it can allow to detect abnormal behaviour based on the functionality implied by the chair function;
- The opportunity to save security personnel effort by automatically searching for dangerous objects or a particular CCTV events in a video sequence linked to dangerous objects;
- The design of a better and more intelligent system able to understand a scene as if the camera acted as a human witness of what is happening within the scene;
- The possibility to fill the gap between academia and industry in terms of video surveillance;
- The creation of a system that can be adapted to be used in other real-world applications such airports, shopping malls, streets, shops etc;
- The opportunity to improve CCTV systems for prevention of incidents and then to enable them to predict incidents in the future (cf. chapter 8 for more details) ;

The main contributions of this approach are:

- Acquisition of 2D and 2.5D data from real-world data in useful environments;
- Extraction of 2D and 2.5D features from above data;
- Robustness to complex environments with noise, change in exterior conditions such as lighting, weather etc.
- Partly real-time and automated method;
- A method used for a new application that can be applied to real-world applications;
- The potential to provide a solution to meet police requirements;
- Low-cost compared to other systems;
- Easy to implement and combine with existing systems.

1-3-Thesis Overview

The next chapter relates to current CCTV systems. First, a timeline representing the evolution of CCTV and some advanced current technologies in the field is presented. Then, a succinct description of a case study intelligent CCTV system: that of Aralia Systems Ltd, is provided. Finally, an overview of current CCTV systems including Aralia's is given. In addition, Chapter 3 comprises a literature review from the research community about methods to extract VPs, segment 2D images, acquire 2.5/3D data and put them into context. These two chapters show how this research relates to the state-of-the-art, partially fills an essential void in the literature and helps to bridge the gap between academic research and industry requirements.

Chapters 4 to 6 describe the technical aspects of the proposed approach. The first of these chapters presents the VP detection method and segmentation of 2D images; the second describes two selected devices to acquire 3D data and a method to extract planes; and finally the latter, explains how surface normals can be used to infer an object's type/function from its geometry.

Chapter 7 formally states the requirements for CCTV systems, police expectations and requirements from CCTV systems to improve security and the public opinion. This chapter also links together the various contributions with the above requirements to clarify the contribution and beneficiaries of this research. It also contains application examples.

Finally, Chapter 8 concludes the thesis and provides a summary of the proposed approach as well as its strengths and limitations and considers avenues for future work.

Chapter 2: Visual Surveillance Systems

This chapter describes the evolution of CCTV and provides an overview of current CCTV systems in terms of what they can and cannot achieve yet.

First, a brief overview of computer vision methods and their applications is provided.

Computer vision [2] consists in using a computer to automatically process images in order to extract, interpret and reconstruct information.

Computer vision has various applications such as:

- Control processes (e.g. an industrial robot);
- Navigation (e.g. by an autonomous vehicle or mobile robot);
- Events detection (e.g. for visual surveillance or people counting);
- Indexing (e.g. for databases of images and image sequences);
- Objects or environments modelling (e.g. medical image analysis or topographical modelling);
- Interaction (e.g. as the input to a device for computer-human interaction);
- Automatic inspection (e.g. in manufacturing applications);
- Face Recognition etc.

Those applications are made possible thanks to computer vision methods such as:

- Camera Calibration [3], which involves finding intrinsic and extrinsic parameters of a camera that enable determination of the distance between an object within the scene and the camera.
- Image Segmentation, which aims to divide an image into multiple sub-regions that share similar properties such as colour, texture etc [4];
- Recognition, which determines whether or not an image contains some specific objects, features, or activities such as object recognition [5], identification [6], detection [7], content-based retrieval [8], pose estimation [9], facial recognition [10];
- Motion analysis, which recovers motion information by processing two or multiple consecutive images from an video sequence [11][12][13] such as egomotion [14][15], tracking [16] and Optical flow [17];
- Scene reconstruction , which allows the recovery of a 3D scene geometry from a single or multiple images or a video sequence of a given scene [18];
- Scene Understanding, which corresponds to contextualisation, i.e. it provides a better understanding of what is happening within a given scene by associating context to features extracted from the scene [18];

Many of these techniques are commonly used for video surveillance applications. Indeed, video surveillance systems are significantly improved thanks to advances in computer vision research.

In the next section, important events concerning CCTV are presented as well as some important advances in technology.

2-1-History

In this sub-section, a timeline summarising the main events that led from the first step of the use of images for monitoring, to visual surveillance nowadays, is presented. The timeline is shown below[1].

[1] The images in the timeline are redacted for copyright reasons.

1913

Secretive photography of imprisoned suffragettes.

1942

First CCTV system installed by the engineer Walter Bruch for Siemens in Germany to test the Stand VII rocket launch site.

1949

Publication of Georges Orwell's 1984, set in London.

First commercial CCTV system available in USA.

***1951:* First Video Tape Recorder to capture live images from a television camera.**

1956

Trial operation of a street camera system demonstrated by police officers on the "Magic mirror" in Hamburg, Germany.

1959

In Hannover: Use beginning of CCTV for temporary monitoring of the increased inner city traffic coming in for the annual industrial trade fair.

1960

First "photographic and automatic red light-surveillance" in Frankfurt to investigate violations of traffic regulations and observe rallies or public gathering.

Use of 2 temporary cameras by the metropolitan police in Trafalgar Square to monitor crowds ahead of the Thai royal family arrival and "Guy Fawkes Day".

1961

First permanent installation of a CCTV system at a London Transport train station.

1964

Police experiment of 4 covert CCTV cameras in Liverpool city centre.

1965

First reports of law enforcement use of surveillance in the US.

Cameras installation in British railways to monitor tracks near Dagenham since vandalism.

1967

Photo-Scan (UK Company) markets a video surveillance system to catch shoplifters.

***1966:* NASA uses analogue signals to map the surface of the moon, sending digital images**

1968

September: Olean, New York was the first city in the United States to install video cameras along its main business street in an effort to fight crime.

October: Installation of temporary cameras by the Metropolitan police in Grosvenor Square to monitor anti-Vietnam War demonstrators.

1969

Installation of permanent cameras in Grosvenor Square, Whitehall and Parliament Square. (67 cameras in total across UK).

First CCTV cameras installed in the New York City Municipal Building near City Hall.

2nd December: Marie Van Brittan Brown and her husband, Albert Brown, are granted a patent for the first home security system using television surveillance (4 poles+1 camera sliding up and down to look at each one).

***1972:* Texas Instruments patents first electronic camera that does not require film.**

1973: Charged Coupled Device image sensor chip technology is invented.

Videocassette Recorder Technology becomes available.

1974

First installation of video surveillance systems to monitor traffic on the main roads in and through London.

1975

First installation of video surveillance systems in 4 London Underground train stations.

Use of CCTV systems to monitor football stadium to monitor violence before, during and after matches.

1980: Expansion of surveillance systems to businesses prone to theft or fraud.

1984

Installation of surveillance cameras at major rallying points in Central London as a response to miners' strikes.

1986: First Megapixel sensor invented by Kodak and able to record 1.4 million pixels.

1987

Use of CCTV at parking garages owned by local authorities.

1988

Installation of CCTV at "council estates" run by local authorities.

1989

Publication of "Who's watching you? Video surveillance in public places" by civil Rights group Liberty.

1992

Installation of street-based video surveillance system in Newcastle using microwaves to link to the city's main police station.

Use of speed cameras and red-light enforcement cameras on the national road network.

Invention of the "nanny Cam". The inspiration of the development of even-smaller and hi-res cameras starts.

1993

26th February: First attack on the World Trade Center, creating a frenzy of increased security.

August: Bombing of Bishopsgate in London by the IRA leads to the construction of the "Ring of steel" around the city.

1994

Publication of "CCTV: Looking Out for You" by the Home Office.

July: Use of covert video surveillance systems at Automatic Teller Machines (ATM).

October: Foundation of the World Wide Web Consortium that pioneered the Internet.

1996

All of England's major cities except Leeds have video surveillance systems installed in their city centre.

First IP camera released by Axis Communications

1997

10 May: Public demonstrations against surveillance cameras in Brighton organised by South Downs Earth First.

July: London police announcement about the installation of surveillance cameras systems to automatically read, recognise and track automobile by their license plates.

1998

October: Newham Council introduces face recognition software in the London Borough to track repeat offenders.

Second wave of adoption of video surveillance.

1999

Apex of Digital Video Recorders.

2001

January: Trial of a facial recognition software to the Super Bowl in Tampa to search for potentials criminals and terrorists in attendance at the event.

11th September: Attack of the World Trade Centre which changed the view on video surveillance from big-brother to individual safety and led to an increase of surveillance network and cameras installation all around the world.

2002

May: The United States Parks Service installed a facial recognition software on the video surveillance cameras at the Statue of Liberty and Ellis Island.

Facial recognition system installed at Sydney International Airport in Australia.

2003

December: Royal Palm Middle School in Phoenix, Arizona installed face recognition video surveillance as a pilot program for tracking missing children and registered sex offenders.

2005

Release of the first IP camera with onboard video content analytics by Intellio.

2006

The most extensive surveillance system "Operation Virtual Shield" is announced in Chicago.

2007

Estimation of more than 97% of all tele-communicated information being carried over the internet.

2011

By this year, the number of CCTV cameras in UK reached 1.85 million that is to say 1 for every 32 people.

Today

Video surveillance systems are common in homes to prevent from break-ins or unwelcome intruders, and in many public places such as airports, railways, cities, industry, schools etc to record any suspicious activity.

Future

More intelligent and automated systems that will be everywhere and able to prevent and predict abnormal behaviour due to a better understanding of what is in the scene.

2-2-Current systems

CCTV systems are still often used at modern launch sites to record rocket flight and determine possible causes of malfunctions etc., but their usage has changed a lot during the last 15 years. They are now used for various applications such as crime prevention, traffic monitoring, industrial processes, transport safety, retail's control, law enforcement etc.

In the 1990s and 2000s especially, their number increased significantly in public places due to public concern of the growth of general crimes in countries such as United Kingdom. This unexpected rise defined video surveillance as the use of video cameras to watch a scene and monitor behaviour, activities or any change within the scene to influence, manage or protect people. This can only be done by processing and analysing CCTV footage. However, the monitoring and searching of traditional CCTV footage has largely remained an intensive manual process which can involve searching painstakingly through vast quantities of raw video footage for important evidence. Thus, the introduction of computer vision techniques in surveillance tasks made possible an attempt to recognise and track objects, and to minimise the quantity of information stored by automatically searching for key events. However, the ability to automatically identify and monitor important events or objects from real CCTV data remains extremely challenging due to noise and change in lighting, shadows, weather conditions, occlusions etc.

This chapter illustrates what current CCTV systems are able or not able to achieve yet.

First, the company involved in this research project and a brief overview of its history are presented. Then, a comparison table of existing CCTV systems including those from the company involved in this project is provided. The information provided in the rest of this chapter has been directly provided by the company and its CEO as well as the other companies included in the comparison table.

2-2-1-Aralia systems Ltd

This section presents the industrial collaborator for this project and a succinct overview of the company history and its solutions through an interview with the CEO. The information below has been provided by the company.

Aralia Systems Ltd. [19] was founded in 1995 by Dr. Glynn C. Wright, who is the CEO. The headquarters are based in Horsham, West Sussex, England and are connected to an office based in Baltimore, Maryland, USA.

The company has been specialised in surveillance and has provided “intelligent” systems since 1997 in order to enhance the global security of public places such as railway and underground stations, airports, marine ports, urban environments and other public buildings etc. It has also provided solutions to protect utilities valuable assets and for forensic crime investigations.

The rest of this section concerns Dr. Wright's opinion about CCTV, its evolution and his company.

The information provided below has been provided by Dr Wright during the interview from the 3rd of March 2014[2].

In 1998, Aralia introduced its first automated surveillance commercial product which was the very first digital video recorder and video content analysis. It appears it was near or at the same time as the low cost compression for images (MPEG-1) which allowed networks to handle and store the data generated by the larger available bandwidth.

According to Dr. Wright, image processing was made possible when storage and digital images recording technology became available. Previously, computers had to be very close to cameras due to a high cost and installation difficulties of long range analogue systems. Most of Aralia's initial analogue algorithms came from remote sensing while others came from robotics and machine vision. However, as the latter were built for controlled environment, they did not give successful results.

In 2002, Aralia was the first company to introduce searchable image databases. This was made possible by storing image content with metadata that were accessible through SQL queries. Such systems needed a database maintenance system that could operate simultaneously on a number of servers to allow access to different geographical locations.

At the time, image processing was mainly performed inside cameras instead of inside central servers whereas nowadays, this is unpopular. This unpopularity is due to the limited computational capability of cameras and big advantages of having all metadata content readily available to make objects tracking through a scene possible.

Since the early 2000s, a large number of companies with scene content analysis went out of business, which left only around 10 remaining companies world-wide offering professional solutions in 2013. The main reason for so many commercial failures was that the solution promised much more than they delivered due to limitations of the camera quality, restrictions on processing and a lack of 3D data interpretation

Aralia is involved with public as well as private and governmental organizations. During a meeting between Aralia and the US authorities, Department for Homeland Security officials reported that, in 2012, there was no commercial solution to identify left objects that were a threat or to reliably track an individual in a busy area such as an airport or a rail terminus.

According to Dr. Wright, the best ways to attain the next stage for visual surveillance systems to be more successful are:

- The use of more processing power, specifically shorter intervals between scene Analysis;
- The use of higher definition wide dynamic range cameras;
- And the recovery of 3D information from the scene to get a much better chance of interpreting the image with given input.

[2] The author recognizes that some of these points are specific of the opinion of the CEO of Aralia and not necessarily as objective as the rest of the thesis.

Furthermore, his evaluation of what current systems cannot do is the tracking of left luggage, and behavioural understanding. This is due to the incapacity of current systems to cope with the complexity of the scene. This shows a need for 3D information to be recovered and for understanding the context of a scene and the relationship between static background and dynamic elements.

The next section provides a comparison table of current CCTV systems, including Aralia’s.

2-2-3- State-of-the-art in terms of current CCTV systems

In this section, table 1 below summarizes what current intelligent CCTV systems can and cannot achieve as well as what differentiates them from each other. The author recognizes that information provided in table 1 is specific to the opinion of each company and not necessarily as objective as the rest of the thesis.

Companies Systems abilities			Aralia [20]	VCA [21]	Ipsotech [22]	NICE [23]	Indigo Vision [24]	Genetec [25]	Synectics [27]
Integrated	Only Video analytics				✓				
Full package	Storage&recorder&analytics		✓	✓		✓	✓	Third party for video analytics AVI[26]	✓
Tasks	Calibration		✓	✓	NC	✓	NC[3]	✓	✓
	Counting	Vehicles	✓	✓	✓	NC		✓	✓
		Objects	✓	✓	✓	NC		Static	✓
		Human	✓	✓	✓	NC		✓	✓
	Loitering		✓	✓	✓	✓		✓	✓
	Event detection	Perimeter	✓	✓	✓	✓	✓	✓	✓
		Fall	✓	✓	NC	NC	NC	NC	NC
		Smoke			✓				
		Abnormal behaviour	✓			✓			
		Intrusion	✓	✓	✓	✓	✓	✓	
	Object detection	Humans	✓	✓	✓	✓	✓	✓	✓
		Vehicles	✓	✓	✓	✓	NC	✓	✓
		Objects	✓	✓	✓	✓	✓	✓	✓

[3] Non communicated.

		Abandoned or suspicious objects or abandoned machinery	✓	✓	▨	▨	✓	✓	✓
		left or lost luggage	✓	✓	✓	✓	✓	▨	✓
	Facial Recognition		✓	▨	✓	✓	▨	▨	▨
	Tracking	Vehicles	✓	✓	✓	✓	▨	✓	▨
		Objects	✓	✓	✓	✓	✓	✓	▨
		Humans	✓	✓	✓	✓	▨	✓	▨
	Feature Extraction	Edges	✓	✓	✓	▨	▨	▨	▨
		Motion	✓	✓	✓	✓	✓	✓	▨
		Speed	✓	✓	✓	✓	✓	✓	▨
		Size	✓	✓	✓	✓	✓	✓	▨
		Orientation/ direction	✓	✓	✓	✓	✓	✓	▨
		Colour	✓	✓	✓	✓	▨	✓	▨
		Texture	✓	▨	▨	▨	▨	▨	▨
		2.5D/3D	✓	▨	▨	▨	▨	▨	▨
		Logos/Tattoos/ Graffiti or vandalism	✓	▨	▨	▨	▨	▨	▨
		Skin colour	✓	✓	▨	▨	▨	▨	▨
	With change in exterior conditions	lighting	Calibration strategy to reduce sensibility	Sensitive to darkness	Sensitive	Sensitive	Sensitive	Sensitive	Very sensitive
		shadows	Calibration strategy to	Sensitive	Sensitive	Sensitive	Sensitive	Sensitive	Sensitive

			reduce sensibility						
		Weather	Calibration strategy to reduce sensibility	Sensitive to extreme (snow, fog)	Sensitive	Sensitive	Sensitive	Sensitive	Very sensitive
		Temperature	NC	Sensitive	NC	NC	NC	NC	NC
		Occlusions	Sensitive when numerous	Sensitive when numerous	Sensitive when numerous	Sensitive	Very sensitive	Very sensitive	Very sensitive
		Crowded	sensitive	Really sensitive	Really sensitive	Sensitive	Very sensitive	Very sensitive	Very sensitive
	Applications	Retail stores and shopping malls	✓	✓	✓	▨	▨	✓	✓
		Airports, bus, train stations, car parks, industries, prisons, military buildings, city centre (surveillance)	✓	✓	✓	✓	✓	✓	✓
		Bars, clubs or casinos	▨	✓	▨	✓	✓	▨	▨
		Museums, art galleries	✓	✓	▨	▨	✓	▨	✓
		Leisure facilities, sports	▨	✓	▨	▨	▨	▨	▨

		stadia	▨		▨	▨	▨	▨	▨
		Tourist attractions	✓	✓	▨	▨	▨	▨	▨
		Real-time	✓	✓	✓	✓	✓	✓	✓
	Data access, extraction&storage		**database**	**Hard drive**	**Third party**	**Database**	**MP4/USB**	▨	**DVR**
	Particularity		**3D**	**NC**	**Smoke detection &boolean reasoning applied to conditions stringed together**	**Avatar used to detect a specific person**	**NC**	**Processing split between camera&server data extraction before compression**	**NC**

Table 1: State-of-the-art in terms of current intelligent CCTV systems on the market

A description of the requirements for current CCTV systems as well as its perception by the public can be found in appendix 1.

2-4-Summary

According to the comparison table, the main current CCTV systems do not differ considerably from one company to another among the leaders in terms of video analytics. Indeed, most of the products available on the market are able to achieve the same type of abilities such as tracking, event detection, loitering, counting, object detection etc. However, they do have significant differences despite being similar overall. On one hand, from what they provide; some only propose video analytics modules whereas others offer a whole package including video analytics as well as storage and recording modules which can be a benefit in the way it allows customers to extract easily recorded data in case of crime to provide it to the police for any investigation. On the other hand, they can all be distinguished according to their own specific method to deal with one module from all those they offer. Indeed, it is difficult to explain how they all differ in details as companies do not disclose details about their algorithms. However, greatest difference seems to be the *means* by which they deal with a specific task rather than their available features. Moreover, they can also differ in the type of applications they address.

To conclude, it is obvious no one can say exactly which current CCTV system is the best due to the lack of information from the companies for secrecy reasons and also as it is difficult to compare systems that do not exactly deal with identical tasks. However, from an interview of each company represented in the table above made at IFSEC 2014 and Aralia Systems Ltd, they all encounter the same issues that is to say noise and extreme changes in exterior conditions such as lighting (too dark or too bright), weather (snow, fog etc.), occlusions (crowded places) etc. Furthermore, it appears that most of them try to deal with simpler situations than those they are ideally required for by the police or other organisations to deal with real-world crimes. Indeed, the detection of left luggage in the absence of people seems easy whereas it is not as easy in presence of a crowd. From the table above, a conclusion can be established; the modules presented above give better results in the presence of simpler environments or exterior conditions as well as using simpler rules when the environment becomes more complex.

From the current surveillance systems presented in the previous section, only Aralia's is currently investigating 3D information to deal with certain tasks. However, they all focus more on the dynamic aspect of the scene rather than the static aspect of it. Moreover, as mentioned in [28], the attention on surveillance increased significantly recently but there is still a lack of integration between technical practitioners and those who study real-world surveillance installations. This, in addition to information gathered in the previous table, show there is a need for systems to become more practical and realistic to be robust to real world complications such as weather and occlusion. Furthermore, it is our belief that the use of 3D data which does not appear on the market, seems to be one solution to reach this goal due to additional and richer information providing more information about the 3D real scene and its surroundings.

Chapter 3: Literature Review

The broad field of computer vision was arguably first introduced by Larry Roberts [29] in the 1960s when he started discussing methods to extract 3D geometrical features from 2D perspective blocks. It can be argued that these methods pioneered the problem of automated scene understanding, the main focus of this thesis. However, it was only in 1978 that the notion of "scene understanding" was first formalised by David Marr [29]. Since then, it has become of progressively more interest for the research community and has led to new applications in robotics, safety, video surveillance and many more areas. This thesis is primarily concerned with methods for high and low level scene understanding for CCTV-based surveillance technology. This is motivated by the tremendous gap between the capability of existing so-called intelligent systems reported by industry/academic literature and the capabilities desired by industry experts in law enforcement, as will be described in Chapter 7. Indeed, real-world environments, where the gap is particularly apparent, are so complex and noise-ridden that it makes many tasks, such as tracking, object detection and recognition, event detection, classification, feature extraction and scene understanding unfeasible with today's state-of-the-art.

While the advances of computer vision methods and technologies are rapid and continuous, there remains a need for systems to become more robust and efficient to real-world conditions in the presence of noise, shadows, change in lighting, weather conditions, unpredictable behaviour or events etc. These problems can only be overcome if systems become more intelligent to act as if they were witnesses of the scene, allowing for a significant improvement of features extracted and then a better high-level understanding of what is actually happening in the scene. However, this will probably take many more years to achieve. For these reasons, the aim of this thesis is to present a system that will solve every problem encountered in various real-world applications in this thesis, but rather evaluate the capabilities of existing methods in detail and present a suite of novel methods that advance the state-of-the-art of the field for specific environments. In particular, this contribution combined with others will help to make systems more robust, efficient and intelligent and will provide a better understanding of a scene in order to be able to prevent and predict abnormal behaviour or incidents such as left luggage/explosives or aggressive behaviour.

In terms of low- and intermediate-level vision, the principle tasks of scene understanding for surveillance are concerned with camera calibration and segmentation so the focus of the literature review is on these areas first. The main focus of the former being on vanishing point detection – perhaps the most common approach. Higher-level research in the field has been more limited (the lack of prior research is a major motivation for this thesis). Nevertheless, this literature review will conclude with an overview of early efforts at high-level understanding using 3D information and contextualisation methods.

3-1-Vanishing Point Detection

Vanishing point (VP) detection is a well-known task in computer vision, most commonly used to assist other tasks such as camera calibration, to determine if the camera has been moved from its original position, or for 3D reconstruction etc. The research associated with this task consists of various methods to recover the perspective of a scene from vanishing point extraction. Indeed, geometrical elements and their properties provide relevant information such as feature interpretation in images or relationships between the scene and the camera that are significantly helpful for scene analysis and understanding. Moreover, most man-made environments are composed of a huge number of simple geometrical elements such as straight lines, VPs, planes etc which define the perspective. These straight lines, when parallel in the 3D world space, intersect at a common point in the image space and are known as vanishing points. Once the coordinates of VPs have been estimated, they can be used to determine lines' orientation or lead to vanishing lines to assist with camera calibration, 3D Reconstruction, 3D Scene Understanding etc. In this section, a literature review on vanishing point detection methods is provided.

3-1-1-General VP detection methods

VPs correspond to the spatial position where parallel lines from the real-world converge under perspective geometry. The extraction of lines from the image space that correspond to parallel lines from the real-world is a common approach to detect VPs. An example of VPs is provided by the figure 1 below for illustration purposes.

Figure 1[4]: Example of a cube and 3 VPs (highlighted by red circles) [30].

The notion of VPs was first introduced in 1980 by Haralick [31] who established the relationship between camera parameters, 3D scene structure and VPs, called perspective transformation as follows [31]:

$$\boldsymbol{(u, v, w, 1) = (x, y, z, 1) \cdot T^{-1} \cdot M} \qquad (1)$$

Where $(u, v, w, 1)$ is the world coordinate system, $(x, y, z, 1)$ is the image coordinate system, $T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_c & y_c & z_c & 1 \end{bmatrix}$ is the translation matrix and $M = \begin{bmatrix} cos\,\theta\,cos\,\psi + sin\,\theta\,sin\,\phi\,sin\,\psi & -\,sin\,\theta\,cos\,\phi & sin\,\theta\,sin\,\phi\,cos\,\psi - cos\,\theta\,sin\,\psi & 0 \\ sin\,\theta\,cos\,\psi - cos\,\theta\,sin\,\phi\,sin\,\psi & cos\,\theta\,cos\,\phi & -\,cos\,\theta\,sin\,\phi\,cos\,\psi - sin\,\theta\,sin\,\psi & 0 \\ cos\,\phi\,sin\,\psi & sin\,\phi & cos\,\phi\,cos\,\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the rotation matrix with θ, ϕ , ψ the respective pan, tilt and swing angles of the camera.

This approach shows the utility of VPs to determine camera parameters as well as 3D objects' coordinates.

[4] This figure is redacted for copyright reasons.

However, VP detection can become problematic when lines intersect at infinity or outside the image boundaries. So, in 1982, Barnard [32] pioneered the projection of an image into a Gaussian sphere [33] [34], i.e. a unit sphere centred at the focal point as illustrated on figure 2 below.

Figure 2[5]: Gaussian Mapping [32], where $\boldsymbol{n} = (\sin\boldsymbol{\alpha}\cos\boldsymbol{\beta}, \sin\boldsymbol{\beta}, \cos\boldsymbol{\alpha}\cos\boldsymbol{\beta})$.

Thus, planes formed by the focal point and lines, intersect the Gaussian sphere in great circles and their intersections allow VPs to be detected as illustrated by figure 3 below.

Figure 3[6]: Vanishing points on the Gaussian Sphere [32].

The use of a bounded space allowed representing any line configurations whether their intersections were finite or infinite. Nevertheless, the partition of the Gaussian sphere limits the accuracy of the method as it needs to be small enough to avoid clustering lines that should be in different clusters.

[5] This figure is redacted for copyright reasons.

[6] This figure is redacted for copyright reasons.

Moreover, accuracy, memory and computational cost increase proportionally with resolution so they need to be balanced to avoid any major issues. An example of such a scenario is the method presented by Magee et al. [33] which is based on the computation of intersection of pairs of line segments by cross-product operations where clusters of intersections on the Gaussian sphere correspond to VPs. Another example is based on a pyramidal data structure to divide the Hough space and was proposed by Quan and Mohr [34].

However, a disadvantage of previous methods is they dealt with noise by estimating VPs location depending on the accumulator cell dimensions. As a consequence, Collins and Weiss [35] approached VP detection as a Bayesian statistical estimation problem; they assumed VPs to be the polar axis of an equatorial distribution on the unit sphere for each cluster of convergent lines and determined the statistical error of this estimation. In this case, the "interpretation plane" is considered as a random sample of an equatorial distribution on the Gaussian sphere which can be modelled by Bingham's distribution as defined by equation (2) [35] that follows:

$$\boldsymbol{B(x;K,U) = B(K)exp^{\{tr(KU^t xx^t U)\}}} \quad (2)$$

Where $B(K)$ is a normalising constant, $K = diag(k_1\,, k_2\,, k_3) = -\frac{1}{2}S^{-1}$ with $U = [u_1, u_2, u_3]$ is a 3 by 3 matrix and $S = diag(\sigma_1\,, \sigma_2, \sigma_3)$ is a diagonal matrix of variances obtained from the decomposition of the covariance matrix $\Sigma = USU^t$, and illustrated on figure 4 below:

Figure 4[7]: Bingham's distribution – representative contours for varying shape parameter magnitudes [35].

In 1993, Tai et al. [36] based their method on the use of the compound probability (as defined below by equation (3) [36]) of a group of lines to converge at the same point to measure the likelihood for estimated VP candidates to be VPs.

$$\boldsymbol{P(x,y) = \prod_{i=1}^{k} \frac{1}{z_i} \int_{-\pi}^{\pi} p_i(\delta\rho, \delta\theta) \sqrt{1 + \left(\frac{d\rho}{d\theta}\right)^2}\, d\theta} \quad (3)$$

[7] This figure is redacted for copyright reasons.

Where $p_i(\delta\rho, \delta\theta)$ is the error distribution $\delta\rho$ and $\delta\theta$ in ρ_i and θ_i for a given line denoted by a vector $\omega_i(\rho_i, \theta_i)$, and $P(x, y)$ is the compound probability of a line passing through a point (x, y) in the image. This performance measure allowed comparing and discarding VP hypotheses. However, the difficulty of detecting VPs increases when the size of convergent groups decreased due to an increase of the probability of accidental coincidence. To overcome this issue, they proposed to use additional geometrical cues such as vanishing lines to discard false VPs.

Lutton et al. [37] proposed the following method. They first looked at intersection of circles on the Gaussian sphere as in [32] and then used a "semi-regular" rectangular cells quantization based on a regular quantization in ϕ and an irregular quantization in θ as illustrated on figure 5 [37] below:

Figure 5[8]: Almost equal-size cells quantization [37].

After these steps, they obtained a list of directions where each direction corresponds to a possible VP in the image plane of which vote have been attributed by counts in the proposed accumulator space. Using a second and identical Hough Transform (HT), they looked for the most represented three orthogonal directions by looking for a vertical direction orthogonal to the greatest number of directions among the list based on an additional assumption from the observation of man-made environment. They first drew an orthogonal great circle on a second Gaussian sphere for each list of directions and determined the vertical direction with the highest probability at the most represented intersection point on the Gaussian sphere. Then, they determined the two other directions by selecting directions of the list that are orthogonal to the considered vertical direction. To improve their approach, they analysed errors due to image size limitation, image quantization and line detection inaccuracy.

McLean and Kotturi [38] based their VP detection method on the equation of the lines instead of line segments end-points, thus they addressed the clustering problem without any accumulator space, using a line quality measure to optimise VPs location without any a priori knowledge about the number or location of VPs. First, they clustered connected regions of pixels with similar gradient orientations and computed a line equation that fits the cluster's data as illustrated by figure 6 below:

[8] This figure is redacted for copyright reasons.

Figure 6[9]: Relationship between cluster's data and the line equation [38].

To quantized gradient orientations, they used the histogram defined by equation (4) below:

$$\boldsymbol{H}(\boldsymbol{n}) = \sum \boldsymbol{\nabla}_{\boldsymbol{ij}} \tag{4}$$

$$\forall(i,j) | \big((n-1)\theta - \pi\big) < \Phi_{ij} < (n\theta - \pi), 0 < n < k$$

They performed a connected components analysis to analyse the histogram and obtained a segmented image of M Line Support Regions (LSR). Each LSR contained a set of pixel locations and respective gradient information. Then, they weighted each pixel's contribution by its gradient magnitude and applied a principal component analysis (PCA) on the LSR data to obtain line equation. They defined the centroid of a line by the location of the centroid of the gradient weighted LSR data and the orientation by the largest eigenvalue eigenvector of the covariance of the gradient weighted pixel location. The eigenvalues of the covariance matrix $(\lambda_{max}, \lambda_{min})$ provided a quality measure of the line. Then, they assumed all VPs to be on a circle of arbitrary radius r and used a criterion based on each line's unique quality and the angle swept out between a point and a line to assign each line to one VP. Once lines associated to a common VP were clustered, the estimate VP location was updated by computing the point in the image, which minimizes the sum of normalized angular errors, to the lines sharing this common VP.

At the end of the 90s, Tuytelaars et al. [39] used a cascaded Hough Transform with a new parameterisation based on three subspaces instead of one space as illustrated on figure 7 [39] below:

[9] This figure is redacted for copyright reasons.

Figure 7[10]: Illustration of the split of the unbounded space into three bounded spaces with respective coordinates (a, b), $\left({}^{1}/_{a}, {}^{b}/_{a}\right)$and $\left({}^{1}/_{b}, {}^{a}/_{b}\right)$ [39].

[10] This figure is redacted for copyright reasons.

The flowchart of their approach is shown on figure 8 below:

Figure 8[11]: Flowchart of the proposed cascaded Hough transform [39].

According to the flowchart, they applied the HT three times: first from the image into a parameter space where peaks corresponded to straight lines in the image, then into line intersections or VPs after a second HT and into vanishing lines when three VPs were found. Moreover, they applied appropriate filters in between each stage to only keep the strongest peaks for the next stage and obtain a better result.

Van Den Heuvel [40] developed a VP detection method based on combinations of the intersections of three interpretation planes and the orthogonality constraints of the three orthogonal directions of a Manhattan world. For this purpose, they followed the procedure below:

1) Computation of statistical test values for all combination of three interpretation planes; the intersection constraint is defined with the normals to the interpretation planes of the three lines involved such as on equation (5) below:

$$\left[\boldsymbol{n^i}, \boldsymbol{n^j}, \boldsymbol{n^k}\right] = \boldsymbol{det}\left(\boldsymbol{n^i}, \boldsymbol{n^j}, \boldsymbol{n^k}\right) = \mathbf{0} \tag{5}$$

Where n^i, n^j and n^k are the three normal vectors to the interpretation plane of the respective image line i, j and k such as $n^i = x^a \times x^b$ with a and b the i^{th} line end-points.

2) Clustering of lines that converge to a VP based on the test values previously determined.
3) The largest clusters are adjusted based on all independent constraints in the cluster, a line error hypothesis is performed for each line and rejected lines are removed from the cluster. The adjustment is repeated until all lines are accepted.
4) Selection of the largest cluster as the VP cluster.
5) Steps 1) to 4) are repeated for the next two VPs using the orthogonality constraints between the three VPs. Thus, (5) becomes (6):

$$\left[\boldsymbol{n^i}, \boldsymbol{n^j}, \boldsymbol{v_1}\right] = \mathbf{0} \tag{6}$$

[11] This figure is redacted for copyright reasons.

Where $v_1 = n_1^i \times n_1^j$ and is the orientation of the first vanishing point. Another constraint is introduced allowing the detection of the second and third vanishing point simultaneously and is defined by the equation (7) as follow:

$$\left(\boldsymbol{n^i} \times \boldsymbol{v_1}\right) \cdot \left(\boldsymbol{n^j} \times \boldsymbol{v_1}\right) = \mathbf{0} \tag{7}$$

6) Statistical testing of all condition equations for intersection and orthogonality once the three VPs have been detected.

Liebowitz and Zissserman [41] introduced a maximum likelihood estimate (MLE) to detect VPs, In order to minimize the errors that occur in the image. To automate this process, they used orthogonality constraints between the two dominant directions of lines in the image. They obtained these directions with the help of a frequency histogram on line orientation and the frequency weighted by segment length. They assumed lines from two dominant directions to be parallel to determine a VP. Then, they follow the following stages. Assuming there are $n > 2$ line segments l_i, they estimated the VP v and the line segments $\hat{l}_i$ such that $\hat{v}$ lies on each line $\hat{l}_i$ and the line set $\{\hat{l}_i\}$ minimizes the Mahalanobis distance[12] from $\{\hat{l}_i\}$ as it is implied by the ML estimate of the VP $\hat{v}$. They modelled the error in the fitted line segments by a cost function (here an isotropic mean zero Gaussian noise on the end points is used) such that if the end points of l are x^a and x^b, the MLE minimizes:

$$\boldsymbol{C} = \sum_i \boldsymbol{d}_\perp^2\left(\hat{\boldsymbol{l}}_i, \boldsymbol{x}_i^a\right) + \boldsymbol{d}_\perp^2\left(\hat{\boldsymbol{l}}_i, \boldsymbol{x}_i^b\right) \tag{8}$$

Subject to the constraint $\hat{v} \cdot \hat{l}_i = 0, \forall\, i$, where $d_\perp(x, l)$ is the perpendicular image distance between the point $x$and the line l. Figure 9 below illustrates the ML cost function they used.

Figure 9[13]: Geometry of the ML cost function [41].

Thus, given $\hat{v}$, $C(\hat{v})$ can be obtained in closed form and minimized over $\hat{v}$ using the Levenberg-Marquart numerical algorithm. An initial solution for $\hat{v}$ is then determined from the null vector of the matrix $(1, l_2, \dots, l_n)$ via Singular Value Decomposition (SVD). According to them, their method is not well suited for VP detection as it is for line detection from the two dominant directions and state they could obtain a more robust result with the HT.

In 2000, Schaffaliztky and Zisserman [42] proposed a method to cluster pairs of lines using RANdom Sample Concensus (RANSAC) according to one of the following geometric constraints:

[12] It measures the distance between a Point P and a distribution D so here it measures the distance between a line and the line set (cf. http://en.wikipedia.org/wiki/Mahalanobis_distance).

[13] This figure is redacted for copyright reasons.

1) A group of coplanar and equally spaced parallel lines such as steps can be obtained by sampling a family of parallel lines, defined by equation (9) below, at equally spaced values of the parameter λ which gives equation (10).

$$\boldsymbol{L_\lambda : ax + by + \lambda = 0} \tag{9}$$

Where $(a, b)^T$ is the common normal vector.

$$\boldsymbol{L_\lambda = \begin{pmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ 1 \end{pmatrix}} \tag{10}$$

Then, under perspective imaging, the point transformation is $x = MX$ and the corresponding line map is defined by equation (11) below:

$$\boldsymbol{l_\lambda = M^{-T} L_\lambda = M^{-T} \begin{pmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ 1 \end{pmatrix} = A \begin{pmatrix} \lambda \\ 1 \end{pmatrix}} \tag{11}$$

Where A is a 3 by 2 matrix determined up to scale only. Its first column corresponds to the vanishing line of the plane l_∞ and the second column to the line $\boldsymbol{l_0}$.

Thus:

$$\boldsymbol{l_\lambda = l_0 + \lambda l_\infty} \tag{12}$$

The closed form solution for the vanishing line when three lines are considered is:

$$\boldsymbol{l_\infty = [l_1, l_2, l]l_3 - [l, l_2, l_3]l_1} \tag{13}$$

Where $[a, b, c]$ indicates the determinant of the 3 by 3 matrix whose columns are a, b, c and l is any 3-vector for which the determinants in the above equation are nonzero.

2) A pattern generated by translation of some element in a plane as wallpaper can be obtained by repetition of a pattern on a plane by translation and defined by the transformation H in the equation (13) below.

$$\boldsymbol{H = MTM^{-1}} \tag{14}$$

Where M^{-1} corresponds to the operations of back-projection, T the translation and M the reprojection.

On the scene plane, the translation $X' = TX$ can be defined as follow:

$$\boldsymbol{T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (0 \quad 0 \quad 1)} \tag{15}$$

Thus, the image points are related according to equation (16) below:

$$\boldsymbol{x' = HX} \tag{16}$$

And (13) becomes (17):

$$\boldsymbol{H} = \boldsymbol{I} + \boldsymbol{M}\begin{pmatrix}\boldsymbol{1}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix}(\boldsymbol{0} \quad \boldsymbol{0} \quad \boldsymbol{1})\boldsymbol{M}^{-\boldsymbol{1}} = \boldsymbol{I} + \boldsymbol{v}\boldsymbol{l}_{\infty}^{\boldsymbol{T}} \qquad (17)$$

Where v represents the point which is the image projection under M of the direction of translation, l_∞ represents the projection of the line at infinity and is the vanishing line. So, v is the vanishing point of the translation direction and satisfies $l_\infty^T v = 0$.

3) A set of elements disposed in a regular planar grid like tiles on the floor corresponds to a repetition in two directions and can be defined by equation (17) as follow:

$$\boldsymbol{H}_{\boldsymbol{mn}} = \boldsymbol{I} + \boldsymbol{mu}\boldsymbol{l}_{\infty}^{\boldsymbol{T}} + \boldsymbol{nv}\boldsymbol{l}_{\infty}^{\boldsymbol{T}} \qquad (18)$$

Where u, v are the vanishing points of the two translation directions and l_∞ is the vanishing line.

Two years later, Rother [43] applied a heuristic method to architectural environments to detect finite as well as infinite VPs from the three mutually orthogonal directions. They used the unbounded image space as accumulator space and the intersection of all pairs of image line segments as accumulator cells. They addressed the non-convergence issue of the perspective projection of a line segment from a 3D scene with the line segment detected into a 2D image due to noise and lens imperfections. For this purpose, they determined how close a projected line segment s' with a vanishing point vp is to its corresponding detected line segment s. So, they defined a line segment according to the midpoint representation (m_x, m_y, l, α_s) and the perfect line segment s' of a line segment s such as it has the same midpoint as s and vp as vanishing point. Then, they introduced:

- a distance function $d(vp, s)$ between a vanishing point vp and a line segment s defined as the angle α between the corresponding line segments s' and s;
- and a distance function $d(l, s)$ between a line l and a line segment s defined as the tuple (d, α), where d is the distance between l and the midpoint of s and α the angle between s' and l.

These are illustrated on figure 10 below.

Figure 10[14]: (a) Illustration of the distance function between a line segment and a finite vanishing point; (b) Illustration of the distance function between a line and a line segment.

Then, they only defined a vote of a line segment s for an accumulator cell a, if the distance $d(a,s)$is below a threshold t_a. Therefore, the total vote of an accumulator cell a is defined as follow:

$$\boldsymbol{vote(a)} = \sum_{\boldsymbol{all\ s\ vote\ for\ a}} \boldsymbol{w_1}\left(\mathbf{1} - \frac{\boldsymbol{d(a,s)}}{\boldsymbol{t_a}}\right) + \boldsymbol{w_2}\left(\frac{\boldsymbol{length\ of\ s}}{\mathbf{max}\,\boldsymbol{length\ of\ s}}\right) \quad (19)$$

Where w_1 and $w_2$are weights.

To determine vanishing points from the three mutually orthogonal directions, they introduced three different criteria:

- The orthogonality criterion such that :

$$\boldsymbol{\langle cv_1, cv_2\rangle = 0, \langle cv_1, cv_3\rangle = 0, \langle cv_2, cv_3\rangle = 0} \quad \textbf{(20)}$$

Where $\langle\cdot,\cdot\rangle$is the scalar product.

- The camera criterion such that: the principal point and the focal length are inside a certain range, in case they are calculable.
- The vanishing line criterion such that: two different vanishing points have a finite vanishing line if not both vanishing points are at infinity.

Based on the previous criteria, they followed the steps described on figure 11 below to determine VPs:

Figure 11[15]: Algorithm adopted for the search step where the accumulator cells $\boldsymbol{a_1}, \boldsymbol{a_i}$, and $\boldsymbol{a_j}$ correspond to the vanishing points of the three mutually orthogonal directions.

They stated their method is computationally expensive as it is applied to architectural applications and not in real-time. Moreover, they mentioned it would be faster by randomly selecting pairs of

[14] This figure is redacted for copyright reasons.

[15] This figure is redacted for copyright reasons.

accumulator cells $(a_i, a_j)$and checking the satisfaction of the three criteria using RANSAC algorithm. However, they required a partly calibrated camera with unknown focal length and principal point.

In 2006, He et al. [44] presented a method to detect finite as well as infinite vanishing point on a normalised unit sphere to bind the search space. They applied the least squares method to extract straight lines from edges and adapted the K-means clustering method to cluster points on the unit sphere, according to the following conditions:

- "Clusters should be compact within a small circle": This criterion allows removing outliers (clusters that do not satisfy the criterion) and improving the accuracy detection by using a mean update defined as follow:

$$\boldsymbol{U_n^k = \frac{1}{N_k} \sum_{\|x_i^k - u_0^k\| < \rho} x_i^k} \tag{21}$$

Where U_n^k is the update mean for cluster k, U_0^k is the current mean for cluster k, the parameter ρ is the limit of the search area, N_k is the number of points within the search area of cluster k.

- "Clusters should be dense": This criterion is based on the cluster acceptance probability defined as follow:

$$\boldsymbol{\pi^k = \frac{N_k}{N}} \tag{22}$$

Where N is the total number of points. The use of this formula allows discarding VPs from sparse clusters, i.e. clusters with a small probability.

Schmitt and Priese [45] addressed the intersection point clustering problem by combining and improving existing methods [43][46][47] and introducing an intersection point neighbourhood combined to the Automatic Grouping of Semantics. They used an intersection frequency function $\Psi(p)$ defined as:

$$\boldsymbol{\Psi(p) = \left(\Phi^2(p) - \Phi(p)\right)/2} \tag{23}$$

With $\Phi(p)$ is the line frequency function which represents the number of lines that run through $p \in \mathbb{Z}^2$**.**

Then, they defined the intersection point neighbourhood $N(p)$ as below:

$$\boldsymbol{N(p) \coloneqq c_{r_p}(p)\, with\, r_p \coloneqq \hat{r} \cdot \frac{\widehat{\Phi}}{\Phi(p)}} \tag{24}$$

Where $c_r(p)$ is a circle of radius r_p around p.

Finally, they used the Automatic Grouping of Semantics (AGS) to clusters intersection point neighbourhoods that share similar semantics based on the algorithm illustrated by figure 12 below.

Figure 12[16]: Steps of the AGS method used [45].

Finally, they determined VP candidates as the centre of each cluster using the equation (25) below.

$$\boldsymbol{w_G - 1 \le |G| \le \frac{w_G^2 - w_G}{2}} \tag{25}$$

Tardif [48] developed a non-iterative algorithm using the J-Linkage technique [49] to simultaneously detect finite and infinite VPs, considered as output and cluster edges, i.e. as inlier[17] if associated to a VP or as outlier[18] otherwise. The proposed approach followed the steps below:

1) Random selection of a minimum of M pair of edges and computation of a vanishing point hypothesis for each of them;
2) Construction of the "preference matrix" (cf. Figure 13 below as example) of which each row represents an edge and each column a vanishing point hypothesis.

Figure 13[19]: An example of a "preference matrix" [48][20].

During this step, a consistency measure is used to determine the line that minimizes the maximal distance to end points based on equation (26) below:

$$\boldsymbol{D(v, \epsilon_j) = dist(e_j^1, \hat{I})} \tag{26}$$

[16] This figure is redacted for copyright reasons.

[17] Inliers correspond to data which even in presence of noise fit the model parameters considered as input of the method.

[18] Outliers are data that do not fit the model parameters and are often considered as noise.

[19] This figure is redacted for copyright reasons.

[20] Used with permission of the author/publisher.

With $\hat{l} = [\bar{e}_j] \times v$, $\bar{e}_j$ is the centroid and v is the vanishing point. It is illustrated by figure 14 below.

Figure 14[21]: An illustration of the consistency measure [48].

3) Edge classification according to their preference, defined as the intersection of the preference sets of its members [48], and the Jaccard distance defined as below:

$$\boldsymbol{d_j(A,B)} = \frac{|\boldsymbol{A} \cup \boldsymbol{B}| - |\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|} \qquad (27)$$

Where A, B are the preference sets of each edge and are equal to 0 if the sets are identical or 1 if they are different. Moreover, clusters with a minimal Jaccard distance are merged and this step is repeated until the distance between all clusters is 1.

4) Computation of a vanishing point for each edge cluster and dismissal of clusters with less than three edges.
5) Update of vanishing points and classification of each edge cluster using Expectation-Maximization (EM).

Moreover, the computation of VPs during steps 2, 4 and 5 is performed according to the following equation:

$$\boldsymbol{V(S,w)} = \begin{cases} \boldsymbol{l_1 \times l_2 \ if\ S\ contains\ 2\ edges} \\ \boldsymbol{\hat{v} = \arg\min_{v} \sum_{\mathcal{E}_j \in S} w_j^2\, dist^2([\bar{e}_j] \times v, e_j^1)} \end{cases} \qquad (28)$$

With knowledge of the intrinsic parameters of the camera, VPs from the three mutual orthogonal directions of the Manhattan world are identified. A comparison between Ground Truth (GT), the method proposed in this thesis and Tardif’s method is provided in chapter 4 section 4-1-3-3-Quantitative analysis and discussion.

Chen et al. [50] presented a VP detection algorithm during which they applied the HT twice. They first mapped straight lines to a circle with the help of the HT as illustrated on figure 15 below.

[21] This figure is redacted for copyright reasons.

(a) (b)

Figure 15[22]: (a) Line parameterization in polar coordinate system, (b) Illustration of the mapping of a line to a circle where $(\boldsymbol{x_v}, \boldsymbol{y_v})$ correspond to a VP coordinates [50].

Then, they introduced a transform such that: $\{x, y\} => \left\{{}^{k}/_{\lambda_1}, \theta\right\}$ to associate points from this circle to points from a line according to the following equation (29):

$$\boldsymbol{{}^{k}/_{\lambda_1} = x_v \cos\theta + y_v \sin\theta} \qquad (29)$$

Where ${}^{k}/_{\lambda_1}$ is the length of the perpendicular from the origin to the line.

Thus, they defined $s = \lambda_1\, cos\,\theta\, , t = \lambda_1\, sin\,\theta$ to obtain equation (30) below, which corresponds to the equation of the line that represents the circle in the Cartesian coordinate system.

$$\boldsymbol{k = x_v s + y_v t} \qquad (30)$$

Finally, they used a second HT to detect this straight line that and determined VPs coordinates by translating the result of the second HT from the space $\{S, T\}$ to the space $\{R, W\}$as illustrated by figure 16 below.

(a) (b)

Figure 16[23]: (a) Design of the straight line that corresponds to the circle, (b) Second HT resulting straight line in the polar coordinates system.

[22] This figure is redacted for copyright reasons.

From the polar coordinate system $\{R, W\}$, they obtained:

$$\boldsymbol{r} = \frac{\boldsymbol{k}}{\sqrt{x_v^2+y_v^2}} \tag{31}$$

$$\boldsymbol{w} = \boldsymbol{arctan}\left(\frac{y_v}{x_v}\right) \tag{32}$$

Thus, VPs coordinates can be determined thanks to equation (33) and (34) that follow:

$$\boldsymbol{x_v} = \frac{\boldsymbol{k}}{r\sqrt{1+tan^2 w}} \tag{33}$$

$$\boldsymbol{y_v} = \frac{\boldsymbol{k\ tan\ w}}{r\sqrt{1+tan^2 w}} \tag{34}$$

Recently, a range of techniques have been developed.

In 2011, Nieto and Salgado [51] proposed an Expectation-Maximisation (EM) algorithm which compared to other methods detects simultaneously converging lines and their corresponding VPs whether they are finite or not; While the method provides more accurate results as it only clusters line segments from the dominant lines of the scene structure, it does not go without computation expenses; this is due to the algorithm's complexity as the number of required operations is proportional to the amount of input data and the number of iterations of the algorithm.

Bazin et al [52] addressed the line clustering and orthogonal VP estimation using a combination of a Branch-and-Bound (BaB) procedure and the interval analysis theory with a rotation parameterization by Euler angles. Their method has been proven: to perform as well or higher than existing techniques, based on the HT [34], RANSAC with sequential search for VPs, RANSAC with simultaneous search for VPs and a combination of the HT and Expectation-Maximization algorithm; to be robust to various to various elements such as noise, the search region etc and to be faster than previous methods using BaB procedure. However, while the HT and RANSAC remain the fastest, the author stated them not to be as accurate and optimal as their method.

In 2012, Ebrahimpour et al. [53] based their VP detection on the combination of visual information, the Hough transform and the K-means clustering without computation of lines intersection. To reduce the processing time and change in exterior conditions such as lighting, they resized the input image, converted it into grey-scale and applied histogram equalization and applied a 45° filter. So, they extracted edges, then lines with the HT, clustered them using K-means clustering by only using their end-points, then applied the K-means again to find the final cluster of which the centroid corresponds to the VP which is only validate if its y coordinate is close to half of the image height.

Gerogiannis et al. [54] presented a technique to detect VPs in structured environments based on the Direct Split-and-Merge (DSaM) algorithm and a voting scheme. Lines were first detected using the DSaM algorithm. First, the set of extracted edges was split into various clusters of edge points by the long axes of highly eccentric ellipses. Then, neighbouring clusters with collinear major axes were

[23] This figure is redacted for copyright reasons.

merged. Corresponding line segments were obtained by the major axis. VP candidates were then obtained by computing the intersection of all combination of detected lines among which intersection that were outside the image boundaries were ignored. Each VP candidate was then assigned a weight corresponding to the product of the line segments' length that converged to this point. This allowed only promoting VPs candidate that originated from longer line segments. A voting scheme was then applied to select VPs among the VP candidates, where a vote was only attributed to line segments that produced the given intersection. For this voting scheme, they designed a grid divided into equally sized cell over the image space and attributed weights to VP candidates using a kernel function. They stated they obtained accurate results with an average detection error of 2%. They compared their DSaM Algorithm with the HT using their voting scheme with both and showed their method provided more accurate results with an error of the order of 1.54 pixels compared to an error of the order of 15 pixels with the HT.

So far, the evolution and a variety of VP detection methods throughout time in relation to the issues encountered have been presented. The next section describes methods based on VPs applied to camera calibration and 3D reconstruction. It only focuses on these two applications as they fit well with the proposed research project and allows to have a better knowledge of what is achievable in terms of feature and 3D data extraction from VPs coordinates.

3-1-2-VP detection applied to calibration and 3D reconstruction

VP detection methods are of significant use as VPs, if accurately detected, allow automating tasks such as camera calibration, 3D reconstruction, pose estimation, motion analysis etc. So, in this section, the main focus is on papers about VP extraction used for camera calibration and/or 3D reconstruction purposes as these are directly related to our subject. While some of the most famous methods based on VPs for camera calibration and 3D reconstruction will be described in details, others will only be succinctly described. Indeed, camera calibration has not been fully investigated for this research work as the proposed approach only concerned the VP detection stage. However, it was necessary to show how it is possible to calibrate a camera or a computer vision system from VPs as it is considered to be the next logical step to investigate as future work to build a complete video surveillance framework.

Camera calibration is an important task in computer vision which requires a certain amount of information or a priori knowledge about the scene and its geometry to recover intrinsic (focal length, optical centre and factor of distortion) and extrinsic (rotation matrix and translation vector) parameters of the camera. This is possible due to the relationship between VP coordinates and camera parameters. Moreover, VPs can also be useful in motion analysis as they provide 3D orientations and are invariant to 3D translations between the camera and the scene.

In 1986, Liou and Jain [55] assumed VPs' locations remained the same in consecutive frames of a video sequence and used a template to fit them. They applied this method to road tracking based on image sequences. In 1990, B. Caprile and V. Torre [56] were one of the first to use VP properties for camera calibration of a system composed of two or more cameras. Figure 17 below illustrates the calibration set-up they used to proceed.

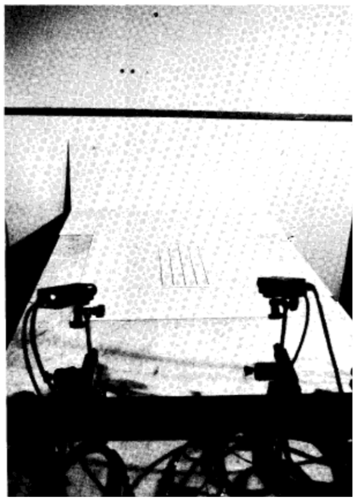


Figure 17: Calibration set-up used in [56].[24]

They first used a single image of a cube to recover the intrinsic camera parameters, i.e. the focal length and the location of the intersection between the optical axis and the image plane. To compute the intrinsic parameters of the camera, they used the following image as can be seen on figure 18 below:

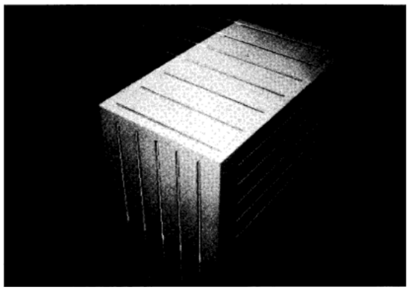


Figure 18: Image of a cube used to recover the intrinsic camera parameters [56][25].

And the equations (35) and (36) below:

$$\begin{cases} \boldsymbol{l} \cdot \boldsymbol{m} = \boldsymbol{0} \\ \boldsymbol{l} \cdot \boldsymbol{n} = \boldsymbol{0} \\ \boldsymbol{m} \cdot \boldsymbol{n} = \boldsymbol{0} \end{cases} \tag{35}$$

Where l, m, n are the three unit vectors associated to the three VPs of three mutually orthogonal directions.

[24] Used with permission of the author/publisher.

[25] Used with permission of the author/publisher.

$$\begin{cases} \boldsymbol{(X_3 - X_1)a + k^2(Y_3 - Y_1)b} \\ \boldsymbol{+(X_1 - X_3)X_2 + k^2(Y_1 - Y_3)Y_2 = 0} \\ \boldsymbol{(X_3 - X_2)a + k^2(Y_3 - Y_2)b} \\ \boldsymbol{+(X_2 - X_3)X_1 + k^2(Y_2 - Y_3)Y_1 =} \end{cases} \tag{36}$$

Where $k = k_1/k_2$ and k_1, k_2 the focal lengths of the camera in x and y-pixel units, a, b are the coordinates of the intersection between the optical axis and the image plane and $(X_i, Y_i), i = 1, 2, 3$ are the coordinates of the three VPs.

Then, they computed the extrinsic parameters of a stereovision system composed of two cameras, i.e. the rotation matrix and the translation vector. First, they estimated VP coordinates following the steps below:

- Edge extraction using a Laplacian-of-Gaussian filter.
- Line extraction using a mean square technique.
- Recovery of the first and second VPs' location.
- Recovery of the third VP using a cross-product and VPs' orthogonality constraint.

Then, they recovered the rotation matrix $\boldsymbol{R}$ by matching the corresponding VPs in the pair of images and using the following equation (37):

$$\boldsymbol{R = V'V^{-1}} \tag{37}$$

Where V is a 3 by 3 matrix whose columns are the three unit vectors associated to VPs, $V' = RV$.

Finally, they recovered the translation vector $\boldsymbol{T}$ using a triangulation method as can be seen on figure 19 below:

Figure 19[26]: Illustration of the triangulation method used to estimate the translation vector [56].

And the following equation (38):

$$\boldsymbol{T = P - R^{-1}P'} \tag{38}$$

Where $\boldsymbol{P}$is a vector that represents a point P in space in the camera coordinate system.

[26] This figure is redacted for copyright reasons.

Caprile and Torre not only demonstrated the accuracy of their method to recover the extrinsic cameras parameters and estimate the length of known segment but also the significance of an accurate calibration as it affects the accuracy of the epipolar transformation that allows to the stereo matching process.

The same year, Wang and Tsai [57] introduced a camera calibration approach based on the use of vanishing lines. As Caprile and Torre, they used a single view of a cube as target which allowed them detecting the three principle VPs as well as the orthocentre of the triangle formed by the three recovered VPs to determine the centre of image. So, they recovered the camera orientation and the focal length of the camera using the slope of lines of the triangle formed by the three recovered VPs. A year later, they proposed another calibration method [58] using this time a hexagonal target, as illustrated on figure 20 below.

Figure 20[27]: Hexagonal used as target for their calibration method [58].

They first extracted the target's edges and vertices after they applied a binary thresholding method on the grey-scale image of the hexagon. The use of the HT allowed them to estimate edges' location and a least-square-error fitting, to determine accurate line equations for each of them. Then, they computed the edges' intersection to obtain the hexagon's vertices. To determine the three VPs, they computed the intersection of each pair of parallel vertices of the hexagon. To determine the vanishing line's equation, they applied a least-square-error fitting method to VPs. Then, they proposed two methods to update VPs location:

- By computing the middle of the intersection between the vanishing line and the pair of parallel lines that produced the given VP. This is illustrated by figure 21 below.

[27] This figure is redacted for copyright reasons.

Figure 21[28]: Illustration of the correction process of VPs' location.

- By selecting a target shape that allows obtaining VPs coordinates not too large in magnitude. Assuming the pan, tilt and swing angles of the camera are small, it is then possible to approximate the $\boldsymbol{u}$ coordinate of a VP as defined by equation (39) below:

$$\boldsymbol{u} \approx \frac{\boldsymbol{A \cdot f}}{\boldsymbol{m \cdot E}} \tag{39}$$

Where $\boldsymbol{m}$ is the slope of two parallel lines that produced the considered VP, f is the focal length of the camera, $A = \cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi$ and $E = \cos\theta\cos\phi$, two terms of the rotation matrix of the camera.

Moreover, they established the relationship between the VL and the pan, tilt and swing angles of the camera as can be seen on figure 22 below:

Figure 22[29]: (a) Relationship between the vanishing line and the pan angle of the camera, (b) Relationship between the vanishing line and the tilt angle of the camera, (c) Relationship between the vanishing line and the swing angle of the camera.

Moreover, they defined each of these relationships with the following equations:

- Configuration (a) :

$$\boldsymbol{tan\,\theta_1} = \frac{\boldsymbol{rm_1 - m_2}}{\boldsymbol{r-1}} \tag{40}$$

$$\boldsymbol{tan\,\theta_2} = \frac{\boldsymbol{rm_1 + m_2}}{\boldsymbol{r+1}} \tag{41}$$

Where $\boldsymbol{m_1}, \boldsymbol{m_2}$ are the slopes of a pair of parallel lines that produced VPs v_1, v_2, $\boldsymbol{r}$ a known ratio value that relates to the relation between the vanishing line and the pan angle of the camera and θ_1, θ_2 two possible values for the pan angle from which one will be determined after computation of the tilt angle $\boldsymbol{\phi}$.

- Configurations (b) and (c):

$$\boldsymbol{u \cdot sin\,\psi + w \cdot cos\,\psi = -\,tan\,\phi} \tag{42}$$

[28] This figure is redacted for copyright reasons.
[29] This figure is redacted for copyright reasons.

Where $(u, w)$are the coordinates of any point in the image space and $\psi, \boldsymbol{\phi}$ are swing angles of the camera, i.e. zoom and tilt respectively.

They also recovered the focal length $\boldsymbol{f}$ of the camera using the following formula:

$$\boldsymbol{f = -w_0 \cdot cos\,\psi / tan\,\theta} \tag{43}$$

Where $\boldsymbol{w_0}$ is the intercept value of the W axis.

While Caprile and Torre recovered the extrinsic camera parameters with an error of less than 1%, Wang and Tsai recovered the focal length and the pan, tilt and swing angles of the camera with an average error of 5%. However, it is difficult to conclude as for the best methods between various with similar error rate as it can be tolerable depending on the application as Wang and Tsai mentioned in their paper.

While Wang and Tsai provided two solutions to avoid the detection of spurious VPs due to line uncertainty, another was proposed by Brillault-O'Mahony [59] in 1992, based on the use of an isotropic accumulator space where the probability of erroneous VP detection would be uniformly distributed throughout all accumulator cells.

Shigang et al. addressed the issue of estimating the change of the camera azimuth using VPs only originating from horizontal and non-parallel lines within the scene [60]. For this purpose, they proceeded as follow:

- Thinned Edge extraction using Sobel operator transform and finding peaks in the edge image;
- Line extraction using the HT keeping only lines longer than 30 pixels.
- Edge tracking and establishment of correspondence between consecutive frames;
- Estimation of the camera rotation. Using VL properties, they defined the VP (x_v, y_v) as:

$$\boldsymbol{x_v = f\,tan\,\alpha} \tag{44}$$

$$\boldsymbol{y_v = 0} \tag{45}$$

Where $\boldsymbol{\alpha}$ is the angle between a horizontal line in the scene and the camera axis and $\boldsymbol{f}$ is the focal length.

Using a vertical rotation axis for the camera, the vanishing line is invariant to the camera rotation which implies the motion of the VP along the vanishing line. They showed the camera motion between time t_1 and t_2 implies the motion of the projection of a horizontal line L in the scene from l_1 to l_2. So, they estimated the camera rotation $\boldsymbol{\theta}$ as follow:

$$\boldsymbol{\theta = tan^{-1}\frac{x_1y'_1 - x'_1y_1}{(y'_1 - y_1)f} - tan^{-1}\frac{x_2y'_2 - x'_2y_2}{(y'_2 - y_2)f}} \tag{46}$$

Where (x_1, y_1), (x'_1, y'_1) and (x_2, y_2), (x'_2, y'_2) are the terminals of l_1 and l_2.

However, to perform this stage, they only considered horizontal lines so they clustered horizontal and non-horizontal lines into two different groups and used only the horizontal cluster to estimate the camera rotation.

- Estimation of the camera locus and location of vertical lines. They computed the former by integrating the camera velocity vector and the estimated rotation. Finally, a triangulation process and the establishment of correspondences allowed them to recover the 3D location of objects as well as vertical lines.

They demonstrated that their method estimated camera loci with an error of less than 0.1m and the position of vertical edges with an error of around 15%. Moreover, they showed that the error rate for the camera rotation estimation is inversely proportional to the number of horizontal lines detected.

Several other calibration techniques were then proposed that involved the estimation of the principal point and the focal length of the camera as well as the adjustment of the aspect ratio of the image based on the properties of extracted VPs and VLs computed from them[61]; the combination of a likelihood function and a stochastic approach to automatically estimate focal length and alignment of the camera from a single image of a scene that satisfies a Manhattan world assumption (i.e. the imaged scene contains three mutually orthogonal directions) [62]; the simultaneous estimation of camera constant, principal point location, two coefficients of the radial symmetric lens distortion and three VPs of orthogonal directions based on a priori information about object geometry [63]; the recovery of the focal length and the optical centre of the camera from two extracted VPs from the geometrical properties of a rectangular prism used as calibration target [64].

More recently, Lee and Nevatia [65] addressed the calibration process in the context of video surveillance based on VP detection. As they stated, the extraction of VPs in such context is a difficult task due to the complexity of the scene that can require user intervention. So, they proceeded the following way:

- A set of parallel lines is drawn by the user;
- Lines' intersection are then computed using a Singular Value Decomposition (SVD) method;
- Recovery of two (or three when possible) VPs;
- Recovery of the third VP when impossibility to do it interactively, using the two other extracted VPs and the centre of the image assuming to be the principal point as can be seen on figure 23 below:

Figure 23[30]: Recovery process of the third VP using the other two and the centre of the image [65].

- Recovery of the rotation matrix and the focal length based on the following equation:

$$\boldsymbol{R_{wc}} = \begin{pmatrix} \frac{\lambda_1(\boldsymbol{u_1}-\boldsymbol{u_0})}{\boldsymbol{f}} & \frac{\lambda_2(\boldsymbol{u_2}-\boldsymbol{u_0})}{\boldsymbol{f}} & \frac{\lambda_3(\boldsymbol{u_3}-\boldsymbol{u_0})}{\boldsymbol{f}} \\ \frac{\lambda_1(\boldsymbol{v_1}-\boldsymbol{v_0})}{\boldsymbol{f}} & \frac{\lambda_2(\boldsymbol{v_2}-\boldsymbol{v_0})}{\boldsymbol{f}} & \frac{\lambda_3(\boldsymbol{v_3}-\boldsymbol{v_0})}{\boldsymbol{f}} \\ \boldsymbol{\lambda_1} & \boldsymbol{\lambda_2} & \boldsymbol{\lambda_3} \end{pmatrix} \qquad (47)$$

Where $\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}$ are the scaling factors, $\boldsymbol{f}$ is the focal length and (u_0, v_0) is the principal point which can be estimated as the orthocentre of the triangle formed by the three orthogonal VPs and $(u_1, v_1), (u_2, v_2), (u_3, v_3)$are the coordinates of the three orthogonal VPs.

- Recovery of the focal length $\boldsymbol{f}$ and the scaling factors $\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \boldsymbol{\lambda_3}$ using the orthogonality constraint between the other columns of the rotation matrix.
- Estimation of the position of the camera using the rotation matrix $\boldsymbol{R_{wc}}$ and correspondence between 3D and 2D points to define the following equation:

$$\boldsymbol{C_{di}} = \begin{pmatrix} \boldsymbol{x_{di}} \\ \boldsymbol{y_{di}} \\ \boldsymbol{z_{di}} \end{pmatrix} = \boldsymbol{R_{wc}} \begin{pmatrix} \boldsymbol{x_i} - \boldsymbol{u_0} \\ \boldsymbol{y_i} - \boldsymbol{v_0} \\ \boldsymbol{f} \end{pmatrix} \qquad (48)$$

Where $C_{di}(x_{di}, \ y_{di}, z_{di})$ is a 3D directional vector and $p_i = (x_i, \ y_i)$ is an 2D image point.

They have demonstrated their method provides better results with mid-range cameras than with near-range cameras, with an error rate of 1 to 4 pixels for the former to one of 4 to 10 pixels for the later.

VPs and their properties have also been used to calibrate a system and perform a 3D reconstruction. A few methods are described here as they represent a potential task that could be added or combined to the proposed approach of this thesis as future work.

[30] This figure is redacted for copyright reasons.

Svedberg and Carlsson [66] addressed the camera calibration, pose estimation and 3D reconstruction tasks thanks to the geometrical constraint of a right angle corner of two rectangular planes as can be seen on figure 24 below.

Figure 24[31]: Diagram illustrating the two rectangular planes or wedge used as target for the calibration process [66].

By projecting the intersection of the parallel edges of the rectangular 3D planes used as target, they obtained VPs. To determine the camera rotation matrix $\boldsymbol{R}$**,** they used geometrical constraints and the projective transformation process between the world and the image space with a pin-hole camera model defined by the following equation:

$$\boldsymbol{u} \sim \boldsymbol{AR}(\boldsymbol{I}|-\boldsymbol{P}_{\boldsymbol{0}})\boldsymbol{P} \qquad (49)$$

where $\bar{u} = (\bar{x}\ \bar{y})^T$ and $u\ \sim\ (\bar{x}\ \bar{y}\ 1)^T$ are the image coordinates with $\sim$ standing for proportionality, $\bar{P}, P$ are the world coordinates, $P_0 = \left(\hat{X}_0\ \hat{Y}_0\ \hat{Z}_0\right)^T$is the 2D centre of projection, $\boldsymbol{R}$ is the rotation matrix of the camera and $A = \begin{pmatrix} \sigma & 0 & \bar{x}_0 \\ 0 & \sigma & \bar{y}_0 \\ 0 & 0 & 1 \end{pmatrix}$represents the internal camera parameters with σ the scale factor along the image axes and $\bar{u}_0 = (\bar{x}_0\ \bar{y}_0)^T$ the coordinates of the principal point in the image space.

Finally, once the internal camera parameters were determined, they used them to estimate the rotation matrix of the camera, computed the centre of projection and synthesized view of the wedge using equation (49), geometrical constraints from their coordinate system's choice, the Singular Value Decomposition as well as other mathematical operations.

Cipolla et al. [67] proposed a method to recover 3D data from uncalibrated images of architectural scenes. For this purpose, they used the strong rigidity constraints of parallelism and orthogonality present in such environment, a user intervention and the perspective projections $\boldsymbol{P}$ for a pin-hole camera model as:

[31] This figure is redacted for copyright reasons.

$$\boldsymbol{\lambda_i}\begin{bmatrix}\boldsymbol{u_i}\\\boldsymbol{v_i}\\\boldsymbol{1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{p_{11}} & \boldsymbol{p_{12}} & \boldsymbol{p_{13}} & \boldsymbol{p_{14}}\\\boldsymbol{p_{21}} & \boldsymbol{p_{22}} & \boldsymbol{p_{23}} & \boldsymbol{p_{24}}\\\boldsymbol{p_{31}} & \boldsymbol{p_{32}} & \boldsymbol{p_{33}} & \boldsymbol{p_{34}}\end{bmatrix}\begin{bmatrix}\boldsymbol{X_i}\\\boldsymbol{Y_i}\\\boldsymbol{Z_i}\\\boldsymbol{1}\end{bmatrix} \quad (50)$$

Where:

$$\boldsymbol{P}=\boldsymbol{K}[\boldsymbol{R}\ \boldsymbol{T}] \quad (51)$$

Where $K=\begin{bmatrix}\alpha_u & s & u_0\\0 & \alpha_v & v_0\\0 & 0 & 1\end{bmatrix}$is the camera calibration matrix with α_u,α_v the scale factors, s the skew parameter and u_0,v_0 the coordinates of the principal point, R the rotation matrix and $\boldsymbol{T}$ the translation vector that relate to the orientation and position of the camera relative to the world coordinate system.

Then, they followed the steps below:

- User intervention to select edges and VPS computation;
- Recovery of the internal camera parameters K using VPs from the three mutually orthogonal directions and the Singular value Decomposition method to solve the equation (52) below:

$$\boldsymbol{KK^T}=\begin{bmatrix}\boldsymbol{u_1} & \boldsymbol{u_2} & \boldsymbol{u_3}\\\boldsymbol{v_1} & \boldsymbol{v_2} & \boldsymbol{v_3}\\\boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda_1^2} & \boldsymbol{0} & \boldsymbol{0}\\\boldsymbol{0} & \boldsymbol{\lambda_2^2} & \boldsymbol{0}\\\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\lambda_3^2}\end{bmatrix}\begin{bmatrix}\boldsymbol{u_1} & \boldsymbol{u_2} & \boldsymbol{u_3}\\\boldsymbol{v_1} & \boldsymbol{v_2} & \boldsymbol{v_3}\\\boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1}\end{bmatrix}^{\boldsymbol{T}} \quad (52)$$

The rotation matrix is then directly computed from VPs and the translation vector is determined from a pair of corresponding points and the epipolar constraint;

- Computation of a projection matrix using the combination of estimated internal and external camera parameters;
- Determination of the epipolar geometry from the projection matrices to find more correspondences and the 3D point location;
- 3D reconstruction combining a triangulation method to the 3D points previously determined.

Guillou et al. [68] proposed a method to calibrate a camera and recover the 3D geometry and photometry of objects using the length of one line segment, at least two VPs and a user intervention from a single image. They first extracted 2 VPs from two sets of parallel lines assuming they are perpendicular to one another and form a rectangle. They used the two extracted VPs from this rectangle to compute the focal length as follow:

$$\boldsymbol{f}=\ \boldsymbol{OP}=\sqrt{\boldsymbol{OPuv^2}-\boldsymbol{PPuv^2}} \quad (53)$$

And illustrated on figure 25 below.

Figure 25[32]: Configuration adopted to recover the focal length [68].

Then, they computed the rotation matrix and the translation vector of the camera using the following configurations as can be seen on figure 26 below:

(a) **(b)**

Figure 26[33]: Configurations adopted to recover (a) the rotation matrix, (b) the translation vector [68].

Finally, they proceeded to the 3D reconstruction following the steps below:

- Design of a planar reference grid;
- Placement of a unit cube in the scene and display in the image space.
- Creation of a parallelepiped that fits the considered object, here a cube, using the motion, rotation and translation of the cube;
- Addition of texture on each face of the parallelepiped according to the image;
- Detection of missing data that corresponds to occluded viewpoint of the considered object and filling of the holes based on a min-max method.

A more recent method was proposed by Criminisi et al. [69] to recover a partial or complete 3D reconstruction from an uncalibrated single view of a given scene exploiting orthogonality and parallelism constraints. They explored how a set of planes parallel to a reference plane and a reference direction not parallel to the reference plane can allow the partial or complete recovery of a 3D scene. Their method is based on recovered VPs, VL and the perspective transformation defined as follow:

[32] This figure is redacted for copyright reasons.

[33] This figure is redacted for copyright reasons.

$$\boldsymbol{x} = \boldsymbol{PX} = [\boldsymbol{p_1}\, \boldsymbol{p_2}\, \boldsymbol{p_3}\, \boldsymbol{p_4}]\boldsymbol{X} \qquad (54)$$

Where x, X are vectors that respectively represent the coordinates of a point in the image space and in the world space, $p_1 = v_X$, $p_2 = v_Y, p_3 = v_Z$ with v_X, v_Y, v_Z the respective VPs to the $\boldsymbol{X}, Y, Z$ directions and $\boldsymbol{p_4} = 0$ is the projection of the origin of the world coordinate system.

To reconstruct partially or completely the scene, they estimated the following measurements:

- Determination of the distance between two parallel planes relative to the distance of the camera centre from one of the two planes as can be seen on figure 27 below:

(a) **(b)**

Figure 27[34]: Illustration of the distance between two parallel planes between two different points $\boldsymbol{x}$ and $\boldsymbol{x'}$, (a) in the world space, (b) in the image space [69].

Using the following equation:

$$\frac{\boldsymbol{d}(x,c)\boldsymbol{d}(x',v)}{\boldsymbol{d}(x',c)\boldsymbol{d}(x,v)} = \frac{\boldsymbol{d}(X,C)\boldsymbol{d}(X',V)}{\boldsymbol{d}(X',C)\boldsymbol{d}(X,V)} \qquad (55)$$

- Determination of the distance between two parallel planes relative to the distance between two other planes as can be seen on figure 28 below:

(a) **(b)**

[34] This figure is redacted for copyright reasons.

Figure 28[35]: Illustration of the distance between two parallel planes to two other planes, (a) in the world space, (b) in the image space [69].

- Estimation of the ratios lengths of parallel line segments on the plane and of the ratios areas on the plane from image measurements by computing the homology $\widetilde{\boldsymbol{H}}$ that defines the relationship between a pair of planes as can be seen on figure 29 below and is defined by equation (55) below.

$$\widetilde{\boldsymbol{H}} = \boldsymbol{I} + \boldsymbol{\mu}\frac{\boldsymbol{v}\boldsymbol{l}^{\boldsymbol{T}}}{\boldsymbol{v}\cdot\boldsymbol{l}} \tag{56}$$

Where $\boldsymbol{v}$ is the vanishing point which is the vertex of the homology, l the vanishing line of the considered plane which corresponds to the axis of the homology and $\boldsymbol{\mu}$ is the scale factor. The computation of the matrix $\widetilde{\boldsymbol{H}}$ allows them to associate each point on a plane into corresponding point on a parallel plane such as $x' = \widetilde{\boldsymbol{H}}x$

(a) **(b)**

Figure 29[36]: Illustration of the Homology transformation between two parallel planes, (a) in the world space, (b) in the image space [69].

The determination of the measurement presented above not only allowed to determine the camera position but also to recover the affine 3D geometry of the scene.

Even though these last few papers showed potential to recover the 3D scene geometry, they are all based on architectural environments and would need to be modified to be adapted to video surveillance. Indeed, in such environment, the complexity of the scene and change in exterior conditions is recurrent and needs to be dealt with to avoid poor performances when applying such methods.

This section described a few methods based on VPs to perform camera calibration and 3D reconstruction. The following section summarizes the findings of the literature review relating to VP detection techniques and such applications and concludes as the feasibility and usefulness of such technique in the proposed approach.

[35] This figure is redacted for copyright reasons.

[36] This figure is redacted for copyright reasons.

3-1-3-Discussion

All the papers described above are based on a large variety of techniques such as the famous SHT or other variants, various accumulator designs, the RANSAC paradigm, the EM algorithm, the MLE technique, the combination of existing methods adding some improvements and some have even been applied to tasks such as camera calibration and/or 3D Reconstruction. All these techniques prove that VP detection has been well researched in the past 50 years but mostly only focus on solving this task based on cubes, parallelepipeds, simple scenes or architectural environments where the perspective is recurrent which makes the VP detection task easier to complete. Moreover, most of the considered environments do not have a geometry as complex as a real-world environment as they do not have many (or often any) occlusions, noise due to weather condition, change in lighting, shadows etc. Indeed, in several papers, these conditions have been stated to make VP detection difficult.

Moreover, the methods previously described have been stated to be accurate by their authors but no significant improvement has been demonstrated as for their possible and successful application in the context of video surveillance. Only one method proposed by Lee and Nevatia [65] addressed VP detection in the context of video surveillance. They actually mentioned the difficulty of such task in such context and only proposed a method that requires user intervention and geometrical assumptions to recover the third VP. This really shows there is a need for VP detection to be applied in the context of video surveillance for the following reasons:

- VPs can be extracted automatically thanks to a variety of proposed methods;
- VP detection needs to be applied to more realistic environment to fulfil its own potential;
- VPs allows automating tasks such as camera calibration and 3D reconstruction without any user intervention or a priori knowledge provided that they have been extracted automatically;
- VPs refer to the scene geometry and are essential to perform 3D Scene understanding.

The key limitation of existing methods therefore is the inability to achieve 3D understanding of *realistic* scenes using an algorithm able to compute vanishing points automatically or semi-automatically and in real-time, giving access to information about the 3D geometry of a scene in the presence of the complications mentioned above (noise, shadows, etc). As seen in the previous section, VPs define the perspective in a scene, allow for the recovery of 3D data which than can lead to a better understanding of what is happening in the scene to help predict or prevent any incidents or abnormal behaviour (cf. chapter 6 and 8 for more details). Motivated by the successes and failures of the above methods, this thesis does not aim to propose a complete new method that overcomes all issues above, but an algorithm that can robustly and automatically extract VPs without any user intervention and in near-real-time from a real video surveillance scene. One of our novelties is the extensive use of real-world scenes which are much more complex than those that have been used previously in the literature.

In the next section, a brief literature review of segmentation methods is given to show how it can help to recover more information from the scene in addition to those provided by lines and VPs extracted.

3-2-Segmentation

While successful VP detection methods have proved invaluable at assisting vision systems in understanding the large scale perspective structure of CCTV scenes, they do little to determine which areas of the scene are of interest and which types of human behaviour should be deemed abnormal. It is thus important for a reliable vision-based CCTV system to robustly segment CCTV images into constituent areas of interest.

In this section, an overview of the state-of-the-art in terms of image segmentation methods is presented. First, the term "segmentation" is defined as it is essential to understand before applying such methods. Image Segmentation [70] consists in the division of an image into distinct regions (set of pixels or super-pixels) where pixels from each region share similar properties such as colour, intensity or texture. However, adjacent regions do not share the same characteristics. This allows simplifying the analysis of an image by analysing part of the image that share similar properties. This is an essential step in image analysis as it enables location of objects and their boundaries in images and also gives a better understanding of the scene organization in terms of safe and dangerous areas. An illustration of such area differentiation obtained after segmentation has been created manually, for visualisation and comprehension purposes, and can be seen on figure 30 below:

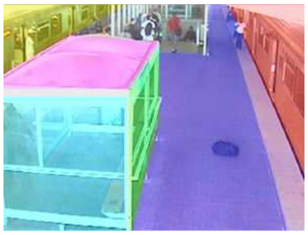


Figure 30: An example of useful segmented regions in the context of video surveillance[37].

Where the colours represent the area of the scene explained below:

- Purple: ground level – expect people standing and walking;
- Pink: above ground horizontal – technically possible to have people standing or sitting but unlikely;
- Yellow, blue, green, orange: barriers to human movement.

This illustration shows how regions that share similar properties within the scene can be clustered according to their characteristics, i.e. colour, texture, dimensions, function, functionality etc. The combination of such segmented information with VPs and 3D data would then provide a good understanding of the static background of the scene and could be of help to human detection

[37] Used with permission of the author/publisher.

systems to improve their accuracy and robustness due to the use of richer data and a variety of features.

Next, follows a comparison table illustrating the principle, advantages and drawbacks of the most common segmentation methods.

Methods		Principle	Advantages	Limitations	Solutions
Histograms = measurement space clustering	***Monochrome Images***	Clusters determined by the interval between histogram's valleys; a specific index and unique grey intensity value applied to pixels of each cluster **[75]**.	• Works well for images with a few distinct objects with very different grey intensities (or grey intensity vectors for colour images) on a nearly uniform background; • Lowest computational time compared to other measurement space clustering method;	• Non-wanted results with segmentation into homogeneous regions (e.g. division of regions or edges that should belong to the same cluster) **[75]**; • No use of spatial information **[71]**; • Computationally expensive for colour images **[71]**;	• Recursive clustering method based on a mask defined as the whole image **[75] [cf. [75] for paper [36] Ohlander]**; • More efficient methods in terms of storage and processing required when applied to colour images **[71]**;
	Colour Images	⇨ Histogram applied to similar components to the Karhunen-Loeve transform, differs from histogram for each RGB channel **[75] [cf. [75] for paper [38] Ohta, Kanade and Sakai]**; ⇨ Apply Region growing around peaks detected in the multi-dimension measurement space **[75]**; **[cf. [75] for paper [34] Narendra and Goldberg] ;**			

Thresholding	***Monochrome Images***	A threshold divides the image into regions with different grey levels by analysing valleys of the image's histogram (simplest case) **[75]**;	• Easy for simplest case (e.g. a bright object on a dark background) **[75]**; • No a priori knowledge **[76]**; • If requirement satisfied: well-defined segmented regions and not complex computationally **[76]**;	• Images with ambiguous peaks or huge and flat valleys **[76]**; • Not based on spatial features => Segmented regions might not be closely connected **[76]**; • Division of multiple dimensions for colour images **[71]**;	**More complex cases:** • Threshold defined by a combination of spatial and grey intensity information **[75]** E.g. (cf. [75] for the following papers) • Spatially adapted threshold for each **pixel (cf. [75] for paper [7] Chow Kaneko)**; • Histogram of only pixels with a high laplacian magnitude **(cf. [75] for paper [48] Weszka)**; • Threshold = percentage of pixels of which neighbour threshold value is different, called busyness **(cf. [75] for paper [49] Weszka and Rosenfeld)**; • Threshold = the value which maximizes the sum of gradients from all pixels of which the grey intensity is equal to the threshold **(cf. [75] for paper [46] Watanabe)**; • Threshold = value which detects more high contrast edges and fewer low contrast edges **(cf. [75] for paper [24] Kohler)**;
	Colour Images	⇨ multiple histogram-based thresholding method **[71][cf. 71 for papers [11] and [12] Goldberg and Shlien]**; ⇨ a threshold to divide the colour space for each histogram's component. ⇨ A line to separate			

			3D points from their projection on this line when using information from the three colour component at a time.			
Region based: Group pixels into homogeneous regions	***Region Growing (RG)*** 1) Seed = selected region ; 2) Seed expansion to include all homogeneous neighbours thanks to an homogeneity criterion; 3) Previous steps repeated until all pixels clustered;	***Single Linkage* [75]**	⇨ Pixels = nodes of a graph ; ⇨ Neighbour pixels with similar properties are joined by an arc ⇨ Simplest case : two pixels are considered similar enough if the absolute value of the difference between their grey level intensity is small;	• Simple method **[75]**; • Spatially accurate placements of boundaries **[75]**;	• Edge gap issue: One leaking arc results into merger of a region with another **[75]**; • Can lead to over-segmentation **[74]**;	• **A criterion can be used to merge regions and then avoid false boundaries [74]**; • **Regions information (colour, texture) also allows to avoid false boundaries [74]**; **Other Similarity definition [75] :** • Difference in grey-level intensity between two pixel normalized by the quantity (square root of 2) times the root mean square value of neighbouring pixel differences over the whole image **[Bryant 1972]** ; • Use of the vector norm of the pixel difference vector in case pixels are represented by vectors;

						• Absolute value of the difference between two pixels has to be small compared to the average absolute value of the centre pixel minus neighbour pixel for each of the neighbourhoods it belongs to **[cf. [75] for paper [1] Asano and Yokoya]** ;
		Hybrid Linkage **[75]**	⇨ Each pixel is assigned a property vector which depends on the K*K pixel's neighbourhood; ⇨ Similarity is a function of the values of neighbour pixels; ⇨ An edge operator can be applied on the whole image to determine if two pixels are joined by an arc;	• More powerful than single Linkage **[75]**; • Homogeneous segmented regions on noisy data due to the similarity definition **[75]**;	• The type of edge operator highly influences results' quality **[75]**;	• Decrease of the gradient threshold of the edge operator can sometimes reduce the gap problem **[75]**; • Improvement can be obtained by combining edge operators and other segmentation methods **[75]**; **e.g.: cf. [75] for paper [36] Ohlander, [50] Yakimovsky, [16] Haralick, [25] Levine and Leemet, [20] Jarvis and Patrick, [43] Pong;**

		Centroid Linkage **[75]**	⇨ Scan (left/right, top/bottom) of the image ; ⇨ Comparison of each pixel value to the mean of a (in)complete neighbouring segment; ⇨ If they are close enough, i.e. a few pixels from each other, pixel are added to the segment and the mean is updated; ⇨ In case of various neighbouring regions, pixel added to the closest; ⇨ In case of too similar means between two close regions: merger of regions and pixel added;	• Does not require a large memory **[75]** so tracking of means easy; e.g.: a hash table, combination of RG methods (cf. next two columns on the right solutions); • Placement of boundaries placed in small gradient areas **[75]**;	•	• Combination of various RG methods **[75]**; e.g.: single band RG using T-test **(cf. [75] for paper 40 Pavlidis and 13 Gupta, Kettig)**, multi-band and F-test **(cf. [75] for paper [22] Kettig and Langrebe, [33] Nagy and Tolaba and [26] Levine and Shaheen)**, RG partitioning the image into same intensity pixels segments **(cf. [75] for paper [2] Brice and Fennema)**.

			⇨ If no neighbouring region: creation of a new segment;			
	Splitting (RS) ***[75] [71]***		1) seed = whole image; 2) if seed not homogeneous, division into four squared subregions = new seed region; 3) Repeated until all subregions homogeneous. Homogeneous if the difference between the largest and smallest grey intensity values is small. e.g.: cf. [75] for papers 45 Robertson, [23] Klinger, [22] Kettig and Landgrebe and [10] Fukada 10.		• resulting image similar to data structure (squared) **[71]**;	

	Merging (RM) ***[75] [71]***	Often combined to RG or RS to merge regions with similar characteristics and obtain the largest possible homogeneous regions e.g.: cf. [75] for papers 19 Horowitz and Pavlidis, [32] Muerle and Allen, [6] Chen and Pavlidis.			
Summary ***[75][71][76]***		+ Better results for an easy homogeneity criterion definition; + Less sensitive than edge detectors to noise due to statistical approach to define homogeneity; + Feature space and pixels' spatial relation considered => Better than space thresholding or clustering methods; + For RG, the edge map helps to decide for merger;		– Seed region selection dependence for RG; – Order of analysis of pixels and regions; – Sequential ; – Computationally expensive in time and memory; – Difficult to decide when the difference between a pixel and the region is small enough to merge for RG; – Avoid boundaries between regions as a seed position for RG;	
Spatial clustering ***[75]***		Combination of histogram and region growing or spatial linkage technique [75]. e.g.: cf. [75] for papers	• Easy implementation **[71]**; • **Direct classification [71]**;	• Undefined way to determine clusters' number **[71]**; • Dependant on image	

		[14] Haralick Kelly, [27] Matsumoto Naka and Yamamoto and [28] Milgram.	**Some methods:** • do not require a difficult measurement space search **[75]**; • Use the gradient edge image more than others **[75]**;	**[71]**; • No spatial information **[71]**; **Some methods:** • requires to try several distinct intervals for each segment **[75]**; • are restricted by small convex segments **[75]**;	
Edge detection [71] (mostly used for grey-scale images)	***Sequential***	⇨ Dependency between result on a set of points and previous set of points' result **[71]**; ⇨ some methods use heuristics and dynamic programming **[71]**;	• For colour images, two regions with similar brightness but that differs from another property can be distinguished **[71]**;	• **Performance depends on the initial point selection [71];** • **Stopping criteria difficult to define [71];** • **Requires a difference in brightness between two regions for monochrome images [71];** • **Only provides region boundaries information or can be combined to other methods to complete segmentation.**	• Current method used : watershed transform similar to real life flood situation **[74]**;

	Parallel	⇨ Can be applied all over the image simultaneously; ⇨ Determination of a set of point to be on an edge does not depend on if other sets of points are or not on one **[71]**; ⇨ Edge detector can be applied on the whole image **[71]**; ⇨ Differential Operators: first-difference operators: Roberts, Sobel and Prewitt ; second-difference operators: Laplacian.		• **Operators requires a distinct change in gray level between two adjacent points [71];** • **Requires very abrupt edges between two regions [71];** • **Sensitive to noise [71];**	
	On colour images	Edge can be defined as: 1) A metric based on distance discontinuities and a colour space **[71]**; 2) Split the image into three monochromes images and merge the results after edge detection		• For 1)&2), similar result as for a monochrome image as in 1D space **[71]**;	

	[71]; 3) Constrain the three colour channels to use them at the same time keeping the edge detection independent from one another **[71]**;			
Summary **[71] [74][76]**		• Provides information about region boundaries ; • Similar to human perception; • Best for images with high contrasting regions; • Can be used to decide the initial position of the seed for RG; • Edges help to decide in which order to process for RG;	• Requires to be combined to other approaches to complete the segmentation ; • Requires well-defined and not too many edges; • More sensitive to noise than thresholding or clustering methods;	
Snake **[74]**: Boundary refinement	⇨ Locate objects boundaries**[74]**; ⇨ Object boundary obtained by minimising the energy function defined on deformation of initial contour **[77]**;	• Solve the problem of false boundaries detection **[74]**; • Effective for the closest contours to the real boundary **[74]**; • Linear object shape description without additional processing **[77]**;	• Do not detect all salient image boundaries **[74]**; • Sensitive to initial conditions **[74]**;	• Use of other approaches (e.g. integrated methods, region segmentation) to detect the real boundaries when not correct **[74]**;
Fuzzy **[71]**	⇨ Deal with uncertainty and ambiguity; ⇨ Iterative optimization method using distances between points and clusters centres to attribute a membership to a point in each cluster according to the degree to which it belongs to an image region or boundary. ⇨ Clusters centres refined according to resulting	• Information from low-level remains to a higher-level **[71]**; • Allows ambiguous clusters boundaries **[71]**; • independent of results from previous levels **[71]**;	• Way to select the number of clusters **[71]**; • Difficult membership determination **[71][76]**; • High computational	

	clusters. ⇨ During iteration: minimization of the distance between a point and a cluster centre and maximization of the distance between clusters centres.			cost for large datasets **[71][76]**;	
Physics models based [71]: To locate objects boundaries by discarding edges from shadows or highlights in colour images.	***Dichromatic reflection model***	⇨ related to materials nature;	• help to identify human faces **[71]**; • Efficient only for known material's reflection properties and easy material to model **[71]**;	• Best to identify or classify material in a scene before processing **[71]**; • Limited to some applications due to use of many assumptions about material type, light source and illumination **[71]**;	• Characterized by the use of distinct reflection models for colour images **[71]**;
	Approximate colour reflectance model (ACRM)	⇨ Shows the Independence between the spectral structure and geometrical scaling of the light reflected.			
Neural Networks (NN) [71]: Used for classification or clustering	***Artificial (ANN)***		• Parallel processing and nonlinear properties **[71][76]**; • Simultaneous examination of various competing hypotheses **[71]**; • Fast classification after training **[71]**; • Easy implementation, based on parallel nature of neural networks **[76]**;	• Very long training time **[71][76]**; • Results can be affected by initialization **[76]**; • Avoid overtraining **[76]**.	
	Hopfield (HNN)				
	Self-organizing Map (SOM)				

	Others	⇨ Back-Propagation (BP);
		⇨ Local Linear Map Network (LLM);
		⇨ Oscillatory Cellular Neural Network (OCNN);
		⇨ Constrain Satisfaction Neural Network (CSNN)			

Table 2: Overview of segmentation methods

3-2-3- Discussion

From the review presented in the table above, it is clear that the choice of a segmentation method depends on the application as well as the type of images to deal with. Moreover, as summary of what is mentioned in many papers such as [71] to [77], it seems that:

- Segmentation is more reliable on colour images than grey-scale images as colour corresponds to richer information than grey intensity;
- The choice of colour space is significant and can allow differentiation of regions from others due to additional property to colour information such as hue etc;
- A more complete knowledge of the scene is required to obtain a well-defined segmentation as well as to design a robust and efficient algorithm;
- The definition of parameters for a chosen method need to be very fine-tuned to provide fine and well-defined segmentation results;
- The choice of initial conditions is significant.

In the case of railway stations and similar environments, due to high presence of noise and change in exterior conditions such as lighting, weather, shadows etc., the best technique to adopt is one that is not too sensitive to noise and that allows performing segmentation without a too strict criteria to avoid merging regions that should be two distinct clusters (perhaps with vague boundaries). One of the methods that appear to be of interest for our application is a fuzzy method such as C-means clustering as it can deal with uncertainty and ambiguity better than any other method. Moreover, before this method has been tested, experiments have been performed with two of the most common segmentation methods: namely, thresholding and a combination of the latter with histograms. This allows demonstrating the difference between these techniques as presented in the table above as well as how they perform on both grey-scale and colour images.

Two of the key requirements for a well-defined segmentation is a detailed knowledge of the scene as well as the use of richer data (e.g. greyscale vs. colour and 2D vs. 3d) so it appears clearly that what could benefit both VP detection and segmentation is the use of additional information from the scene. One way to obtain richer information is to acquire 2.5/3D data from the scene and combine it to 2D VP detection and segmentation results for refinement or confirm the reliability of 3D data once analysed as both 2D and 3D should lead to common information. Thus, the acquisition of 2.5/3D data seems to be the missing step from 2D feature extraction to lead to 3D scene understanding. Furthermore, 2.5/3D data not only provides specific knowledge about the scene itself but also additional information such as depth, texture and scene geometry.

In the next section, various methods that allow capturing 2.5/3D data are described and compared.

3-3-2.5D/3D Methods

This section describes the most common methods used to capture 2.5D[38]/3D data in order to recover 3D scene geometry. Such methods can be classified as active, where a particular light is introduced to the scene, such as a laser, or passive where only raw images are used for information extraction. The most common methods include: laser triangulation, time-of-flight, stereovision method, structured light methods and shape from shading methods. For each category, a description of the general principle and examples of devices based on them is provided.

3-3-1- Laser Triangulation

The laser triangulation method allows recovery of the height of an object or surface observed as a line of laser light is swept across the field of view. The simple computation of depth is then usually applied once the laser has been segmented from the rest of the image. Moreover, the name “triangulation” comes from the triangle formed by each element of the system.

Active triangulation-based systems or laser triangulation [78] [79] follow the principle illustrated on figure 31 and described below:

- A laser emits a light on a surface or object;
- A camera looks for the location of the laser dot or line on the surface or object targeted.

Figure 31[39]: Laser triangulation principle

[38] This refers to methods that do not explicitly calculate 3D geometry, but acquire a different form of three-dimensional information, typically in the form of surface orientation maps/surface normals.

[39] This figure is redacted for copyright reasons.

The distance from the baseline to the object or surface targeted can be established by knowing the baseline[40] b, the viewing angle of the laser α, the viewing angle of the camera β and using trigonometry as in the formula below:

$$\boldsymbol{d} = \frac{\boldsymbol{b\, sin\, \alpha\, sin\, \beta}}{\boldsymbol{sin}(\boldsymbol{\alpha}+\boldsymbol{\beta})} \tag{57}$$

It has various applications in machine vision. Interestingly, it has been applied to:

- visual surveillance as in [80] where the constrained Delaunay triangulation method is used to partition human posture into triangular meshes and extract skeletal features such as body shape, body parts etc allowing its analysis;
- Industrial inspection and object reconstruction as in [81] which reviews active laser range scanners based on triangulation, robotics etc.

However, it suffers from the obvious problem of the need to direct laser light onto the scene which can be both disruptive and time consuming for CCTV applications.

3-3-2- Time-of-Flight

The time-of-flight (ToF) cameras [82] [83] are active 3D image-based scanners composed of an active illumination unit, a lens and an imaging sensor. For each pixel, they allow capture of the intensity (grey-value) and the distance between the object observed and the camera; also called depth.

The working principle is described and illustrated by the figure 32 below:

- The illumination unit emits an intensity-modulated light in the near-infrared range which travels with a constant speed ;
- When the light irradiates an object or a surface, it is reflected back to the camera;
- Then, the lens allows projecting the reflected light onto the imaging sensor and to suppress background light, an optical band-pass filter with the same wavelength as the illumination unit is used;
- Finally, depth information for each pixel can be retrieved by relating the phase of the emitted and received signals.

Figure 32[41]: Time-of-Flight principle

[40] Distance between the laser and the camera here.

[41] This figure is redacted for copyright reasons.

Moreover, as mentioned previously, it can provide the intensity of each pixel which corresponds to the amount of light reflected from a specific point of the surface or object observed.

However, ToF only allows recovery of 2.5D surface of an object in real-time[42] as it does not permit the reconstruction of its full geometry from any point of view.

This is not problematic as some application only requires 2.5D data such as:

- automotive,
- human-machine interfaces,
- gaming, measurements as in [84] where both triangulation and Time-of-Flight are reviewed,
- industrial machine vision,
- real-time 3D-imaging and people tracking as in[85] where tracking is applied on the plan view maps obtained from ToF depth maps based on geometrical features,
- robotics,
- object detection as in [86] where graspable objects are identified from segmented surfaces of point clouds acquired from a ToF,
- Human detection as in [87] where once segmented by a local variation method 3D data are used to extract shape features from each object with GIST SIFT and HOG to identify the most appropriate method and classified as pedestrian or non-pedestrian by a SVM etc.

Another example of device based on the Time-of-Flight principle is the LiDAR.

LIght Detection And Ranging (LIDAR) [88][89][90] is usually composed of a laser range finder, scanner and optics, photodetector and receiver electronics, navigation and positioning systems . Its working principle follows stages below:

- A laser emits a light in the ultraviolet, visible or near-infrared ranges, towards the target;
- The light is reflected by backscattering[43] and scanned into the system using specialised scanners and optics[44];
- The photodetector reads and records the returning signal;
- Navigation and positioning systems[45]determine the absolute position and orientation of the sensor;
- A sensor analyses the time it took the signal to return.

LiDAR allows the recovery of ranges based on the time measured as in the following formula:

$$\boldsymbol{d} = \frac{\boldsymbol{c * t}}{\mathbf{2}} \tag{58}$$

[42] 20-50 frame per seconds.

[43] It consists in the reflection of signals to their original source. Various type can be used depending on the application.

[44] such as azimuth and elevation, dual oscillating plane mirrors, dual axis scanner or polygonal mirrors depending on the purpose of the application.

[45] translate sensor data into static points.

Where $c[m/s]$ is the speed of light, $t[s]$ the time it takes to the signal to return and $d[m]$ the distance between the laser and the target.

This remote sensing technology has many distinct applications such as: agriculture, meteorology, archaeology, forestry as in [91] where aerial images and airborne LiDAR (a technique based on a similar principle) allow the estimation of tree heights in forests, geography, astronomy, military, law enforcement, robotics as in [92] where they addressed the problem of 3D objects segmentation, classification and tracking in the context of ground robot mobility, video surveillance as in [93] where it allows access to 3D data , autonomous vehicles etc.

Despite its inherent ease of use however, the method is expensive and suffers from high levels of noise – especially in real-world settings. Furthermore, the ToF technology is range-limited and has been largely surpassed by the Microsoft Kinect, which is described in section 3-3-4-Structured light, for most application areas. It is thought that the expense and prevalence of noise render these technologies inadequate for our needs in intelligent surveillance.

In the two previous sections, the principle of two range finding methods based on the analysis of the reflected signal has been described. In the next section, the stereovision method that allows extraction of 3D information based on the principle of human binocular vision is presented.

3-3-3-Stereovision

Stereo vision [94] [95] is a passive method that consists in extracting 3D information from two or more digital images captured from two or more different viewpoints. The images can be obtained from an optical sensor (camera) which can be moved so its relative position is known at all times or from two or more fixed or moving sensors that are commonly placed horizontally from one another and remain at the same position at all times. They play the same role as human eyes for human binocular vision that is why when two cameras are used, it is called binocular stereo vision.

Binocular stereo vision systems provide points' spatial location by finding where two lines, passing through the optical centre, and the projection of each point in each image converge. 3D coordinates of an object can be recovered by means of trigonometry from the imaged points of object surfaces and the relationship between optical sensors. In essence a binocular stereo system works by finding correspondence between two views (the most challenging step) and computing depth directly using triangulation.

One of the main methods to tackle this issue is based on structured light projection and will be explained in more detail in the next section.

Stereo vision principle is defined by the way it is set up and it is illustrated by figure 33 below:

Figure 33[46]: Stereovision principle

Stereo vision methods can be divided into two categories: geometric and photometric stereo vision methods. The former is based on the use of two or more cameras and triangulation method to determine the distance from the object to the camera. On the contrary, the latter uses only one camera but requires several images with different illumination conditions and will be described in more details in chapter 6.

One of the devices investigated during this research work is the Bumblebee camera which is based on the stereovision principle. An example of use of this device is described in [96].

[46] This figure is redacted for copyright reasons.

In [96], Xing-Zhe et al. proposed a method to reconstruct a 3D terrain for a patrol robot. Once they captured a disparity map of the environment with a Bumblebee stereo vision system, they estimated the ground plane parameters in real-time using RANSAC algorithm, where inliers are parameters that best fit the ground plane model. As the ground plane is not always flat, the relative position of the ground to the camera can vary which can be problematic. A Kalman filter is then applied to estimate ground plane parameters of the next frame and select inliers in the next frame according to ground plane parameters from the previous frame. Those inliers allow calculating ground plane parameters. The distance between a 3D point and the ground plane can then be obtained by calculating the distance from a point to a plane. The conversion of this distance measure in grey value intensity allows the recovery of the elevation image. Finally, the use of some image processing methods such as thresholding, median filter and mathematical morphology were applied to remove noise from the elevation image and determine the size and locations of obstacles via connected component analysis method.

Stereovision methods can be applied to various areas such as:

- industrial applications as in [97] where they describe the factors that affect the accuracy of a stereovision system taking into account its calibration and measurement processes and they applied this system to two industrial cases such as the quality control of railway sleepers and a car's frame measurement;
- real-time applications as in [98] where they addressed the correspondence problem and the extraction of dense and highly accurate disparity maps in real-time improved thanks to a fuzzy inference system discarding false matches;
- Human-machine interactions as in [99] where the system is able to detect and track multiple people using both colour and position information to improve accuracy and robustness of the system, robotics etc.

Despite some limitations in accuracy, the low cost and flexibility of such methods make the methods potentially suitable for surveillance applications, hence this method has been tested empirically in this thesis thanks to the Bumblebee2 camera bases on this principle (cf. chapter 5 for more details).

In this section, basic notions associated with the stereovision method have been presented. In the next section, a method to recover 3D information from structured light where standard stereo vision fails is described.

3-3-4-Structured light

Structured light 3D scanners [100][101] are active devices that use projected light patterns to recover the 3D shape of objects. They are usually composed of a camera and a projector. This method is similar to stereo vision but here the second camera is replaced by a projector. Furthermore, they are more easily able to solve the correspondence problem as the projector is effectively introducing additional salient features into a potentially featureless scene. Structured light scanners follow the principle illustrated by figure 34 and described below:

- The projector corresponds to the light source as it projects a known light pattern on the surface of a 3D object;
- The camera captures images of the illuminated scene;
- The analysis of the deformation of the imaged pattern with respect to the projected one allows the recovery of 3D information such as shape, position, orientation and texture.

Figure 34[47]: Structured light principle

Then the distance d from the camera to the object can be recovered using the following formula:

$$\boldsymbol{d} = \boldsymbol{B}\frac{\sin(\boldsymbol{\beta})}{\sin(\boldsymbol{\alpha}+\boldsymbol{\beta})} \tag{59}$$

Where B is the baseline, α and β the respective viewing angle of the projector and the camera.

However, the correspondence between the imaged and the projected pattern has to be solved. Some of these methods can be found in [102]. Details about the correspondence issue will not be covered here as it does not relate directly to our research subject.

Two methods allow generating stripe patterns: laser interference and projection. On one hand, the former consists in regular and equidistant stripe patterns created by the intersection of two planar laser beams of which the size is proportional to the angle separating the two laser beams. On the other hand, the latter is based on the use of non-coherent light and the basic principle of video projector. Patterns are generated by a display within the projector, usually a liquid crystal display (LCD) or Liquid crystal on silicon (LCOS).

An example of device based on projected light patterns is the Microsoft Kinect sensor (See chapter 5 for details of its operation). Another example [103] involves two cameras that effectively use the

[47] This figure is redacted for copyright reasons.

pattern as a means to solve the stereo correspondence problem discussed in the previous section. This method is highly accurate but the technology currently struggles to operate near to real-time and at range. It also requires a precise calibration procedure.

The structured light method can be applied to different applications such as:

- 3D imaging as in [104] where they recovered the shape of static and dynamic scenes or objects from a single projection of a structured light pattern (here composed of a simple grid with distinct vertical and horizontal lines) and coplanarity constraints providing simultaneously the position of all connected grid points,
- surface reconstruction as in [105] where they designed an interactive reconstruction system that captures depth maps in real-time with help of a user holding the Kinect and moving within the indoor space to recover a 3D model of an indoor scene, Reverse engineering [106],
- Accident scene investigation,
- gaming, human-machine interaction as in [107] used to detect moving objects by extracting SURF descriptors from the two images and look for matches in the 3D point cloud using RANSAC and refining the results with the ICP algorithm,
- real-time as in [108] where they proposed a method based on the continuous projection of a single composite pattern combining several structure light patterns allowing a real-time 3D reconstruction etc.

Unfortunately, the range of such methods renders their use limited in intelligent surveillance.

In the last section, the basic principle of structured light method has been described. In the next section, the principle of shape from shading methods including photometric-stereo which is one of the methods investigated for this project, are presented.

3-3-5-Shape from shading based methods

Another method for shape (2.5D) recovery is Shape from shading [109]. It is based on photometry method and consists in recovering a 3D surface geometry based on the variation of pixel intensities and assumptions on surface geometry, illumination conditions and reflectance properties. The surface is commonly considered as Lambertian which means that only the albedo[48] and foreshortening[49] determines the reflected intensity.

This method uses shading information from a single image to estimate the surface orientation of the surface or object targeted. This can be done by modelling the observed image intensity in terms of surface orientation and use this relationship to solve for the surfaces' slopes. This was first researched by Horn for his PhD thesis in 1970 [110] and then further analysed [111] [112]. In presence of only a single image, it is difficult to separate gradient from colour or textural information. It implies an ambiguity as to determine if an intensity gradient comes from a slope or

[48] Ratio of reflected to incident light at normal incidence.
[49] Proportional to the cosine of the zenith angle.

colour and pattern change or shadow. Therefore, standard shape from shading is inherently limited to CCTV scenes whereby material properties are already known – a highly limiting prerequisite.

A useful extension to shape from shading is photometric stereo which solves these ambiguities by using two or more images of the same object, captured from one or more known viewing angles under different lighting conditions. Photometric-stereo then permits the recovery of surface normal at each pixel coordinates. An extensive use of this method is proposed in chapter 6 (where further technical details are furnished) due to (1) its ease of implementation, (2) its robustness to shadows and noise and (3) its efficiency of operation. Limitations include small in depth of field and distortions introduced when converting from 2.5D to 3D.

Both can be used for a variety of applications such as:

- face recognition as in [113] where they used a set of images acquired from the same viewpoint under different illumination directions to recover the 3D geometry of a face,
- video surveillance as in [114] where they used only two light sources to detect hidden objects ,
- astrophysics as in [115] where they recovered the relief map of the moon,
- shape and materials extraction as in [116] where they extract BRDFs and 3D shape from a set of images captured under different illumination conditions,
- Capture of 3D geometry of moving objects as in [117] where they recovered surface normal from images acquired with a PS system and recovered the 3D shape by derivation of mesh sequences.

In this section, a description of basic principles and applications of five distinct methods that allow recovery of 2.5D or 3D information have been provided. At this point, the determination of the most appropriate method for the final goal of this project is based on the evaluation of the advantages and limitations of the methods described above. This evaluation is given by the table below.

3-3-6-Discussion

In this section, advantages and limitations of each method described above are shown. The table below summarizes what each method can and cannot reliably achieve and some solutions that can help overcome or compensate for some of them. All the information presented below has been extracted from the literature and are referenced accordingly.

Methods	Advantages	Limitations	Solutions
Triangulation	• High-resolution and high-accuracy **[118][119]**; • Correspondence problem and 3D scene geometry recovered simultaneously **[120]** ;	• Sensitive to perturbation of image coordinates **[118]**; • Sensitive to noise and occlusions **[119]**; • Choice of input is critical as it can lead to incorrect or missing features **[120]**; • Size constrained by baseline **[121]**;	**Invariance to affine and projective transformations [118]:** • Use of a Gaussian noise model; **Reduction or removal of noise/occlusions:** • post-processing on raw data; • multiple triangulation systems to cover occluded area ;
Time-of-Flight	• Better resolution in absence of noise **[121]**; • No post-processing **[121]**; • No baseline or alignment required **[121]**; • No scene lighting conditions required **[121]**; • Reconstruction independent of texture; • Low-cost **[121]**; • Real-time use (20 fps) **[122]**; **LiDAR:** • Accessibility even in difficult environments for other scanners **[124]** ; • Process at night **[124]**; • Large structures and longer ranges usage **[123]**; • Range accuracy nearly constant for a whole	• Reflectance of objects (saturation or no visible data captured) **[121]**; • Noise **[121]**; • Ambient light **[121]**; • Aliasing effect **[121]**; • Motion artefacts **[121]**; • Lack of information from a single depth map **[122]**; • Limited side field of view (FOV) **[122]**; • Objects away from camera occluded by the closest (shadows etc.) **[122]**; **LiDAR:** • Limited resolution due to feature accuracy	**Resolution and quantity of light reflected improvements [121]:** • Higher-power light source ; • Post-processing such as a median filter ; • Longer exposure time ; • Smaller area observed ; • Higher modulation frequency; **FOV improvements [122]:** • Multiple ToF used;

	measurement volume **[123]**;	required and occlusions (obstructions, bridges, tunnels etc.) **[124]**;	
Stereovision	• Real-time use since the 90s **[126] ;** • Works better on rough surfaces with surface orientation discontinuities and textured surfaces with reflectance variations **[127]**; **Global methods [126]** : • Less sensitive to ambiguous regions ; • Establish correspondences difficult to match with local methods; **Local methods [126]**: • Very efficient ;	• Intensive post-processing for depth maps reconstruction **[121]**; • Sensitive to textural surfaces and those with many discontinuities **[125]**(cf. Bumblebee camera) ; • Performance can depend on thresholds' choice and rate constants that control iterative algorithm's convergence **[128]**; • Type of algorithm depends on application **[128]**; • Improvement of accuracy and false matches required **[128]**; • More robust algorithms needed to solve real-world problems **[128] ;** • Inefficient to measure continuous surface with only a few reference points **[125]**; **Global methods [126]** : • Expensive computationally ; **Local methods [126]**: • Sensitive to ambiguous regions ;	**Improvements of point clouds accuracy:** • Addition of texture on targets; Selection of objects with sufficient textural information ; • Use of structured light codification pattern **[125]**;
Structured light	**Compared to ToF, stereo cameras:** • Lower-cost (cf. Microsoft Kinect) **[129]**; • Higher-accuracy point clouds (cf. Microsoft Kinect) **[129]**; • Usually better or similar performance (cf. Microsoft Kinect) **[129]**;	• Limited resolution when object are too far from the sensor (cf. Microsoft Kinect) **[129]**; • Sensitive to occlusions, low surface reflectance or reflection out of camera's scope (cf. Microsoft Kinect) **[129] [125]**;	**Camera/Projector correspondence problem [125]:** • Information of point of emission provided by light reflected ; **Codification projected pattern [125]:**

	• Depth resistant to change in lighting and environment clutters => Better object detection and human hands understanding; • Classification accuracy for object recognition and human activity analysis improved by RGB-D **[129]**; • Provide synchronized colour and depth images ; • Known light source **[125]** ; • Recover objects' shape from deformation of the emitted pattern ; • Coded light reflected provides information about its original point of emission;	• Requires a high illumination contrast **[121]**; • Light sourced sometimes dangerous and more expensive **[121]**; • Limited range (cf. Microsoft Kinect) ; • Expensive computationally and unproductive if only based on geometrical constraints between the camera and the projector **[125]**; • Mutual illumination **[130]**;	• Significant choice; • Depends on application due to advantages and limitations of each codification ; • Comparison **[131] [132]** ; **Ease of Segmentation process [125]:** • New light sources usage; **Algorithm example with similar results as Gray code technique[130]:** • Faster acquisition; • Wider applications;
Shape from shading	• Usage of known as well as unknown **[134]** light source direction **[133]**; • Usually better results on smooth objects **[133]**; • Real-time applications ; **Photometric-Stereo [128] [135]:** • Better on smooth surfaces with few discontinuities; • Better on surfaces with uniform properties; • Full 3D reconstruction; • Combination of silhouette with shading cue;	• Evolution of the objects surface's reflectance **[133]**; • Significant constraints' choice: brightness, smoothness etc., to avoid imprecise results**[133]**; • Sensitive to noise **[133]**; • Poor results on synthetic images and worse on real images **[133]**; **Photometric-Stereo [128] :** • More expensive and sometimes dangerous light sources required for long range; • Results depend on objects' reflectance;	**Improvements of Shape from shading[133]:** • Orthographic projection replaced by perspective projection ; • Combination of shading with other features such as shape, texture, depth map or with stereo; • Usage of multiple images : various camera positions or various light sources positions; **Photometric-Stereo :** • Avoid eye contact when too close to light sources';

Table 3: Table summarizing advantages, limitations and solutions related to 2.5/3D methods described in this section

To summarize, the above table shows advantages and limitations (and potential ways to solve them) for all 2.5/3D methods considered in this section. It appears that they all have many advantages and limitations of which some are common to several techniques. However, it seems the choice of one or the other would depend more on the applications, choice of parameters and the environment where it will be used. Related to the subject, a choice has been made to further investigate the Microsoft Kinect, the bumblebee camera as well as photometric-stereo with two light sources, as they have a reasonable price and they all are well-suited for our application. In the next chapters, details of the experiments conducted with the selected methods are presented. The Microsoft Kinect has been selected for plane extraction from 3D point clouds due to its high speed and direct capture mechanism. These results will be compared to results obtained with the bumblebee camera, which could be more easily put in place in a video surveillance environment because of its longer range. Finally, the photometric stereo represents the main contribution of this thesis in order to build a new algorithm to recognise different shapes from surface normal evolution along a surface of a selected object this is due to its ease of operation and range – a compromise between the Kinect and Bumblebee sensors.

3-4-Summary on the literature

The literature clearly shows what has been done and what needs to be achieved in terms of robustness, accuracy and performance, to be able to solve video surveillance challenges due to the complexity of the scene and change in exterior conditions to address 3D scene understanding.

Moreover, a few papers involve discussion concerning the application of well-researched methods to more realistic scenes.

While VP detection has been well researched until now as mentioned previously, the goal here is to design an algorithm robust to noise and change in exterior conditions to detect lines and VPs from railway station images as it corresponds to the first significant step to be able to intelligently, automatically and rapidly access information about the scene geometry and objects contained within. Moreover, VP detection can be extended to calibrate a system allowing the determination of the distance between an object and the camera or the recovery of 3D scene geometry.

Due to the lack of “realistic” applications for such methods, an algorithm is proposed. The purpose of this algorithm is to be able to detect VPs in a more generic way that could be applied to different types of railway stations scenes with more or less occlusions, noise, shadows, and changes in lighting or weather conditions, as well as directly based on the geometry of the scene and then specific to the type of scenes analysed. Of course, this was intended to be also applied to other types of video surveillance scenes in the future without too much difficulty.

Once VPs are recovered, the best way to recover 3D data for this study case was selected. For this purpose, three of the five methods described in the previous sections have been investigated to choose the one that would allow the acquisition of 2.5 or 3D with an accuracy and resolution that is best suitable for scene understanding. Due to the amount of time and money available as well as features such as: as accuracy, ease of implementation, the ability to use in real-time applications, recovery of point cloud or surface normal and acquisition speed, only the Microsoft Kinect, the Bumblebee camera and photometric stereo have been selected to conduct further experiments.

To address the limitations found during this investigation, photometric stereo is used due to range issues for the Kinect that would not be suitable for video surveillance applications and due to missing 3D data recovered when in the presence of lack of texture with the Bumblebee camera. Both of these weaknesses are also stated in the table from the previous section and were confirmed by our experiments. Moreover, due to the required additional presence of two light-sources in the photometric stereo system compared to video surveillance system already in place, it would be much easier, less expensive and time consuming to combine them.

To conclude, the proposed approach to achieve the final goal might not be “the best”, but it is our belief that it represents a more intelligent, intuitive and generic method as it is based on the scene structure; and that the combination of 2D and 3D features allows obtaining a better understanding of a scene and can be combined without too many complications to real-time systems to improve their robustness and efficiency to lead to crime prevention and prediction and enhanced forensics. What appears to be as a key for future systems, is the combination of features; and this has also been well demonstrated by the literature.

Chapter 4: 2D Image analysis

Image analysis [136] consists in the extraction of meaningful features such as colour, shape, texture, size and orientation, from images. These features are essential for scene understanding and to perform subsequent tasks such as object detection, tracking, classification, object recognition, event detection etc. Thus, they can prove to be indispensable for video surveillance applications. Features such as clothes, hair or skin colour can aid to track a person that appears and disappears from a scene; surfaces' normals and size information can help classify objects by detecting how one differs from another; lines, VPs, VLs and planes can help recover 3D scene geometry.

Therefore, the development of intelligent surveillance systems proves to be intractable in the absence of image analysis. Furthermore, the omnipresence of noise due to distortion from the camera and change in exterior conditions such as lighting, weather, shadows, occlusions, in video surveillance scenes make image analysis even more difficult. Thus, video surveillance systems require richer information to become more intelligent and be able to perform specific and complex tasks such as the detection of left luggage or objects that represent threats, the recovery of 3D scene geometry to differentiate flat from non-flat surfaces, the prevention and prediction of abnormal behaviour to apprehend criminals and avoid incidents etc, in such environment.

The goal of this chapter is to show how the extraction and clustering of 2D features such as lines and VPs, as well as colour information can be extracted from 2D images to improve image analysis in spite of such scene changes and complexity. First, a detailed description of the proposed VP detection algorithm as well as a qualitative and quantitative analysis of results obtained is presented. Then, three common segmentation methods such as: binary thresholding, histogram and Fuzzy c-means clustering are investigated and evaluated to be applied in the context of video surveillance.

The 2D image analysis expected achievements are the following:

- A robust and efficient line and VP detection algorithm even in presence of noise and change in exterior conditions;
- A robust and efficient clustering method to differentiate dangerous from safe areas within the scene based on colour information;
- The possibility to combine geometrical and colour features to be able to recover 3D scene geometry;
- Discussion about the limitations of 2D feature extracted and proposed solutions to address them.

4-1-Line and Vanishing point detection

As discussed previously in chapter 3, the literature reveals a tremendous amount of techniques that are applicable to simple objects and environments such as cubes, parallelepipeds and architectural scenes where perspective cues are so numerous that VP detection becomes relatively straightforward. On one hand, many methods have been proposed for various specific goals such as to address boundary issues, to improve VP detection accuracy without increasing memory and computational cost and to deal with noise. On the other hand, only a few have been applied to real scenes with a greater complexity such as video surveillance environments. Furthermore, in spite of their usefulness to perform camera calibration and 3D reconstruction, lines, VPs and colour information remain essential cues for contextualisation[50], behaviour analysis and object classification in the context of video surveillance. Indeed, the knowledge of geometrical cues and colour information can contribute to obtain a better and clearer understanding of physical events as they allow recovering the scene geometry and dividing the scene space into various areas according to what it is composed of. This point will be described in more details at the end of the chapter in section 4-3-Discussion.

Lines and VPs are recurrent in man-made as well as video surveillance environments. Indeed, the former is made of a great number of buildings, roads, streets etc, which are composed of a large number of parallel lines, simple geometrical cues such as cubes, parallelepipeds etc; themselves composed of lines and planes, which provide additional perspective information. In such environments, the perspective is so recurrent and the space so well organised that the automatic line and VP extraction becomes comparatively straightforward. Scenes such as railway stations and other busy concourses tend also to have a large number of parallel lines. However, as mentioned previously, the recurrence of noise within such environment makes the automatic and real-time VP extraction problematic. Therefore, algorithm parameters and thresholds need to be optimised to detect VPs coordinates accurately.

For the above reasons, the proposed approach is based on a noise removal and clustering method specific to railway and underground stations but can be adapted for use elsewhere. The complexity and variation in exterior conditions for such environment required several refinements of our algorithm to reach the current version. However, only the final version will be discussed here; refinements will only be described when necessary for the comprehension the proposed approach.

To summarize, the proposed approach to detect VPs in railway and underground stations follows steps below:

1. Pre-processing: Image enhancement.
2. Feature extraction:
    a. Edges using the Canny edge detector.
    b. Line parameters using the Standard Hough Transform.
3. Line clustering into those emanating from each VP.
4. Computation of intersections of each pair of lines for each cluster.
5. Intersections clustering.
6. Extraction of VPs coordinates from each intersections cluster.

The novelty of this approach resides in the use of a specific voting scheme allowing dissociating lines that lead to VPs from those that lead to simple intersection points by keeping only the intersection

[50] It consists in reasoning about a set of connected feature, either 2D or 3D features, extracted from a scene and put them in context to have a better understanding of what is happening within the scene; e.g. geometrical cues and colour information can help to recover a 3D scene geometry, classify objects and then can help to infer behaviour when connected to human detection system output.

points of the former as VP candidate. Moreover, another contribution of this approach is the adaptation of two common 2D feature extraction methods, the Canny edge detection filter and the Standard Hough Transform, to more complex environments. Indeed, the amount of lines and then intersections in video surveillance environments is so important that it creates additional noise to the one already present within the scene due to exterior conditions.

In the next section, a description of the Canny edge detection filter and the Standard Hough Transform are given to facilitate the comprehension of the proposed approach in sections that follow.

4-1-1-Theoretical background

In this section, two common methods allowing extracting 2D features are described: the Canny edge detection filter that is used in the proposed approach to extract edges and the Standard Hough Transform, which is used to extract line parameters.

4-1-1-1-Canny edge detection filter: Origin and theory

The Canny edge detection filter was developed by John F. Canny in 1986 [137].

The goal of this filter is to detect the optimal edge when the three following conditions are satisfied:

- A good detection such that only real edges must be detected;
- A good localization such that the distance between detected and real edges must be minimum ;
- A minimal response such that edges can only be detected once and image noise should not be detected as false edges.

This filter is most commonly known as the optimal detector due to its widespread use and successes in many areas of computer vision. Canny succeeded to meet the above conditions by using the sum of four exponentials, approximated by the Gaussian's first derivative, as the optimal function determined by the calculus of variations[51].

The algorithm is composed of multi-stages [138] [139] [140] described below:

1. Noise reduction: Application of a 5x5 Gaussian filter on the image, e.g. the kernel of a 5x5 Gaussian filter with a standard deviation $\sigma = 1.4$ would be:

$$\boldsymbol{B} = \frac{\mathbf{1}}{\mathbf{159}} * \begin{bmatrix} \mathbf{2} & \mathbf{4} & \mathbf{5} & \mathbf{4} & \mathbf{2} \\ \mathbf{4} & \mathbf{9} & \mathbf{12} & \mathbf{9} & \mathbf{4} \\ \mathbf{5} & \mathbf{12} & \mathbf{15} & \mathbf{12} & \mathbf{5} \\ \mathbf{4} & \mathbf{9} & \mathbf{12} & \mathbf{9} & \mathbf{4} \\ \mathbf{2} & \mathbf{4} & \mathbf{5} & \mathbf{4} & \mathbf{2} \end{bmatrix} \tag{60}$$

[51] This method determines a function which can optimize a given functional.

2. Gradient magnitude and angle calculation: Gradient can be determined at each pixel using the Sobel operator; this can be done by estimating the gradient in the x and y direction respectively by applying the two kernels below:

$$\boldsymbol{K_{G_x}} = \begin{bmatrix} -\mathbf{1} & \mathbf{0} & \mathbf{1} \\ -\mathbf{2} & \mathbf{0} & \mathbf{2} \\ -\mathbf{1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (61)$$

$$\boldsymbol{K_{G_y}} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{1} & -\mathbf{2} & -\mathbf{1} \end{bmatrix} \quad (62)$$

Then, the gradient magnitudes can be calculated using the Euclidean distance measure by applying Pythagoras theorem as follow:

$$|\boldsymbol{G}| = \sqrt{\boldsymbol{G_x^2} + \boldsymbol{G_y^2}} \quad (63)$$

Where G_x and G_y are the gradients in the x and y directions respectively.

Finally, the direction of the edges can be calculated thanks to the following formula:

$$\boldsymbol{\theta} = \mathbf{tan}^{-1}\left(\frac{|\boldsymbol{G_y}|}{|\boldsymbol{G_x}|}\right) \quad (64)$$

3. Non-maximum suppression: It consists in removing pixels that are not edges by keeping only pixels on an edge which have the highest gradient magnitude;
4. Hysteresis thresholding: Use of two thresholds, t_{low} and t_{high} to determine real edges:
    a. If a pixel has a gradient magnitude of intensity higher than t_{high}, it is considered as a real edge;
    b. If a pixel has a gradient magnitude of intensity lower than t_{low}, it is discarded;
    c. If a pixel has a gradient magnitude of intensity included between t_{low} and t_{high} and any of its neighbours, in a 3 by 3 region, has a gradient magnitude higher than t_{high}, is it considered as a real edge;
    d. If only the first of the two previous conditions is satisfied, it is considered as an edge only if any of its neighbour, in a 5 by 5 region, has a gradient magnitude higher than t_{high}; otherwise, it is discarded. This stage allows removing noise by assuming edges to be like long lines.

The main limitations [139] of this algorithm are its computational cost and effectiveness as they are proportional to:

- The size of the Gaussian filter: While smaller filters would detect small sharp lines that are not significant and would smooth less the image, larger ones would smooth the image too much and suppress noise better;
- The hysteresis thresholds: If too high, information can be missing whereas false detections (noise) can arise if too low.

According to our knowledge of the literature, no technique has yet been identified to select a generic value for the hysteresis thresholds that would permit Canny edge detector to detect edges on all or most type of images. This reason justifies the difficulty encountered to select and explain a choice for the Hysteresis thresholds in the proposed approach

While this section gave an overview of how the Canny filter detect edges, the next one describes the standard Hough Transform principle to detect lines.

4-1-1-2-Standard Hough transform

This section first gives a brief history of the Hough Transform and then describes the theory behind the Standard Hough Transform used in the proposed approach to detect lines.

- Origin

The Hough transform appeared for the first time in 1962 in a patent filed by P.V.C. Hough [141]. At the time, it was used to recognise complex patterns in order to detect and plot subatomic particles' tracks in bubble chamber photographs. This transform was then defined as a transformation able to map a point to a straight line. Hough thus invented the famous Hough Transform that is popular today. However, his transform required many other steps to get to the one that is used nowadays and that has been for decades by the computer vision community.

In 1969, the Hough patent was referenced for the first time in a book entitled "Picture Processing by Computer" [142] published by Rosen A. Rosenfeld. There, he presented an alternative technique based on a point-line transformation defined by the following equation:$y = y_i x + x_i$, where (x_i, y_i), i=1,…,n are a set of points. Then, if points are collinear, their corresponding lines in the Hough space all go through a single point. Moreover, he stated that nearly parallel lines intersect at infinity if points belong to a line almost parallel to the x axis. To address this problem, he recommended to exchange x and y. He then described the Hough space as an array of counters and was the first who gave an explicit algebraic form to the transform used today.

The first polar parameterization, more commonly used today, was first described by R.O. Duda and P.E. Hart in [143]. The polar parameterization came from an alternative mathematics approach based on integral geometry to study the probability of random geometric events. It was then defined by the equation below:

$$\boldsymbol{y = -x\frac{\cos\theta}{\sin\theta} + \frac{\rho}{\sin\theta}} \quad (65)$$

Duda and Hart realised this equation allows mapping a point (x_i, y_i) in the image plane to a curve in the Hough space according to the following equation, obtained from (9):

$$\boldsymbol{\rho = x_i \cos\theta + y_i \sin\theta} \quad (66)$$

in a $\rho - \theta$ transform space.

Since, their paper has been cited and referenced many times in the literature and the research on the Hough Transform and its variation has increased significantly. The Hough transform became especially popular in computer vision in 1981 thanks to Ballard [144]. Moreover, nowadays, the introduction of the Hough transform by Duda and Hart [143] is still a standard technique in computer vision. A recent paper [145], published in 2009 by P.E. Hart, describes the invention of the Hough transform, as known nowadays.

The next section describes the theory behind the famous Standard Hough Transform.

- Theory

In the image space, a line can be expressed with two variables and various coordinate systems, i.e. with the parameters (m, p) in the Cartesian coordinate system or with the parameters (ρ, θ) in the polar coordinate system. In the proposed approach, the coordinate system used is the polar parameterization.

In the polar coordinate system, lines are defined by the Hough Transform as illustrated on figure 35 below:

Figure 35[52]: Definition of a line in the polar coordinate system

And the following equations:

The equation (9) can then be expressed as follow:

$$\boldsymbol{\rho = x\cos\theta + y\sin\theta} \tag{67}$$

Or:

$$\boldsymbol{\rho = ax + by} \tag{68}$$

Where

$$a = \cos\theta \ \text{ and } b = \sin\theta$$

Where ρ and θ are defined as can be seen on the figure 6 above.

In other words, for a given point(x_0, y_0), a family of lines going through this point can be identified by the following equation:

$$\boldsymbol{\rho = x_0\cos\theta + y_0\sin\theta} \tag{69}$$

Where each pair of coordinates (ρ, θ) corresponds to a line that goes through the point(x_0, y_0).

[52] This figure is redacted for copyright reasons.

The plot of each couple (ρ, θ) is a sinusoid which represents all possible lines passing through the point(x_0, y_0). So the plot of the couple (ρ, θ) for each image point is a set of sinusoids which only intersect when points belong to the same line in the image. The couple (ρ, θ) of sinusoids intersection corresponds to line parameters.

The detection of lines in the Hough space can then be defined as the search of the number of intersections between curves. The larger this number, the larger is the number of points belonging to the line. The Hough Transform is usually applied with a threshold which corresponds to the number of intersections. If the number of intersections is higher than the threshold selected then the line is deemed significant in the image.

As mentioned previously, through years, many variations of the Hough transform have arisen. The most famous method is the Standard or Generalised Hough Transform (SHT) [141] [143] [144] as described above, and among different variations, such as the Randomized Hough Transform (RHT) [146], the Probabilistic Hough Transform (PHT) [147], the Progressive Probabilistic Hough Transform (PPHT) [148] etc. However, the proposed approach only uses the SHT to detect lines as described previously.

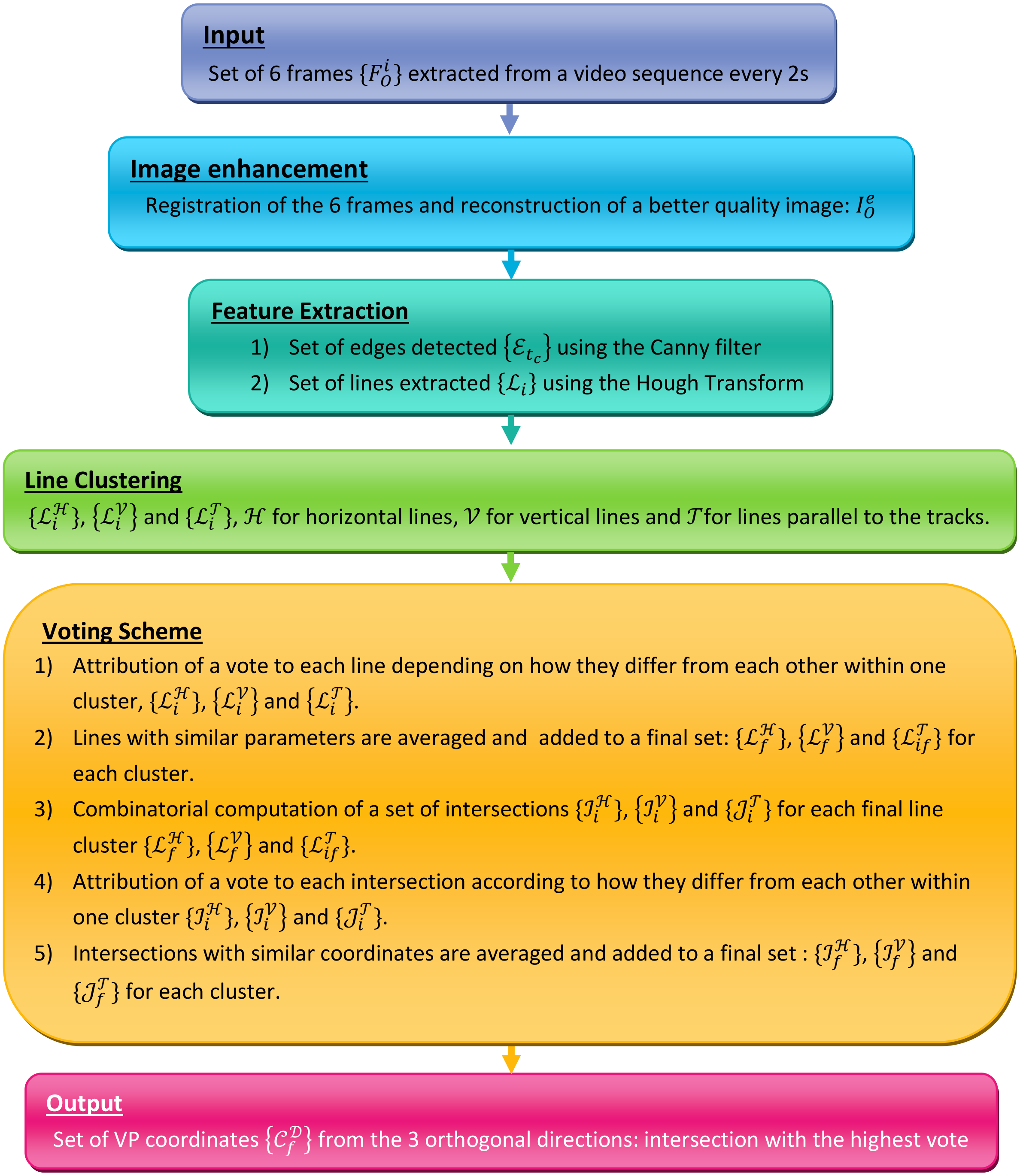
The next sections describe in detail every stage of the proposed approach to detect VPs and provide a qualitative and quantitative analysis of results obtained.

4-1-2-Method proposed

This section presents the details of the proposed approach to detect VPs in the context of railway and underground stations. An overview of the algorithm is first given; then, every stage is explained in detail; finally, results are presented and analysed quantitatively and qualitatively.

4-1-2-1- Algorithm Overview

The proposed method is an automatic VP detector, using the Canny edge detection filter and the Hough Transform to extract edges and lines respectively, and based on a specific voting scheme to cluster lines emanating from VPs. As mentioned in the introduction to this section, the type of scene to which the algorithm is applied, is quite complex and noisy due to the recurrence of geometrical cues and simultaneous change in exterior conditions such as weather, lighting, shadows, and mixture of dynamic and static people and objects causing occlusions. For this reason, a voting scheme, specific to observations that are common to most or all scenes observed, has been designed. Indeed, this scheme allows differentiating lines that emanates from VPs to those that lead to simple intersection points while removing or reducing noise. Its specificity then allowed improving lines and vanishing point detection considerably. The proposed algorithm follows steps illustrated by the diagram below:



Input

Set of 6 frames $\{F_O^i\}$ extracted from a video sequence every 2s

Image enhancement

Registration of the 6 frames and reconstruction of a better quality image: I_O^e

Feature Extraction

1) Set of edges detected $\{\mathcal{E}_{t_c}\}$ using the Canny filter
2) Set of lines extracted $\{\mathcal{L}_i\}$ using the Hough Transform

Line Clustering

$\{\mathcal{L}_i^{\mathcal{H}}\}$, $\{\mathcal{L}_i^{\mathcal{V}}\}$ and $\{\mathcal{L}_i^{\mathcal{T}}\}$, $\mathcal{H}$ for horizontal lines, $\mathcal{V}$ for vertical lines and $\mathcal{T}$ for lines parallel to the tracks.

Voting Scheme

1) Attribution of a vote to each line depending on how they differ from each other within one cluster, $\{\mathcal{L}_i^{\mathcal{H}}\}$, $\{\mathcal{L}_i^{\mathcal{V}}\}$ and $\{\mathcal{L}_i^{\mathcal{T}}\}$.
2) Lines with similar parameters are averaged and added to a final set: $\{\mathcal{L}_f^{\mathcal{H}}\}$, $\{\mathcal{L}_f^{\mathcal{V}}\}$ and $\{\mathcal{L}_{if}^{\mathcal{T}}\}$ for each cluster.
3) Combinatorial computation of a set of intersections $\{\mathcal{I}_i^{\mathcal{H}}\}$, $\{\mathcal{I}_i^{\mathcal{V}}\}$ and $\{\mathcal{I}_i^{\mathcal{T}}\}$ for each final line cluster $\{\mathcal{L}_f^{\mathcal{H}}\}$, $\{\mathcal{L}_f^{\mathcal{V}}\}$ and $\{\mathcal{L}_{if}^{\mathcal{T}}\}$.
4) Attribution of a vote to each intersection according to how they differ from each other within one cluster $\{\mathcal{I}_i^{\mathcal{H}}\}$, $\{\mathcal{I}_i^{\mathcal{V}}\}$ and $\{\mathcal{I}_i^{\mathcal{T}}\}$.
5) Intersections with similar coordinates are averaged and added to a final set : $\{\mathcal{I}_f^{\mathcal{H}}\}$, $\{\mathcal{I}_f^{\mathcal{V}}\}$ and $\{\mathcal{I}_f^{\mathcal{T}}\}$ for each cluster.

Output

Set of VP coordinates $\{\mathcal{C}_f^{\mathcal{D}}\}$ from the 3 orthogonal directions: intersection with the highest vote

Figure 36: algorithm overview

In summary, the proposed algorithm uses a sequence of six successive frames (of resolution 640*480) as input, extracted every 2s from the same video sequence of a railway or underground station. These images are then combined to form a better quality image which contains the same information as the set of original images with an enhanced quality. The use of only one and better quality image reduces the time of computation and allows removing noisy features from the input image to facilitate line and VP detection. The new input is a higher resolution (1280*960) image from the same scene from which edges and lines are extracted using the Canny edge detector and Hough transform respectively.

The images used to design and test the proposed approach, have been extracted from CCTV footage provided by Aralia. This CCTV footage has been captured by cameras with fixed or variable focal lengths. However, it did not affect the processing applied by the author as video sequence were not recorded when the camera was moving or its settings were changing. If it was the case, it is easy to understand it would create noise due to movements of the camera and a change in settings that would affect the perspective in the scene and would require a calibration process updated every time the camera has moved or changed settings.

Moreover, the proposed processing applied on the images such as the conversion into grey-scale, image enhancement, the edge extraction using Canny filter, line extraction using Hough Transform and intersection points coordinates computation using Singular Value Decomposition, have all been implemented by the author, in C/C++ based with the use of pre-written functions available in the OpenCV library or in matlab.

Due to a tremendous quantity of lines in such environment, a specific a priori knowledge allowed clustering lines into three distinct directions. This step will be explained in section 4-1-2-4-Feature extraction and line clustering. Once lines are clustered, a voting scheme is applied to each cluster to select lines with the highest probability to lead to a VP in a given direction. A set of intersection points is then computed from all pairs of remaining lines in a combinatorial way. Finally, the voting scheme is applied to resulting intersection points allowing distinguishing simple intersections from VPs. These steps are essentials as they make VP detection easier and reduce the computational cost significantly. VPs from the three orthogonal directions of the Manhattan world are then extracted from each resulting cluster as intersections with the highest score.

The next section first describe preliminary observations from the scene that allowed to detect edges, lines and VPs from railway and underground stations without being too affected by noise and changes in exterior conditions such as lighting, weather, shadows, occlusions etc.

4-1-2-2-Preliminary observations

In this section, important observations made from the scene analysed are described as they allowed deciding which step to pursue next.

Railway and underground stations video sequence provided by the company appeared to be really complex and noisy in some cases due to changes in exterior conditions such as lighting, shadows, weather, distortion of the camera and also dynamic changes within the scene such as human motion. Crowds etc. creating occlusions. Preliminary tests showed that these changes were not the only one responsible for noise, thus creating incorrect edges or removing edges leading to the detection of missing or incorrect lines and then VPs.

Indeed, not only changes in exterior conditions and dynamic changes within the scene could create noise or remove relevant features, but also, too many lines originating from the geometry of the scene, as can be seen on figure 37 below:

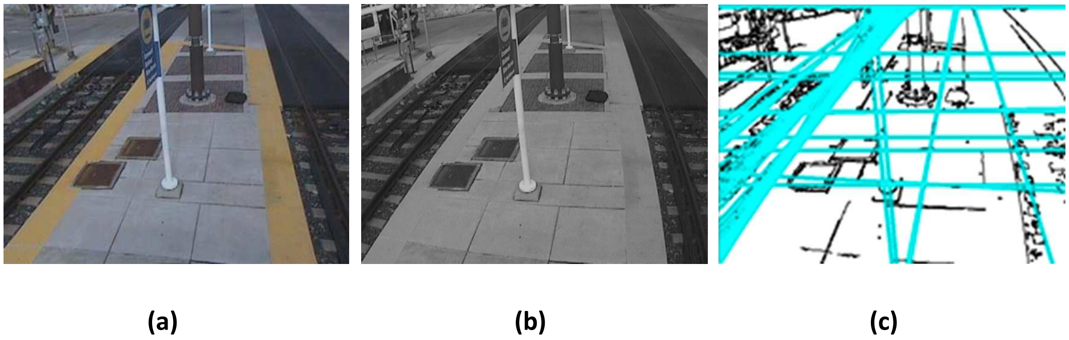


Figure 37: (a) Example of an original colour image, (b) Example of the same original colour image converted into grey-scale, (c) Example of lines detected by the Hough Transform from the grey-scale image after edge detection was performed[53].

It appears clearly on these images that lines in some direction intersect with lines in other directions making the line extraction process more difficult. This is the example of lines in the direction of the tracks and lines that are perpendicular to the tracks. The number of lines in the direction of the tracks is so important that it creates a line cluster and makes it almost impossible to differentiate one line from another. Moreover, the distinction is neither facilitated by the addition of lines from other directions that intersect with this line cluster. Furthermore, the amount of lines in the vertical direction, as can be seen on figure 8 above, is very low compared to the amount of lines in the other directions which can then prevent from extracting enough lines, i.e. more than one in a given direction, to detect a VP in this direction. Thus, the extraction of relevant lines that emanate from VPs and the dismissing of others can become problematic to successfully detect VPs due to the issues approached above.

Other tests have then been performed to understand how it would be possible to address these issues to detect only or almost only an adequate amount of relevant lines at the same time as discarding those that are not of interest. These tests were performed on 64 frames extracted from the video sequence of each scene. Then, the number of lines to detect was selected to be 50 as a too low number would lead to an insufficient amount of lines while a too high number would lead to too many lines, including false lines originating from noise. The detection of 50 lines on 64 frames can be restraining computationally; however, it was chosen empirically and only for preliminary test purposes. The tests involved the edge extraction and line detection stages for arbitrary threshold values: the hysteresis thresholds used were $t_1 = 50$ and $t_2 = 180$ and the Hough transform's threshold used was $t_H = 100$ .Then, another threshold was introduced to only extract relevant lines, from the 50 lines extracted, that are most likely to intersect at accurate VPs at the same time as reducing or removing noise without losing any relevant feature. This threshold represents the total number of lines detected on 64 frames. It is defined as follow: $1 \leq t_{\mathcal{L}} \leq 64$, where 1

[53] Used with permission of the author/publisher.

represents the minimum number of frame and 64 the maximum. Thus, when $t_{\mathcal{L}} = 1$, only the maximum number of lines detected on 64 frames remains whereas when $t_{\mathcal{L}} = 64$, it is the minimum. It then allowed to visualise the variation of the amount of lines and noise by manually determining the best threshold for each scene corresponding to a number of lines in each direction neither too high to add noise nor too low to miss relevant information. An example of this experiment can be seen on figure 38 below.

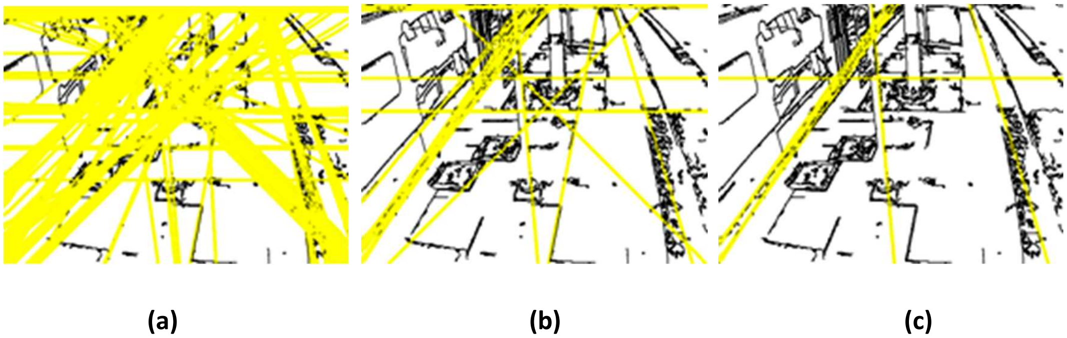


Figure 38: Example of lines detected and displayed for a different threshold value$t_{\mathcal{L}}$: (a) $t_{\mathcal{L}} = 1$, (b) $t_{\mathcal{L}} = 50$, (c) $t_{\mathcal{L}} = 64$[54].

In most scenes provided by Aralia, this manual testing revealed the best threshold to be around 50 and was therefore not overly dependent on manual fine-tuning. This number permitted to address noise issues at the same time as not confounding or excessively slowing the system down. However, the goal of the proposed approach was not only to detect VPs robustly but also automatically. Even if this manual testing appeared to be a potential solution to the issues described above, it allowed removing noise in the direction of the tracks but made the number of lines insufficient in the vertical direction and vice versa. While the removal of noise without dismissing too many lines can be potentially improved, an insufficient number of vertical lines cannot be solved entirely as it originates from how railway and underground stations are designed. Indeed, in the real-world, some public places do not have many lines in some direction due to the lack of objects with perspective in this or these direction(s). This can usually be the case in video surveillance when the considered direction is in the field of view of the camera as it is the case here for the vertical direction. Moreover, other observations about the scenes showed that the images were distorted due to the radial distortion of the camera lens and the image quality was rather poor.

To summarise, these observations made from the scenes to analyse revealed several issues:

- Distortion ;
- Low-quality ;
- Noise from changes within the scene;
- Noise from a multitude of lines in some direction due to scene geometry.

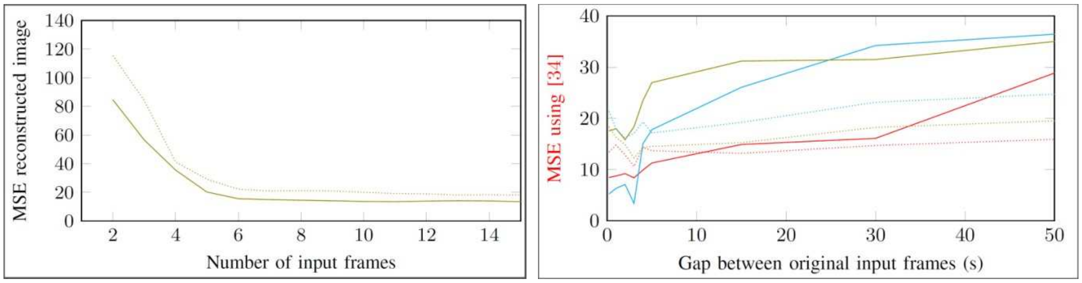
[54] Used with permission of the author/publisher.

An obvious solution to solve some of these issues then appeared to be pre-processing stages such as Image enhancement and a Smooth filter.

In the next section, the pre-processing stages are described.

4-1-2-3-Pre-processing

After careful observation of the scenes to analyse, a multi-frame enhancer as in [148] was applied to the algorithm's input set. This latter was composed of 6 frames extracted from the same video sequence and separated by an interval of 2s. The choice of the number of frames is justified by the graphs as can be seen on figure 39 below.



(a)

(b)

Figure 39: Justification for the choice of number of frames and the interval between frames used: (a) Graph showing the error during the reconstruction process according to the number of frames, (b) Graph showing the error during the reconstruction process according to the interval of time between the frames selected

Figure 39 (a) represents the mean-square error (MSE) of the reconstructed images as a function of the number of original frames used. This graph clearly shows the reconstruction stage is better when five to seven frames are used which explains why six frames were used as input to the proposed approach. Figure 39 (b) represents the MSE of the reconstructed image as a function of the time interval used to select each frame. According to this graph, the best results are obtained when the input frames are taken every two or three seconds. The justification is that shorter gaps do not improve image quality whereas larger gaps introduce noise due to changes within the scene such as lighting etc., which explains why six frames used as input were extracted every two seconds in the proposed method.

The image enhancement and distortion removal performed consisted in several stages:

1. Image registration: it is based on a subpixel method in the frequency domain proposed by Vanderwalle [148], to align common features from original frames of the input set.
2. Image reconstruction: it used interpolation to reconstruct a better quality image from the frames registered previously.
3. Removal of the radial distortion of the camera lens: it followed Grammatikopoulo's method [149].

The first step allowed removing undesirable effects caused by small vibration of the camera due to exterior conditions such as the train arriving and leaving the station, wind etc.

An example of a better quality image obtained after image enhancement and distortion removal can be seen on figure 40 below.

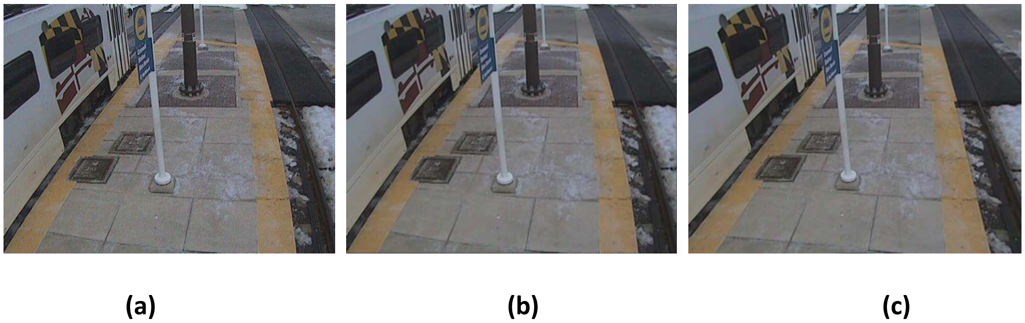


Figure 40: Example of an enhance image: (a) First of the six original frame, (b) High-Resolution Enhance image, (c) High-Resolution and undistorted Enhance image[55].

After image enhancement and distortion removal, the new input of the algorithm was then a higher-resolution (1280*960) image I_O^e. Then, the new input image was converted into grey-scale I_G^e, as it is required before applying the Canny filter, and a Gaussian smooth filter was applied on the grey-scale image I_{GS}^e to smooth the image and remove noise even further. The latter was performed since colour information has no proven advantage and CCTV cameras are often subjected to high noise levels. Further to previous experiments, additional experiments have been performed to extract edges and lines from the enhance image. An example of edge and line detection performed on an original frame and the same but enhance frame can be seen on figure 41 below.

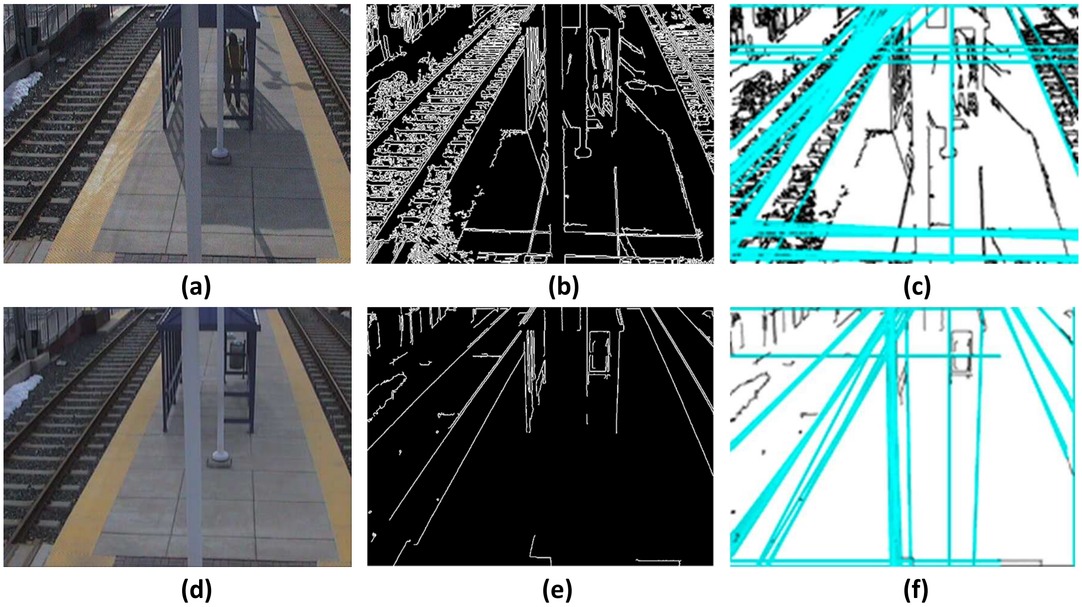


Figure 41: Comparison of lines detected between an original image and its enhanced version: (a) Original frame, (b) Edge detected image, (c) Lines detected superimposed on Canny filtered image, (d) High-Resolution Enhance image, (e) Edge detected High-Resolution Enhance image, (f) Lines detected superimposed on Canny filtered High-Resolution Enhance image[56].

[55] Used with permission of the author/publisher.

[56] Used with permission of the author/publisher.

As can be seen on figure 41 above, it is apparent that the edge and line detection performed on the enhance image is of much better quality and clearer than on the original frame. Moreover, the noise has been reduced and it is then more easily to differentiate one line from another in most of the direction including a sufficient number of lines in the three orthogonal directions of interest.

To summarize, this section described the pre-processing stages adapted to reduce or remove noise and to be able to obtain a sufficient number of relevant number of lines at the same time as dismissing those that are not of interest to detect VPs. These stages have been implemented by the author in Matlab. They have been proven to be significantly efficient as they improve the edge and line detection even further and succeed to reduce noise at the same time as keeping a sufficient number of lines in the three directions of interest making the process of distinction between those that are not of interest and those that lead to VPs even easier.

In the next section, the feature extraction and line clustering stages are described in details.

4-1-2-4-Feature extraction and line clustering

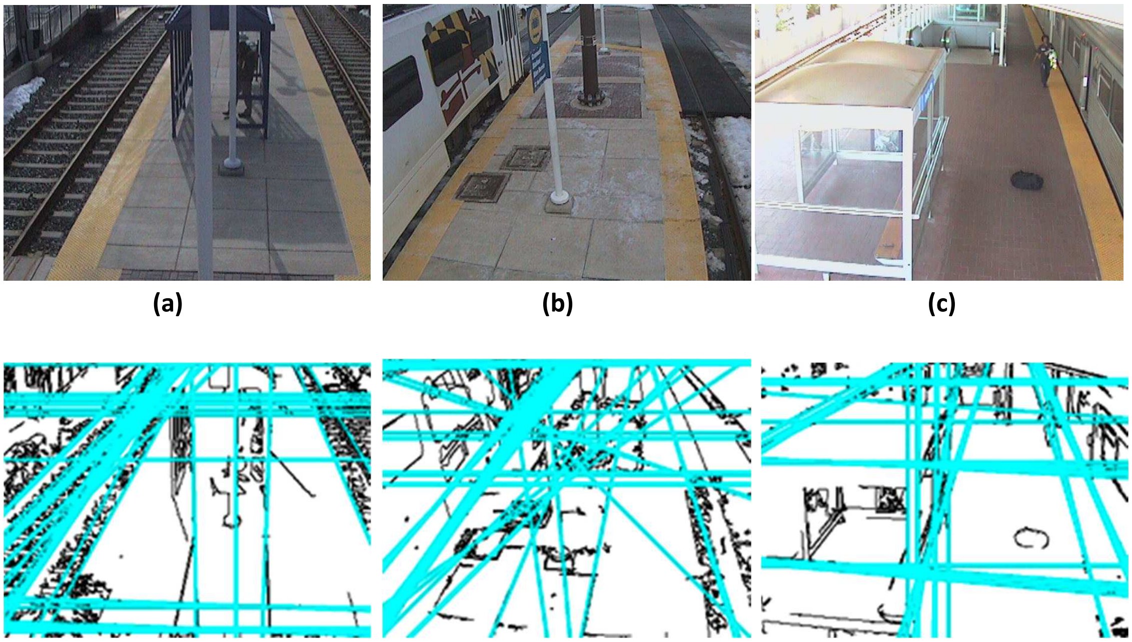
The feature extraction methods used to extract edges and lines that is to say Canny detector and the Hough transform have been performed by the author using pre-written functions of the OpenCV library. More details extracted from the OpenCV website can be found in appendix 2.

After improving the quality of the image and succeeding in reducing noise, the Canny edge detection filter as described in section 4-1-1-1-Canny edge detection filter: Origin and theory, has been applied to I_{GS}^{e}. The Canny edge detector was chosen as it has been stated in [150] to perform better than other edge detectors. As mentioned in this paper, it performs better than other edge detectors in almost all scenarios and is not as sensitive to noise. However, its computational cost is higher which can be a problem for real-time applications unless parameters are fine-tuned. Moreover, the most important reason for choosing this edge detector is its insensitiveness to noise as it is essential in the study case presented here. Indeed, as mentioned several times already, noise is omnipresent in video surveillance scenes and even if the improvement provided by an enhancement method and a smooth filter are significant, not all noise must have been removed so it is important to use a method that is not too sensitive.

For the Hysteresis procedure, the following threshold values were used: $t_1 = 50 \text{ and } t_2 = 200$. After processing of the Canny filter, a set of edges $\{\mathcal{E}_{t_c}\}$ was detected. Then, the Hough transform was performed on $\{\mathcal{E}_{t_c}\}$ to detect lines and extract their parameters. The threshold value used for the Hough Transform was$t_H = 100$. The output of the Hough transform was a set of line parameters $\{\mathcal{L}_i = (\rho_i, \theta_i)\}$, where ρ_i is the distance between the line and the origin and θ_i is the angle of the vector from the origin to the closest point of the line as can be seen on figure 6 in section 4-1-1-2-Standard Hough Transform, Origin.

At this stage, the threshold values used for Canny and SHT were selected empirically based on experiments where the best results for our application were estimated. Furthermore, these thresholds need to be correctly adapted depending on the application, the type of scenes and the result expected. However, there is no generic technique yet allowing the selection of a threshold value for Canny that works well on all type of images. As seen in section 4-1-1-2-Standard Hough Transform, the coordinate system used to express lines in this approach was the polar coordinate system. Moreover, the number of lines detected during the SHT can be adapted but a number of 50 lines were chosen arbitrarily. The prevalence of noise would rise by introducing a greater number of lines including false detections while an insufficient number would not provide enough information to lead to VPs; indeed, the number of lines to use for VP detection has to be superior to 1.

As mentioned in section 4-1-2-2-Preliminary observations, the simultaneous analysis of all lines in all directions would be difficult to be able to differentiate lines from each other as well as simple intersections from potential VP candidate. Moreover, by observing all railway and underground stations video sequences available, it is noticeable than lines are organized accordingly to a same pattern. An example of this pattern can be seen on figure 42 below.



(a) (b) (c)

(d) (e) (f)

Figure 42: Example of real-scene used to test our algorithm: (a), (b) and (c) are three different real railway station images and (d), (e), (f) show the same set of images after the first steps of our algorithm applied (Colour Image->Grey-scale->Smoothed-> edges detected(Background of the images)->Lines detected(superimposed on the canny filter result set[57].

As can be seen on figure 42 above, the lines detected from three distinct scenes all are organized the same and according to the three following direction:

[57] Used with permission of the author/publisher.

- the horizontal direction which corresponds to the direction perpendicular to the tracks' direction and can be defined as lines with an angle θ≈90° or 270° as lines appear to be almost horizontal;
- the vertical direction which corresponds to the direction of the poles and can be defined as lines with an angle θ≈0° or 360° as lines appear to be almost vertical;
- the direction of the tracks which corresponds to all the remaining lines, i.e. lines which with an angle different from 0°, 360°, 90°, and 270° as they appear to be neither horizontal nor vertical.

Once lines were detected, only VP detection was left. Thanks to observations from the scene, it has been made possible by following the steps below:

- Line clustering ;
- Line distinction using a voting scheme ;
- Intersection computation ;
- Intersection distinction using a voting scheme ;
- VP detection.

In the rest of this section, only the lines clustering is described but the other steps will be described in the following sections.

Based on observations from the scene, line clustering was then performed by grouping lines into three distinct clusters according to the three distinct directions they are organised such as:

- A set of horizontal lines $\{\mathcal{L}_i^{\mathcal{H}}\}$;
- A set of vertical lines $\{\mathcal{L}_i^{\mathcal{V}}\}$;
- And a set of lines in the direction of the tracks, i.e. neither horizontal nor vertical $\{\mathcal{L}_i^{\mathcal{T}}\}$.

This was made possible as explained previously by clustering lines according to their angle. The clustering was performed using different parameters for each cluster. Horizontal lines have been selected according to the following conditions: $\theta \approx 90° \pm 3° \; or \;\; \theta \approx 270° \pm 7°$; the Vertical lines with $\theta \approx 0° \pm 3° \; or \;\; \theta \approx 360° \pm 7°$; and the Tracks lines with θ different from $90° \pm 3° \; or \; 270° \pm 7° \; or \; 0° \pm 3° \; or \; 360° \pm 7°$. The line clustering output was then three distinct sets of line parameters $\{\mathcal{L}_i^{\mathcal{H}}\}$, $\{\mathcal{L}_i^{\mathcal{V}}\}$ and $\{\mathcal{L}_i^{\mathcal{T}}\}$.

To summarize, this section described feature extraction methods and line clustering process used in the proposed approach to detect VPs. The next section shows how it was possible to differentiate lines using a voting scheme to lead to a potential VP candidate.

4-1-2-5-Line distinction process

This section explains how lines within a cluster have been differentiated from one other using a voting scheme.

The steps presented below are the same for each line cluster: horizontal, vertical or tracks. However, to illustrate the voting scheme used only an example based on the horizontal cluster is given to facilitate comprehension. The horizontal cluster has been chosen to exemplify this process as it usually contains fewer lines than other direction and make it easier to understand how it works. An example of horizontal line cluster is shown in table 4.

ρ	θ in degree
958	90.000002
466	90.000002
462	90.000002
458	90.000002
470	90.000002
122	90.000002

Table 4: Table representing a cluster of horizontal lines

During line extraction, it appeared that some lines had similar parameters than others which explained why on images shown previously it was noticeable than in some direction, lines were so close to each other than sometimes they were very difficult to distinguish. To avoid using several times lines with similar parameters, a voting scheme has been applied to each distinct cluster. The voting scheme follows several steps:

1. Flag attribution based on line parameters similarity.
2. Mean computation and vote attribution.
3. Intersection computation for each new line cluster in the considered direction.
4. Flag attribution based on coordinates similarity.
5. Mean computation and vote attribution.

In this section, only the first two steps are described but the other steps will be described in the next section.

The first step consisted in:

1. Numbering lines from the considered cluster as can be seen on table 5 below.

Flag	ρ	θ **in degree**
1	958	90.000002
2	466	90.000002
3	462	90.000002
4	458	90.000002
5	470	90.000002
6	122	90.000002

Table 5: Table representing a cluster of horizontal lines after numbering

2. Attributing a flag to lines with similar parameters as can be seen on the table 6 below.

Flag	ρ	θ **in degree**
1	958	90.000002

2	466	90.000002
2	462	90.000002
2	458	90.000002
2	470	90.000002
6	122	90.000002

Table 6: Table representing a cluster of horizontal lines after flag attribution

The attribution of a flag was made by comparing the first line to all the other lines, then the second line with all remaining and so on. The parameter similarity was either based on the angle valueθ, either on the ρ value or both within the following tolerances: 19 for ρ and 15° for θ for the horizontal group, 20 for ρ and 15° for θ for the vertical group and 20 for ρ and 7° for θ for the changing group. The voting scheme was first tested with similar parameters for each direction and then the best parameters for each direction was selected empirically as it led to better results for most of the scenes.

Once a flag has been attributed to all lines, lines with the same flag were temporarily clustered into a new set to compute the mean value and merge lines with similar parameter into a single line as can be seen on table 7 below.

Flag	ρ	θ in degree
1	958	90.000002

(a)

Flag	ρ	θ in degree
2	466	90.000002
2	462	90.000002
2	458	90.000002
2	470	90.000002

(b)

Flag	ρ	θ in degree
6	122	90.000002

(c)

Table 7: (a) Temporary cluster 1, (b) Temporary cluster 2, (c) Temporary cluster 6.

Then, the mean was computed for the temporary clusters containing more than one line using the formula below:

$$\mathcal{M}\left(\rho_{\mathcal{C}_f}\right) = \frac{\sum_1^N \rho_i}{N} \tag{70}$$

$$\mathcal{M}\left(\theta_{\mathcal{C}_f}\right) = \frac{\sum_1^N \theta_i}{N} \tag{71}$$

$$\mathcal{M}\left(\mathcal{L}_{\mathcal{C}_f}\right) = \left\{\mathcal{M}\left(\rho_{\mathcal{C}_f}\right), \mathcal{M}\left(\theta_{\mathcal{C}_f}\right)\right\} \tag{72}$$

Where N is the total number of lines in a temporary cluster, ρ_i the distance between the line and the origin of the ith line in the temporary cluster $\mathcal{C}_f$, θ_i the angle of the vector from the origin to the closest point of the ith line in the temporary cluster $\mathcal{C}_f$, $\mathcal{M}\left(\rho_{\mathcal{C}_f}\right)$ is the mean value of ρ parameters in the temporary considered cluster $\mathcal{C}_f$, $\mathcal{M}\left(\mathcal{C}_f\right)$ is the mean value of θ parameters in the temporary considered cluster $\mathcal{C}_f$ and $\mathcal{M}\left(\mathcal{L}_{\mathcal{C}_f}\right)$ is the mean set for the temporary line cluster associated to the considered line cluster $\mathcal{C}_f$ where f is the flag that allows distinguishing one temporary cluster from another.

Moreover, after mean calculation the flag is replaced by a vote that is related to the number of lines included in the temporary cluster. So for the example clusters considered above, the resulting mean calculation and vote attribution is shown in table 8 below:

Vote	ρ	θ in degree
1	958	90.000002

(a)

Vote	ρ	θ in degree
4	464	90.000002

C

Vote	ρ	θ in degree
6	122	90.000002

Table 8: (a) Temporary cluster 1 after mean calculation, (b) Temporary cluster 2 after mean calculation, (c) Temporary cluster 6 after mean calculation.

Once lines have been differentiated within a considered cluster, they are grouped into a final cluster in the considered direction, here it is$\{\mathcal{L}_f^{\mathcal{H}}\}$, as can be seen on the table 9 below.

Vote	ρ	θ in degree
1	958	90.000002
4	464	90.000002
6	122	90.000002

Table 9: Table representing the final horizontal cluster

To summarize, this section described how lines can be differentiated within a cluster based on the similarity of their parameters and the number of similar lines in the same cluster.

The next section describes how lines differentiation allowed extracting VPs coordinates.

4-1-2-6-From lines to intersections

This section describes in detail how lines differentiation led to potential VP candidate intersection points. The examples given in this section are related to the same horizontal cluster described in the previous section.

Using the final horizontal cluster, intersection of each pair of lines of this cluster is computed using a Singular Value Decomposition (SVD) method based on the following formula:

$$\boldsymbol{AX = b} \;\; \Rightarrow \begin{bmatrix} \boldsymbol{a_1} & \boldsymbol{b_1} \\ \boldsymbol{a_2} & \boldsymbol{b_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\rho_1} \\ \boldsymbol{\rho_2} \end{bmatrix} \qquad (73)$$

Where $a_1, a_2, b_1, b_2, \rho_1$ and $\rho_2$are the parameters of the considered lines and x and y are the coordinates of the resulting intersection point.

For this calculation, the polar coordinate system has been used for line equations as follow:

$$\boldsymbol{a_i x + b_i y = \rho_i} \qquad (74)$$

Where i is the index of the line of the considered cluster.

As described in section 4-1-1-2-Standard Hough transform, a and $\boldsymbol{b}$ are considered as below:

$$\boldsymbol{a} = \cos\boldsymbol{\theta} \tag{75}$$

$$\boldsymbol{b} = \sin\boldsymbol{\theta} \tag{76}$$

So assuming $X = \begin{bmatrix} x \\ y \end{bmatrix}$, equation 18 becomes:

$$\boldsymbol{AX = b} \;\; \Rightarrow \boldsymbol{A^{-1}AX} = \; \boldsymbol{A^{-1}b} \; \Rightarrow \boldsymbol{X = A^{-1}b} \tag{77}$$

Thus:

$$\boldsymbol{A^{-1}} = \frac{1}{det(A)}\begin{bmatrix} \boldsymbol{b_2} & \boldsymbol{-b_1} \\ \boldsymbol{-a_2} & \boldsymbol{a_1} \end{bmatrix} = \frac{1}{\boldsymbol{a_1b_2 - b_1a_2}}\begin{bmatrix} \boldsymbol{b_2} & \boldsymbol{-b_1} \\ \boldsymbol{-a_2} & \boldsymbol{a_1} \end{bmatrix} \tag{78}$$

And:

$$\boldsymbol{A^{-1}} = \frac{1}{\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2}\begin{bmatrix} \sin\theta_2 & -\sin\theta_1 \\ -\cos\theta_2 & \cos\theta_1 \end{bmatrix} \tag{79}$$

So:

$$\boldsymbol{X} = \frac{1}{\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2}\begin{bmatrix} \sin\theta_2 & -\sin\theta_1 \\ -\cos\theta_2 & \cos\theta_1 \end{bmatrix}\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{80}$$

The SVD method used here was implemented by the author in C/C++ with the use of function from the OpenCV library. More details can be found in Appendix 2.

The output of the SVD method is then a set of intersection points for the considered direction, here $\{\mathcal{I}_i^{\mathcal{H}}\}$. Each of these sets were composed of: the number of lines in the final set in the considered direction, the index, the flag and the vote of the first line, the index, the flag and the vote of the second line, used to compute the intersection point, the vote[58] of the intersection point computed and its x and y coordinates. An example of an intersection set for the horizontal direction considered previously $\{\mathcal{I}_i^{\mathcal{H}}\}$ is shown on table 10 below.

$\boldsymbol{\mathcal{N}}$[59]	Line 1 Index	Line 1 Flag	Line 1 Vote	Line 2 Index	Line 2 Flag	Line 2 Vote	Intersection's vote[60]	$\boldsymbol{x}$	$\boldsymbol{y}$
6	0	1	1	1	2	4	$1 * 4$	-0.000025	711
6	0	1	1	2	6	1	$1 * 1$	-0.000019	540
6	1	2	4	2	6	1	$4 * 1$	-0.000010	293

Table 10: Table representing the resulting intersection point after computation from the final set of lines shown previously.

[59] Total number of lines in the considered direction.

[60] This vote will be used later to detect vanishing point candidate in the considered direction as well as for display purposes. It is calculated from the vote of lines considered to compute the considered intersection point using the following formula: $vote\ of\ line\ 1 * vote\ line\ 2$.

To summarize, this section described how line differentiation allowed determining potential VP candidate intersection points.

The next section shows how simple intersection points have been distinguished from potential VP candidate.

4-1-2-7-VP detection

This section explains how the intersection points, determined with the method described in the previous section, allowed differentiating simple intersection points from potential VP candidate.

All examples given in this section are based on the same horizontal cluster as the one used in the previous sections.

In the final set considered, intersection points computed and their vote are the following:

Intersection's vote	x	y
4	-0.000025	711
1	-0.000019	540
4	-0.000010	293

Table 11: Table representing the intersection point after computation from the final set of lines and their vote.

The vote attributed to each intersection point is the criteria that allowed differentiating simple intersection point from potential VP candidate. However, in this case, two of the three intersection points appear to have the same vote. The way to deal with such a case was to apply the same method as the one applied for lines to differentiate intersection points from simple intersection to potential VP candidate. The process followed the same stages as those used to distinguish lines: intersection numbering, attribution of a flag based on intersection points coordinates' similarities, creation of temporary clusters for intersection points with similar coordinates, mean calculation and vote attribution. This case does not always apply but here intersection points are then numbered and compared to each other to attribute the same flag to intersection points with similar x and y.

Those with similar flag are then separated into two temporary clusters according to their vote as can be seen on table 12 below.

Flag	Intersection's vote	x	y
1	4	-0.000025	711
2	1	-0.000019	540
3	4	-0.000010	293

(a)

Flag	Intersection's vote	x	y
1	4	-0.000025	711
1	4	-0.000010	293

(b)

Flag	Intersection's vote	x	y
2	1	-0.000019	540

(c)

Table 12: (a) Intersection points numbered and their vote, (b) Intersection points and their vote in the temporary cluster 4, (c) Intersection points and their vote in the temporary cluster 1.

A flag was first attributed to each intersection according to how similar their coordinates were. As previously for the clustering of lines, the first intersection point was compared with all the others and then the second one with all the remaining, etc. For this comparison, the following tolerances were used: 0.000004 for x and 50 for y for the horizontal group, 200 for x and 1500 for y for the vertical group, 100 for x and 100 for y for the changing group. These values have been chosen empirically from experiments performed on all scenes.

Then, the mean calculation using the same formula presented in section 4-1-2-5-Line distinction process was performed to finally obtain a final intersection points' cluster $\{\mathcal{I}_f^{\mathcal{H}}\}$ as on the table 13 below.

Intersection's vote	x	y
4	-0.0000175	502
1	-0.000019	540

Table 13: Table representing the intersection point after computation from the final set of lines and their vote.

Finally, VPs coordinates are extracted from the final intersection points clusters for each direction: $\{\mathcal{J}_f^{\mathcal{H}}\}$, $\{\mathcal{J}_f^{\mathcal{V}}\}$ and $\{\mathcal{J}_f^{\mathcal{T}}\}$, as the intersection points that has the highest vote. An example of this final stage is illustrated by the table 14 below as well as corresponding lines detected used to computation the final intersection points.

ρ	θ in degree
1032.75	340.500001
625.7	39.33
958	90.000002
464	90.000002
122	90.000002
1278	0
458.7	356.99
391.25	356.750001
62	354.45
97.625	355.25
333.5	356.500002
28.833	354.7
539.7	357.99
201.33	355.7

(a)

Vote	x	y
24	1009.086	-244.281
4	-0.0000175	502
1	-0.000019	540
286654464	1259.87	14941.50
48	899.54	8695.18
573308928	1008.19	10738.07
9	701.15	4614.85
2507653251072	822.33	8186.44
16	322.61	2703.53
12	-1031.077	-11354.99

(b)

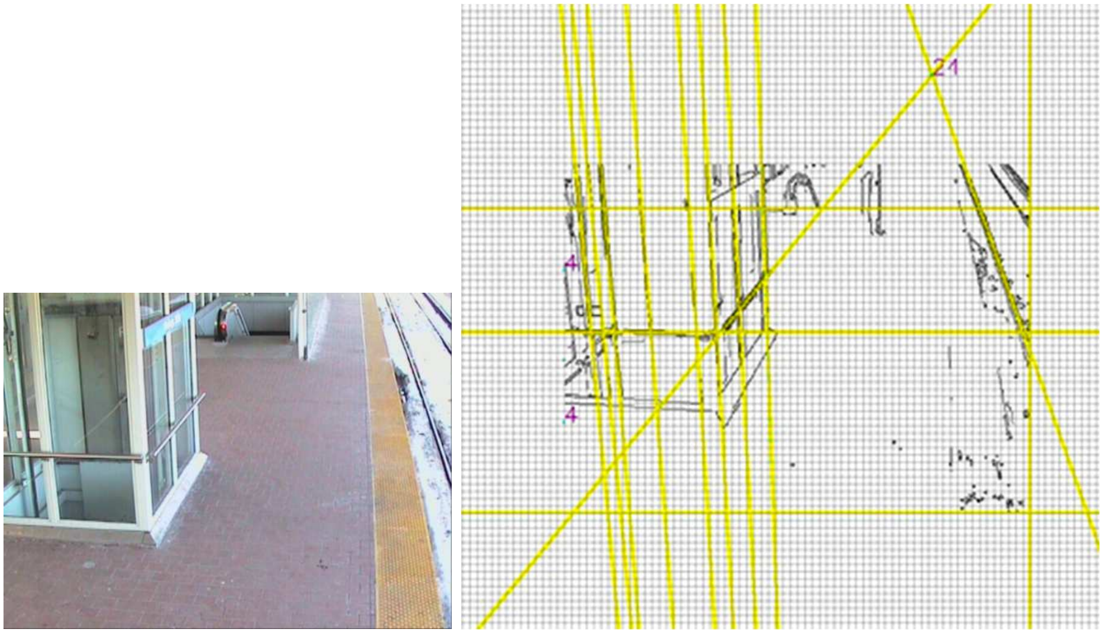
Table 14: (a) Table representing the final set of lines used for display; (b) Table representing the final set of intersection points used for display.

The parameters in both tables that are blue correspond to the lines detected and intersection points computed that are in the direction of the tracks, the parameters that are in green correspond to the lines detected and intersection points computed that are in the horizontal direction and the remaining in pink are those that are in the vertical direction. For each direction, VPs can then be identified as shown in table 15 below.

Vote	x	y
24	1009.086	-244.281
4	-0.0000175	502
2507653251072	822.33	8186.44

Table 15: Final VPs coordinates detected.

It is then possible to display the results for visualisation purposes as can be seen on figure 43 below.



(a) (b)

Figure 43: Example of a real-scene image and the result of our algorithm: (a) Original frame, (b) Image results presenting the line and average and the intersection point found[61].

All lines and VPs detected as well as their vote are displayed on top of the Canny edge detection image, which is superimposed on a high-resolution (9280*1330) black background for visualisation purposed. Unfortunately, VPs are sometimes outside of the image boundaries which explain why in this case only the VP detected in the direction of the tracks is visible. This VP corresponds to the intersection point with a vote equal to 24.

To summarize, this section described how VPs can be extracted from railway and underground station scenes using the proposed approach. The next section presents results obtained with this approach as well as a quantitative and qualitative analysis.

4-1-3-Results

In this section, various results obtained with the proposed approach are presented as well as some preliminary tests to illustrate the advantages of some of the intricacies of the method. First, the results of the line and VP detection approach applied to simple to more complex line drawings is presented. Then, results obtained with the proposed method applied to real scenes are presented. Finally, results obtained with real-scenes and those obtained with a state-of-the-art method based on the J-linkage technique are compared to ground truth. Finally, a detailed analysis of the results is provided.

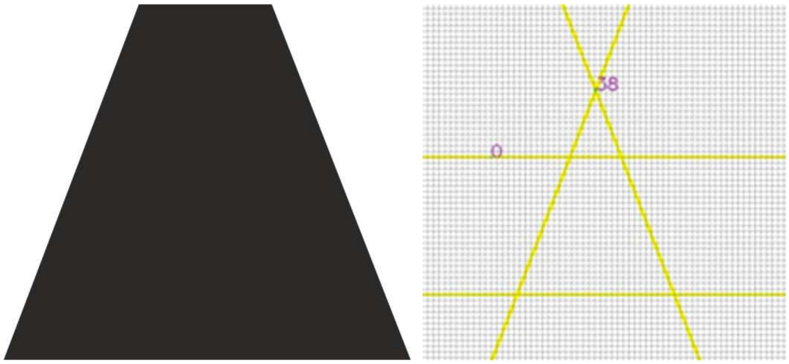
[61] Used with permission of the author/publisher.

All results presented in this section show the original image, the detected edges (for some results) and a final image that corresponds to the result obtained from the original one. The final image shows the lines detected with the proposed approach and for visual purposes, lines with same parameters have been drawn on top of those detected to extend them and show towards which VP direction they tend to. Indeed, detected lines are usually shorter and as they lead to VPs sometimes outside the image boundaries, an extension allows visualising where they lead to.

4-1-3-1-First experiments on simple and more complex drawings

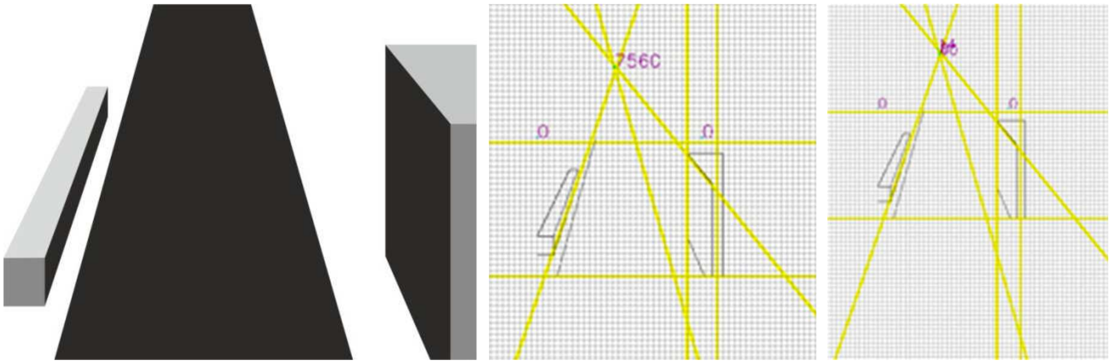
This section first illustrates results obtained on simulated images with various degrees of complexity. For each degree, the efficiency and failed cases of the proposed approach are discussed.

- Simple drawings



(a) **(b)**

Figure 44: Most simple drawing used to test our algorithm: (a) Simulated original drawing, (b) Results obtained from our algorithm



(a) **(b)** **(c)**

Figure 45: Simple drawing used to test our algorithm: (a) Simulated original drawing, (b) Results obtained from our algorithm without intersection averaged, (c) Results obtained from our algorithm with intersection averaged

Figures 44 and 45 show results obtained from simulated images with a really simple and a more complex drawing. It clearly appears that line detection is working 100% on the simplest case as can be seen on figure 44 whereas it is not as accurate as on a more complex drawing as on figure 45. This decrease in accuracy can be explained by the length of lines in the image. Indeed, the detection of the shortest lines tends to be incorrect or missing during edge or line detection processes. However, if lines in some directions are undetected, those that are, are well detected and lead to intersection points and then VPs successfully.

Now, results obtained with simulated image with simple and more complex drawings with a change of perspective are presented.

- More complex drawings

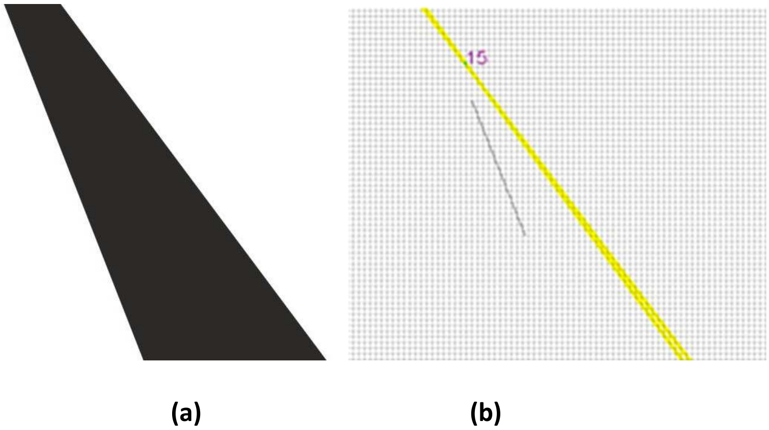


Figure 46: More complex drawing used to test our algorithm: (a) Simulated original drawing, (b) Results obtained from our algorithm

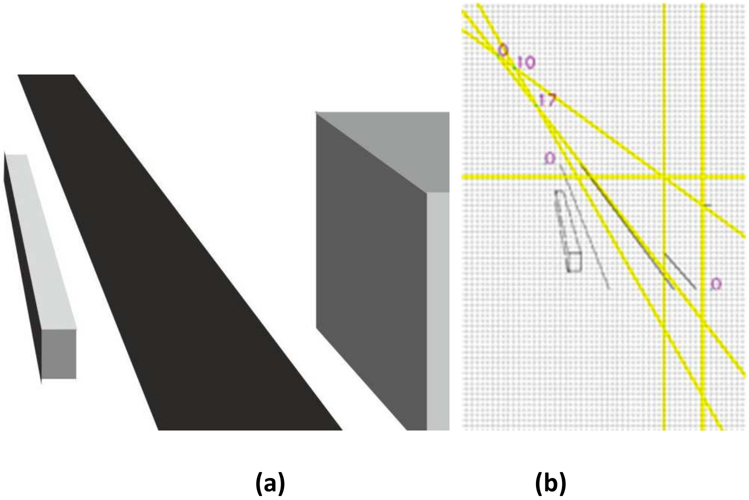
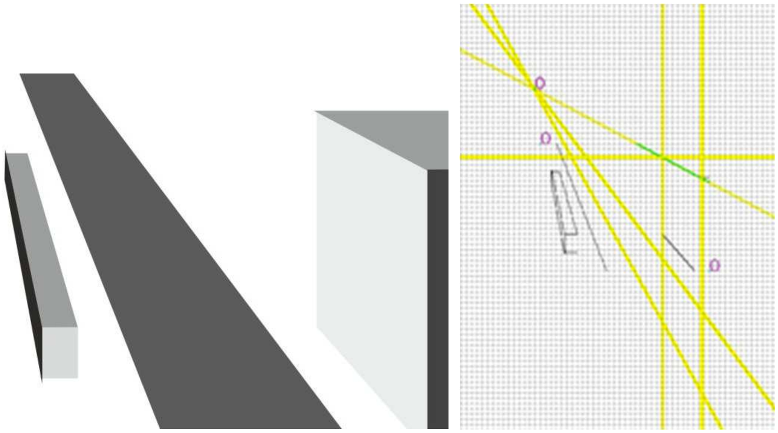


Figure 47: Most complex drawing used to test our algorithm: (a) Simulated original drawing with wrong perspective, (b) Results obtained from our algorithm



(a) **(b)**

Figure 48: Most complex drawing used to test our algorithm: (a) Simulated original drawing with corrected perspective, (b) Results obtained from our algorithm

To evaluate the efficiency of the proposed approach, experiments have been realised to verify its performance on drawings with a wrong perspective (i.e. where not all lines perfectly intersect in a single point, as it is the case for parallel from real images. Similar simulated images as those used at the beginning of the section are used with a change of perspective and exactly the same with a wrong perspective. It is then interesting to compare the results obtained between those shown on figure 47 and those on figure 48 which represent the correct perspective. In both cases, the result is quite accurate and successful for line detection and intersection computation. To conclude, the proposed approach is not too sensitive to perspective distortion.

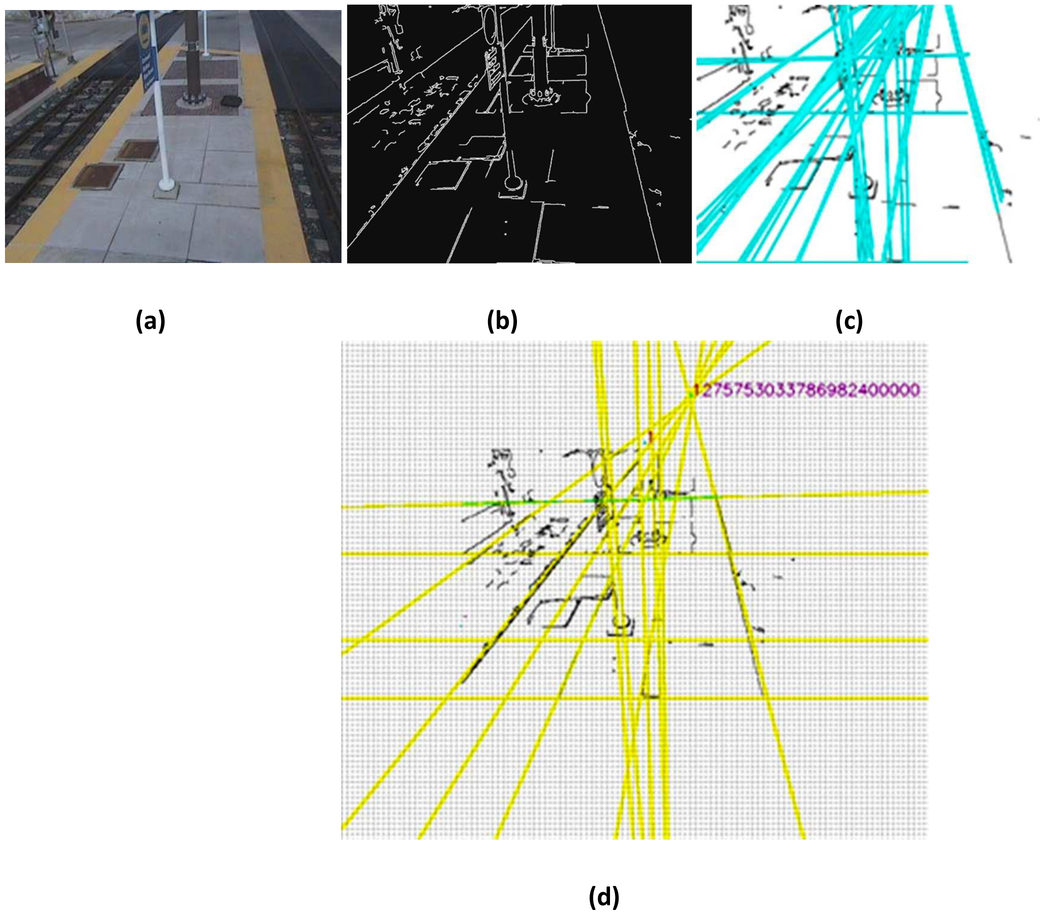
Now, results have been presented about simulated images, the next section focuses on results obtained with real railway and underground station scenes.

4-1-3-2-Last experiments obtained with our current algorithm on simple and more complex real scene: Railway and underground station, our study case

This section presents results obtained with the proposed approach applied to real railway and underground stations scenes with various degrees of complexity, including a variety of indoor and outdoor scenes. For each degree, the efficiency of the method as well as failed cases is discussed.

- Outdoor real scenes

Simple outdoor real scenes



(a)

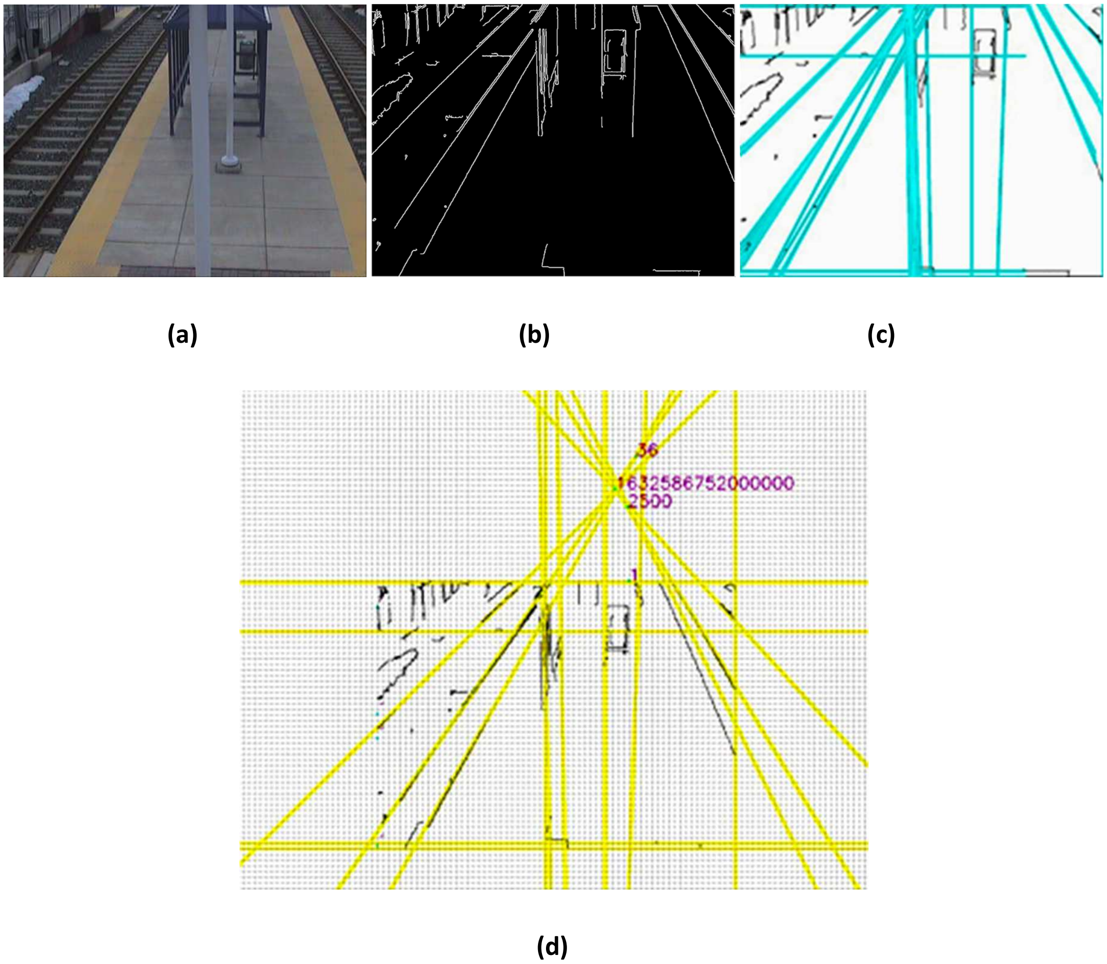
(b)

(c)

(d)

Figure 49: Example of one simple outdoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 3 VPs have been detected[62].

[62] Used with permission of the author/publisher.


(a)

(b)

(c)

(d)

Figure 50: Example of another simple outdoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 2 VPs have been detected[63].

[63] Used with permission of the author/publisher.

Outdoor real scenes with change in weather conditions

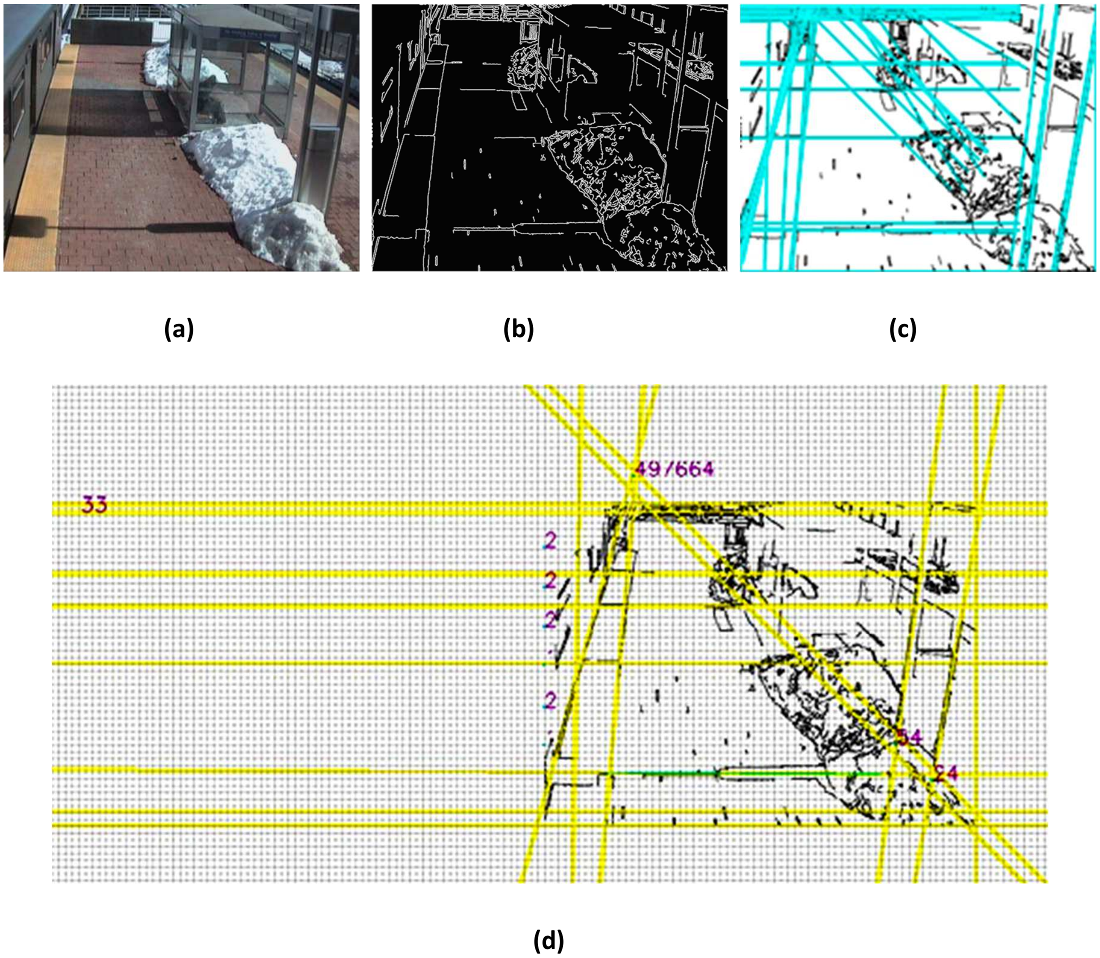
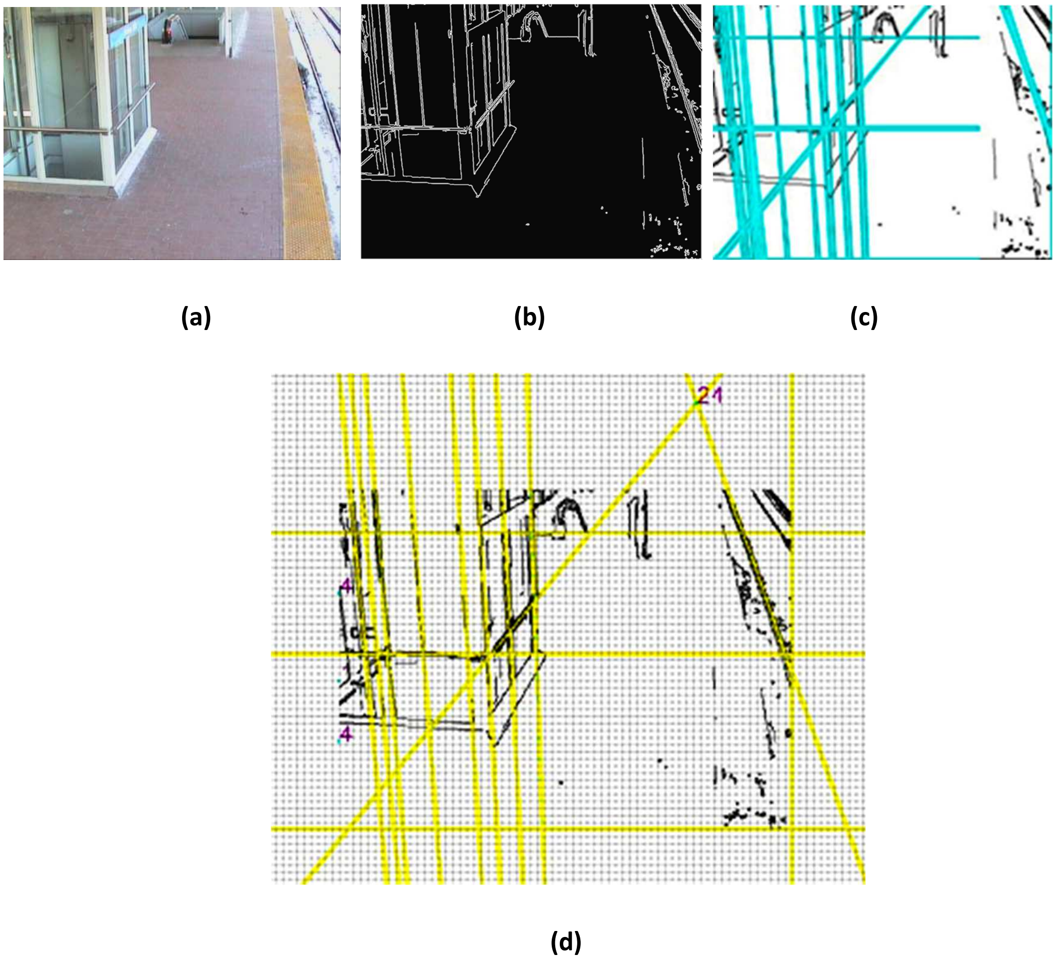


Figure 51: Example of an outdoor scene with some shadows and change of lighting and weather conditions: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 2 VPs have been detected[64].

[64] Used with permission of the author/publisher.

Complex outdoor real scenes

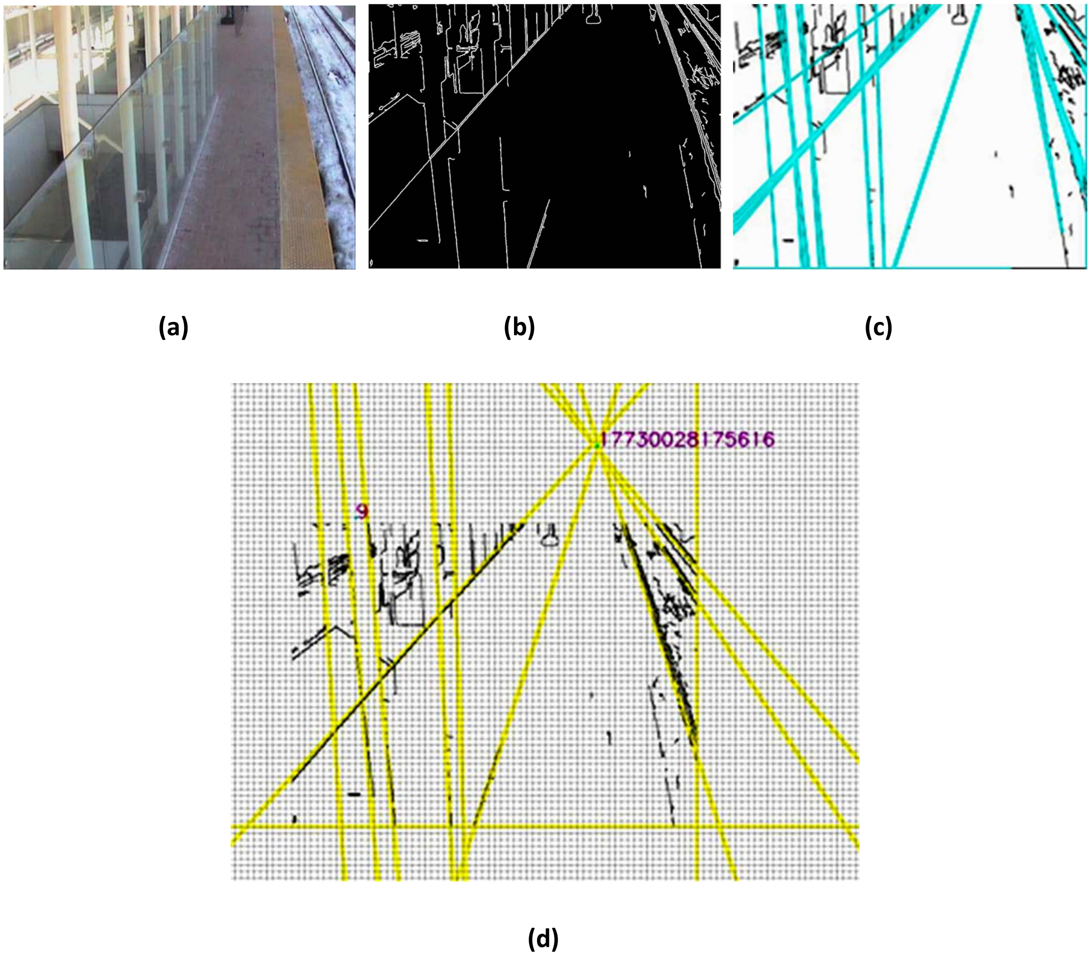


(a) (b) (c)

(d)

Figure 52: Example of a complex outdoor scene with presence of other perspective due to the shelter and escalators: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 1 VP has been detected[65].

[65] Used with permission of the author/publisher.



(a) (b) (c)

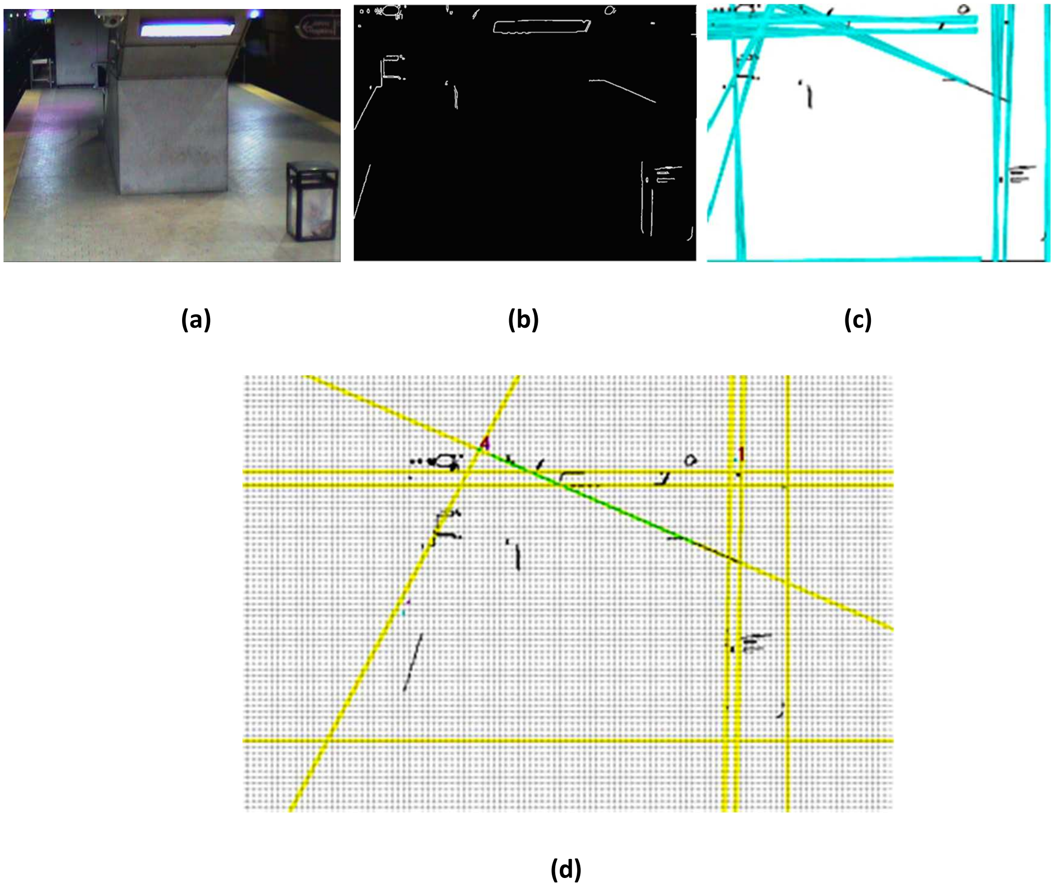
(d)

Figure 53: Example of a complex outdoor scene with presence of other perspective due to the shelter and escalators: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 1 VP has been detected[66].

[66] Used with permission of the author/publisher.

- Indoor real scenes

Simple indoor real scenes

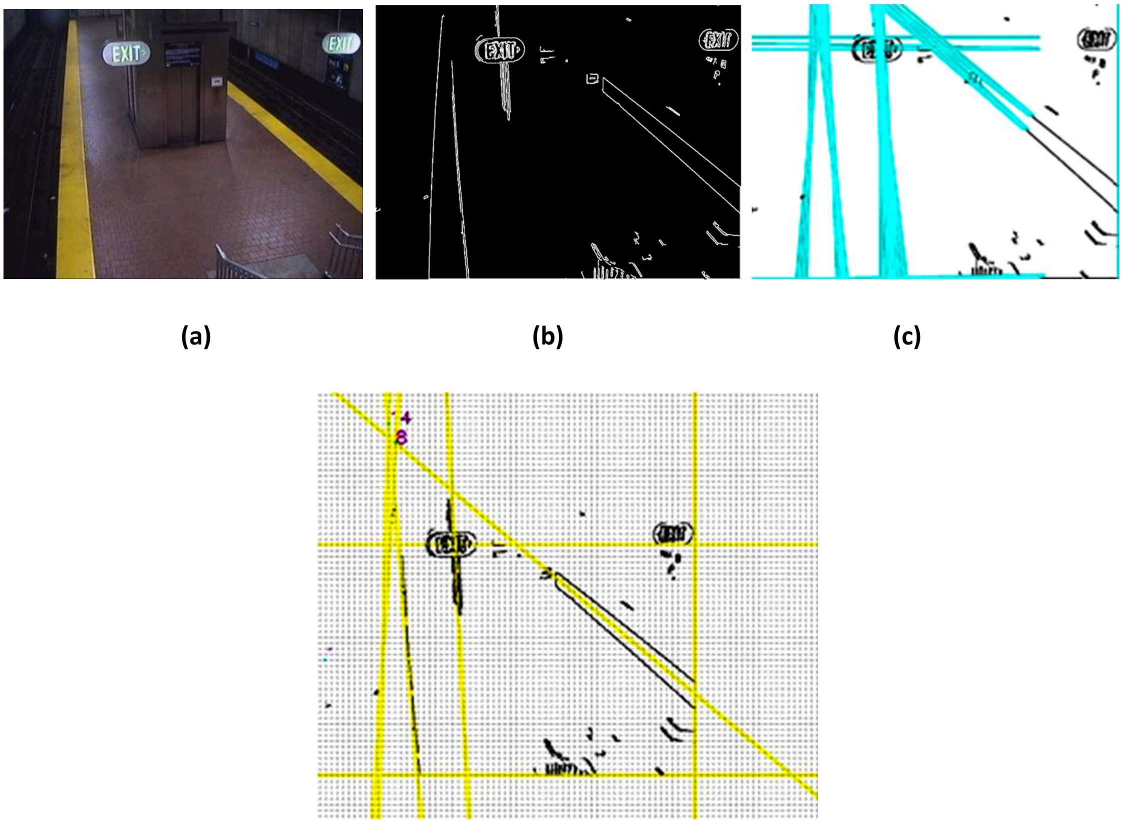


(a) (b) (c)

(d)

Figure 54: Example of a simple indoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 2 VPs have been detected[67].

[67] Used with permission of the author/publisher.



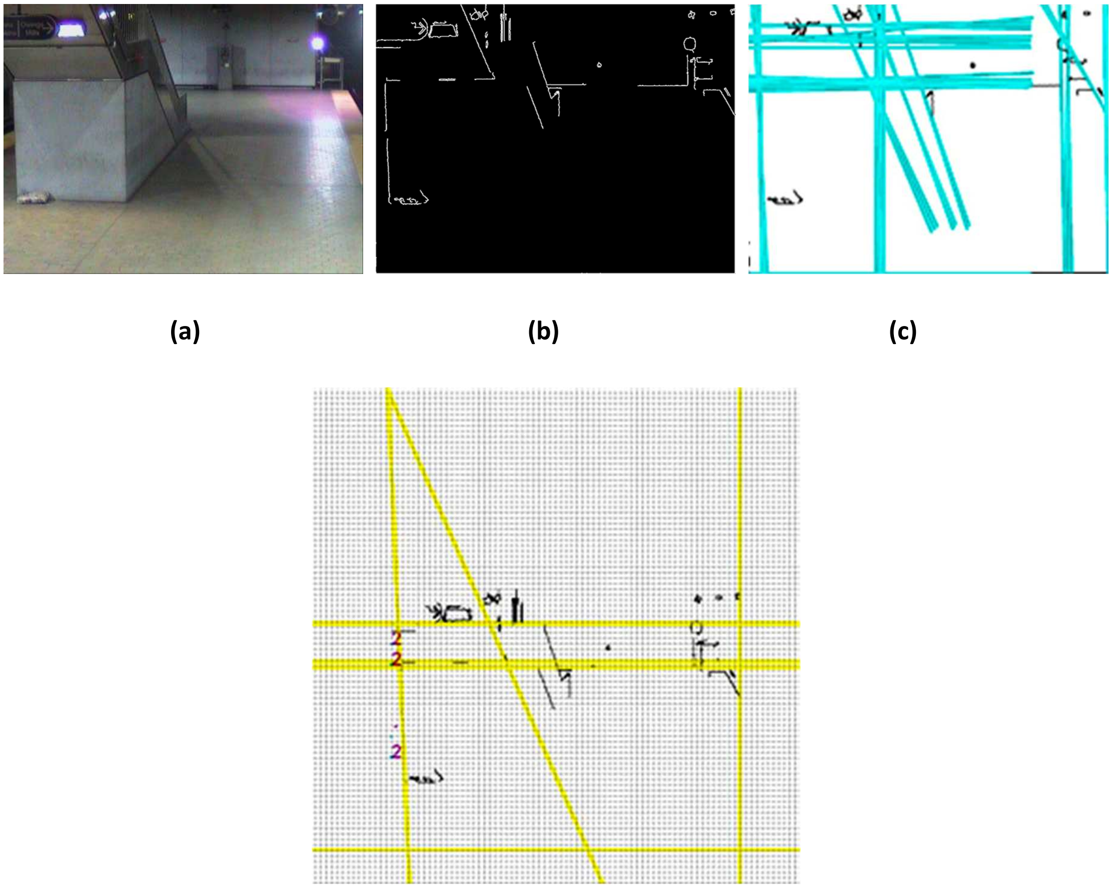
(a) **(b)** **(c)**

(d)

Figure 55: Example of another simple indoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 1 VP has been detected[68].

[68] Used with permission of the author/publisher.

Complex indoor real scenes



(a)

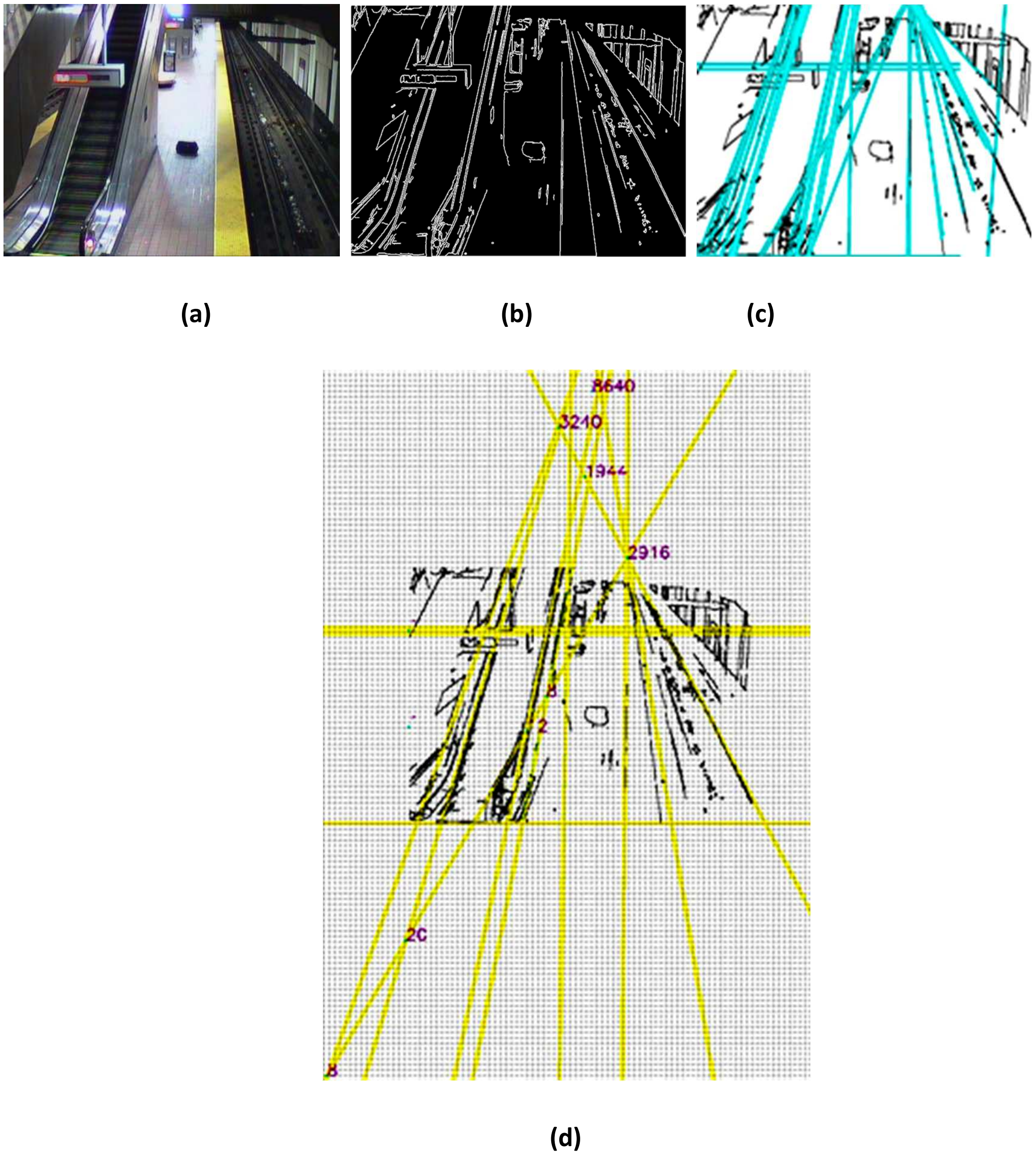
(b)

(c)

(d)

Figure 56: Example of a more complex indoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where no VPs have been detected[69].

[69] Used with permission of the author/publisher.



(a) (b) (c)

(d)

Figure 57: Example of a more complex indoor scene: (a) Original enhance frame, (b) Results obtained from Canny, (c) Results obtained from the HT, (d) Final result of our algorithm where 2 VPs have been detected[70].

As can be seen on Figures 49 to 57, this study case focused on various real railway and underground stations scenes, outdoor as well as indoor, with various degree of complexity due to the scene geometry and objects present as well as changes in exterior conditions such as lighting, weather, shadows, occlusions, perspective etc.

The proposed approach has been tested on 18 different clips but due to the quantity of data, only few images are shown here to best illustrate the strengths and weaknesses of the method and avoid redundancy.

[70] Used with permission of the author/publisher.

The results are classified into three categories:

- The type of scene: outdoor or indoor;
- The degree of complexity depending on scene geometry and exterior conditions;
- The number of VPs detected: from 3 to 1 VP(s) detected as well as a case where it failed.

For the less complex outdoor scenes, two to three VPs have been detected as can be seen on figures 49 and 50 whereas for the less complex indoor scenes, only 0 to 2 VP(s) have been detected as can be seen on figures 54 and 55.

As for outdoor scenes with different lighting conditions as can be seen on figure 52, the number of VPs detected remains around 2 and only decreases proportionally according to the scene complexity, i.e. presence of stairs, escalators or shelters as can be seen on figures 47 and 48. However, scenes in presence of non-flat elements, such as stairs, escalators or shelters, can lead to the detection of additional VPs as can be seen on figure 57. In this example, a VP is detected to be in the direction of the escalators which is not part of a Manhattan world. Furthermore, it represents a useful and significant additional detection as it provides information about the scene geometry and suggests it is not only flat. Intelligent video surveillance systems often struggle to detect non-flat surfaces as the information they use are not as rich as 3D data, i.e. as they do not provide as many information or feature about the scene, to make the system understand the scene geometry accurately. Then, they tend to deal with non-flat surfaces as if they were flat which can be problematic for scene understanding. However, the presence of additional perspective makes VP detection more problematic as it not only requires differentiating simple intersection points from VPs but also VPs from the three orthogonal direction of the Manhattan world to VPs from other directions. This has not been explored here but will be in the future.

To summarize, a major distinction can be noticed between results obtained from outdoor and indoor scenes. Results from outdoor scenes are better than those obtained on indoor scenes due to change in lighting mostly. Indoor scenes are most of the time darker. The darkness confounds different planes which imply incorrect VP detection as can be seen on figures 54 and 56. Moreover, outdoor scenes where the lighting is more important produce a higher objects' reflectance but it does not reduce the efficiency of the proposed approach; this is especially noticeable as only 2 on 18 cases failed for indoor scenes where no VP were detected and no case failed for outdoor scenes. To conclude, VP detection has proven to be difficult in presence of a scene with a higher degree of complexity as expected according to the literature which mostly focuses on simpler scenes.

While this section analysed the proposed approach qualitatively, the next one provides a quantitative analysis.

4-1-3-3- Quantitative analysis and discussion

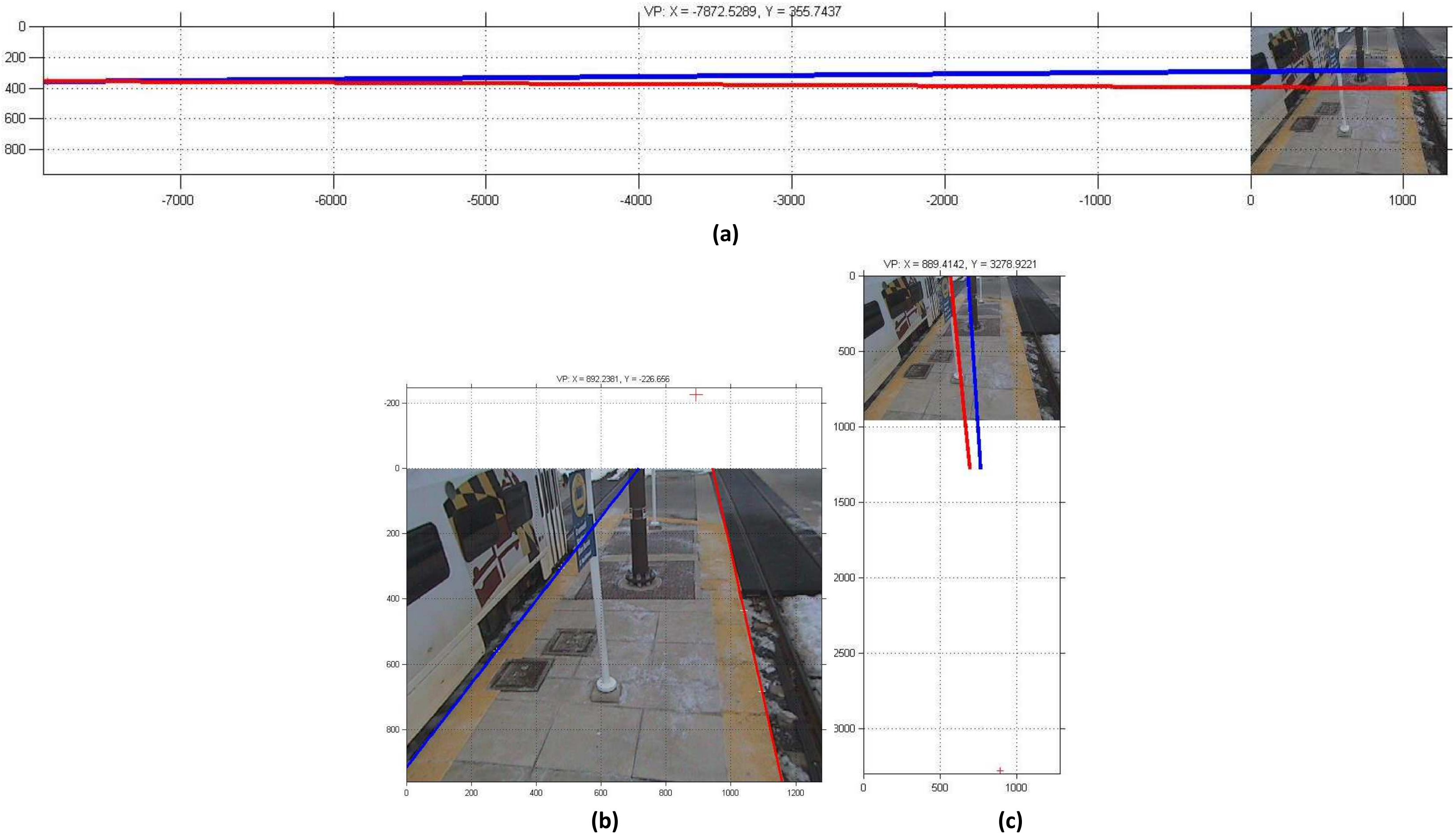
In this section the proposed VP detection method is critically analysed.

In addition to the qualitative analysis provided in the previous sections, a quantitative analysis is made in this one by comparing the proposed approach and a state of the art method proposed by Tardif [48] to Ground truth (GT). This qualitative analysis presents experimental results performed to demonstrate the effectiveness and robustness of the proposed approach compared to a current state-of-the-art method. It has been made possible by computed VPs with the proposed and novel approach (PNA) and Tardif's method (TAR) for 18 different railway and underground station video sequences, including 10 outdoor and 8 indoor scenes to detect VPs.

To the best of our knowledge, there are no existing and similar approaches to the proposed method intending to compute VPs specifically from railway and underground station environments and, as mentioned above, most of the related work focused on architectural or simpler environments. Indeed, the proposed approach only combines existing feature extraction such as The Canny filter and the HT but is based on a novel clustering and voting scheme depending entirely on information acquired from the observation of the scene. The use of observations on the line pattern for the clustering stage is not a unique approach as authors proposed various line patterns to detect lines. However, they all differed from one another including from the one proposed here and did not apply it to various or more realistic environments.

Thus, Tardif's method [48] has been chosen to be compared with the proposed approach as it is recent and provides very accurate results for certain man-made scenes. His method is based on a non-iterative method using J-linkage [49] to cluster lines of the image according to the VP they are directed towards. Thus, the comparison has been made possible using ground truth as reference.

The testing protocol adopted was the following. To avoid any bias, GT VPs were determined by people who did not develop either approach. They were trained to use very basic bespoke software that allowed them to manually select key lines in the images that are known to be directed to a VP (two lines per VP). The software then calculates a VP as the intersection of the manually identified lines. Figure 58 below illustrates results obtained with the manual GT detection software.



(a)

(b)

(c)

Figure 58: Example of ground truth for an outdoor scene: (a) VP from the horizontal direction, (b) VP from the tracks direction VP, (c) from the vertical direction[71].

[71] Used with permission of the author/publisher.

GT data have then been used to classify VP detection as successful, unsuccessful or undetected. The simplest method to do this would be to use a Euclidean distance threshold to determine whether or not the VP estimate is sufficiently close to GT. However, VPs within (or close to) the image are expected to have a smaller Euclidean distance error to be classified as successful compared to those at a greater distance. Therefore, the method adopted was the calculation of a Euclidean distance d_{ref} between the estimated VP $(\mathrm{x_{PNA}}, \mathrm{y_{PNA}})$ or $(\mathrm{x_{TAR}}, \mathrm{y_{TAR}})$ and the image centre $(\mathrm{x_{ref}}, \mathrm{y_{ref}})$, and d_{GT} between the estimated VP $(\mathrm{x_{PNA}}, \mathrm{y_{PNA}})$ or $(\mathrm{x_{TAR}}, \mathrm{y_{TAR}})$ and GT $(\mathrm{x_{GT}}, \mathrm{y_{GT}})$. A ratio has been used as a measure of success. When the ratio is less than a given threshold, *t*, the detection is deemed successful. For the bulk of this work, the value used for this threshold was *t* = 0.02. Moreover, experiments have also been performed with two other threshold values: *t* = 0.05 and *t* = 0.10 to verify if results obtained where those expected when increasing the threshold value.

As seen previously, the proposed approach provides an output containing potential VP candidate for the three orthogonal directions of a Manhattan world. So, both Euclidean distances have been calculated for VPs detected with the proposed approach and Tardif's method. The Euclidean distance calculation was performed using the following formula:

$$\boldsymbol{d_{GT}} = \sqrt{((\mathbf{x_{PNA}} - \mathbf{x_{GT}})^2 + (\mathbf{y_{PNA}} - \mathbf{y_{GT}})^2)} \quad (81)$$

Where $\mathrm{x_{PNA}}$ and $\mathrm{y_{PNA}}$ are the coordinates of the VP detected with the proposed and novel approach, $\mathrm{x_{GT}}$ and $\mathrm{y_{GT}}$ are the ground truth VP coordinates and d_{GT} the Euclidean distance between the estimated VP and GT. This is also calculated with estimated VP $(\mathrm{x_{TAR}}, \mathrm{y_{TAR}})$ with Tardif's method.

$$\boldsymbol{d_{ref}} = \sqrt{((\mathbf{x_{PNA}} - \mathbf{x_{ref}})^2 + (\mathbf{y_{PNA}} - \mathbf{y_{ref}})^2)} \quad (82)$$

Where $\mathrm{x_{PNA}}$ and $\mathrm{y_{PNA}}$ are the coordinates of the VP detected with the proposed and novel approach, $\mathrm{x_{ref}}$ and $\mathrm{y_{ref}}$ are the coordinates of the centre of the image and d_{ref} the Euclidean distance between the estimated VP and GT. This is also calculated with estimated VP $(\mathrm{x_{TAR}}, \mathrm{y_{TAR}})$ with Tardif's method.

The figure 59 below illustrates a case of estimated VP and Euclidean distance.

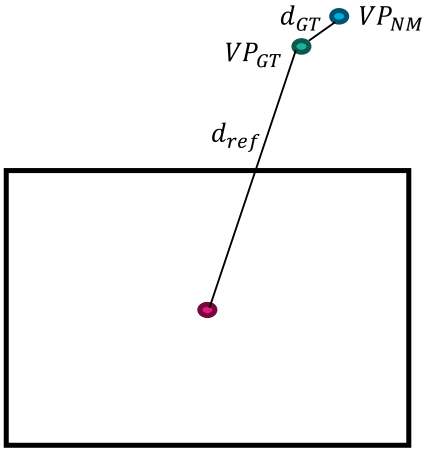


Figure 59: Drawing representing a case of an estimated VP and both Euclidean distances.

Once the two Euclidean distances have been calculated, the following quotient has been determined.

$$\frac{d_{GT}}{d_{ref}} \tag{83}$$

Then, this quotient is compared to the threshold t=0.02 (2%). If the quotient is strictly greater than$t$, the estimated VP is classified as unsuccessful and if the quotient is lower or equal to 2%, the estimated VP is considered as successful. Moreover, VPs have been considered undetected when either method did not detect a VP in the case of a GT VP.

The performance of the methods compared to GT is presented in table 16 for the VP in the direction of the tracks. Unfortunately, results for VPs at the convergence of near-horizontal or near-vertical lines proved too difficult for both algorithms for the scenes analysed. This was due to estimated VPs being too far from the image (often at infinity) to be included. Both methods did however, locate VPs for very simple geometric images (e.g. of cubes) or of highly perspective images of buildings.

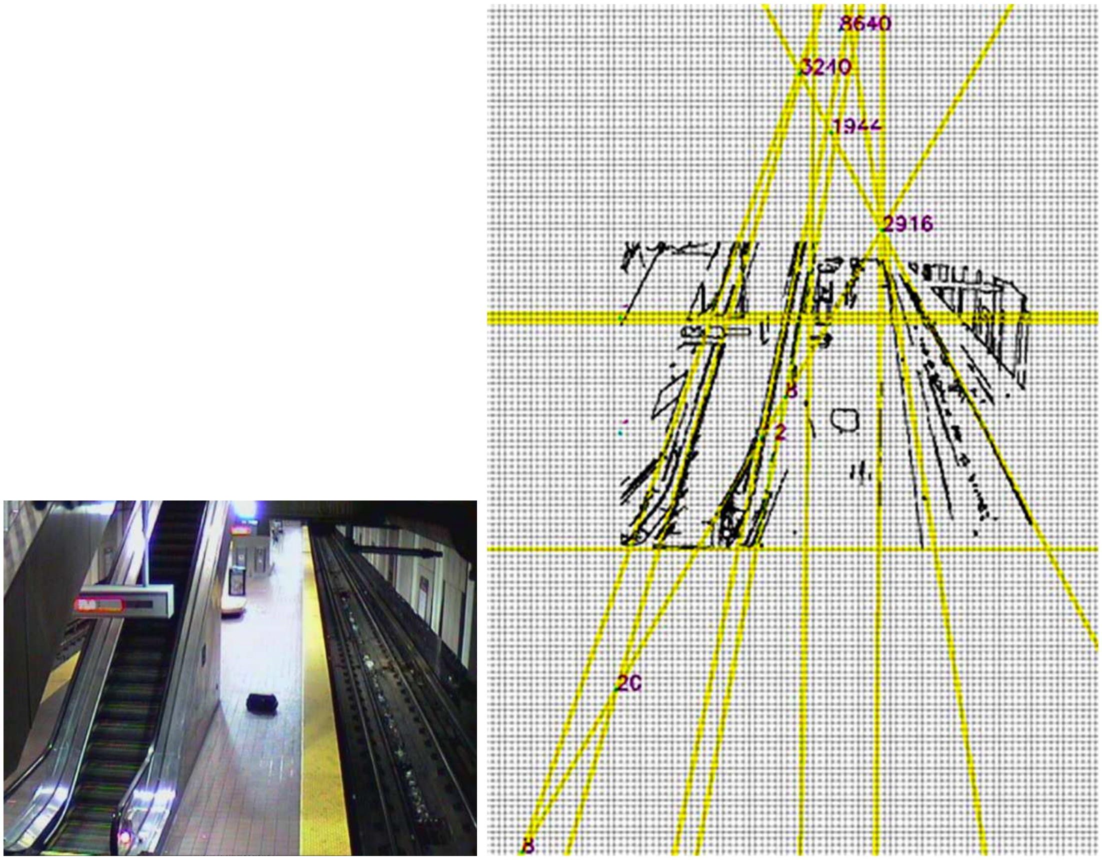
Results are summarized on table 16 below.

Method	VPs successfully detected			VPs unsuccessfully detected			Undetected VPs		
	Indoor scenes	Outdoor scenes	Total	Indoor scenes	Outdoor scenes	Total	Indoor scenes	Outdoor scenes	Total
GT	8	10	18	--	--	--	--	--	--
PNA	5	6	11	1	4	5	2	0	2
TAR [48]	2	2	4	3	8	11	3	0	3

Table 16: Table representing the results obtained from the various approaches for all scenes for VPs located in the direction of the tracks.

As can be seen, 11 out of 18 VPs were detected using PNA, resulting in a 61% hit rate (increased to 78% for t = 0.05). Conversely, TAR provides 4 VPs, a 22% hit rate (increased to 44% for t = 0.05). This can be partly explained by the fact that TAR sometimes uses shorter lines to compute VPs which do not always allow VPs to be detected. Furthermore, TAR does not always detect lines correctly due to the fact that it was originally designed for urban images with almost constant illumination and higher quality images. Overall, the results of the experiments clearly show our method is more robust for this type of scene. This is due to the use of a priori knowledge and thresholds selection specific to railway station scenes.

For certain scenes, additional VPs have been detected due to the prominence of large structures that do not comply with a Manhattan world. Typical causes are stairs or escalators as shown in Figure 60 and table 17 below. While a detailed study of such cases is reserved for future work, only a few scenes have been considered for preliminary tests. The proposed method shows some promise for these cases although only if t is increased to 0.05 (presumably as such lines tend to be more parallel in the image compared to the lines in the track direction). There remains an open question as to how the algorithm should determine how many VPs to seek as it would have to dissociate those from the three orthogonal directions of the Manhattan world to other perspective ones.



(a) (b)

Figure 60: (a) Original image, (b) VPs successfully detected in the direction of escalators[72].

ρ	θ
811.000000	350.666666
554.833333	11.833334
738.111111	332.555557
388.500000	16.700000
596.250000	11.000000
346.750000	19.500000
686.500000	30.000001
958.000000	90.000002
246.000000	90.000002
226.000000	90.000002
811.500000	0.500000
593.000000	1.000000

(a)

Vote	x	y
8640	682.795031	-659.261833
2916	815.327397	-34.204353
360	676.251665	-886.069168
1944	655.754569	-338.816091
60	940.262062	-1782.096458
24	1127.834408	-2677.352355
24	855.265955	-1376.422231
12	438.252097	613.925091
3240	558.770996	-525.561691
40	978.752138	-1910.390563
20	-13.140581	1395.760112
16	898.923290	-1499.706743

(b)

Table 17 : (a) Lines detected in the case of the figure 31 above, (b) VPs detected in the case of the figure 31 above.

[72] Used with permission of the author/publisher.

In this case, VPs in the horizontal and vertical directions have not been detected successfully so only VPs in the direction of the tracks are detected. However, as mentioned previously the additional VP in the direction of the escalator make VP detection from the three orthogonal direction difficult as lines and intersection points from the escalator confuses the program. A solution is then to be found to deal with that and differentiate those from the three orthogonal direction with those from other directions as it shows it is potentially feasible to detect VPs from other directions. Moreover, VPs from non-flat surfaces are of interest for scene understanding as they provide useful information about the 3D geometry of the scene.

To conclude, the performance of Tardif's algorithm for VP detection is almost half less than the proposed approach. This can be explained by the image enhancement which helped to improve line detection as Tardif's method did not succeed to detect a reasonable number of lines as for unsuccessful detected and undetected VPs. On the other hand, the proposed line clustering depends directly on observations from the scene which increases the chance for lines to lead to right VPs and tends to detect longer lines thanks to the threshold chosen for the HT which is not the case for Tardif's approach that also lead his technique to unsuccessful VP detection.

Moreover, the fact that lines are not well detected for indoor scenes can come from the fact the threshold for Canny and then HT must not be well adapted to the scene. A proposed improvement to solve this issue would be to use an auto-adaptive method to select a particular threshold for Canny and the HT for each type of scene as in [151].

Furthermore, the use of a Singular Value Decomposition method can be problematic if $\theta \approx 0$, as it results in infinite values. One solution to overcome this issue in this particular study case would be to only keep θ values when they are non-zero to avoid infinite values to be included as it would confuse the algorithm and lead to false intersection coordinates. However, lines with $\theta = 0$ correspond to vertical lines and can be of interest for the proposed approach especially as lines from this direction tend to be lower than in other direction. So, they should be clustered apart from them rest of the other lines and processed independently to avoid any error during the SVD computation. During the intersection coordinates computation, a matrix inversion formula has been considered but a straight forward algebraic formula such as Gauss-Seidel and Jacobi methods etc, will be used in the future to avoid time consuming computation as it can be an issue for real-time applications.

Another difficulty for this problem is when two of the vanishing points from the three orthogonal directions (typically very far below the bottom of the image for the vertical direction or very far on the left side of the image for the horizontal direction) are very difficult to be reliably detected as lines appeared to be almost parallel. A method needs then to be devised to deal with VPs when they are supposed to be at infinity. So for two lines in the same cluster $l_1(\rho_1, \theta_1)$and $l_2(\rho_2, \theta_2)$ have the same angle $\theta_1 = \theta_2$, the VP will be at infinity. Then VP coordinates can be computed as the furthest point in the accumulator space Δ_{max}, with parameters $\theta = \theta_1 = \theta_2$ and $\rho = \frac{1}{2}(\rho_1 + \rho_2)$ as in [45].

Whilst there is no necessity to display vanishing points, an operational system is likely to offer an option to permit manual verification by a technician during system commissioning, or to select secondary vanishing points, such as those generated by stairs and elevators, that may fall below the detection threshold, but still add value to interpreting the scene context.

To summarize, the experiments have shown the proposed method is more robust and effective to extract VP coordinates from railway station scenes than a state-of-the-art method. In future work, an auto-adaptive method to set thresholds for the edge and line detection could be used. In these experiments, values selected provided clear edges and contours and detected most of the line for most of the data in the available database and so it was possible to use the same value for all data. However, there are certain cases most likely where the chosen parameters would fail. In addition, it may be necessary to adaptively select parameters if the method were to be modified for use in other environments such as airports, shopping centres or sports stadia etc.

Line and VP detection is an essential preliminary task to obtain more information about the scene and then a better understanding of what is happening. However, it requires other tasks to bring a sufficient amount and quality of information to fully understand what is happening within the scene and then be able to associate static and dynamic aspects of the scene.

The next section described another significant task for scene understanding: image segmentation.

4-2-Segmentation

This section considers some of the most popular methods of segmentation that have been proposed in the computer vision community and assesses their potential application to intelligent CCTV technology. Moreover, segmentation methods were not examined in depth as other methods described in this thesis as it was only considered as a pre-step towards 3D and scene understanding and not the main contribution which resides in the use of 3D information to obtain a richer scene understanding.

4-2-1-Techniques considered

This section describes the three following and common segmentation techniques:

1. Thresholding technique [71] [75];
2. Histograms [71][75][76];
3. And The Fuzzy C-Means algorithm [152].

These methods have been considered as they seem to be most applicable to video surveillance applications for the following reasons:

- the choice of parameters needs to be adapted for these methods to perform well;
- they are not computationally expensive;
- they do not require a priori knowledge;
- The last one works well in presence of ambiguities and uncertainties.

The first two methods have been considered to investigate the efficiency of grey intensity for segmentation and because they have been well-researched which can be useful in this study case for some preliminary experiments.

However, video surveillance scenes are commonly very noisy and in presence of many changes in exterior conditions such as lighting, shadows, occlusions etc, which create ambiguous boundaries and uncertain regions within the scene. So, it is essential to choose a method that is not too strict and can deal with uncertainties and ambiguities. This is the reason why the fuzzy C-means algorithm seems to be the most adapted as it works well in presence of ambiguity and uncertainty.

In the next section, a brief description of each of these methods is given.

4-2-1-1-Thresholding

A binary and Otsu [151] thresholding methods have been first investigated. The methods were developed in C/C++ using OpenCV thresholding techniques and followed the algorithm illustrated by the figure 61 below:

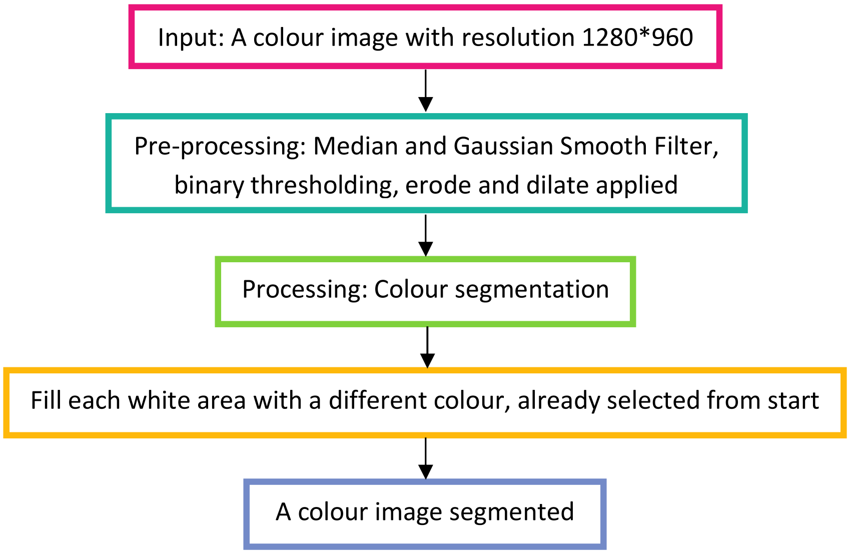

Input: A colour image with resolution 1280*960
Pre-processing: Median and Gaussian Smooth Filter, binary thresholding, erode and dilate applied
Processing: Colour segmentation
Fill each white area with a different colour, already selected from start
A colour image segmented

Figure 61: Algorithm overview based on two thresholding methods

Thresholding is a basic image processing operation whereby pixels in an image are converted to 1 or 0 depending on whether they are above or below a given threshold. The method is primarily used to segment the foreground from the background of an image but also helps to enhance edges, such as those that form lines to VPs discussed above. The Otsu method is an extension of this that aims to adaptively select the threshold such that the intensity variations within the regions of the image either side of the threshold are minimised. Furthermore, the Otsu method permits multi-level thresholding.

A binary thresholding method was first applied to a greyscale image to highlight significant edges and contours of objects within the scene. The threshold value used for this method was $t_b = 75$

which was selected empirically when it highlighted best edges and contours of images. Of course, it would ideally be possible to determine a threshold value automatically but the aim, here is to investigate the best-case scenario. Morphological operations of erosion and dilation [153] have then been applied several times to make contours and edges clearer and reduce or remove noise from the image. These operations have been performed by applying a 3*3 rectangular structuring element to the binary image. Not surprisingly the robustness of this approach improved when a median or Gaussian filter was applied on the image before segmentation. For the median and Gaussian filter, an aperture width of 3 was chosen.

The results of these operations and filter were a grey scale image with dark contours and edges and large white areas representing zones of interest. To highlight this result, the grey-scale image has been converted in colour and distinct random colours have been applied to cover each white area. The result expected was to define safe and unsafe areas within the scene based on each coloured area. However, it did not allow dividing certain regions of the image into sub-regions as can be seen on figure 62 below so the algorithm was modified to intend to solve this issue.

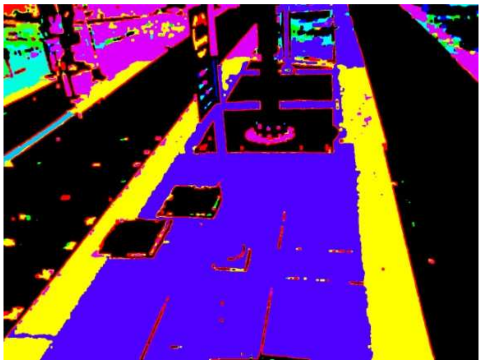


Figure 62: Illustration of the results of the attempt described above[73].

Instead of converting the input image in grey-scale, it was split into RGB channels on which the same stages as described above were applied to each channel. To verify which method from the Otsu and binary thresholding was the most adapted to the type of scenes analysed, various threshold values were used to perform each method on each channels.

The Binary thresholding method was first performed with a threshold $t_b = 110$ for the red channel and $t_b = 80$ for the green and blue channel. Then, the Otsu thresholding method was performed with a threshold $t_o = 75$ for all three channels. Due to the quantity of data, no figures will be presented to illustrate the results of these tests as these methods have been tested with these threshold values on various scenes. However, the results of these tests showed the choice of the best threshold value is not a trivial task as there is no optimal way to select a generic threshold that work best for every type of images. Indeed, images provided by the company differ from each other as they represent a range of very distinct situations with the goal to design a method able to work on most of them.

[73] Used with permission of the author/publisher.

To ensure the scope of our study is sufficiently valid to confidently evaluate the state-of-the-art methods with respect to CCTV processing, an open source code [154] has been acquired allowing performing a large number of modern variations on the thresholding methods**.** The 16 methods are described below:

1. Default: variation of the IsoData algorithm;
2. Huang: Implementation of Huang's fuzzy thresholding method based on the use of Shannon's entropy function;
3. Intermodes: Based on a bimodal histogram;
4. IsoData: Based on the IsoData algorithm of T.W. Ridler and S. Calvard ;
5. Li: Implementation of Li's Minimum Cross Entropy thresholding method;
6. MaxEntropy: Implementation of Kapur-Sahoo-Wong thresholding method ;
7. Mean: Based on the mean of grey levels;
8. MinError: Iterative implementation of Kittler and Illinworth's Minimum Error thresholding ;
9. Minimum: Based on a bimodal histogram similarly to Intermodes method;
10. Moments: Implementation of Tsai's method ;
11. Otsu: Otsu's thresholding clustering algorithm ;
12. Percentile: Based on the fraction of foreground pixels to be 0.5;
13. RenyiEntropy: Using Renyi's entropy;
14. Shanbhag ;
15. Triangle ;
16. Yen .

The results obtained from each of these methods are presented in section 4-2-2-Results and discussed in section 4-2-3-Discussion.

Additional experiments made with the Otsu thresholding method with the same threshold for all of RGB channels $t_o = 75$ have been performed. This method was finally chosen between the others due to more promising results obtained while comparing qualitatively the results obtained for each the sixteen methods tested on 18 clips. However, due to the absence of a generic method to choose the best threshold value, the value used was chosen empirically. Once all stages as described above were processed, the three channels were merged to obtain a final colour image where each white region was filled with a distinct and random colour.

The results obtained are discussed in section 4-2-2-Results. However, it is apparent from the results thresholding methods are not sufficient to divide regions of the image according to safe and unsafe regions correctly in this study case. For this reason, the combination of histogram with thresholding methods is then explored in the next section to deepen the analysis of grey-scale image segmentation.

4-2-1-2- Histograms

This section shows how the combination of thresholding [71] [75] methods and histograms [71] [75] [76] can improve the segmentation process by focusing the analysis on the variation of the intensity values of each RGB channel to segment an image.

The figure 63 below illustrates the method developed:

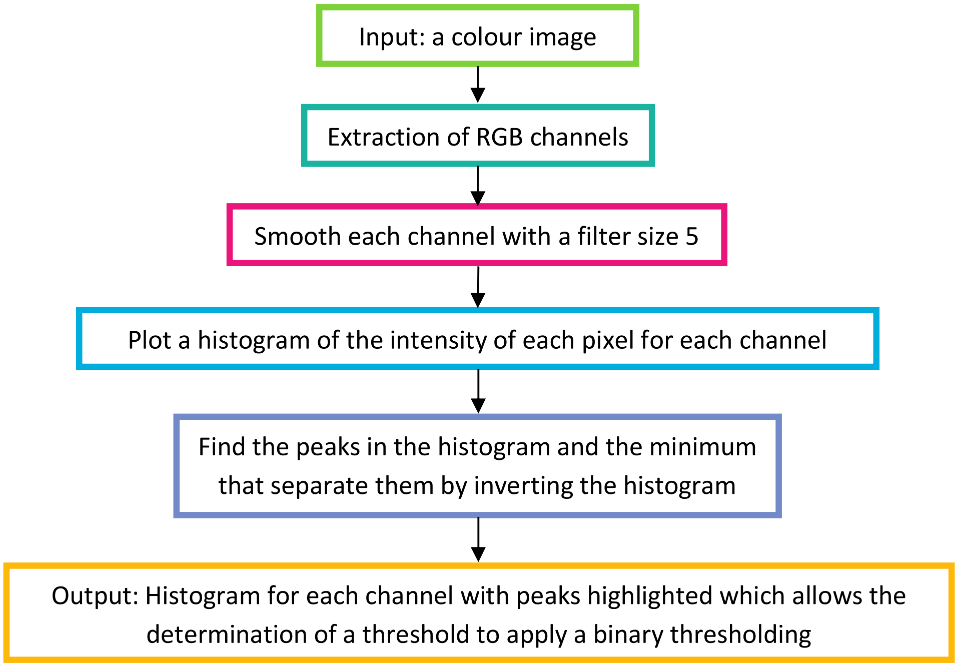
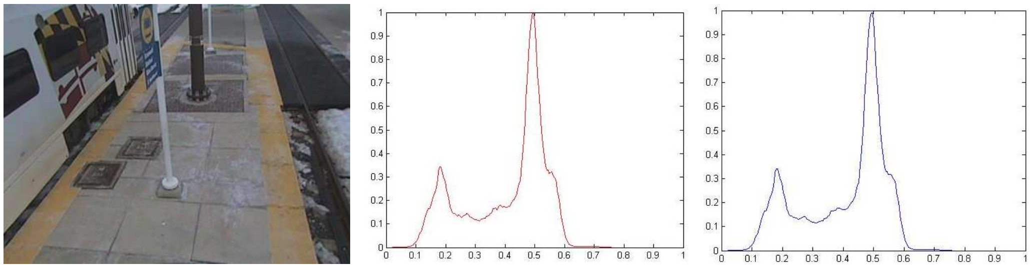

Input: a colour image
Extraction of RGB channels
Smooth each channel with a filter size 5
Plot a histogram of the intensity of each pixel for each channel
Find the peaks in the histogram and the minimum that separate them by inverting the histogram
Output: Histogram for each channel with peaks highlighted which allows the determination of a threshold to apply a binary thresholding

Figure 63: Algorithm overview based on the histogram and thresholding methods combined

The input of our approach being a colour image, each RGB channel was extracted from the input image to obtain three grey-scale images. The steps described below were then applied to each of the channels.

To facilitate the comprehension, the example of the red channel is given, remembering the same approach was applied to the green and blue channels.

The histogram of the intensity of pixels of the grey-scale image corresponding to the red channel was plotted. It resulted in a series of peaks and valleys that represent the repartition of the red colour within the considered image. An example of histogram can be seen on figure 64 below:



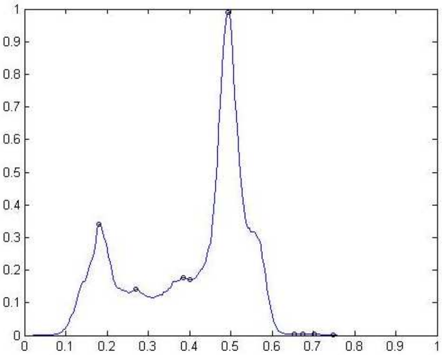
(a) (b) (c)

Figure 64: (a) original image, (b) red channel histogram, (c) (b) red channel histogram smoothed[74].

A median filter has been applied to the histogram as can be seen on figure 64 (c) but in this case it does not appear very different from the histogram on figure 64 (b).

Once the histogram has been generated, it is possible to retain only part of the image that are of interest, i.e. peaks of the histogram which represent part of the image where the red colour varies significantly compared to valleys that correspond to noise or irrelevant information, by determining peaks and valleys of the histogram. To determine peaks, the way adopted here was to find local maxima among the data used to generate the histogram. Then, the number of peaks needs to be determined. It was done in several steps:

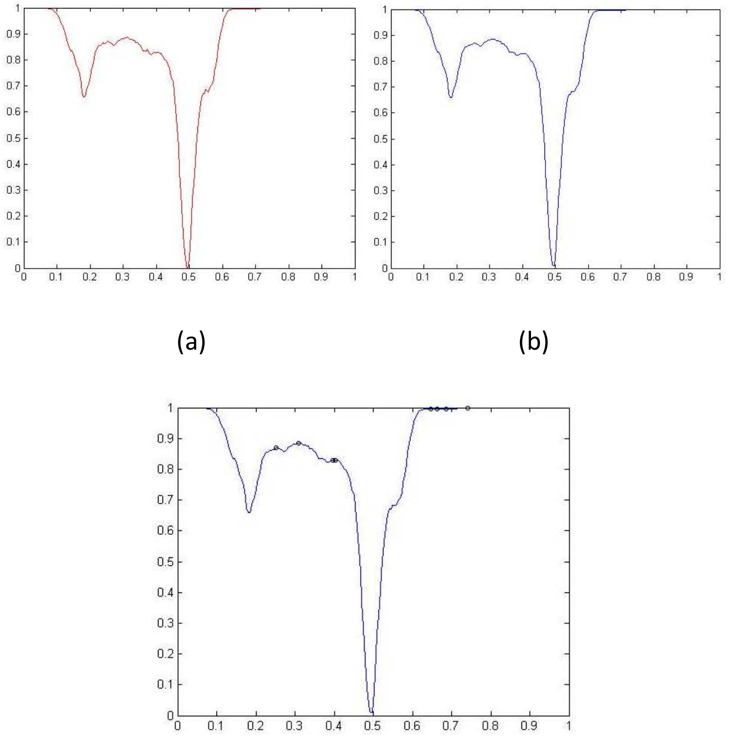
- Determination of the size of each peak in terms of the x and y variation as can be seen on figure 65 below where peaks are represented by a circle;
- Calculation of the first derivative of the histogram;
- Determination of the number of peaks using the following conditions:
  - If the first derivate of the histogram for the x coordinate of peaks is equal to 0;
  - the first derivative of the x coordinate of 5 neighbours points on the side of the peak where x increases are positive;
  - and the first derivative of the x coordinate of 5 neighbours points on the side of the peak where x decreases are negative;
- then a peak has been detected.



[74] Used with permission of the author/publisher.

Figure 65: Smoothed histogram and peaks detected where the circles are on the histogram.

Once peak coordinates have been determined and the number of peaks has been detected, an inverted histogram is generated and peaks of this histogram are then detected as can be seen on figure 66 below.



(a) (b)

(c)

Figure 66: (a)Inverted histogram, (b) smoothed inverted histogram, (c) peaks detected where the circles are on the histogram.

The same steps as those described above are then repeated for the inverted histogram. This allows determining peaks coordinates and the number of peaks of the inverted histogram as these peaks in fact corresponds to valleys in the histogram first generated.

Once valleys have been detected, they are used to remove part of the image that are not of interest to only keep parts of the images that correspond to peaks of the histogram first generated. This then allow to determine a threshold to apply thresholding method on each channel and then only keep the part of the image where relevant objects are gathered.

Additional results obtained on the two other channels will be presented and are discussed in section 4-2-2-Results.

To summarize, the two previous sections concerned an investigation into the thresholding and histogram methods to segment images. However, results obtained from both methods have shown that the analysis of grey images is limited, especially in complex and noisy environments such as railway and underground station scenes where boundaries between objects is so ambiguous and regions are sometimes uncertain that it do not succeed to divide the image into safe and unsafe areas as expected.

The next section then explore a segmentation method is based on colour information that is not too sensitive to ambiguity and uncertainty.

4-2-1-3-Fuzzy C-Means

In this section, the Fuzzy C-means segmentation method is described. This algorithm was introduced in 1973 and according to [152], it can be formulated as follows:

$$\textbf{Minimize}\ \boldsymbol{J_m(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{k_i}^{m} \|x_k - v_i\|^2} \tag{84}$$

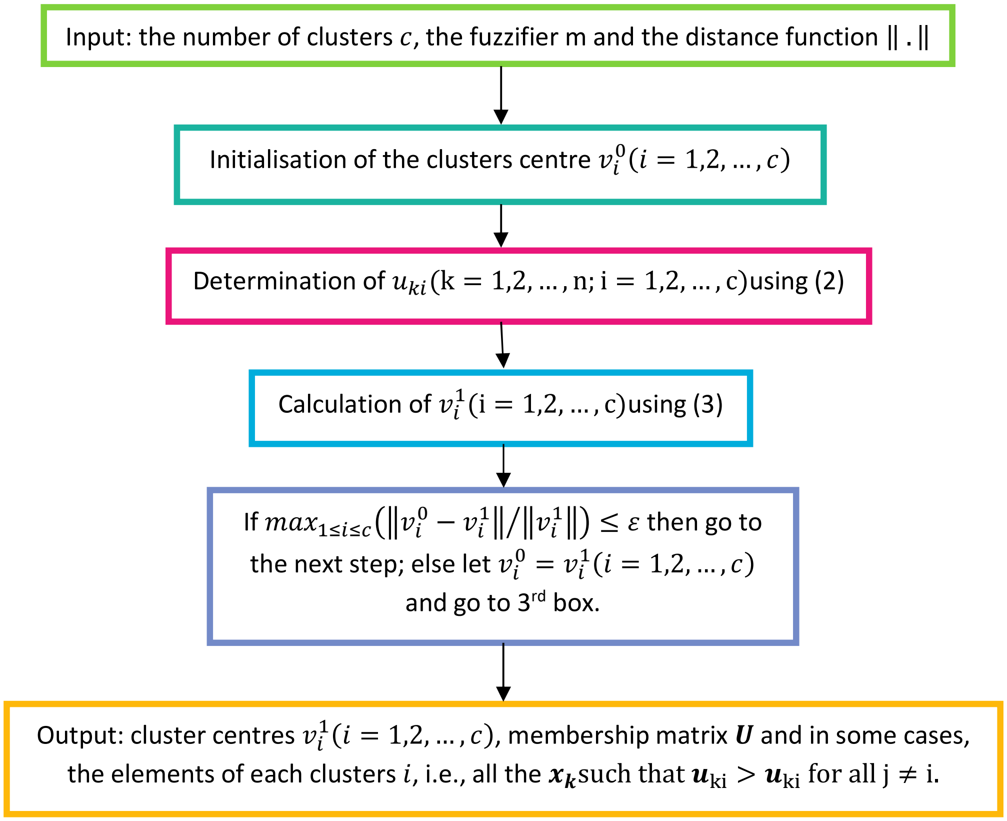
Where n is the total number of data vectors in a given data set, c is the number of clusters, $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^3$the feature data, $V = \{v_1, v_2, \dots, v_n\} \subset \mathbb{R}^3$ the cluster centres and $U = (u_{ki})_{n*c}$ a fuzzy partition matrix composed of the membership of each feature vector x_k in each cluster i. u_{ki} should satisfy $\sum_{i=1}^{c} u_{ki} = 1$for $k = 1,2, \dots, n$ and $u_{ki} \geq 0$ for all $i = 1,2, \dots, c$ and $k = 1,2, \dots, n$. The exponent $m > 1$ is usually called a fuzzifier. To minimize $J_m(U,V)$, the cluster centres v_i and the membership matrix U has to be calculated according to the following iterative formula:

$$\boldsymbol{u_{ki}} = \left\{ \begin{array}{lll} & \boldsymbol{\left(\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{m-1}} \right)^{-1}} & \boldsymbol{if\ \|x_k - v_j\| > 0,} \\ \forall \boldsymbol{j}, & \boldsymbol{1} & \boldsymbol{if\ \|x_k - v_i\| = 0} \\ & \boldsymbol{0} & \boldsymbol{if\ \exists j \neq i\ \|x_k - v_j\| = 0,} \end{array} \right\} \tag{85}$$

For $k = 1, \dots, n$ and $i = 1, \dots, c$:

$$\boldsymbol{v_i = \frac{\sum_{k-1}^{n} u_{ki}^{m} x_k}{\sum_{k-1}^{n} u_{ki}^{m}}, i = 1, 2, \dots, c} \tag{86}$$

It follows the basic algorithm illustrated by the diagram below:



Input: the number of clusters c, the fuzzifier m and the distance function ‖ . ‖

Initialisation of the clusters centre $v_i^0 (i = 1,2,\dots,c)$

Determination of $u_{ki}(\mathrm{k} = 1,2,\dots,\mathrm{n}; \mathrm{i} = 1,2,\dots,\mathrm{c})$using (2)

Calculation of $v_i^1(\mathrm{i} = 1,2,\dots,\mathrm{c})$using (3)

If $max_{1\le i\le c}(\|v_i^0 - v_i^1\|/\|v_i^1\|) \le \varepsilon$ then go to the next step; else let $v_i^0 = v_i^1 (i = 1,2,\dots,c)$ and go to 3rd box.

Output: cluster centres $v_i^1 (i = 1,2,\dots,c)$, membership matrix $\boldsymbol{U}$ and in some cases, the elements of each clusters i, i.e., all the $\boldsymbol{x_k}$such that $\boldsymbol{u}_{\mathrm{ki}} > \boldsymbol{u}_{\mathrm{ki}}$ for all $\mathrm{j} \neq \mathrm{i}$.

Figure 67: Algorithm overview based on the Fuzzy c-means algorithm

To summarise, this algorithm follows the stages below:

1. Random selection of the centre of each cluster (i.e. expected number of clusters defined as input) ;
2. In RGB space, calculation of all minimal distances between the centre and other pixels in its surroundings which then define clusters' members ;
3. Re-calculation of the cluster centre after all members have been determined ;
4. Repeat step 2 and 3 until each cluster remains stable based on the distances calculated and the membership.

This algorithm has been applied to the 18 different clips available for various number of clusters as input, from 2 to 10 every 2 steps, including 5 and 7. The selection of various numbers of clusters allowed finding the prediction of this algorithm as for the best number of clusters. A qualitative analysis of these tests is given and discussed in section 4-2-2-Results and in section 4-2-3-Discussion.

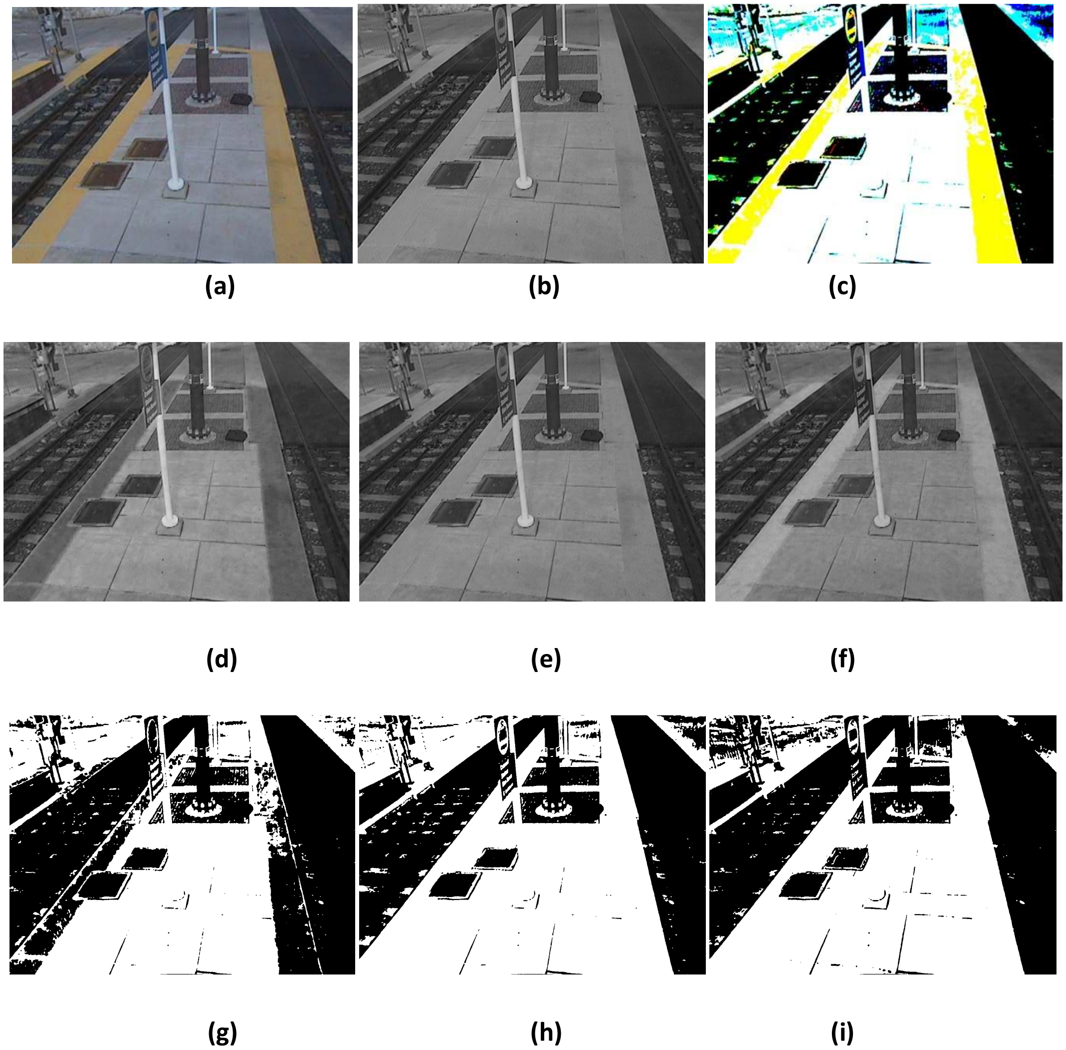
To summarize, this section presented the investigation of three common segmentation techniques, two applied on grey-scale images while the last one was applied on colour images.

The next section provides a few results obtained from these three methods and conclude as for their use in addition to VPs detected for scene understanding.

4-2-2-Results

This section presents results obtained with the three segmentation methods considered namely and in this order: thresholding method, histograms and the Fuzzy c-means clustering.

4-2-1-1-Thresholding



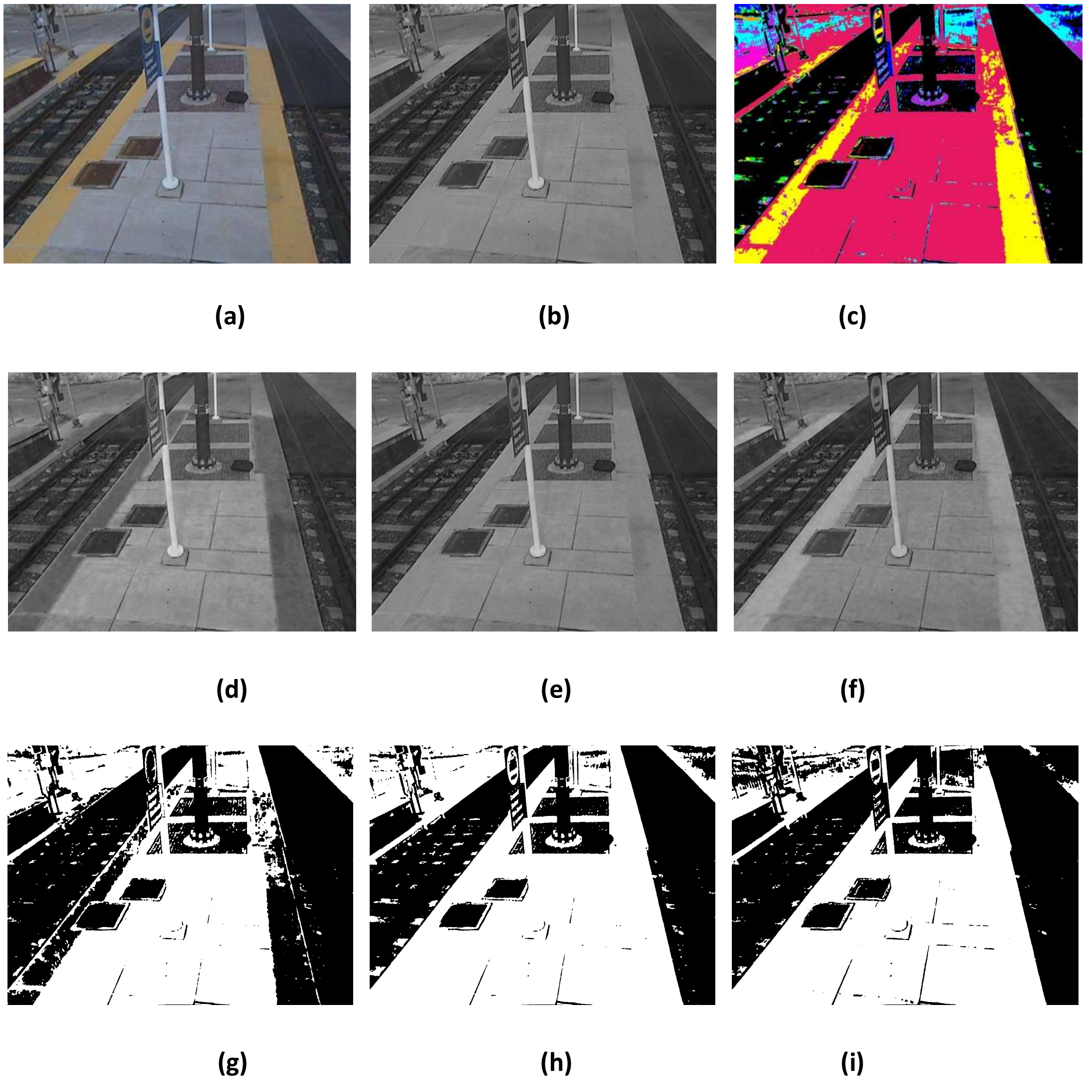
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 68: (a) Original Image, (b) Grey Image, (c) colour Image obtained after merger of the three channels once the thresholding has been applied, (d) Red channel Image, (e) Green channel Image, (f) Blue channel Image, (g) Red channel after Binary thresholding, (h) Green channel after Binary thresholding, (i) Blue channel after Binary thresholding[75].

[75] Used with permission of the author/publisher.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 69: (a) Original Image, (b) Grey Image, (c) Resulting segmented Image after merger of the three channels, (d) Red channel Image, (e) Green channel Image, (f) Blue channel Image, (g) Red channel after thresholding, (h) Green channel after thresholding, (i) Blue channel after thresholding[76].

[76] Used with permission of the author/publisher.

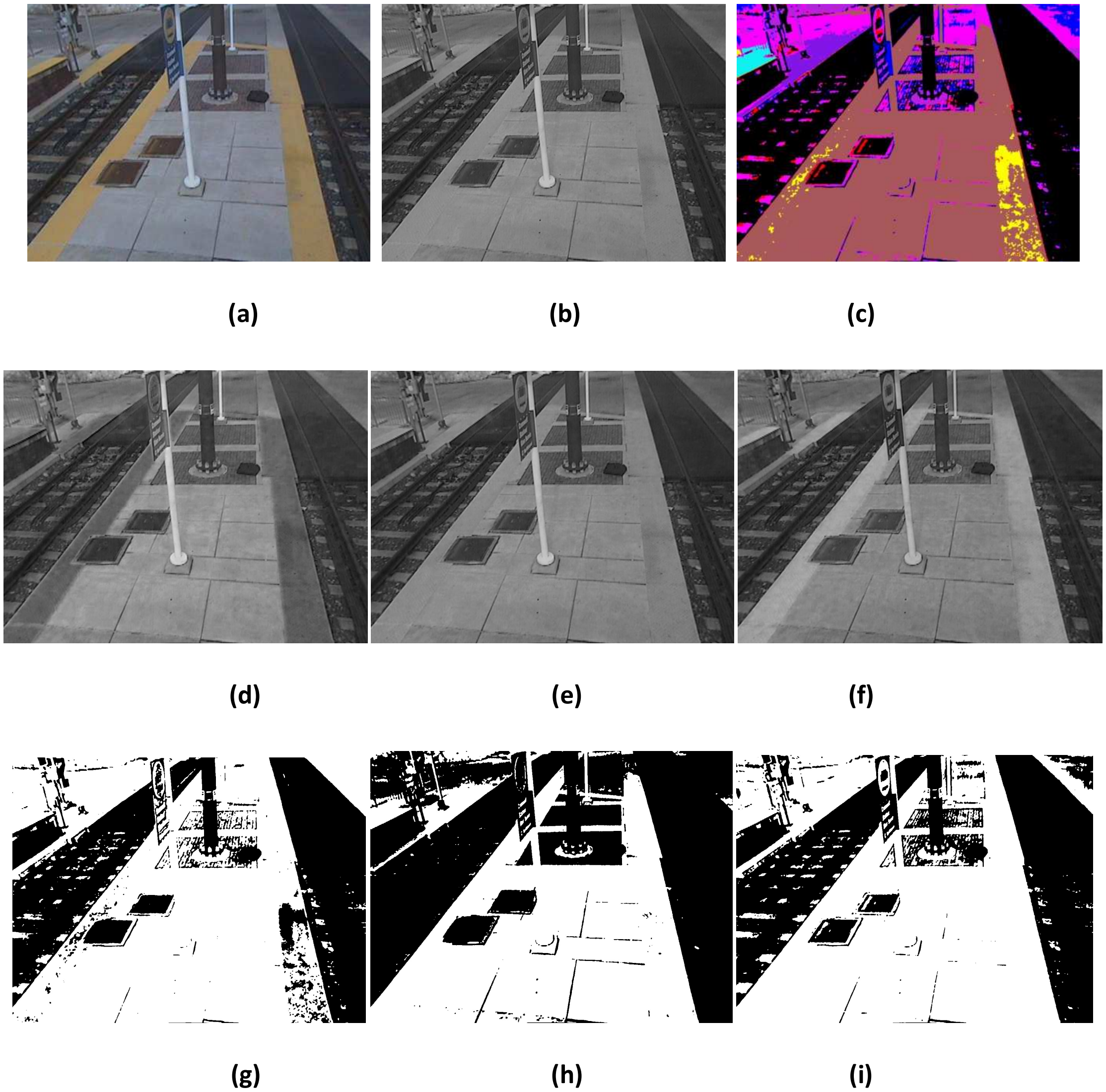


Figure 70: (a) Original Image, (b) Grey Image, (c) Resulting segmented Image after merger of the three channels, (d) Red channel Image, (e) Green channel Image, (f) Blue channel Image, (g) Red channel after thresholding, (h) Green channel after thresholding, (i) Blue channel after thresholding[77].

[77] Used with permission of the author/publisher.

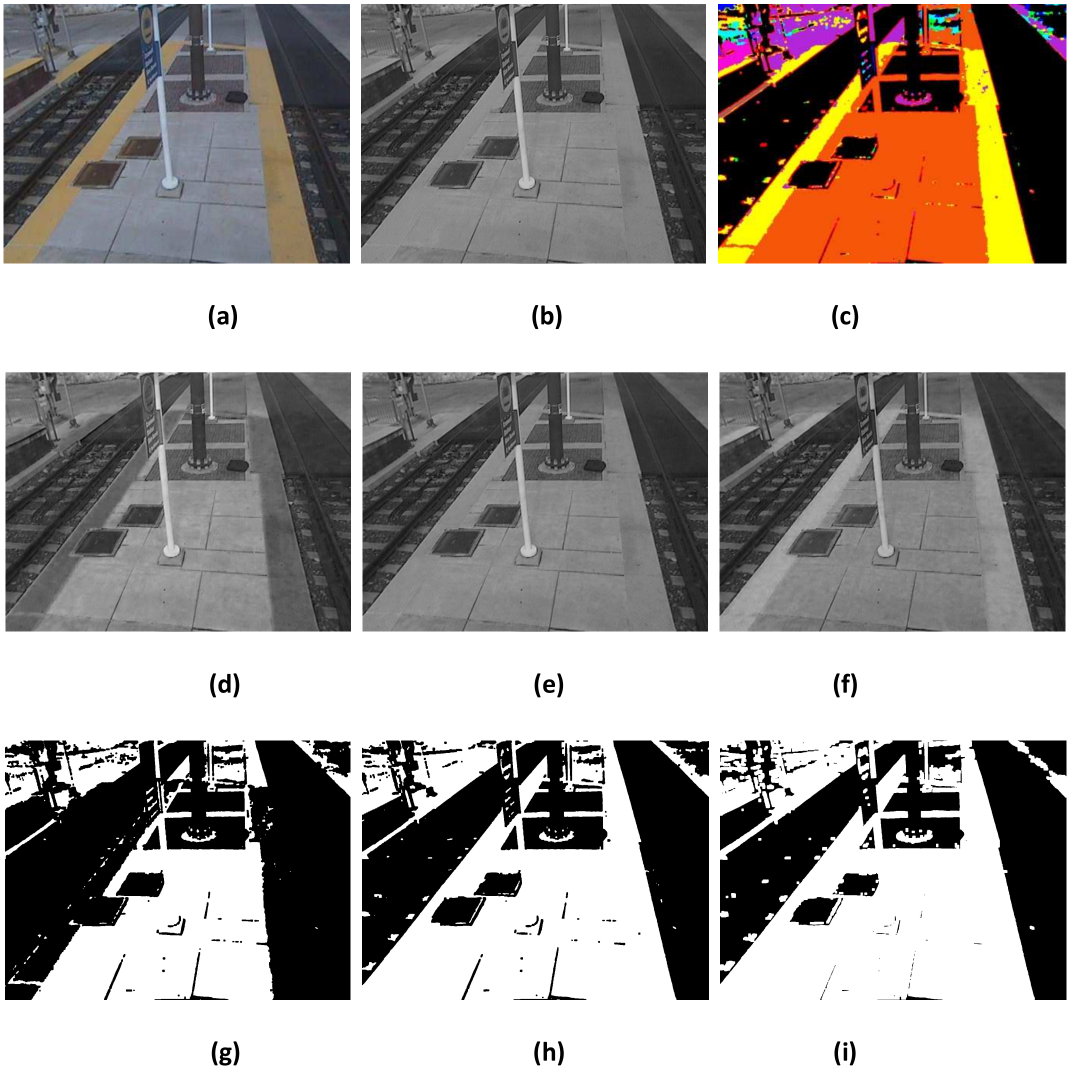


Figure 71: (a) Original Image, (b) Grey Image, (c) Resulting segmented Image after merger of the three channels, (d) Red channel Image, (e) Green channel Image, (f) Blue channel Image, (g) Red channel after thresholding, (h) Green channel after thresholding, (i) Blue channel after thresholding[78].

The Otsu and binary thresholding methods have been tested first. The goal was to be able to segment as many clusters as possible according to those expected. These clusters are: tracks, the yellow band on the platform, the platform, benches, bins, poles, shelters, escalators or lifts. Intensity variations alone are unlikely to offer complete segmentation but public places tend to be “themed” (i.e. all bins, pillars, etc. are matching). However, the segmentation process was expected to segment regions in as many as clusters as possible from those listed above. However, the main ones to detect as a priority are: tracks, the platform, benches, bins, shelters.

[78] Used with permission of the author/publisher.

As for the yellow band on the platform and the escalators or lifts, it would useful to differentiate them from other regions of the image as these clusters provide information about dangerous zone as well as non-flat surfaces which are required to complete scene understanding accurately. The yellow band on the platform represents a dangerous area of the scene for human to be as it is close to the train and the end of the platform. So, its clustering would allow raising an alarm every time someone is detected in this region of the image. Moreover, in spite of the fact that stairs and escalators can be dangerous areas too where someone can fall down, they represent non-flat surfaces. So, their clustering would help the proposed VP detection method to distinguish VPs from the three orthogonal directions of a Manhattan world and those from non-flat surfaces as they would obviously be in different clusters.

Figures 68 to 71 show results obtained with a thresholding method applied and various combination of the use of OpenCV erode and dilate functions. The goal was to exaggerate some contours while reducing others to be able to segment the scene according to the following clusters: the platform, the yellow band if possible, tracks, poles if possible and the left luggage if possible. However, even if the scene on which it has been applied was simpler than other due to the presence of not many objects it did not provide the result expected as the left luggage, the poles and the yellow band on the platform were either combined with other cluster or not clustered at all. An example of this case can be seen on figure 70, the left luggage is in a different cluster than the platform but combined with the poles.

The following results were obtained with the code allowing testing 16 different thresholding techniques.

- Tests realised with an open source code performing sixteen thresholding techniques:

Each result presented below from left to right and top to bottom concerns the following thresholding techniques:

1. Default.
2. Huang.
3. Intermodes.
4. IsoData.
5. Li.
6. MaxEntropy.
7. Mean.
8. MinError.
9. Minimum.
10. Moments.
11. Otsu.
12. Percentile.
13. RenyiEntropy.
14. Shanbhag.
15. Triangle.
16. Yen.

A brief overview of these methods has been described in section 4-2-1-1-Thresholding.

Indoor scenes:

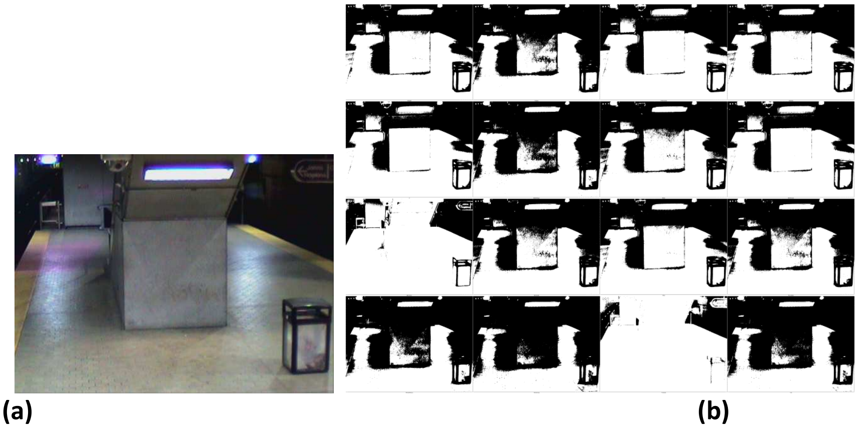


Figure 72: (a) Original Image, (b) Resulting Image for the several thresholding techniques[79]

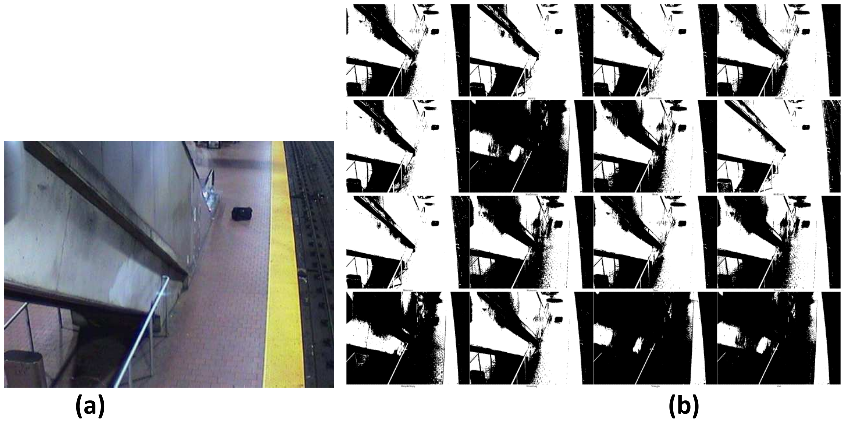


Figure 73: (a) Original Image, (b) Resulting Image for the several thresholding techniques[80]

[79] Used with permission of the author/publisher.

[80] Used with permission of the author/publisher.

Outdoor scene:

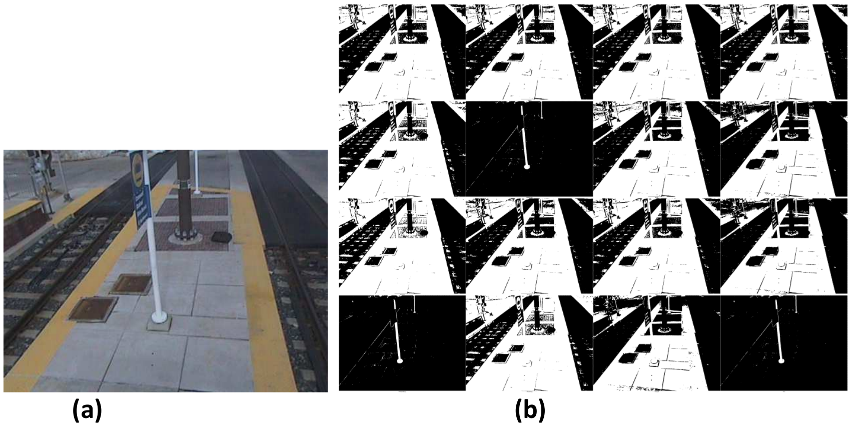


Figure 74: (a) Original Image, (b) Resulting Image for the several thresholding techniques[81]

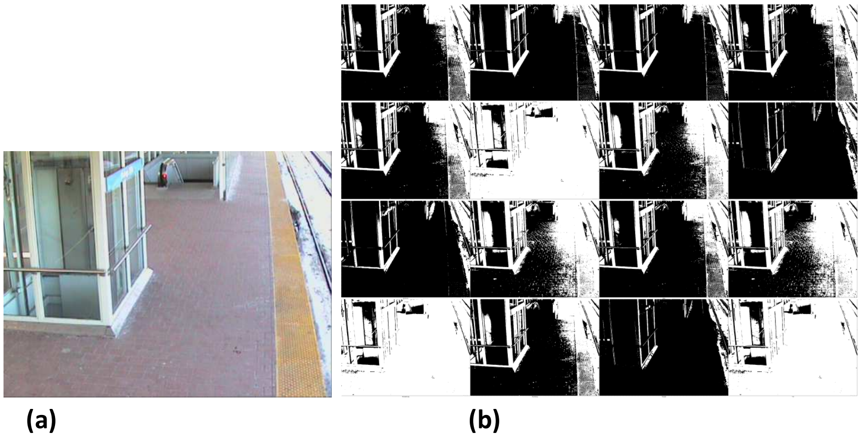


Figure 75: (a) Original Image, (b) Resulting Image for the several thresholding techniques[82]

The sixteen thresholding methods have been tested on the 18 available clips in the hope to find out which one was the best for most or all of the scenes available.

However, as can be seen on figures 72 to 75, it is obvious that the best thresholding technique differs from a scene to another due to their difference in complexity, lighting, shadows, weather conditions etc. Moreover as noticed for the VP detection task, it is also much more difficult to obtain well-defined segmented regions on indoor scenes due to poor lighting and darkness which is ubiquitous compared to outdoor scenes. This is clearly noticeable on the results obtained from various thresholding techniques as indoor scenes appear to be either too black or too white in most of the image which does not allow any distinction between background and objects within the scene.

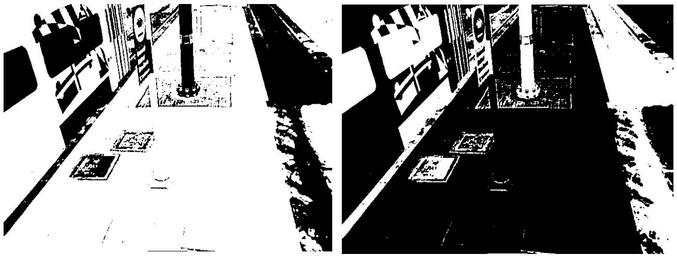
[81] Used with permission of the author/publisher.

[82] Used with permission of the author/publisher.

To address these issues, the histogram method is proposed as it permits to analyse the repartition of colour within the image to keep only information of part of the image that are of interest. Results of the histogram method are presented in the next section.

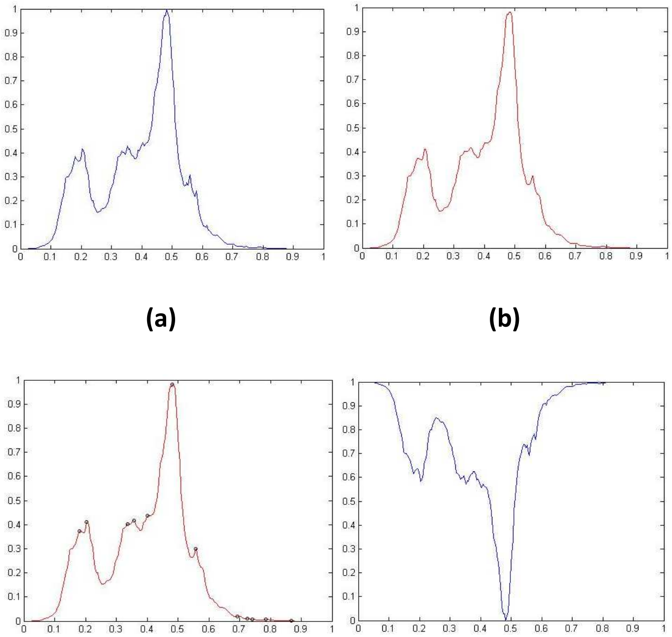
4-2-1-2- Histograms

- Example of the results obtained on the blue and green channels of the outdoor scene considered in section 4-2-1-2-Histograms



(a) (b)

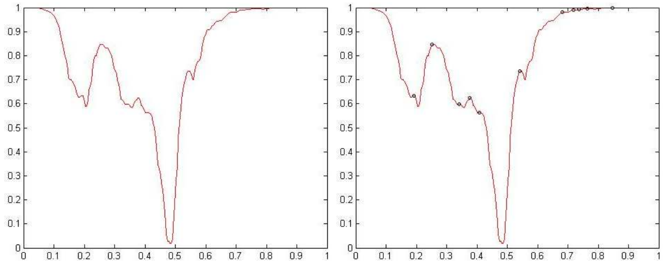
Figure 76: (a) Binary image obtained with the threshold determined from the analysis of the histogram of the red channel, (b) negative of the binary image (a)[83].



(a) (b)

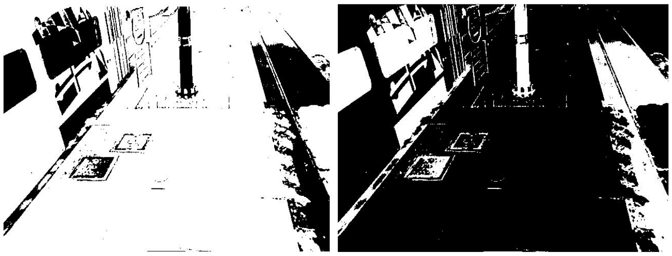
(c) (d)

[83] Used with permission of the author/publisher.



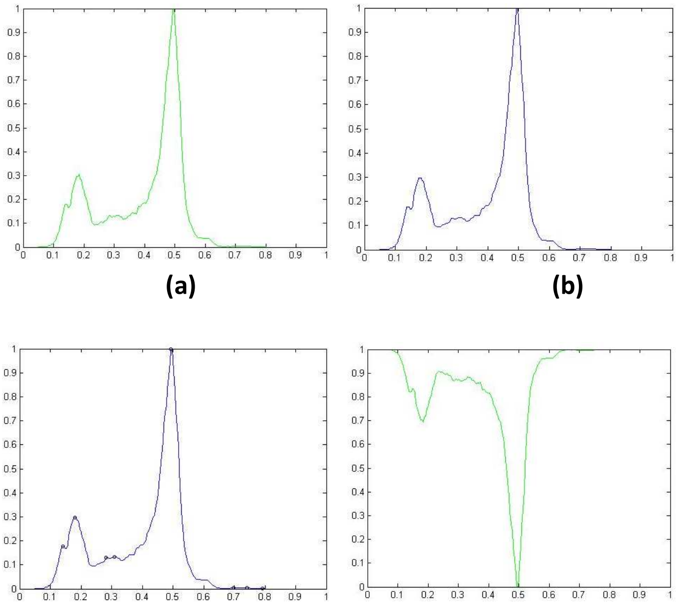
(e) **(f)**

Figure 77: (a) histogram for the blue channel, (b) smoothed histogram, (c) peaks detected on the smoothed histogram, (d) inverted histogram, (e) smoothed inverted histogram, (f) peaks detected on the smoothed inverted histogram that correspond to valleys of the histogram.



(a) **(b)**

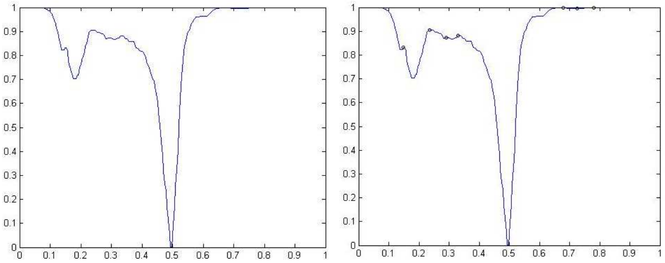
Figure 78: (a) Binary image obtained with the threshold determined from the analysis of the histogram of the blue channel, (b) negative of the binary image (a)[84].



(a) **(b)**

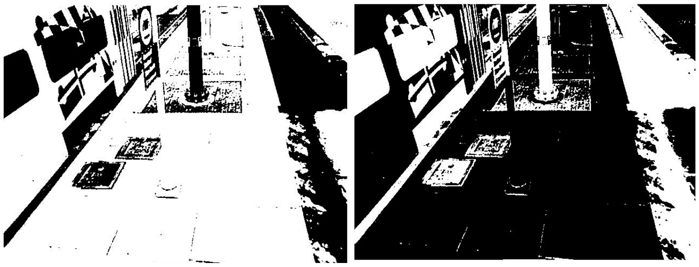
[84] Used with permission of the author/publisher.

(c) (d)



(e) (f)

Figure 79: (a) histogram for the green channel, (b) smoothed histogram, (c) peaks detected on the smoothed histogram, (d) inverted histogram, (e) smoothed inverted histogram, (f) peaks detected on the smoothed inverted histogram that correspond to valleys of the histogram.



(a) (b)

Figure 80: (a) Binary image obtained with the threshold determined from the analysis of the histogram of the green channel, (b) negative of the binary image (a)[85].

As can be seen on figures 77 and 79, there are only two peaks for the histogram of each channel of the considered image so the determination of the threshold can be done by determining the valley there is between those two peaks and can then be used as a threshold value. Furthermore, the binary thresholding result for each channel obtained with the threshold determined based on the valley detected for the histogram of each channel is illustrated on figures 77 and 79 for the red, blue and green channel respectively. The results obtained do not highlight clearly the distinct cluster that are of interest. This must be due to change in lighting and shadows, the presence of the train and ambiguous boundaries that allow separating one region from another into different clusters.

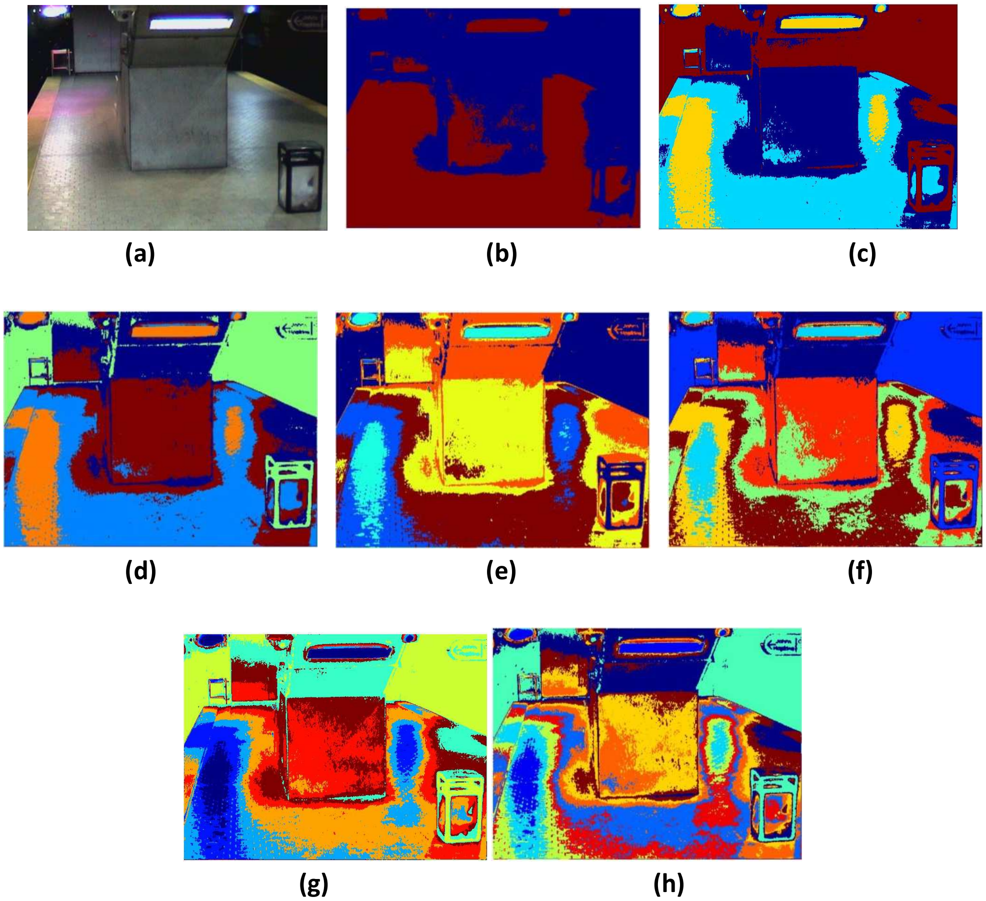
For this reason, another method has been investigated in order to solve the issues of ambiguous boundaries and uncertain regions due to the complexity of the scene. The results obtained with this other method are presented in the next section.

4-2-1-3-Fuzzy C-Means

[85] Used with permission of the author/publisher.

This section presents results obtained with the FCM algorithm.

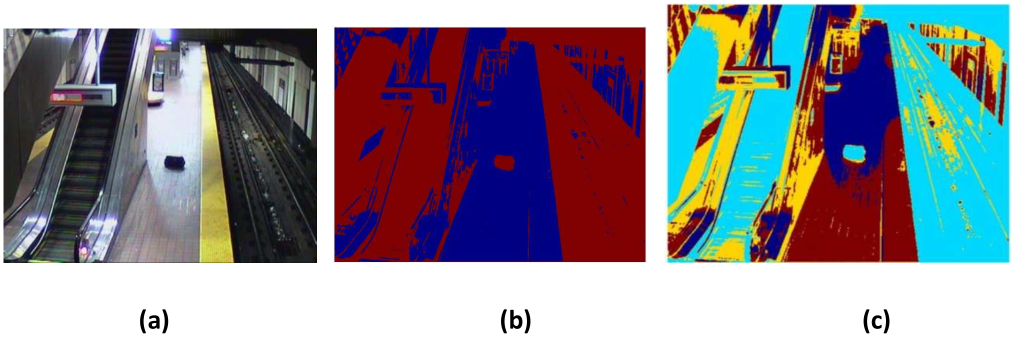
- Indoor scenes:


(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 81: (a) Original Image, (b) 2 clusters, (c) 4 clusters, (d) 5 clusters, (e) 6 clusters, (f) 7 clusters, (g) 8 clusters, (h) 10 clusters[86].


(a) (b) (c)

[86] Used with permission of the author/publisher.

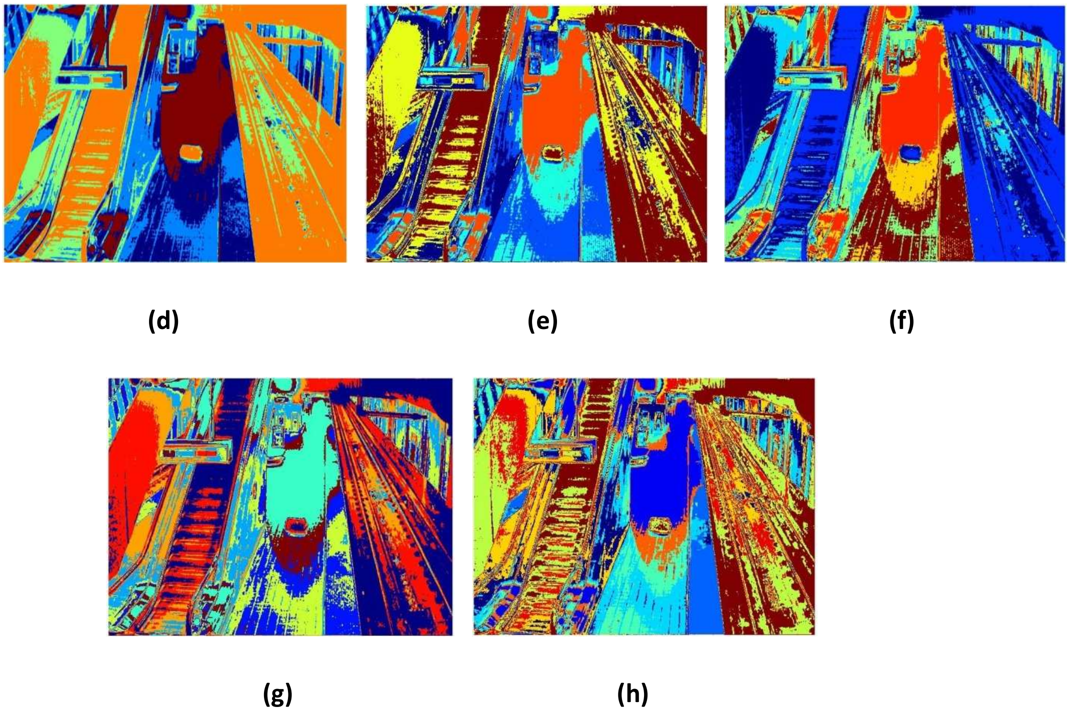
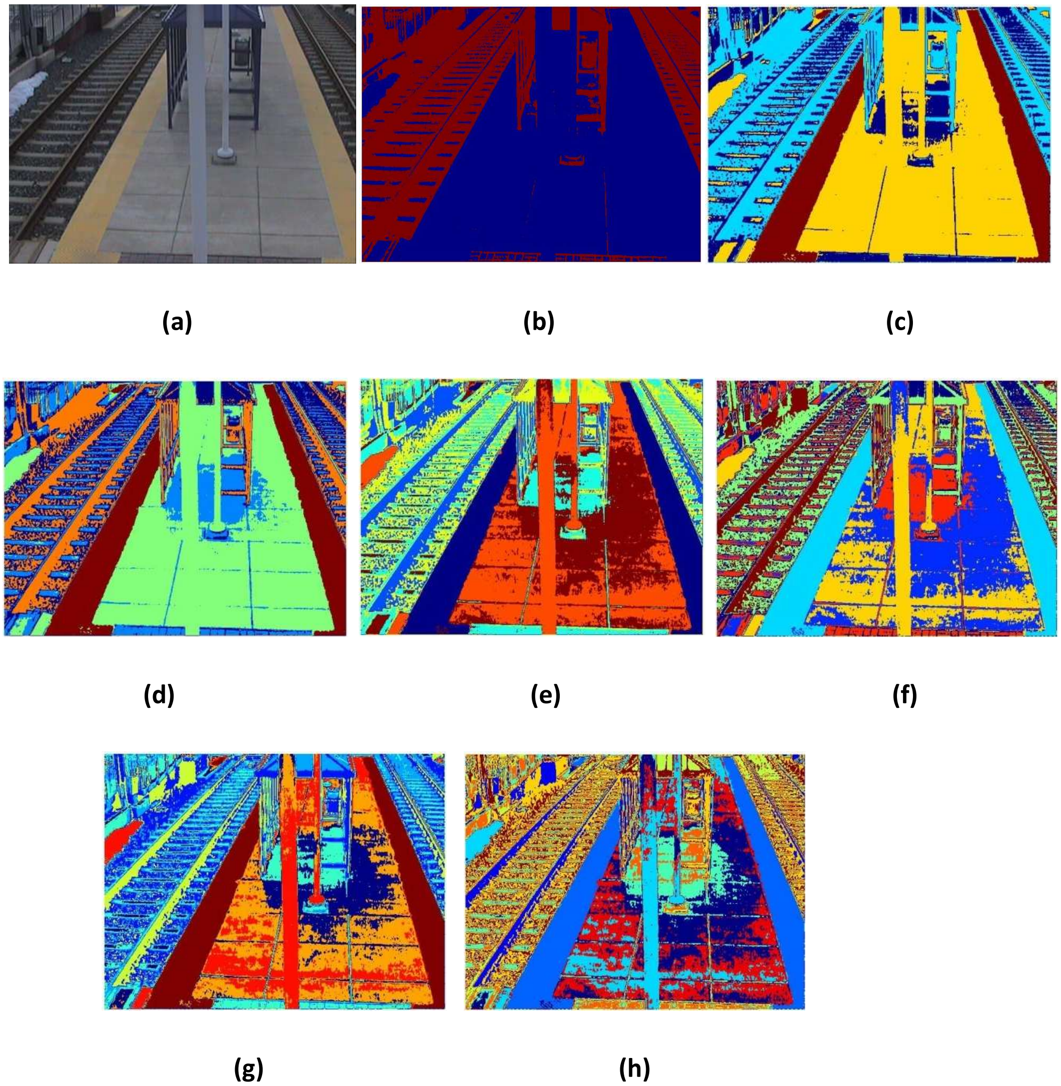


Figure 82: (a) Original Image, (b) 2 clusters, (c) 4 clusters, (d) 5 clusters, (e) 6 clusters, (f) 7 clusters, (g) 8 clusters, (h) 10 clusters[87].

[87] Used with permission of the author/publisher.

- Outdoor scenes:



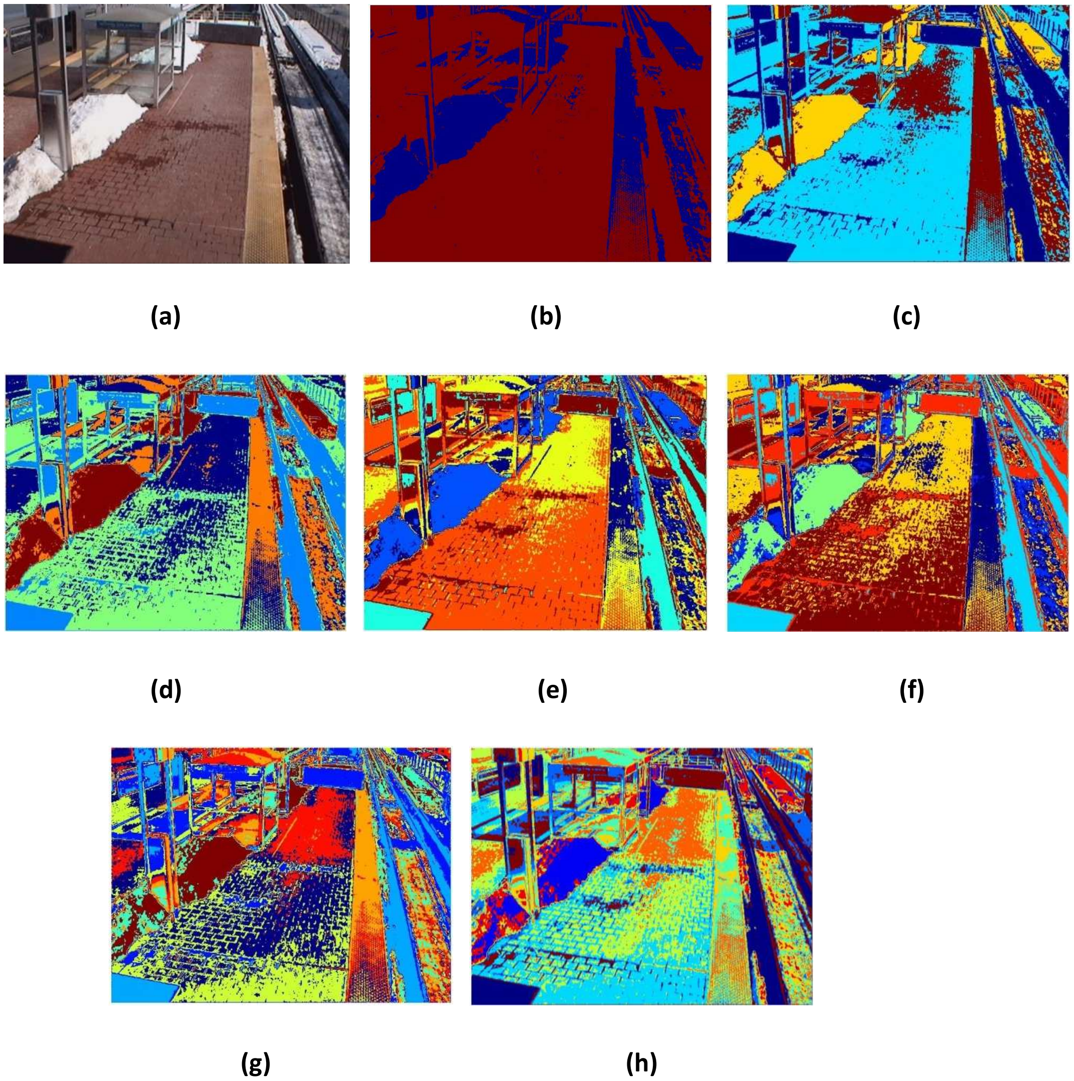
(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 83: (a) Original Image, (b) 2 clusters, (c) 4 clusters, (d) 5 clusters, (e) 6 clusters, (f) 7 clusters, (g) 8 clusters, (h) 10 clusters[88].

[88] Used with permission of the author/publisher.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 84: (a) Original Image, (b) 2 clusters, (c) 4 clusters, (d) 5 clusters, (e) 6 clusters, (f) 7 clusters, (g) 8 clusters, (h) 10 clusters[89].

[89] Used with permission of the author/publisher.

Number of Video surveillance Scene	Nb of clusters expected	Best nb of clusters FCM
1	**7**	**5**
2	**5**	**4 or 6**
3	**4**	**5**
4	**4**	**4**
5	**4**	**5**
6	**7**	**5**
7	**5**	**6**
8	**6**	**6**
9	**4**	**6**
10	**5**	**5**
11	**4**	**6**
12	**5**	**6 or 7**
13	**7**	**6**
14	**7**	**5**
15	**7**	**6**
16	**4**	**5**
17	**6**	**6**
18	**7**	**6**

Table 18: Summary of the number of expected clusters to detect and those that have been the best detected with FCM; The clusters considered are: Tracks, Yellow Band Platform, Platform, Bench, Bin, Poles(lift), Shelter, Escalators(Stairs) when within the scene.

The results presented above have been obtained as output of the FCM algorithm for different number of clusters: 2, 4, 5, 6, 7, 8 and 10.

As can be seen on figure 81 to 84, the results obtained vary in function of the number of clusters used as input of the algorithm:

- Two clusters are not even sufficient to differentiate correctly the tracks from the platform;
- Four clusters allow dividing the scene into additional clusters but are not as precise as results obtained with more clusters;
- Five and six clusters usually provide the best segmentation as they not only cluster the main elements of interest but also additional ones such as differentiating the platform from the yellow band;
- Finally seven to 10 clusters provide noisier and noisier segmentation due to a lack of such a number of objects or boundaries between regions.

Furthermore, these results show the difference between five and six clusters remains difficult to point out.

Moreover, once more the FCM algorithm performed a better segmentation on less complex and outdoor scenes due to the presence of a brighter light as can be seen on figure 83 and 84. However, the segmentation result obtained on figure 84 is not as accurate of the segmentation result obtained on figure 83. This can be explained due to a change in weather condition where the presence of snow confuses the clustering method and does not allow segmenting the image as precisely as in absence of snow. As for figure 81, the contrast between really dark and really bright areas (due to light reflection)confuses the algorithm and create clusters where there should not be one due to noise created by extreme change in lighting even if the scene is not as complex as others. Figure 82 is actually an example of one of the most complex scenes as it not only has really bright areas due to light reflection but also shadows contrasting the bright areas and non-flat surfaces such as escalators. The complexity of the scene in addition to extreme changes in lighting has also been proven to confuse the clustering method.

The table 18 above represents a qualitative analysis of expected clusters and those that subjectively seemed to perform segmentation better on each scene. The second column of the table corresponds to the number of cluster expected for each scene and has been counted subjectively. The third column corresponds to the number of cluster that has been selected (again subjectively) to perform the best results compared to the number and type of clusters expected, Although the "perfect" number of clusters clearly depends on the scene, the consistency of results indicates only small variation in the number of regions required. This table shows that 8 out 18 scenes obtain better results with 6 clusters, 7 out of 18 obtain better results with 5 clusters, only one with 4 clusters and two are undecided between 6 and 7 or 4 and 6 clusters. However, it did not allow determining which number of cluster would be best for every scene.

So, this method performs better results than thresholding and histogram and allows segmenting the image according to the main clusters of interest and eventually additional ones but it still requires a number of clusters as input. This is an inconvenient that would need to be adapted to be applied to any type of scenes without having to decide which number of clusters should be detected but rather detect a different number of clusters automatically according to the scene itself. This can be investigated as future work.

In this section, results obtained from three distinct segmentation methods have been presented and discussed. The next section then concludes on the use of 2D image analysis for 3D scene understanding in the context of video surveillance.

4-3-Summary on 2D image analysis

This chapter described the VP detection approach as well as the investigation of three segmentation methods. Each of these methods have potential to be used as preliminary tools for 3D scene understanding as they provide meaningful 2D features such as lines, VPs and colour information that are related to the scene geometry and organisation. However, the experiments and detailed evaluation presented in this chapter shown that VP detection and segmentation remain two significant but very difficult tasks, especially in realistic, complex and noisy environments. Indeed, this is due to their lack of information which do not permit to use one or the other on their own or even combined to perform scene understanding accurately. This is due to the insufficiency of 2D features to deal with the complexity of video surveillance environments and the presence of noise due to change in exterior conditions such as lighting, shadows, weather etc.

However, the exploration of these approaches demonstrated that such methods can act as an essential starting point by providing 2D information such as geometry and colour as these attributes, in turn, are essential for object classification, event detection, object recognition and scene understanding. However it is required to be combined with other sources of information in order to build a reliable picture. Therefore, one major conclusion to this chapter is that current methods fall far short of reliably attaining the industry goals set out in Chapter 2 for general application.

VPs or segmentation would not be sufficient to understand and interpret a scene on their own or even combined even if they were performed perfectly. Therefore, the combination of 2D with richer information provided by 3D data can overcome this issue. The next chapter explores ways to acquire 3D data and extract meaningful features such as planar surfaces to assist scene understanding. Moreover, VP detection and segmentation are then envisaged to be fused in order to generate a more robust scene understanding – e.g. solid walls may be detected by 3D vision which is then used to aid both reliable VP detection and segmentation. Other examples of this combination will be explained in more details in chapter 8.

Chapter 5: From 2D to 3D

As highlighted in the previous chapter, 2D methods are limited in that they do not provide sufficient information to perform scene understanding on their own. Here, methods are introduced that aim to acquire 3D data which, in turn, add the ability to better segment, calibrate and fully understand an image. Of course, the wealth of information that 3D vision can introduce is not free. There have been a range of attempts at 3D data acquisition as discussed in the literature review above. However, most of the methods fall short due to several reasons (e.g. cost, lack of real-time ability, insufficient accuracy, etc). In this chapter an empirical investigation of some of the methods that are most suited to CCTV is presented, with a particular emphasis on plane extraction, as this is the central step in 3D segmentation and scene understanding. Unfortunately, the quality of data obtained is not always sufficient and these methods seem inadequate at present. However, the next chapter presents a novel method for 3D data acquisition that overcomes many of the weaknesses of the state-of-the-art.

5-1-3D point cloud acquisition

The previous chapter showed the limited potential of 2D feature extraction and segmentation to contribute for scene understanding. However, as explained previously, the limitation comes from the use of 2D data which are not as rich as 3D data. Furthermore, it is possible to combine VP detection and segmentation to recover 3D information as presented in the literature by extracting volumes such as planes (walls, roofs, etc) that are composed of lines and VPs, and segment 3D data into relevant regions or highlight objects of interest based on colour information and lines as boundaries.

This chapter shows how 3D data can be recovered to extract planar surfaces that will assist the 2D methods for scene understanding.

Among the literature, many methods allow acquiring 3D information. However, they all have strengths as well as limitations that need to be overcome and do not make all of them suitable for video surveillance applications. In this chapter, two devices based on only two of these existing methods are investigated: Microsoft Kinect and the Bumble Bee binocular stereo camera.

While the first section of this chapter will describe their working principle and characteristics device, the other sections propose a method to extract planar surfaces from 3D point clouds acquired from one of these two devices – an essential step towards scene understanding and a goal that proved elusive using the previous chapters.

5-1-1-Cameras considered

This section describes two well-established methods among a variety presented in the literature as they seem more adapted and efficient for this study case, less sensitive to noise and at an affordable cost compared to others. Moreover, more technical specifications can be found for the Microsoft Kinect in [159] and for the Bumblebee camera in Appendix 3. More information about the comparison of the most common methods to acquire 3D data is described in chapter 3. However, the Microsoft Kinect and the Bumble bee binocular stereo camera will not be quantitatively compared as they do not work on the same principle and have different characteristics. The aim of this investigation here, is not to define which one is the best but which one will be the most suitable for our application to reach our final goal.

5-1-1-1-Kinect

The Kinect [155][156] is a device motion sensor designed by Microsoft for video games consoles and windows computers.

The first generation was introduced in November 2010 as an accessory to Xbox 360 Console and a windows version was released in February 2012. A Kinect software development kit for windows 7 was released on June 2011 allowing the implementation of Kinect applications in C/C++ and other programming languages.

As can be seen on figure 85 below, it is composed of an RGB camera, an infrared (IR) emitter, an IR depth sensor, an accelerometer, a motor and a multi-array microphone which provides full-body 3D motion capture, facial and voice recognition and allow the tracking of objects or human movements in a 3D environment. This sensor is a horizontal bar connected to a small base with motorized pivot. In use, it has to be positioned lengthwise above or below the video display.

Figure 85[90]: Kinect sensor composition [157]

The RGB camera operates at 30 Hz and can store the three channels in a 640*480 resolution with 8-bit per channels or a higher one such as 1280*960 which results in colour images. The depth sensor is composed of an IR emitter or IR laser projector and an IR camera. A known noisy pattern of structured IR light is projected by the emitter in the scene. The deformation of this pattern due to the surface of objects observed within the scene is then captured by the IR camera as it is not visible by the RGB camera. This distortion allows the computation of the distance between objects and the sensor which provides depth information and permits to recover depth maps of the scene (cf. chapter 3 section on 2.5D/3D methods for more details)[91].

[90] This figure is redacted for copyright reasons.

[91] Note that not all details of the Kinect's workings have been disclosed by Microsoft.

The Kinect's depth sensor has two ranges: the default and the near range which are illustrated on the figure 86 below. However, the near range is only available in the Kinect sensor for Windows which is not the one used for the experiments.

Figure 86[92]: Depth range for Kinect sensor [158]

The Kinect sensor allows the acquisition of RGBD data defined by the three colour channels (red, green and blue) and depth information.

As seen in chapter 3 and table 1, it is stated in the literature that Kinect is sensitive to noise or occlusions [129] [125] and has a limited range [129]. So, it requires pre-processing such as filtering (Gaussian filter) to improve data quality, calibration to align the colour image to depth information and is limited by the area it covers. However, it is cheaper, captures point clouds with a higher accuracy and usually gives better or similar results than other cameras [129]. Another advantage is that it combines depth and colour information which has shown improvements for tasks such as object detection, classification etc [129]. The resolution rendered (640*480) is inferior to the resolution used by most of the scanners but it is sufficient for many applications.

Additional characteristics are summarized in the table 19 below:

Kinect	Specifications
Area covered	Up to 6 m^2
Field of view	43° vertically and 57° horizontally
Vertical tilt range	±27°
Frame rate (depth and colour stream)	30 fps
Minimum distance field of view	63 cm vertically and 87 cm horizontally
Operating range of the sensor	0.5 (1.8 m in case of using it for interaction) m to 5.0 m or 0.8m to 3.5m [K1]?

Table 19: Kinect sensor specifications [156][158]

[92] This figure is redacted for copyright reasons.

A next generation Kinect or XboX One Kinect [160] has been released on November 22, 2013. It is composed of a 1080 pixels camera as can be seen on figure 87 below.

Figure 87[93] : Xbox One Kinect [161]

On the contrary the one described previously, this new generation of Kinect sensor is based on the Time-of-Flight principle as described in chapter3 section 3-4-2-Time-of-Flight and can be used on smaller environments than the generation described above.

This section described the Kinect sensor working principle and gave an overview of its characteristics. The next section presents the same type of information for another device the bumblebee camera which on the contrary to the Kinect based on structured light is based on stereovision principle.

5-1-1-2- Bumblebee2 camera

The Bumblebee2 camera [162] is a binocular stereo vision camera designed by Point Grey Research which can be used for applications such as people tracking, gesture recognition, mobile robotics etc.

The Bumblebee2 is a stereo vision camera which provides a library for real-time applications with a simple user interface allowing user to access in real-time, 3D coloured point cloud, disparity map, corrected images etc. The camera uses 2 CDD image sensors and can capture images with 2 different resolution and frame rates such as: 640*480 at 48 fps sand 1024*768 at 20 fps.

This device is composed of two cameras with a baseline of 12 cm as can be seen on figure 88 below.

Figure 88[94]: Bumblebee2 camera components [162][95]

[93] This figure is redacted for copyright reasons.

The camera sensors can capture black and white as well as colour images with a resolution of 640*480. It is pre-calibrated against distortion and misalignment and calibration data are stored onboard the system. As the Kinect sensor, it is a horizontal bar and has to be lengthwise in use. Moreover, it is physically compact: 157mm*36mm*47.4mm.

Additional specifications for this camera are summarized in the table 20 below:

Bumblebee2	Specifications
Area covered	NC
Field of view	66.17° horizontally
Frame rate	20 or 48 fps
Minimum distance field of view	50 cm

Table 20: Bumblebee camera specifications provided by Point Grey

The previous sections presented basic characteristics of two famous 3D range cameras: the Microsoft Kinect and the Bumblebee2 camera. The next section explains 3D information have been acquired for both of these device.

5-1-2-3D point cloud acquisition

This section describes how 3D point clouds can be acquired from Kinect sensor and the bumblebee camera.

To start with, the notion of 3D point cloud is explained. A 3D point cloud is composed of a set of points in a 3D space. The acquisition of a 3D point cloud then consists in the recovery of a set of 3D points describing the surface of an object/scene, along with their corresponding coordinates expressed as (x, y, z) in a Cartesian coordinate system. Moreover, it is possible to view the point cloud from different orientations by rotating or translating it, as well as under different lighting conditions. Features such as colours or textures can also be added to 3D point cloud surfaces which can be useful for tasks such as classification, object detection and recognition, scene understanding as well as visualisation purposes. However, it does not correspond to a full 3D model of a scene and objects as it recovers information in the field of view of the sensor. So 3D point cloud data can be referred as 2.5D instead of 3D.

There are wide-ranging methods for allowing the acquisition of 3D data in real-time such as time-of-flight cameras [82] [83], laser triangulation [78] [79], geometric stereo [94] [95], LIDAR [88] [89] [90] Shape from Shading [109], Bumblebee cameras [162], Kinect [156] [159], structured light [100] [101] etc. However only the Microsoft Kinect [156] [159] and the Bumblebee cameras [162] are investigated here. The selection of these two methods is explained in more details in chapter 3.

[94] This figure is redacted for copyright reasons.

[95] Used with permission of the author/publisher.

5-1-2-1-With Kinect sensor

This section presents an example of 3D point cloud obtained with the Microsoft Kinect sensor as can be seen on figure 89 below.

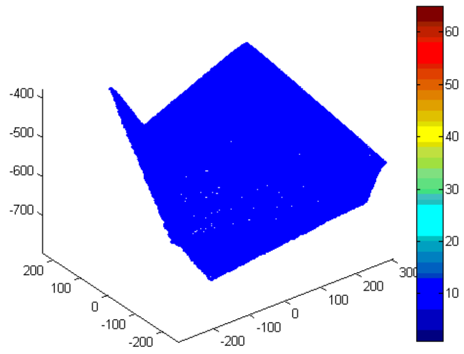


Figure 89: 3D Point cloud acquired with the Microsoft Kinect sensor

Another example of 3D point cloud obtained is shown on figure 90 below.

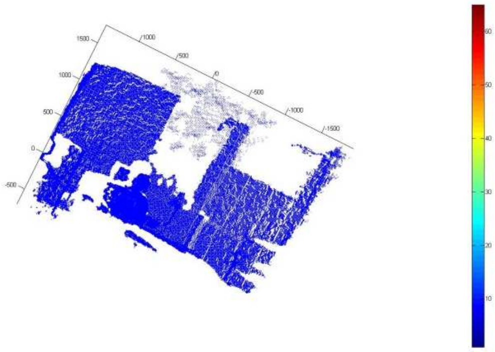


Figure 90: Another example of 3D Point cloud acquired with the Microsoft Kinect sensor

In this section, an example of how depth maps and 3D point clouds can be obtained from the Kinect sensor has been provided. In the next section, an example of how the Bumblebee can acquire depth maps and 3D point clouds is given.

5-1-2-2-With Bumblebee2 camera

This section presents an example of 3D point cloud obtained with the Microsoft Kinect sensor as can be seen on figure 91 below.

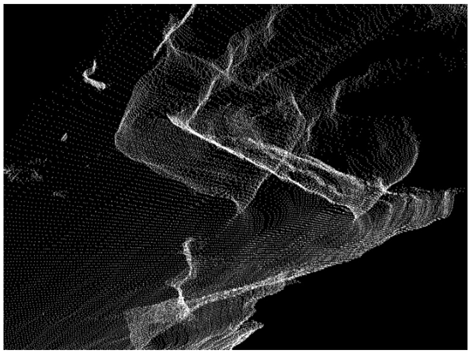


Figure 91: 3D Point cloud acquired with the Bumblebee2 camera

Another example of 3D point cloud but this time with colour information is shown on figure 92 below.

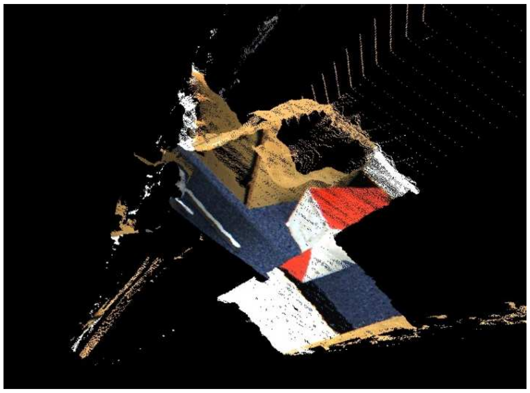


Figure 92: Coloured 3D point cloud acquired with the Bumblebee2 camera

The following section compares the quality of 3D point cloud obtained with both devices.

5-1-3-Discussion

Point clouds obtained from both cameras are high quality but should be used for different applications due to their differences. Indeed, the Kinect sensor is better to be used in a shorter range than the bumblebee camera as mentioned previously which makes sense as it has been designed for video games.

Moreover, both devices have been used to acquire 3D point clouds and coloured 3D point clouds as colour information can be very useful for scene understanding allowing segmentation, classification, detection and object recognition. The results obtained were different than those obtained without colour information as can be seen on figure 89 and 90 for the Kinect and on figure 91 and 92 for the Bumblebee2 camera.

As can be seen on figure 90, Kinect sensor has difficulties to distinguish surfaces with different depth when colour is added as it creates a junction between two surfaces that should not be connected as they do not belong to the same plane. Apart from this issue, certainly due to its range limitation and the addition of colour information, the resulting point cloud is dense but remains better without colour. Moreover, even if in presence of colours it does not render the 3D geometry the way it is realistically, it provides realistic colour of 3D objects present in the considered scene.

On the contrary, as can be seen on figure 92, coloured 3D point clouds obtained with the Bumblebee camera do not have depth issue but have difficulty to retrieve the whole 3D geometry of objects. Indeed, figure 92 represents the 3D point clouds of a cube but it appears clearly that the Bumblebee2 struggled to recover all faces of the cube accurately as it does not appear as a cube due to the lack of several 3D points. According to a conversation with Point grey, this can be due to the lack of textural information and can be problematic for some applications. Point cloud without colour.

Furthermore, additional experiments are required to verify the exactitude of this information provided by point grey. One method would be to capture the cube again by adding different texture on several faces and keeping one without any texture. This would allow comparing the quality of the 3D points obtained depending on the type of texture and in absence of texture to verify if it really comes from this reason. However, due to better results obtained with the Kinect sensor, this issue will not be investigated further but it does not mean the Bumblebee2 is not efficient but only that it is not the most adapted for this study case compared to the Kinect which in our opinion is better adapted to realise preliminary tests for plane extraction from 3D point clouds.

To summarize, the Kinect sensor is very fast, very cheap and its performance is already well established so further investigation of its application for plane extraction are described in the next section. However, it is limited by its range and will only be useful to start preliminary experiments and observations to process plane extraction from 3D point clouds. Indeed, it would not be ideal in the case of video surveillance as range can be quite important in some circumstances; but a solution to address this issue will be proposed in chapter 6.

This section compared qualitatively the benefits and limitations of each device investigated. The following one describes preliminary and quantitative experiments on how planes can be extracted from 3D point clouds acquired with the Kinect sensor.

5-2- plane extraction from 3D point clouds

In the previous chapter, feature extraction and image segmentation based on 2D data have been investigated and methods have been proposed as initial steps towards 3D scene understanding. In this section, feature extraction based on 3D data is explored. The goal of this investigation is to show how the combination of 2D and 3D features can achieve an approximate 3D model of a scene, classification, object detection or recognition or scene understanding tasks.

As there is no need to acquire complete 3D of the scene for this study case, the focus has been mainly on methods that only allow recovering partial information of objects and the scene that only corresponds to what is in the field of view of the camera. As explained previously, the key characteristics of man-made and video surveillance environment is the omnipresence of cubic objects and geometry which is what motivated the investigation on plane extraction from 3D point cloud. Indeed, once planar surfaces have been extracted it can be combined with VPs to refine VP detection as 3D data are richer than 2D information. Thus, the use of colour segmentation can be performed to then assist for scene understanding by clustering the scene or objects into the categories safe and unsafe areas or objects to detect to prevent from crimes.

The following section evaluates and describes common methods used to extract plane from 3D point clouds.

5-2-1-Common techniques for plane extraction

Among the literature, the most common techniques [163] to extract plane from 3D data are: the SHT [164], RANSAC algorithm [165] and the Region growing method [165].

All variations of SHT, standard technique used to extract line from 2D data, can be extended to 3D data to extract planes: the standard HT (SHT) [164], the probabilistic HT (PHT) [147], the adaptive probabilistic HT (APHT), the progressive probabilistic HT (PPHT) [148] and the randomized HT (RHT) [166]. However, according to [167] one method is more promising than others for this task.

They all have in common the transformation of a point cloud in the Hough space to detect planes by counting the number of points that lie on one plane represented by one cell in an accumulator. According to [167], advantages of the SHT are its completeness and deterministic characteristics as it is performed for all points. Convergence criterion is also considered as an advantage. However, these rules differ from a technique to another. The RHT, for example, uses a threshold corresponding to the number of points left in the point cloud or the number of planes detected whereas the PPHT is based on the number of points already processed which makes it less sensitive to noise. While the APHT has the most complex rule using a maximum monitored over time during the voting scheme to improve robustness of the algorithm. However, as a consequence, it makes it difficult to implement compared to other techniques. Another common advantage for the PPHT and the RHT is that once planes are detected, they are removed from the point set which allows the algorithm to speed up and false detection to decrease. The only advantage shared by SHT, PHT, PPHT and RHT is an easy implementation. Finally, only the RHT accesses one cell of the accumulator per iteration. Their evaluation shows that RHT is the method to adopt when dealing with 3D data due to its very high computational performance and the quality of the result obtained, due to random selection of points and removal of data once they have been attributed to a plane increasing the accuracy of the next plane detection.

Furthermore, another important aspect of HT is the choice of accumulator design. In [167], they described several types of accumulator such as the accumulator array [143], the accumulator cube [168] and a new design they proposed as the accumulator ball [167] which does not favour planes with specific parameters due to the equal size of patches. Their comparison shows that the ball accumulator is best to detect arbitrary planes and slightly outperforms the cube accumulator even if it has issues to detect planes parallel to the xy-plane; whereas the array accumulator struggles to correctly identify all faces of a cube and fails when it is aligned to the coordinate system. They then conclude differences remain negligible as planes parameters are computed equally with each accumulator design.

They also compared performances of the RHT with a method proposed by Poppinga et al. in 2008 [165] and with a method proposed by Attene et al. in 2006 [163] for plane extraction. On one hand, the second method is based on region growing and incremental plane fitting methods followed by a polygonization. They proceed by randomly selecting a point and its nearest neighbour from the point cloud. Then, for each iteration, they add points to the input set that are the closest to the optimal plane through points from this region when the mean square error of the optimal plane is inferior to a threshold. They tend to ignore regions with only a few points. On another hand, the third method is the HFP method based on triangular mesh segmentation by fitting primitives. They first consider each triangle as a cluster and assign it to a primitive. The cluster then becomes bigger in the direction best represented by the primitive. The closest sets of triangles are clustered if the resulting can be approximated by a primitive. In [166], their experiments demonstrated that RHT outperforms RG and HFP methods for plane extraction as the other two accurately detect planes that are rather smaller.

Tarsha-Kurdi et al. [169] compared the performance of HT to those of RANSAC algorithm to extract planes from 3D data. Their experiments demonstrated that RANSAC provides faster and better quality results than HT with 70% of successful detected planes. So, they concluded RANSAC is the best technique to use to extract planes from 3D data acquired with LIDAR compared to HT and Region Growing that do not give as accurate performances and results.

According to the evaluation presented above, only the RANSAC algorithm has been further investigated to extract planes from 3D point clouds as it provides better results than HT and RG.

The RANSAC algorithm is then described in the next section. More details about how the HT detects lines from 2D data are described in chapter 4 section 4-1-1-2-Standard Hough Transform and details about RG methods are summarised in chapter 3 section 3-2-Segmentation.

5-2-2-Theoretical background: RANSAC Algorithm

The RANSAC algorithm was first introduced by Fischler and Bolles [165] in 1981 to estimate parameters of a model starting from a set of data composed of numerous outliers. The RANSAC algorithm has been chosen for the proposed plane extraction approach due to very accurate performances even in presence of noise as it allows removing it. This is then very convenient as the data used in this study case are very noisy. Moreover, it has been proven to successfully detect planes in 2D as well as in 3D as explained recently in a book by McGlone et al. in 2004 and in a paper by Nguyen et al. in 2005, where it is compared to five other algorithms to extract lines. RANSAC was first used for line fitting problems in 2D but later on it has been applied to extract 3D features such as in [170] (by Bretar and Roux in 2005).

The main goal of RANSAC is to determine a set of parameters of a model from a set of data in presence of many outliers. RANSAC algorithm is very simple and follows steps described below:

1) Random selection of N points (minimum number of points required to determine the model) from a set of M points, in the case of plane extraction N=3;
2) Estimation of model parameters from the N selected points;
3) Determination of the number of points from the input set that fits the model parameters estimated within a tolerance k;
4) If the number of inliers divided by the total number of points exceeds$k$: Re-estimation of model parameters using all identified inliers and exit.
5) Otherwise, repeat steps 1 to 4 a maximum of L times;

L can be determined using the following formula:

$$\boldsymbol{L} = \frac{\log(\boldsymbol{P_{fail}})}{\log(1-\boldsymbol{P_g}^{\boldsymbol{N}})}, \quad (87)$$

where $P_g$is the probability of a randomly selected data to fit the model and P_{fail} the probability of the algorithm to exit without finding a good fit for the model?

In presence of multiple structures and after a successful fit, the fit data are removed and the algorithm is again applied.

An advantage of RANSAC is its robustness to estimate model parameters, i.e., it can estimate parameters with a high degree of accuracy even in presence of a significant number of outliers. This strength represents a key characteristic required to be applied on video surveillance scene due to the presence of noise and changes in exterior conditions. However, a limitation of this algorithm is that there is no boundary concerning the time it takes to estimate parameters. Moreover, if the number of iterations computed is limited the solution obtained might not be optimal, and it may not even be one that correctly fits data. So, RANSAC offers a trade-off as a greater number of iterations increases the probability of a reasonable model to be estimated. Another disadvantage is RANSAC can only estimate one model for a particular data set which implies it can even fail to find either one. Even if RANSAC has limitations, its strength is sufficient to be applied to video surveillance scene as so far what has limited other methods investigated where the complexity of the scene, the presence of noise and change in exterior conditions.

The next section describes the proposed approach based on the RANSAC algorithm to extract planes from 3D point cloud and justify.

5-2-3-Proposed approach

This section describes every step of the proposed approach to extract planes from 3D data.

An open source code developed by Peter Kovesi [171] has been used as a starting point for plane extraction based on the RANSAC algorithm. This source code has then been combined with an open source code provided with the Kinect sensor to acquire 3D point clouds. Then, several experiments have been conducted to first extract one plane, then two, three and finally more. These experiments have been performed in the centre for Machine vision at the University of the West of England from 3D point clouds acquired with the Microsoft Kinect sensor.

Preliminary tests performed with Kovesi's source code have shown that it could determine plane parameters but could not recover the set of points that belonged to the whole plane as it could not detect the plane boundaries. So, this code was then used to determine plane parameters and was then modified to extract the set of points that belong to the corresponding infinite plane detected.

Here, a plane is defined by the following equation:

$$\boldsymbol{\mathcal{P}}: \boldsymbol{ax + by + cz + d = 0} \quad (88)$$

where a, b, c and d are the coefficients that defines the plane $\mathcal{P}$.

This has been made possible following the steps below:

1. Input set: a set of 3D points $\{\mathcal{I}\}$ from the 3D point cloud acquired with the Microsoft Kinect sensor used to extract of k planes.
2. Estimation of the parameters of the k^{th}plane $\mathcal{P}_k$ with Kovesi's method from $\{\mathcal{I}\}$.
3. For the k^{th} estimated plane $\mathcal{P}_k = (a_k, b_k, c_k, d_k)$:
    a. Determination of a set of points $\{i\}$ that belongs to the estimated plane such as $r_{kj} < r_t$ where $r_{ki} = a_k x_i + b_k y_i + c_k z_i + d_k$.
    b. Once a set of point that belong to the k^{th} estimated plane has been found:
        i. it is added to a cluster $\{\mathcal{C}_k\}$.
        ii. And removed from the input set $\{\mathcal{I}\}$.
4. Repeat steps 2 and 3 for $(k+1)^{th}$ plane $\mathcal{P}_{k+1}$ from the new input set $\{\overline{\mathcal{I}_k}\}$ and so on.

Here:

- r_{ki} Gives a measure of closeness of a point to a plane and then allows determining if this point belongs to the considered plane.
- $r_t = 0.02$ And has been determined empirically in the case where the metric used performed best. Smaller r_t is, better the plane detection is supposed to be as it represents a similarity metric. This parameter has been chosen since it provided better results and is easier to use as it is based on the equation of a plane, compared to the formula to calculate the distance from a point to a plane.
- $\{\mathcal{C}_k\}$ is a cluster composed of points from the set $\{i\}$ that belong to the k^{th} estimated plane.
- $\{\overline{\mathcal{I}_k}\} = \{\mathcal{I}\} - \{\mathcal{C}_k\}$, and represents the new input set used to estimate the $(k+1)^{th}$ plane, once the points from the cluster $\{\mathcal{C}_k\}$ have been removed from the original input set $\{\mathcal{I}\}$.

This approach has then been tested on various simple and more complex simulated scenes to first extract three then more planes.

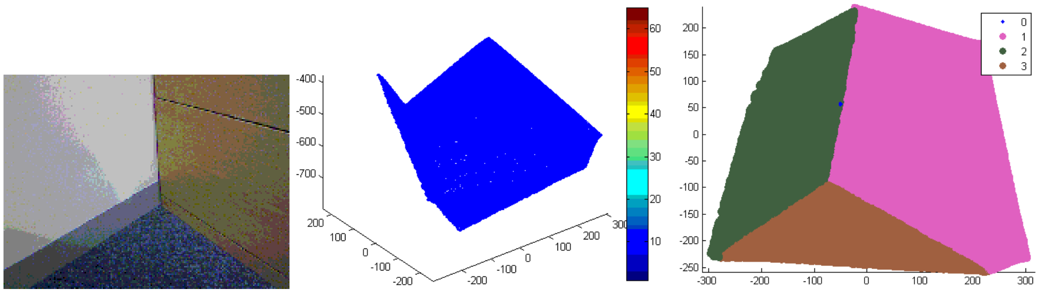
For visualisation and differentiation purposes, a random and different colour has been attributed to each estimated plane on top of the original 3D point cloud.

This section described the proposed approach to extract planes from 3D data based on the RANSAC algorithm and the definition of a plane. The following section presents and analyses results obtained from different experimentations.

5-2-4-Results

This section presents results obtained with the proposed plane extraction method from: the 3D point cloud of a corner the 3D point cloud of a pile of books and a simulated railway station.

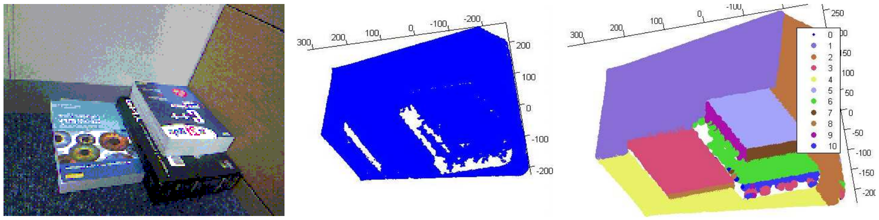
- Extraction of maximum three planes with the example of a corner



(a) (b) (c)

Figure 93 : Result of the proposed plane extraction approach on a simple example of a corner : (a) Original frame acquired from the Kinect sensor, (b) Point cloud acquired from the Kinect sensor, (c) Results of the proposed plane extraction method

- Extraction of several planes with the example of a pile of books



(a) (b) (c)

Figure 94: Result of the proposed plane extraction tool on a more complex example, a pile of books in the corner : (a) Original frame acquired from the Kinect sensor, (b) Point cloud acquired from the Kinect sensor, (c) Results of the proposed plane extraction method

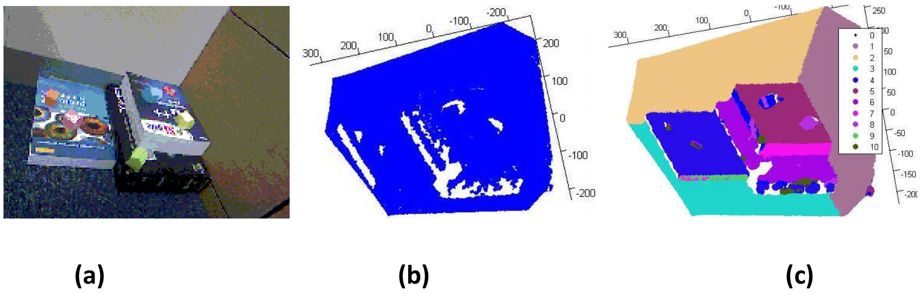


Figure 95: Result of the proposed plane extraction tool on a more complex example, a pile of books in the corner with small items on tops of the books with the shape of small houses : (a) Original frame acquired from the Kinect sensor, (b) Point cloud acquired from the Kinect sensor, (c) Results of the proposed plane extraction method

- Extraction of several planes from a more realistic environment with the example of the simulated platform

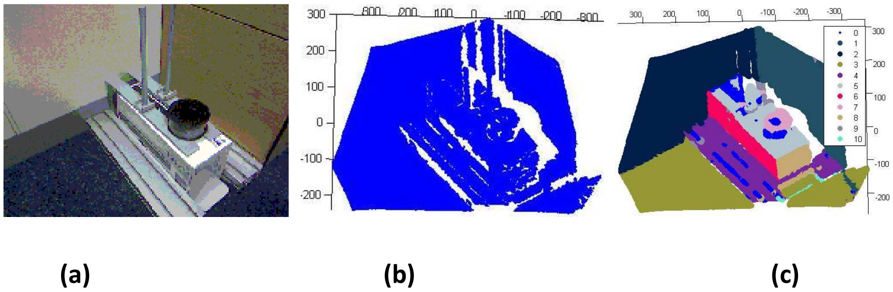


Figure 96: Result of the proposed plane extraction tool on a more complex example, a simulated platform with a circular bench, track and two poles : (a) Original frame acquired from the Kinect sensor, (b) Point cloud acquired from the Kinect sensor, (c) Results of the proposed plane extraction method

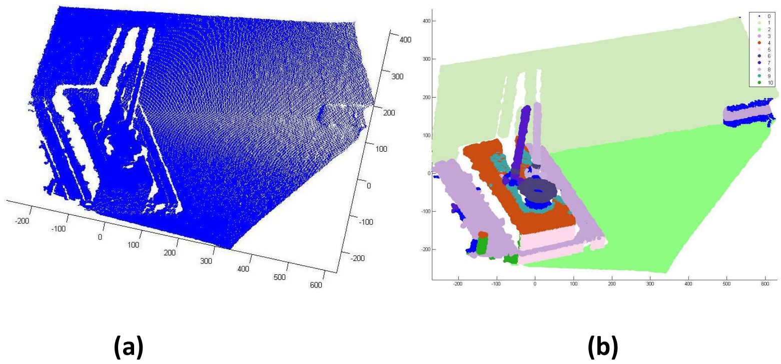


Figure 97: Result of the proposed plane extraction tool on a more complex example, a simulated platform with a circular bench, track and two poles from another viewpoint : (a) Point cloud acquired from the Kinect sensor, (b) Results of the proposed plane extraction method

The results obtained with a corner are very accurate as the proposed approach succeeded to recover the three entire planes without a missing point as can be seen on figure 93. The results obtained with a more complex scene such as a pile of books are not as accurate as those obtained with the corner as can be seen on figure 94, as there are areas with missing points. However, it did not affect the method for the plane extraction part as it detected each plane apart from small planes that are perpendicular to the floor where point are missing certainly due to lighting conditions and the sensor orientation.

To make the scene more complex and tests the efficiency and robustness of the algorithm, experiments have been realised with the same pile of books in addition of small objects with the shape of a house on top of them. As can be seen on figure 95, the addition of these objects did not affect the plane extraction process as the same number of planes have been extracted compared to the previous experiment. However, it shows the difficulty of the algorithm to detect planes from small objects. This can be explained as the Microsoft Kinect sensor was not designed to be used in small environments such as toys but rather objects of human size.

The last experiment performed, was realised with a simulated railway station using tools and boxes from the laboratory. As can be seen on figures 96 and 97, the results have lost in accuracy compared to the other experiments realised. However, between figure 96 and 97, the number and accuracy of planes extracted remain constant even if the orientation of the sensor is different and even circular planes have been detected. So, proposed approach has potential for scene understanding and has be proven to be more efficient due to the use of 3D data compared to previous methods proposed based on 2D data. However, this method needs more experiments on more realistic and human size scenes which would then be difficult to apply to real video surveillance scenes as it would require the installation of an additional device to acquire 3D point clouds from real scene. This represents a really high cost and would require shutting down the station as it is not possible in presence of people. So for the reasons mentioned above, an alternative has been found to make use of 3D information based on a system that can be more easily tested and installed in such environments. This system is described in the next chapter.

In the next section, potential solutions to improve the proposed plane extraction method are proposed.

5-2-4-3-Improvement proposed

The previous section has described a new plane extraction method that provides reasonable and accurate results. However, it is difficult to put into place in real video surveillance scenes and lacks information about objects’ boundaries as in some areas; 3D points were missing or sometimes merged with a plane that did not belong to the object where they originated.

The detection of boundaries would be a useful improvement for plane extraction as it would allow differentiating an object from another and even assist to detect smaller objects as they are easy to distinguish from bigger objects due to their size and can sometimes be of interest for 3D scene understanding.

A suggested solution to address these issues can be to apply pre-processing to improve the quality of 3D point clouds acquired as they are known to be noisy and non-uniform depending on the type of environment observed and to look for planes' intersections.

In [169], the authors compared the RANSAC algorithm with the HT for the automatic detection of 3D planes and proposed improvements and extensions to the RANSAC method. This could be a method to investigate as a future work. Another method to explore could to use lines and VPs extracted and colour information to segment the scene and detect boundaries between objects to differentiate them. As preliminary experiments, the method proposed in [175] could be investigated.

As mentioned previously, the proposed plane extraction method would represent an incommensurable cost and time to be able to extract 3D data from a real railway or underground station. For this, the application of VP detection, colour segmentation and plane extraction to real railway and underground station scenes is described in the next chapter. However, first, a global conclusion as for the quality of the three proposed approaches is given in the next section.

5-3- Summary on 2D and 3D feature extraction

The achievements of the last two chapters concern the extraction of 2D feature such as lines, VPs and colour information as well as 3D feature such as planar surfaces. A subjective summary (extensive numerical analyses have already been presented) of these methods is illustrated by the following table 21.

	Computation	Accuracy	Robustness to change in exterior conditions	Robustness to noise	Practicality
Line detection	Inexpensive	Sensitive to noise, improved by image enhancement.	Lack of robustness for dark places and reflected light, ok otherwise	Robust	No specific requirements
Intersection computation	Expensive	Accurate	Depends on the robustness of the line detection process	Depends on the robustness of the line detection process	Requires to test all combination
VP detection	Average cost	Depends on line detection stage	Depends on the robustness of the line detection process	Average	Do not deal with infinite VPs
Colour segmentation	Inexpensive	Sensitive to noise and exterior conditions	Inversely proportional to the complexity of the scene	Inversely proportional to the complexity of the scene	Too strict, number of clusters required as input
Plane extraction	Inexpensive	Better accuracy if quality of 3D data improved	Robust	Average	No specific requirements

Table 21 : Comparison of proposed approaches to be used as a future product. NB. Practicality refers to the ease by which the methods can be commercially deployed.

This table evaluates strengths and limitations the proposed VP detection, segmentation and plane extraction method. Please note however, that these are relative and subjective only to give an idea as to where they are considered to be close to commercial use ("Good") and still fall far short of requirements ("Bad"). Moreover, 3D plane extraction methods lacked of applications to realistic settings – this is due to practical reasons and limitations of the two 3D cameras used (something that will be addressed in the next chapter).

To conclude on 2D feature extraction, segmentation and 3D plane extraction, these methods all have potential to assist for scene understanding. However, they appear to be insufficient as a whole to overcome video surveillance challenges due to noise and exterior conditions, especially as they only would allow extracting part of objects which would make the object "recognition" and detection, classification and event detection tasks difficult. Considering the limitations of the above, the next chapter presents a novel method that allows the extraction of 3D data under a more controlled environment as the only reason that seems to stop the previous methods to have greater performance was exterior conditions.

The proposed solution to address some of the main video surveillance challenges is based on photometric-stereo where the use of light sources in addition to the lighting environment allows a better control of change in lighting and then light reflection, shadows etc within the scene. The next chapter shows how this new method can contribute to the research community as well as industries to improve crime prevention and help to prosecute criminals making video surveillance systems more accurate and intelligent.

Chapter 6: Photometric-stereo, a new approach for object classification

This chapter investigates a new application for photometric-stereo: the use of controlled light sources for object “recognition” from shape information applied to video surveillance scenes.

As explained in previous chapters, the lack of information from 2D data and the poor quality of 3D point clouds acquired by the Kinect and Bumblebee cameras showed that there is a need for research to focus on rendering of 3D data to assist with 3D scene understanding, especially when applied to real-world scenes.

3D Scene understanding is one of the most difficult and complex task in computer vision as it requires high quality data to obtain a useful and significant understanding of 2D and depth information to achieve scene interpretation.

However, the considered study case do not require as much precision but only information about the shape of an object to be able to “recognise” it based on its geometry. In this case, 2.5D data provided by photometric stereo method as surfaces’ normals were used as it is not feasible to recover full 3D geometry from a single view (combining viewpoints could be an avenue for future work).

As mentioned previously, applications such as archaeology, architecture, medical imaging etc. require the complete recovery of 3D data to enable reconstruction of the full geometry of an object of interest. However, in the context of railway and underground stations, such a precision is not required to determine objects’ shape. Therefore, 2.5D data provided by photometric stereo in the form of surfaces’ normals will be sufficient for the considered application.

Moreover, this method seems well-suited for video surveillance applications for the following reasons:

- it allows projecting the light wherever it is required;
- it can either be applied to static or dynamic objects;
- it does not require too demanding changes of the existing surveillance installations;
- it addresses the main challenges encountered in video surveillance environments namely changes of geometry, i.e. non-flat surfaces, in the scene dynamic, i.e. occlusions due to human motion and in exterior conditions such as lighting, shadows, and to some extent weather.

To summarize, this chapter describes an innovative proposed approach to address some of the main video surveillance challenges as well as 3D scene understanding by using photometric-stereo to extract shape information about an object of interest and infer the type of object it is (e.g. sharp and dangerous or blunt and benign) from its geometry. The next section gives an overview of the photometric-stereo’s principle and describes the proposed approach to recognise shape of objects from surfaces’ normals’ variation.

6-1-Surface normal retrieval

This section first gives a description of photometric-stereo theoretical aspect. Then, the proposed approach is described and results are presented and discussed.

6-1-1- Theory

Photometric stereo (PS) was first introduced by R. Woodham [127] in 1979. This technique allows estimation of objects' surfaces' normals by observing them under different lighting conditions. This method is a specific case of the shape from shading method which differs by the use of more than a single image to exploit shading information. Shape from shading was analyzed by B. K. P. Horn in 1970 [110].

PS is based on the variation of the direction of incident illumination between successive views while the viewing direction remains constant. This implies that each pixel in three images obtained by PS corresponds to the same point of the same object and has the same gradient. This is due to the fact that imaging geometry has not changed whereas the reflectance has, due to variation of the direction of the incident illumination.

Most papers that use photometric stereo rely on three or more sources. Three non-planar sources is the minimum required to recover all three components of the surface normal for a Lambertian surface, while further sources can add robustness or deal with non-Lambertian behaviour. Hansen et al. [172], for example, pose the general operation of photometric stereo P as:

$$\{\mathbf{N}_i\} = \boldsymbol{P}\left(\{\boldsymbol{I}_{1,i}\},\{\boldsymbol{I}_{2,i}\},\{\boldsymbol{I}_{3,i}\},\ldots,\mathbf{L}_1,\mathbf{L}_2,\mathbf{L}_3,\ldots\right) \tag{89}$$

where $\{\mathrm{N}_i; i = 1,2,\ldots M\}$ is the set of unit surface normals for the M-pixel image, $\{I_{k,i}; i = 1,2,\ldots M\}$ is the set of pixel intensities for image k, and L_l is the lth light source vector. For standard three-source PS at pixel i, this can be expanded to [1]:

$$\boldsymbol{\rho_i}\begin{bmatrix}\boldsymbol{N}_{x,i}\\ \boldsymbol{N}_{y,i}\\ \boldsymbol{N}_{z,i}\end{bmatrix} = \begin{bmatrix}\boldsymbol{L}_{1,x} & \boldsymbol{L}_{1,y} & \boldsymbol{L}_{1,z}\\ \boldsymbol{L}_{2,x} & \boldsymbol{L}_{2,y} & \boldsymbol{L}_{2,z}\\ \boldsymbol{L}_{3,x} & \boldsymbol{L}_{3,y} & \boldsymbol{L}_{3,z}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{I}_{1,i}\\ \boldsymbol{I}_{2,i}\\ \boldsymbol{I}_{3,i}\end{bmatrix} \tag{90}$$

where ρ_i is the albedo of the ith pixel, $N_{x,i}$, $N_{y,i}$ and $N_{z,i}$ denote the x, y and z components of the ith surface normal, and $L_{l,x}$, $L_{l,y}$ and $L_{l,z}$ denote the x, y and z components of the lth light source vector. For this thesis, x is defined as a horizontal direction that is parallel to the image plane, y as vertical and z directed towards the camera. This equation can be solved assuming that the light source vectors are known (ρ_i, $N_{x,i}$, $N_{y,i}$ and $N_{z,i}$ are unknown, but $N_{z,i}$ can be calculated from $N_{x,i}$ and $N_{y,i}$ since N_i is a unit vector – hence there are three independent unknowns for a system of three simultaneous equations).

For the considered case, it is necessary to calculate as much of the surface normal as possible using two light sources, which is more practical for real world environments. Therefore:

$$\boldsymbol{\rho}\begin{bmatrix}\boldsymbol{N_x}\\ \boldsymbol{N_z}\end{bmatrix}=\begin{bmatrix}\boldsymbol{L_{1,x}} & \boldsymbol{L_{1,y}}\\ \boldsymbol{L_{2,x}} & \boldsymbol{L_{2,y}}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{I_1}\\ \boldsymbol{I_2}\end{bmatrix}=\begin{bmatrix}\boldsymbol{m_x}\\ \boldsymbol{m_z}\end{bmatrix} \tag{91}$$

where the subscript i has been dropped for the sake of brevity and defined $m_{x,i}=\rho_i N_{x,i}$ and $m_{z,i}=\rho_i N_{z,i}$.

As ρ is unknown, a *non-unit* normal is defined:

$$\boldsymbol{p}=\begin{bmatrix}\boldsymbol{p_x}\\ \boldsymbol{1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{m_x/m_z}\\ \boldsymbol{1}\end{bmatrix} \tag{92}$$

m_x and m_z can be calculated by:

$$\boldsymbol{m_x}=\boldsymbol{L'_{1,1}I_1}+\boldsymbol{L'_{1,2}I_2} \tag{93}$$

$$\boldsymbol{m_z}=\boldsymbol{L'_{2,1}I_1}+\boldsymbol{L'_{2,2}I_2} \tag{94}$$

where

$$\begin{bmatrix}\boldsymbol{L_{1,1}} & \boldsymbol{L_{1,2}}\\ \boldsymbol{L_{2,1}} & \boldsymbol{L_{2,2}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{L_{1,x}} & \boldsymbol{L_{2,x}}\\ \boldsymbol{L_{1,y}} & \boldsymbol{L_{2,y}}\end{bmatrix}^{-1} \tag{95}$$

Therefore

$$\boldsymbol{p_x}=\frac{\boldsymbol{L'_{1,1}I_1+L'_{1,2}I_2}}{\boldsymbol{L'_{2,1}I_1+L'_{2,2}I_2}} \tag{96}$$

Using the Pythagoras Theorem,

$$|\boldsymbol{p}|=\sqrt{\boldsymbol{p_x^2}+\boldsymbol{1}} \tag{97}$$

Hence, the final surface normal is

$$\mathbf{N}=\begin{bmatrix}\boldsymbol{p_x}\\ \boldsymbol{1}\end{bmatrix}\Big/\sqrt{\boldsymbol{p_x^2}+\boldsymbol{1}} \tag{98}$$

This section described the origin and the principle of photometric-stereo using either three or two light-sources. The next section provides details about the proposed method to recognise object's shape to infer their function and functionalities.

6-1-2- Method

The photometric stereo system used for this research work is based on the illumination of a target from two different directions as can be seen on figure 98 below.

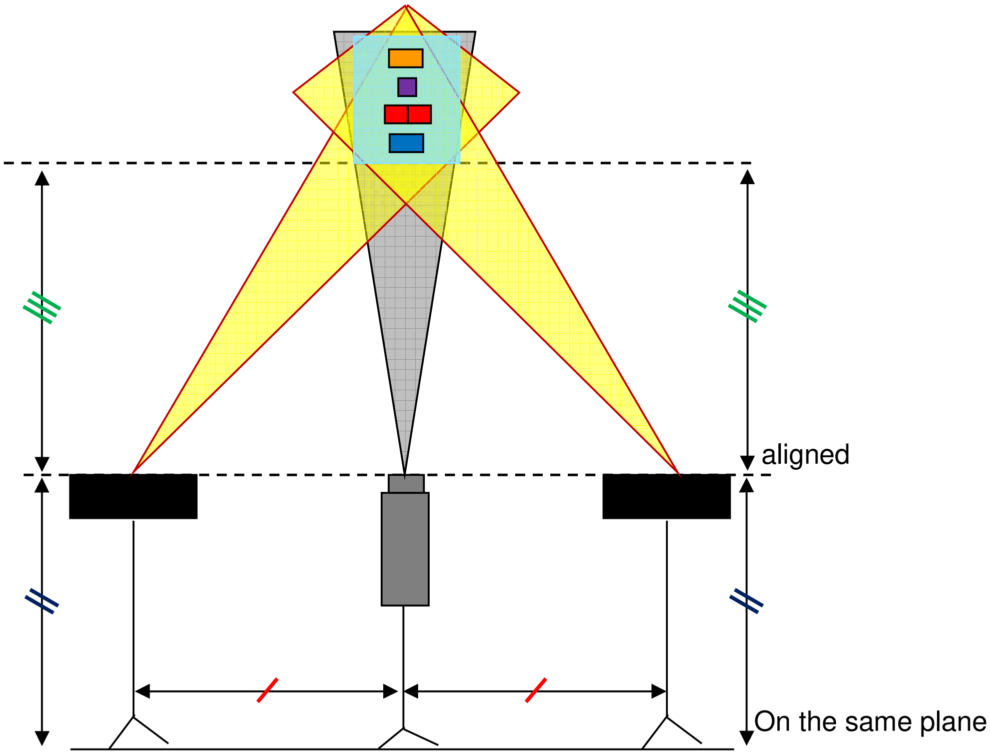


Figure 98: PS system used to recover surface normal.

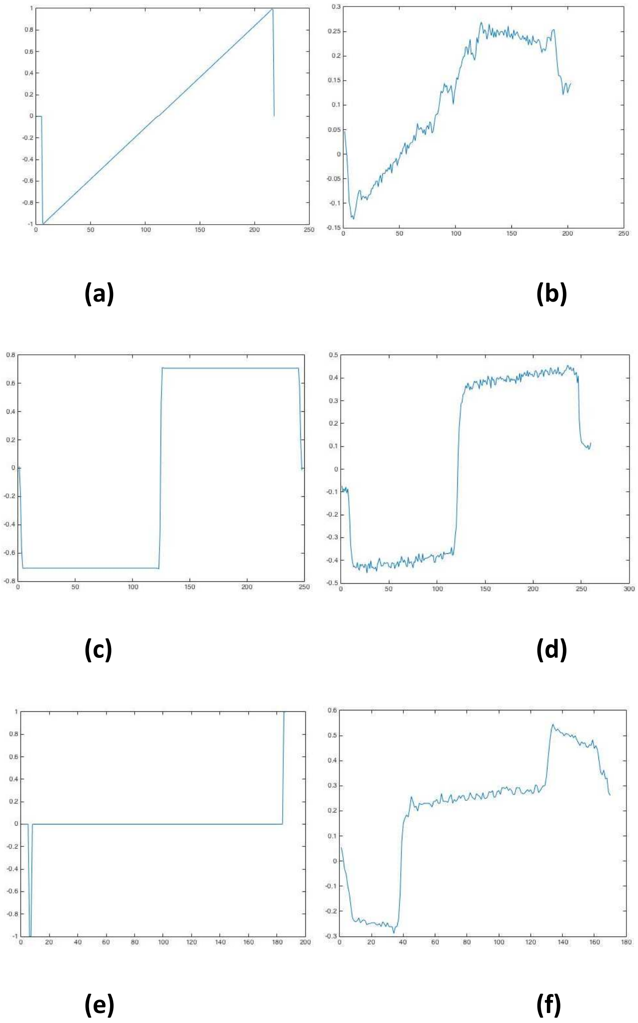
The goal of this system is to capture a scene, containing one or several objects, under two different lighting conditions to recover the surfaces' normals and analyse their variation along the surface of one or several object(s) of interest. Each type of object is characterized by a specific geometry and as the surfaces' normals are related to geometry, their variation along the surface of an object depends on the object shape. Once the image of a scene under two different lighting condition has been captured, the surfaces' normals of each point of the imaged scene are recovered using the equations presented in the previous section.

Preliminary experiments have first been performed using three simple shapes: a half-cylinder, a prism and a cuboid placed on a flat support such as a cardboard. For each of these shapes, the variation of the surfaces' normals along the surface has been analysed. Moreover, each of these shapes has been simulated using Matlab and referenced as GT. The choice of these shapes is designed to illustrate the potential functionalities. For example, the cuboid contains a planar surface that might be a knife, while a half-cylinder is unlikely to be a dangerous object. An example of this case can be found in section 6-1-3-Results for more realistic experiment. Based on a standard 2D image, it is difficult to determine whether the item held by the man is a weapon (e.g. is the item a knife or an umbrella?). However, a 2.5D analysis would show whether the item is rounded (umbrella) or straight (knife).

GT shapes and their respective surfaces' normals have been simulated and recovered as follow:

- the half-cylinder has been simulated as a half-circle repeated in space from which surface s' normals have been recovered.
- the prism has been simulated as two 45 degrees planes with the two opposite orientation repeated in space from which surfaces' normals have been recovered.
- As the cuboid has flat surfaces, the surface s' normals have been simulated directly by a plane with only value 0 for N_x and 1 for N_z.

Then, the variation of the surfaces' normals along the surface of each GT shape was analysed. From the results of these experiments, it was noticeable that the surfaces' normals obtained from GT shapes as well as real shapes, were varying with a specific pattern to the shape analysed. An example of this observation is illustration on figure 99 below.



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 99: (a) Variation of N_x along the surface of the GT half-cylinder, (b) Variation of N_x along the surface of the real half-cylinder, (c) Variation of N_x along the surface of the GT prism, (d) Variation of N_x along the surface of the real prism, (e) Variation of N_x along the surface of the GT cuboid, (f) Variation of N_x along the surface of the real cuboid.

As can be seen on figure 99 above, the surfaces' normals of a half-cylinder are characterised by one period of a sawtooth function, the surfaces' normals of a prism are characterised by a step function and the surfaces' normals of a cuboid are characterised by a inverted followed by an upright Dirac function.

This pattern was then used as a feature to identify an object shape and make the distinction between distinct objects from the recovery of their surfaces' normals. A bench for example tends to be cubic more than circular so it can be differentiated from any other objects that are circular within the same scene. Moreover, in the case of another object with the same shape such as a shelter or a bin, additional features such colour, texture and size are required to make the distinction but it is possible thanks to colour segmentation, camera calibration using VPs to determine the size of objects etc. Furthermore, it is unlikely within a specific environment to detect an object with the same shape, colour, size and texture. Indeed, railway and underground stations like most man-made environments tend to only be composed of a few objects such as benches, bins and shelters, which are usually similar from one station to another. In such environments, shelters are usually bigger than benches, themselves usually bigger than bins so the knowledge of their dimensions in addition of their surfaces' normals can be sufficient to make a distinction between them.

VP detection, colour segmentation and plane extraction take on their full meaning as they are required to assist the proposed method for object recognition and scene understanding.

The proposed approach to recognise objects' shape follows steps illustrated by the diagram on figure 100 below.

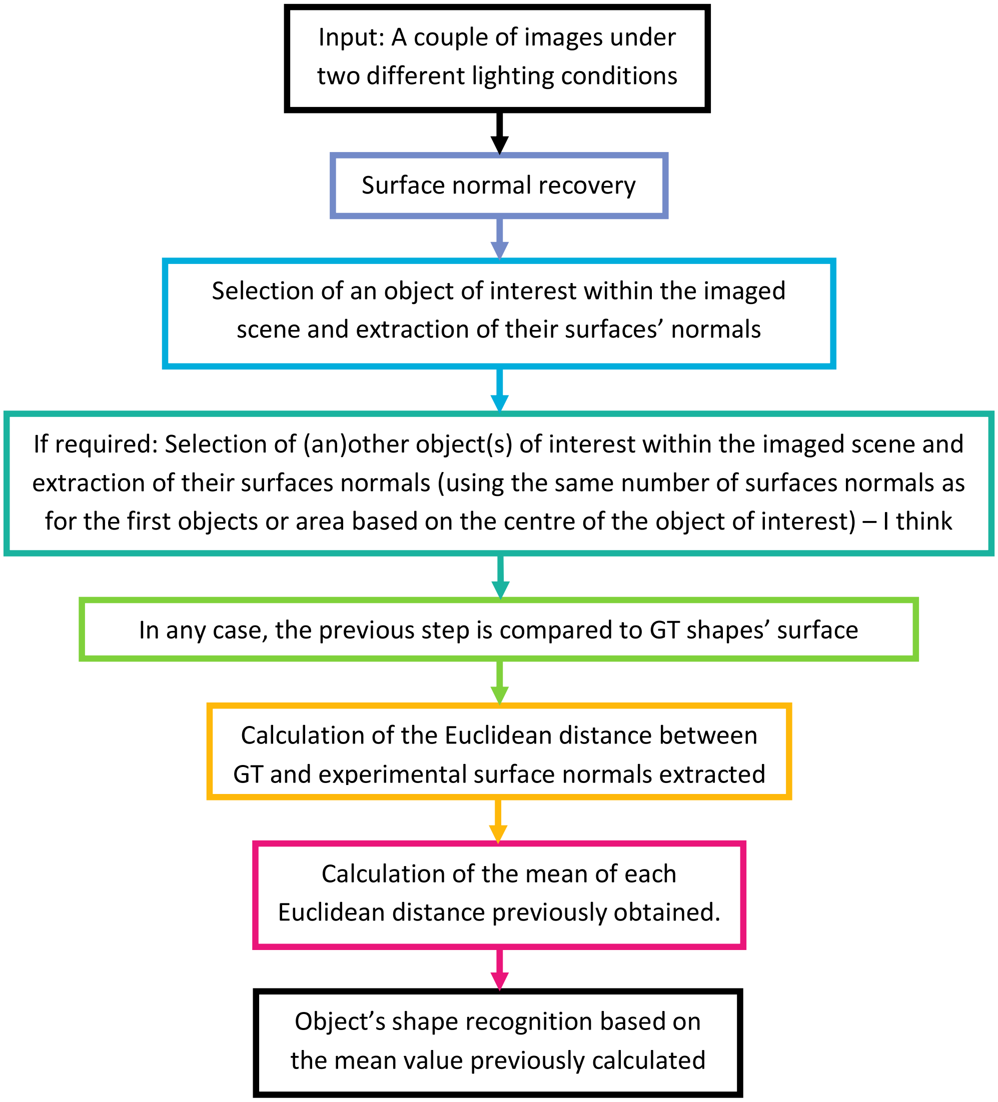
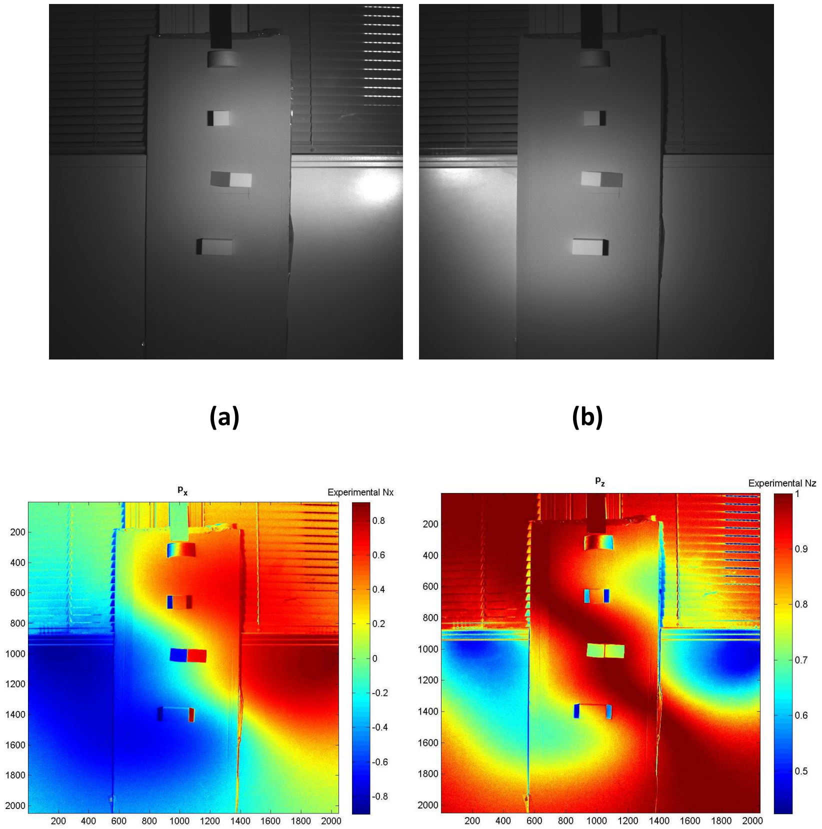

Input: A couple of images under two different lighting conditions
Surface normal recovery
Selection of an object of interest within the imaged scene and extraction of their surfaces' normals
If required: Selection of (an)other object(s) of interest within the imaged scene and extraction of their surfaces normals (using the same number of surfaces normals as for the first objects or area based on the centre of the object of interest) – I think
In any case, the previous step is compared to GT shapes' surface
Calculation of the Euclidean distance between GT and experimental surface normals extracted
Calculation of the mean of each Euclidean distance previously obtained.
Object's shape recognition based on the mean value previously calculated

Figure 100: Algorithm overview.

A description of each step of this algorithm is provided below. First, images of the scene under two different lighting conditions are captured from which surfaces' normals are recovered. An example of this step can be seen on figure 101 below.


(a)

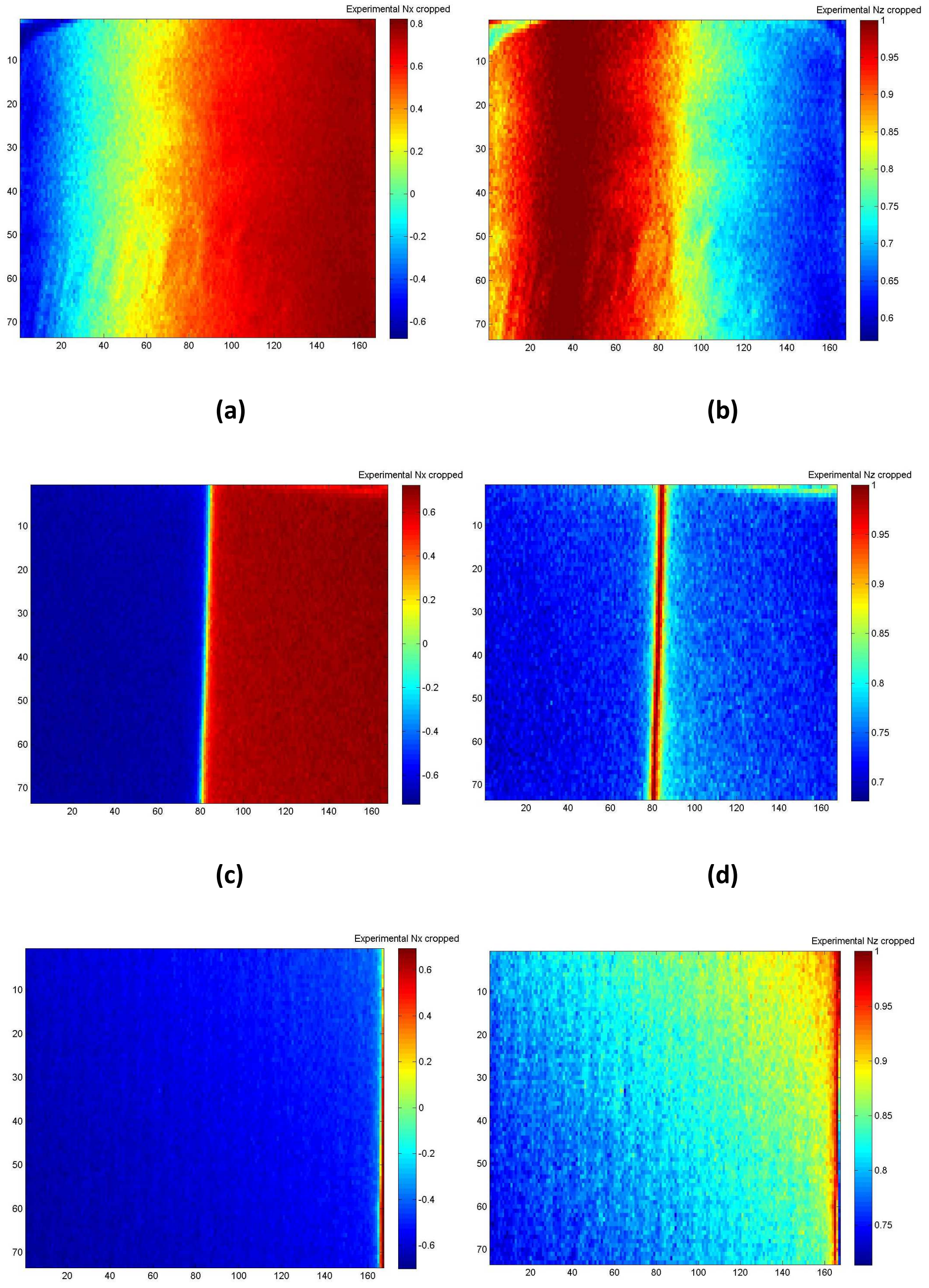
(b)

(c)

(d)

Figure 101: (a)(b) Experimental images obtained under two different lighting conditions (top to bottom: half-cylinder, prism, cuboid). (c) Recovered N_x surface normals, (d) Recovered N_z surface normals (note the non-uniform pattern in the background, which is due to non-uniform illumination).

Then, the surfaces' normals of one of the three shapes of interest, i.e. the half-cylinder, are extracted from the recovered surfaces' normals of monitored scene. This is done manually to allow the user, e.g. the security personnel from a control room, to only extract data he or she is interested in and then analyse it. Once an object or area of interest is identified, the user only needs to select it by drawing a rectangle around it and cropping it. This allows the extraction of surfaces' normals of interest only. Moreover, the dimensions of this rectangle are memorised. This way, for the next object or area of interest, the user only has to select its centre to make its coordinates known to the software and the extraction of its surfaces' normals is done automatically. This is made possible by using the dimensions of the first selected rectangle as a reference to calculate the exact same selection for other objects or areas of interest once their centre has been selected. It not only allows the automatic extraction of surfaces' normals of various object(s) or area(s) of interest but also the extraction of the same amount of data, which is significant as they will be compared at a later stage. An example of the surfaces' normals extracted for each of the three shapes can be seen on figure 102 below.



(a)

(b)

(c)

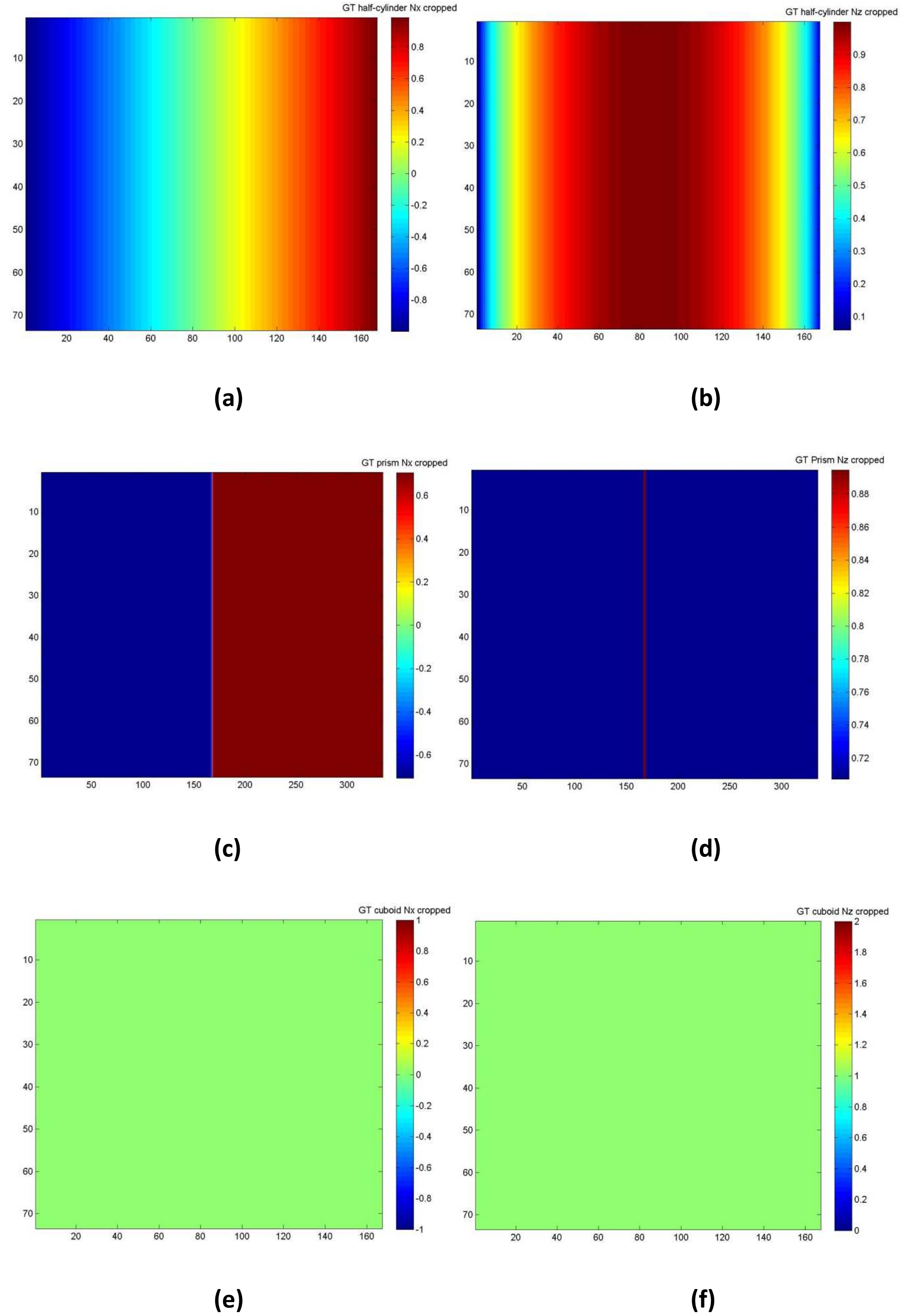
(d)

(e)

(f)

Figure 102: (a) Recovered Nx of the real half-cylinder, (b) Recovered Nz of the real half-cylinder, (a) Recovered Nx of the real prism, (b) Recovered Nz of the real prism,(c) Recovered Nx of the real cuboid, (d) Recovered Nz of the real cuboid.

The exact same steps are then repeated for GT surfaces' normals. An example of surfaces' normals extracted for the three GT shapes can be seen on figure 103 below.



(a) (b)

(c) (d)

(e) (f)

Figure 103: (a) Recovered Nx of the GT half-cylinder, (b) Recovered Nz of the GT half-cylinder, (a) Recovered Nx of the GT prism, (b) Recovered Nz of the GT prism,(c) Recovered Nx of the GT cuboid, (d) Recovered Nz of the GT cuboid.

The next stage is then to compare experimental and GT surfaces' normals of object(s) or area(s) of interest previously extracted. For this purpose, a Euclidean distance measure is calculated. For each object or area of interest, the Euclidean distance between GT N_x and N_z and experimental N_x and N_z surfaces' normals is calculated using the following formula:

$$\boldsymbol{d_{GT\_Exp}^{shape\_shape}} = \sqrt{\left(\boldsymbol{N_x^{GT\_shape}} - \boldsymbol{N_x^{Exp\_shape}}\right)^2 + \left(\boldsymbol{N_z^{GT\_shape}} - \boldsymbol{N_z^{Exp\_shape}}\right)^2} \qquad (99)$$

Where $N_x^{GT\_shape}$ and $N_z^{GT\_shape}$ are the x and z component of the surface normals recovered for a GT shape, $N_x^{Exp\_shape}$ and $N_z^{Exp\_shape}$ are the x and z component of the surface normals recovered for an experimental shape and $d_{GT\_Exp}^{shape\_shape}$ is the Euclidean distance which is used a metric to compare the x and z component of the surface normals recovered for a GT shape to those for an experimental shape.

As three shapes were considered in the considered example, nine Euclidean distances have been calculated, one for each of the following combination between GT and experimental surfaces' normals as can be seen on the table 22 below.

	Experimental Half-cylinder	**Experimental Prism**	**Experimental Cuboid**
GT Half-cylinder	$d_{GT\_Exp}^{\text{Half−cylinder\_Half−cylinder}}$	$d_{GT\_Exp}^{\text{Half−cylinder\_Prism}}$	$d_{GT\_Exp}^{\text{Half−cylinder\_Cuboid}}$
GT Prism	$d_{GT\_Exp}^{\text{Prism\_Half−cylinder}}$	$d_{GT\_Exp}^{\text{Prism\_Prism}}$	$d_{GT\_Exp}^{\text{Prism\_Cuboid}}$
GT Cuboid	$d_{GT\_Exp}^{\text{Cuboid\_Half−cylinder}}$	$d_{GT\_Exp}^{\text{Cuboid\_Prism}}$	$d_{GT\_Exp}^{\text{Cuboid\_Cuboid}}$

Table 22 : Euclidean distance calculated in the case of three objects of interest (i.e. here, the half-cylinder, the prism and the cuboid)

For each Euclidean distance, a mean value has been determined using the formula (100) below:

$$\boldsymbol{\mathcal{M}ean_{GT\_Exp}^{shape\_shape}} = \frac{\boldsymbol{\sum_0^T d_{GT\_Exp}^{shape\_shape}}}{\boldsymbol{T}} \qquad (100)$$

Where $d_{GT\_Exp}^{shape\_shape}$ is the Euclidean distance which is used a metric to compare the x and z component of the surface normals recovered for a GT shape to those for an experimental shape, T is the T number of data values and $\mathcal{M}ean_{GT\_Exp}^{shape\_shape}$ is the mean value determined for Euclidean distance corresponding to a specific GT and experimental shape.

The table below illustrates the mean values that would be calculated in the case of three shapes of interest.

	Experimental Half-cylinder	**Experimental Prism**	**Experimental Cuboid**
GT Half-cylinder	$\mathcal{M}ean_{GT\_Exp}^{\text{Half−cylinder\_Half−cylinder}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Half−cylinder\_Prism}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Half−cylinder\_Cuboid}}$
GT Prism	$\mathcal{M}ean_{GT\_Exp}^{\text{Prism\_Half−cylinder}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Prism\_Prism}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Prism\_Cuboid}}$
GT Cuboid	$\mathcal{M}ean_{GT\_Exp}^{\text{Cuboid\_Half−cylinder}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Cuboid\_Prism}}$	$\mathcal{M}ean_{GT\_Exp}^{\text{Cuboid\_Cuboid}}$

Table 23 : Mean value calculated in the case of three objects of interest (i.e. here, the half-cylinder, the prism and the cuboid)

Finally, the shape of each object of interest can be identified by the lowest mean value for a given GT shape.

Two other comparisons related to the GT prism and cuboid are performed with the same approach.

This section described the proposed approach for object's shape recognition. The following section presents results obtained and their analysis.

6-1-3-Results

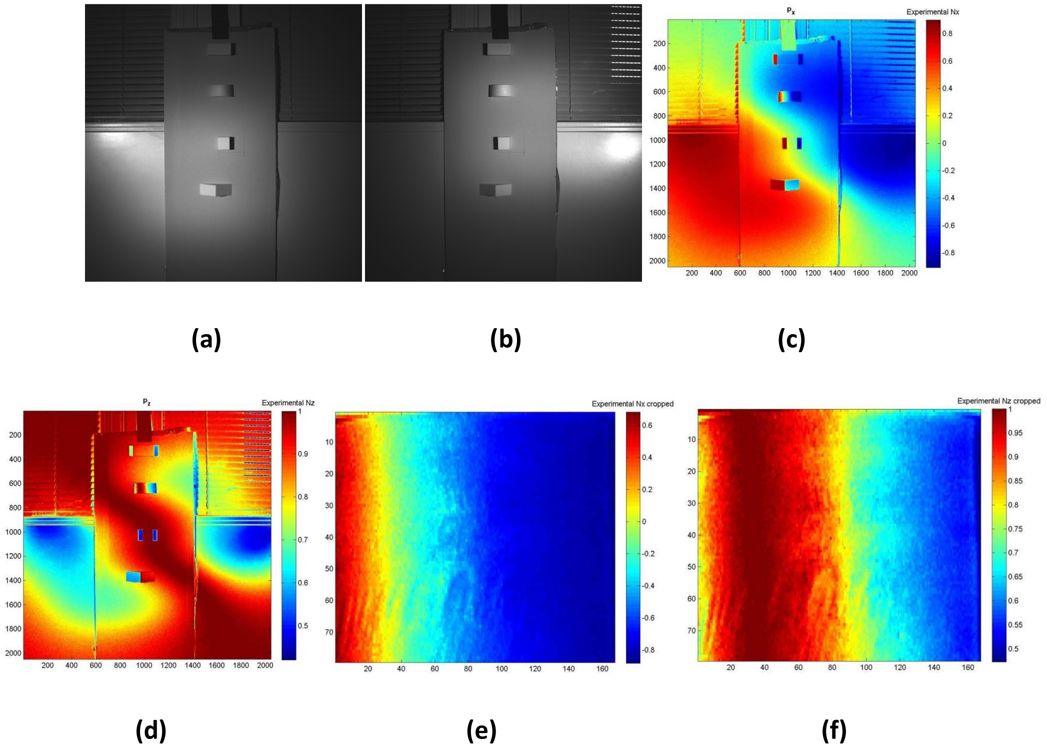
In this section, results obtained within s short and long-range are presented.

6-1-3-1-Short-range results

In this section, qualitative results are provided from experiment performed with the cardboard with three simple shapes (i.e. half-cylinder, prism and cuboid) within a short range.

The position of the shapes has been interchanged throughout the experiment to avoid position bias. As all combinations represent a huge amount of data, only a few examples of various positions for the half-cylinder are presented here. The goal of using a change in position for these shapes was to verify the efficiency of the method to recognise an object's shape from the pattern of the variation of its surface's normals.

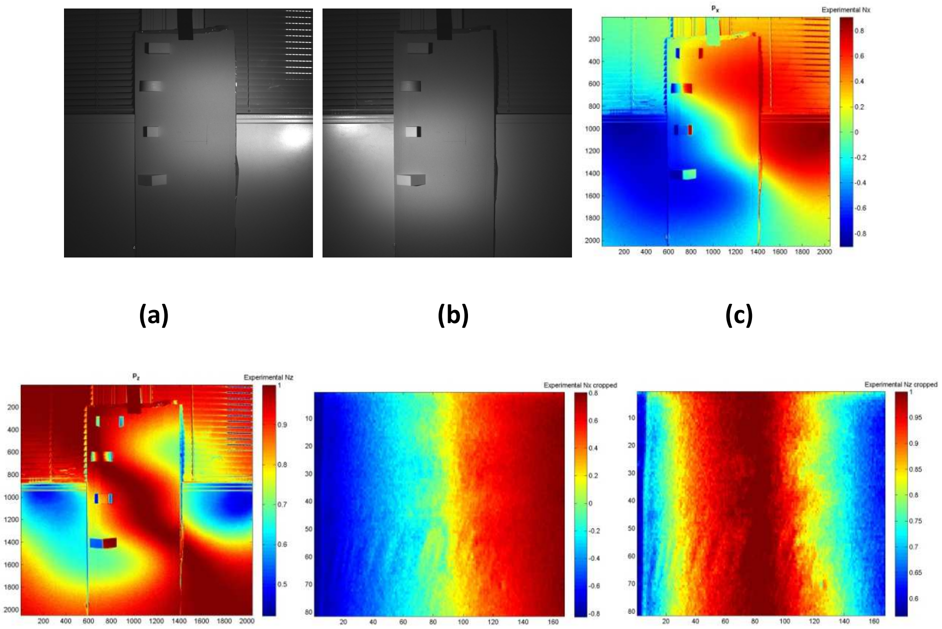
Distance from Light1 to Light2 =1.52 m, distance Camera to model=1.95 m:



(a) (b) (c)

(d) (e) (f)

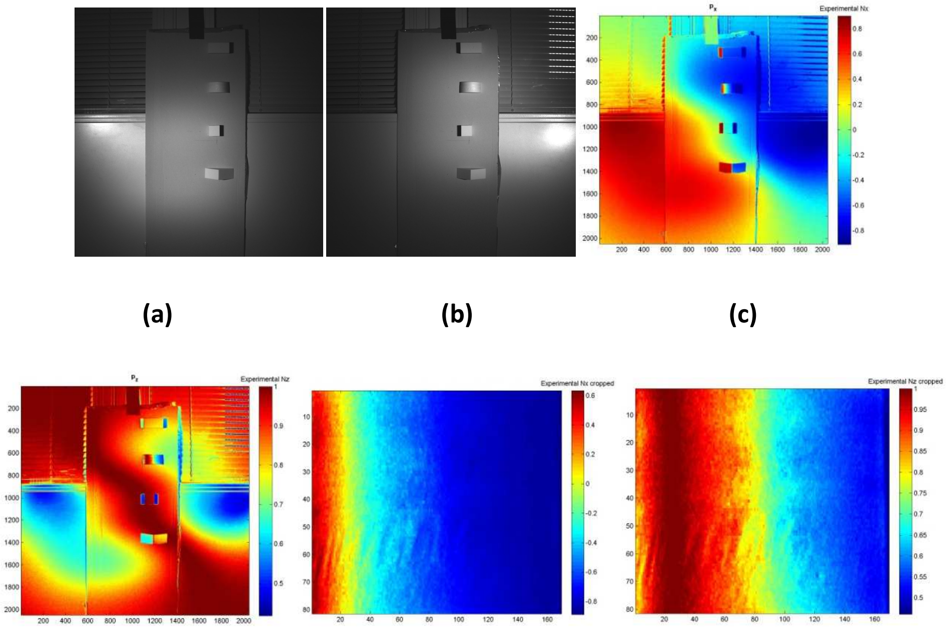
Figure 104 : (a)(b) images under two different lighting conditions with the half-cylinder at the central top of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

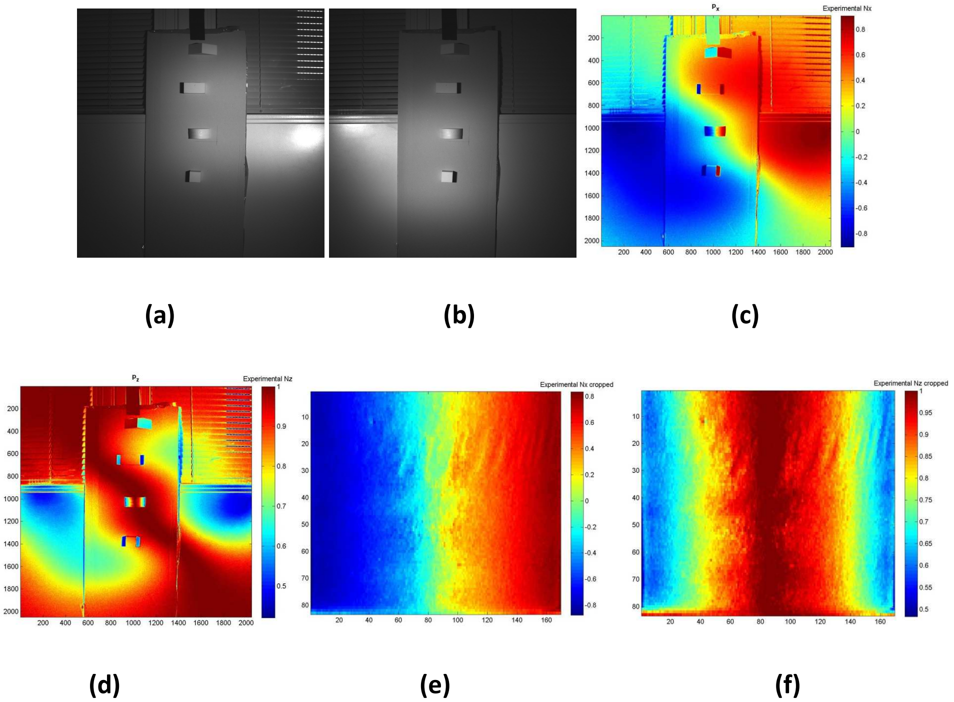
Figure 105 : (a)(b) images under two different lighting conditions with the half-cylinder at the top left of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

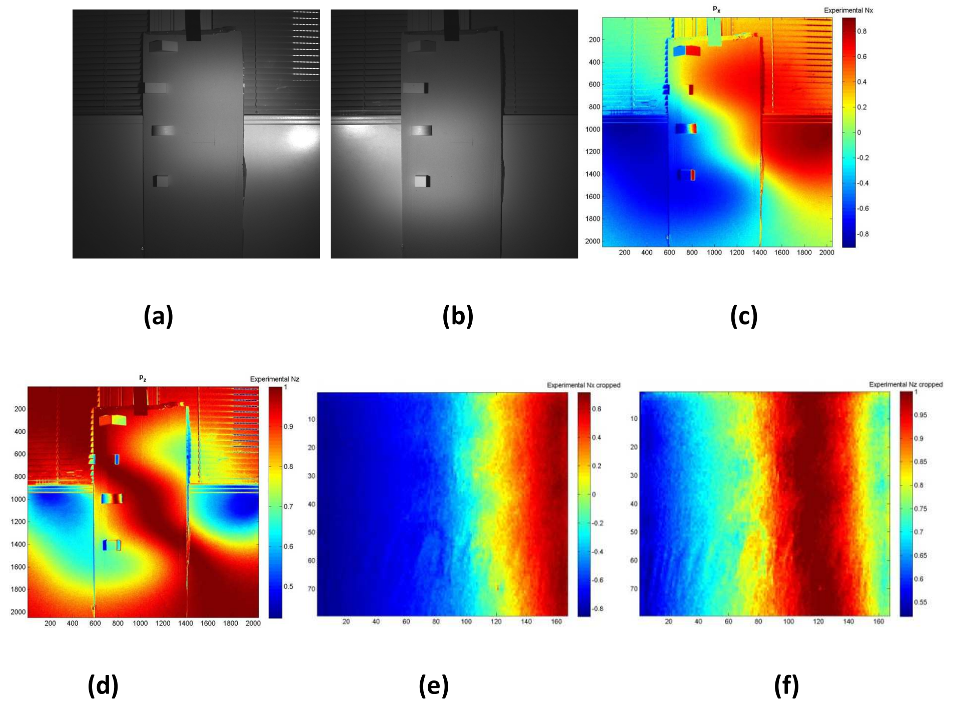
Figure 106 : (a)(b) images under two different lighting conditions with the half-cylinder at the top right of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

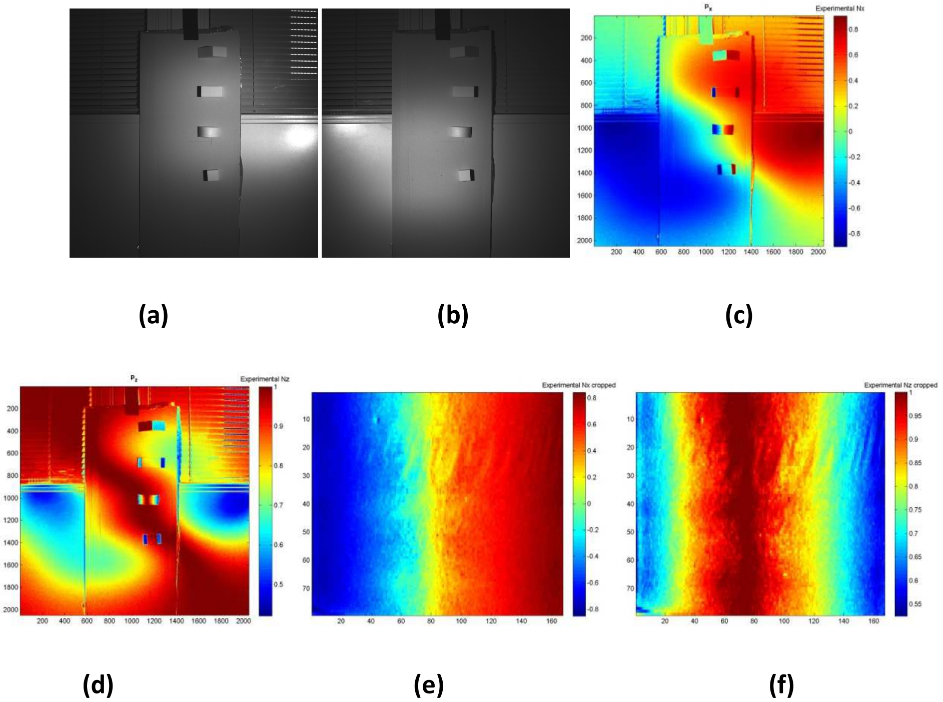
Figure 107 : (a)(b) images under two different lighting conditions with the half-cylinder at the centre of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

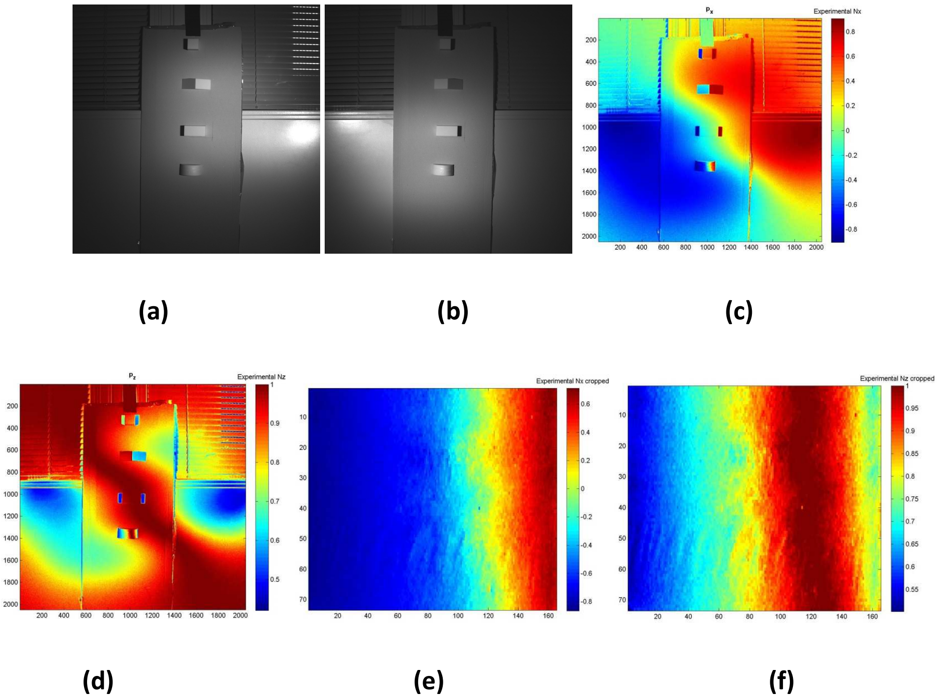
Figure 108 : (a)(b) images under two different lighting conditions with the half-cylinder at the central left of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

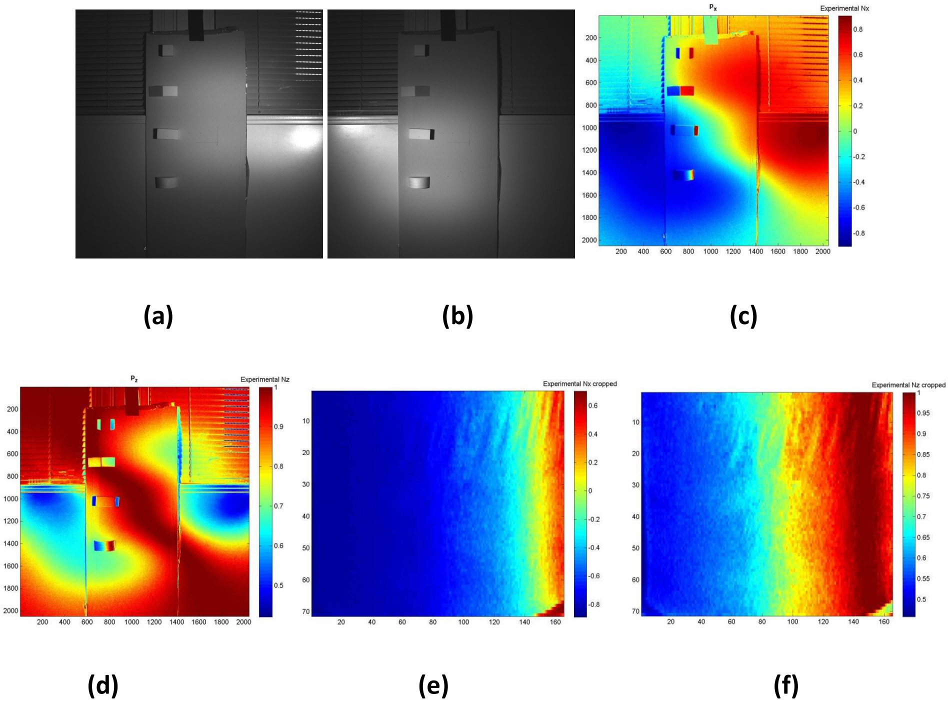
Figure 109 : (a)(b) images under two different lighting conditions with the half-cylinder at the central right of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

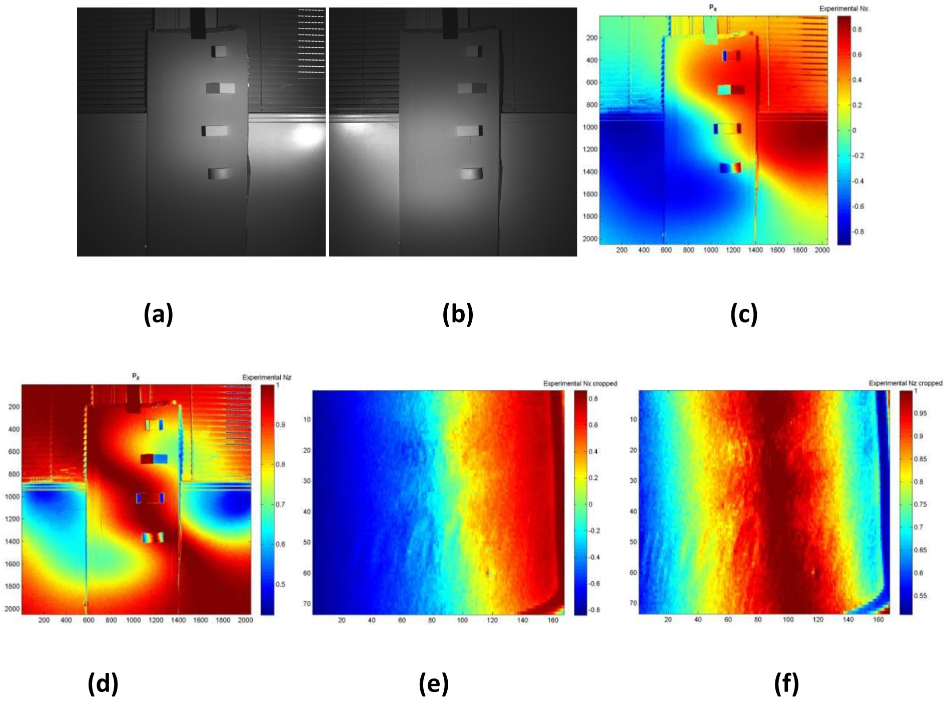
Figure 110 : (a)(b) images under two different lighting conditions with the half-cylinder at the central bottom of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

Figure 111 : (a)(b) images under two different lighting conditions with the half-cylinder at the bottom left of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

Figure 112 : (a)(b) images under two different lighting conditions with the half-cylinder at the bottom right of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

From the figures above, it is noticeable that the position of the shape affects the variation of the surface normals of the half-cylinder.

Figures 104, 105 and 106 show the surface normals of the half-cylinder when it is at the top of the cardboard and either at the left, the centre or the right of the cardboard. These results show mainly the x component of the surface normals is affected by the change in position as the image of the z component of the surfaces' normals seems very similar in the top and right cases.

Figures 107, 108 and 109 show the surface normals of the half-cylinder when it is at the centre, central left and central right of the cardboard. In this case, the x component of the surface normals is affected by the change in position unless for the central right position where their variation is has expected. However, on the contrary to the previous case, the z component of the surface normals is also affected by the change in position here.

Finally, figures 110, 111 and 112 shows the surface normals of the half-cylinder when it is at the central bottom, bottom left and bottom right of the cardboard. In this case, both x and z components of the surface normals are affected by the change in position.

These experiments have shown that the variation of the surface normals depends significantly on the repartition of the light, especially in the case of short-range experiments. Moreover, when the shape is at the extremities of the cardboard it seems to give worse results than when it is more central. This shows it is then essential to focus the light sources in front of the object of interest to be sure the light reflected does not affect the recovery of the surface normals. Indeed, the light sources do not seem to be projected uniformly on the cardboard which can explain why surface normals vary from one position to another. To verify the origin of this issue and deeper investigate the proposed approach, similar experiments have been realised within a long-range and are presented in the next section.

While this is clearly causing major problems here, it does not seem to be a major weakness of the method as different light source types should provide a more uniform distribution. Furthermore, from the result above, one can easily assume that the same light sources at longer range do not cause the same problem.

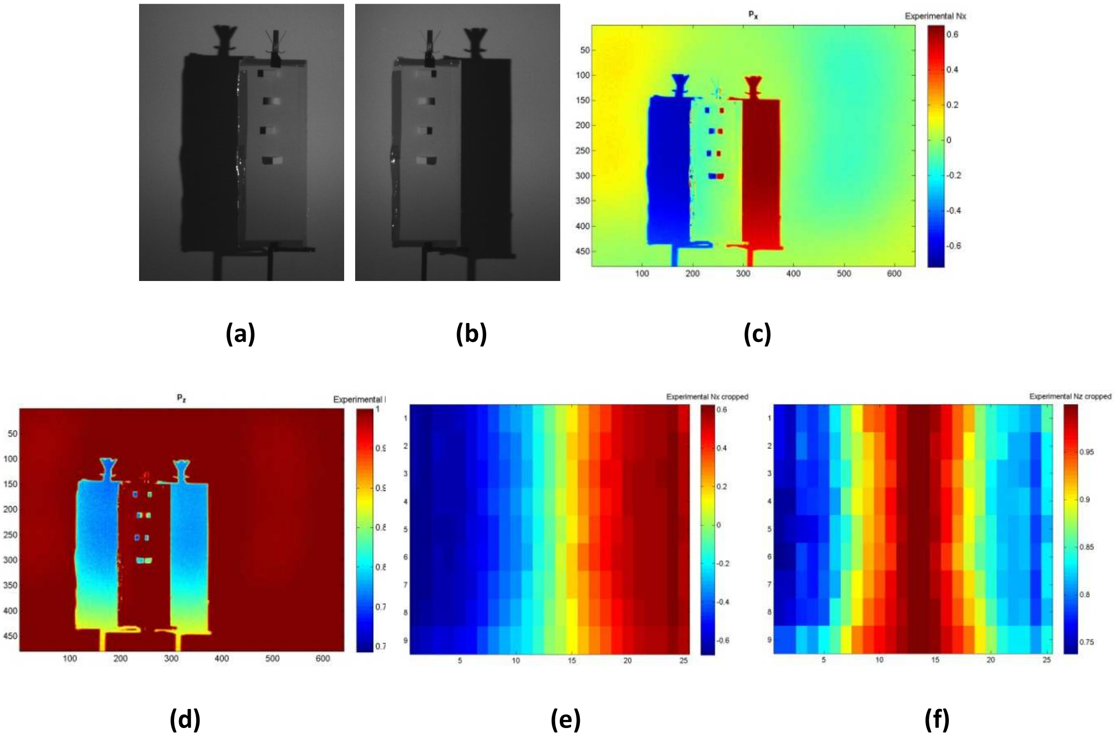
6-1-3-3- Long-range results

In this section, results obtained from a long range are presented and analysed qualitatively. Several experiments were conducted keeping the same distance between the light sources and the cameras but varying the distance from the camera to the object to see its influence on the surface normals' recovery.

- Cardboard with three different simple shapes: a half cylinder, a prism and a cuboid interchanged from top to bottom and left to right:

Distance from Light1 to Light2 =7.9 m, distance Camera to model=5.2 m:

The figure below represents results of tests made with the board from a distance of 5.2 m from the camera to the board.



(a) (b) (c)

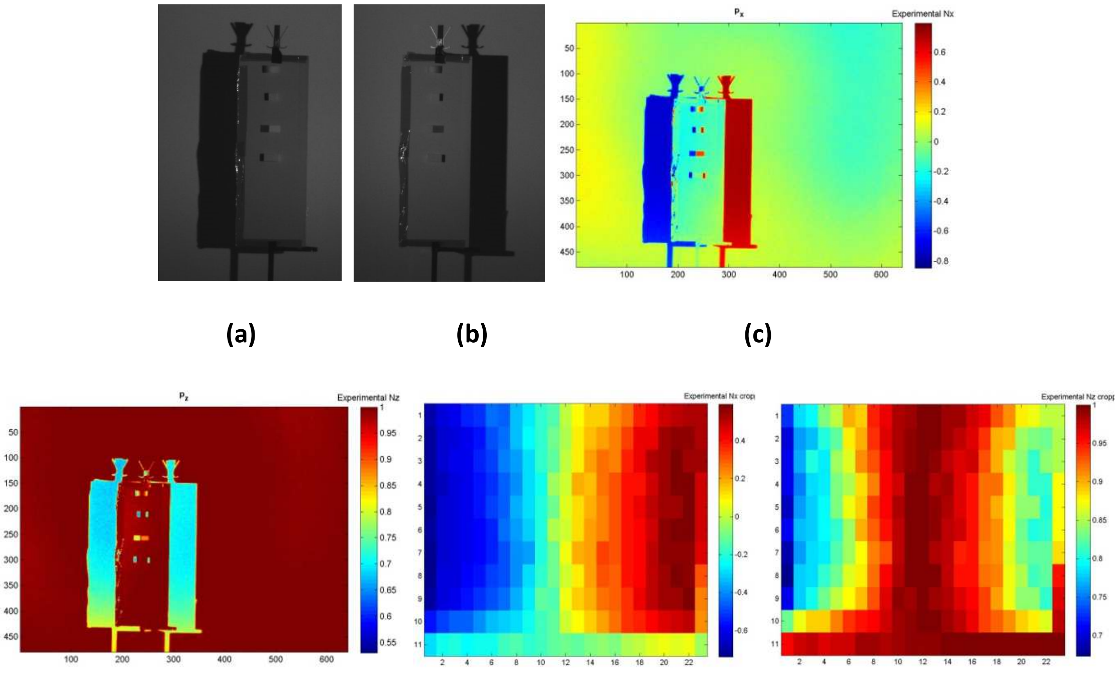
(d) (e) (f)

Figure 113 : (a)(b) images under two different lighting conditions with the half-cylinder at the centre of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

Compared to results obtained within a short-range, these results show that the distance has an influence on the surface normals' recovery as they appear to vary less in this case even if the shape's location varies from one position to another. This confirms that the issues encountered within short-range must depends on the fact the light-sources are too close to the target which then does not allow the light to be distributed evenly. On the contrary to this case as a distance between the camera and the target is larger and permits to distribute the light more evenly and obtain better results.

Distance from Light1 to Light2 =7.9 m, distance Camera to model=7.6 m:

The next figure illustrates results obtained from a longer distance for the half-cylinder at the top and the centre of the board. Here no position variations has been tested as the point of this experiment was only to see if a longer distance would make any difference.



(a) (b) (c)

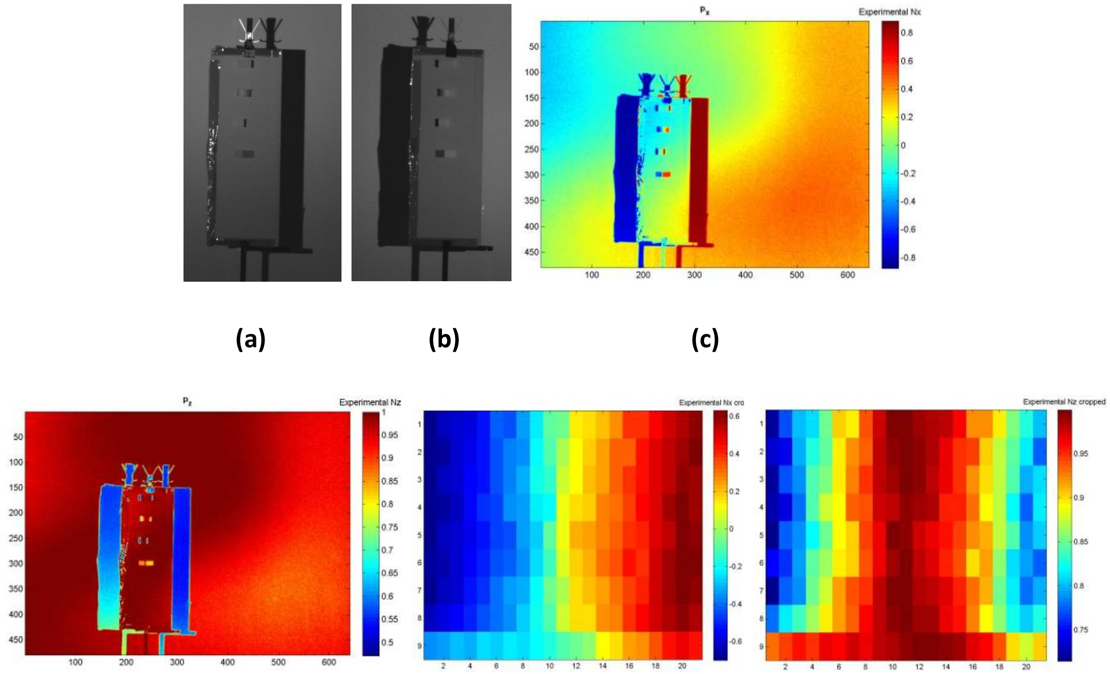
(d) (e) (f)

Figure 114 : (a)(b) images under two different lighting conditions with the half-cylinder at the centre of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

Figure 114 above shows similar results as those obtained from a distance of 5.2 m from the camera to the board.

Distance from Light1 to Light2 =7.9 m, distance Camera to model=10 m:

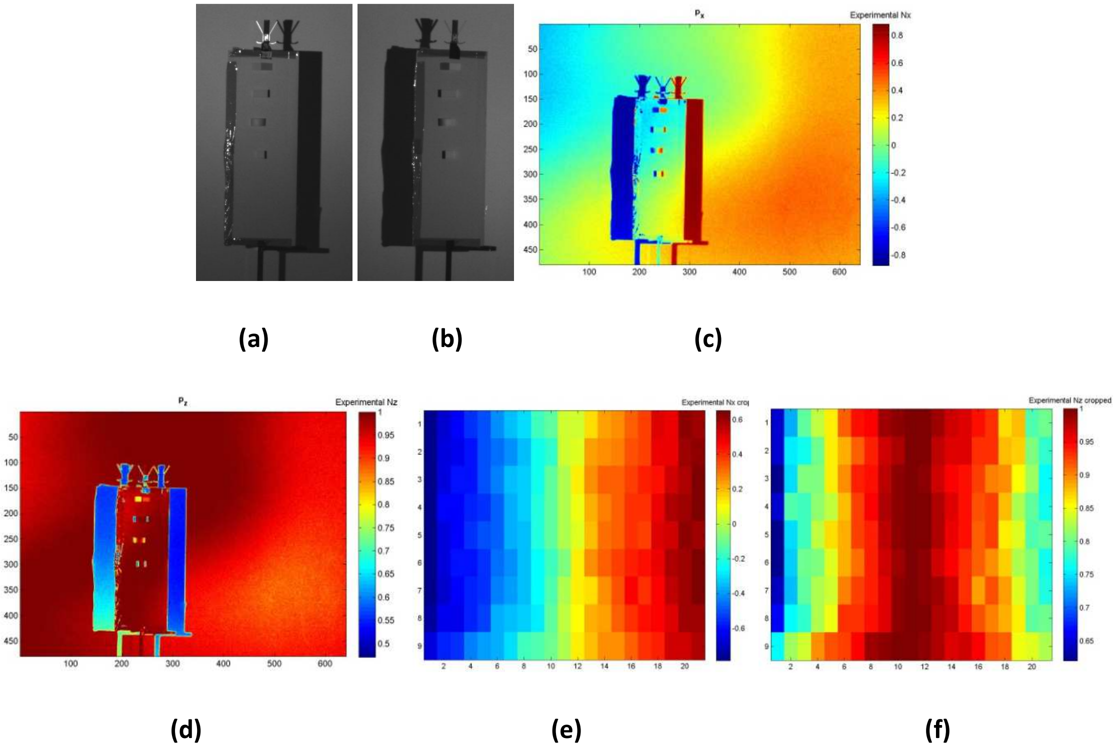
The figures below show results obtained within a longer distance from the camera to the objects, i.e. 10 m. Only a few examples will be provided here as it will permit a better comprehension.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

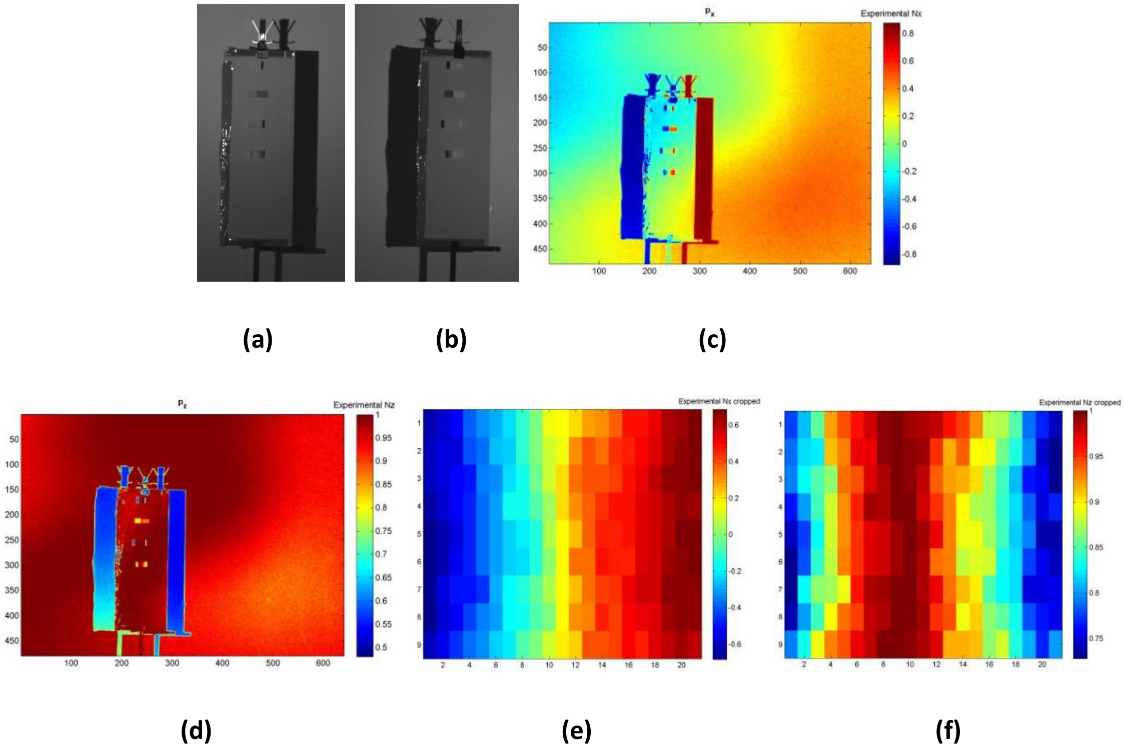
Figure 115 : (a)(b) images under two different lighting conditions with the half-cylinder at the central top of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

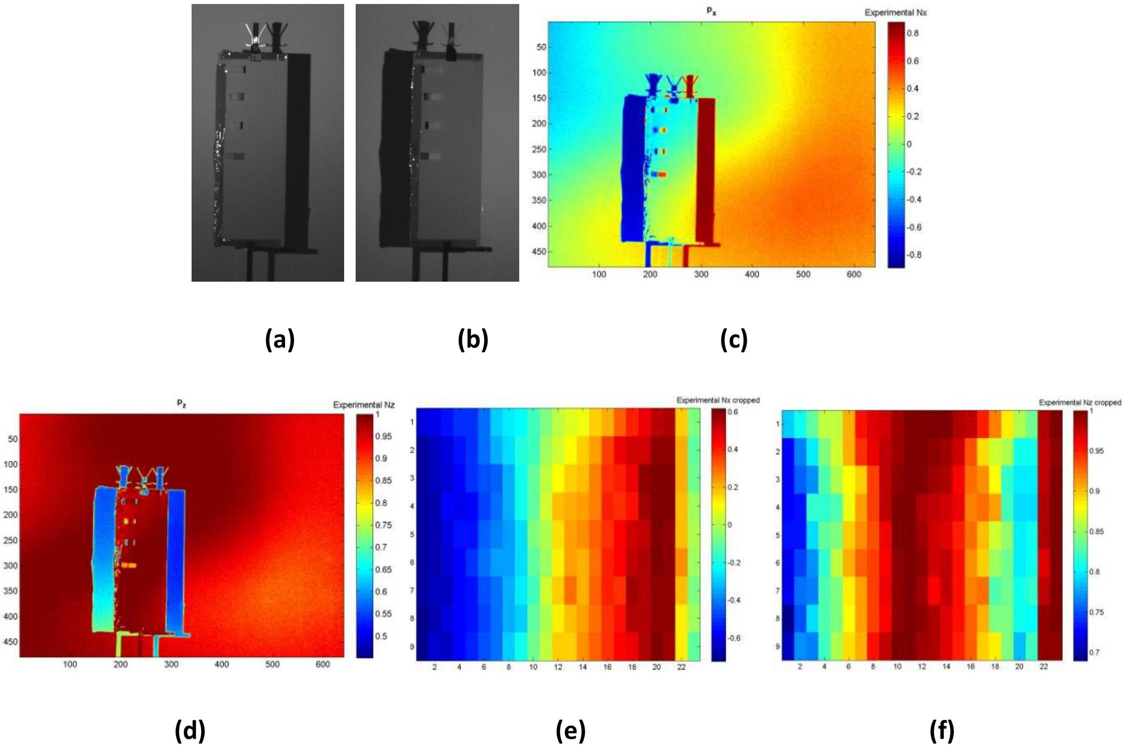
Figure 116 : (a)(b) images under two different lighting conditions with the half-cylinder at the centre of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

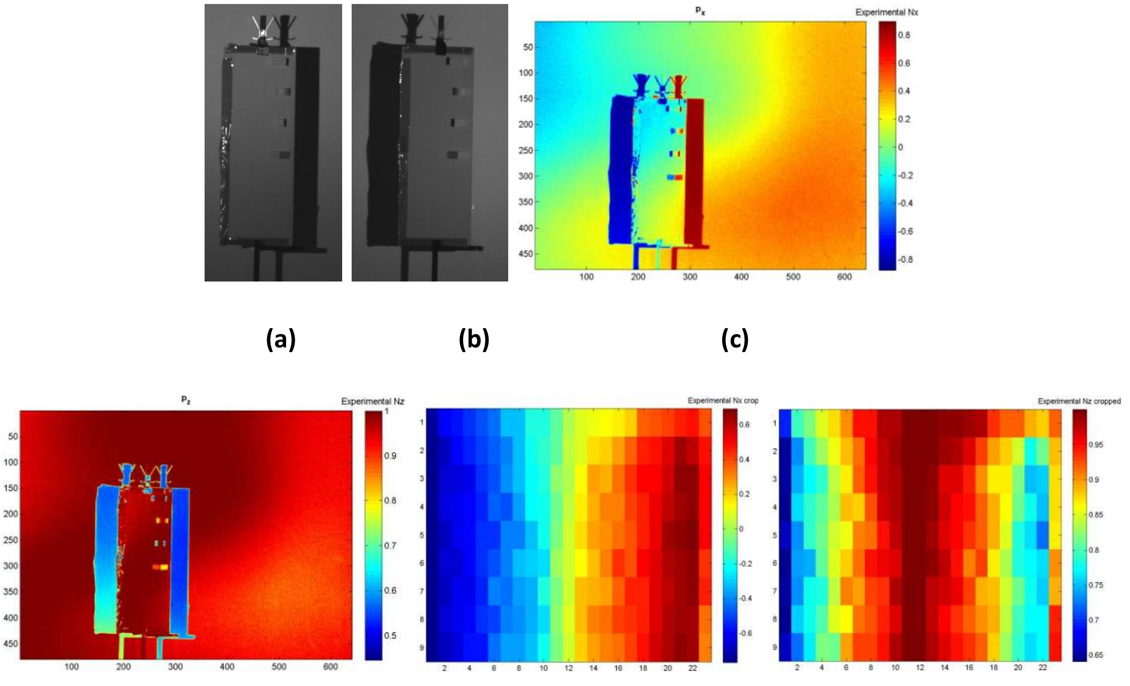
Figure 117 : (a)(b) images under two different lighting conditions with the half-cylinder at the central bottom of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

Figure 118 : (a)(b) images under two different lighting conditions with the half-cylinder at the central left of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.



(a) (b) (c)

(d) (e) (f)

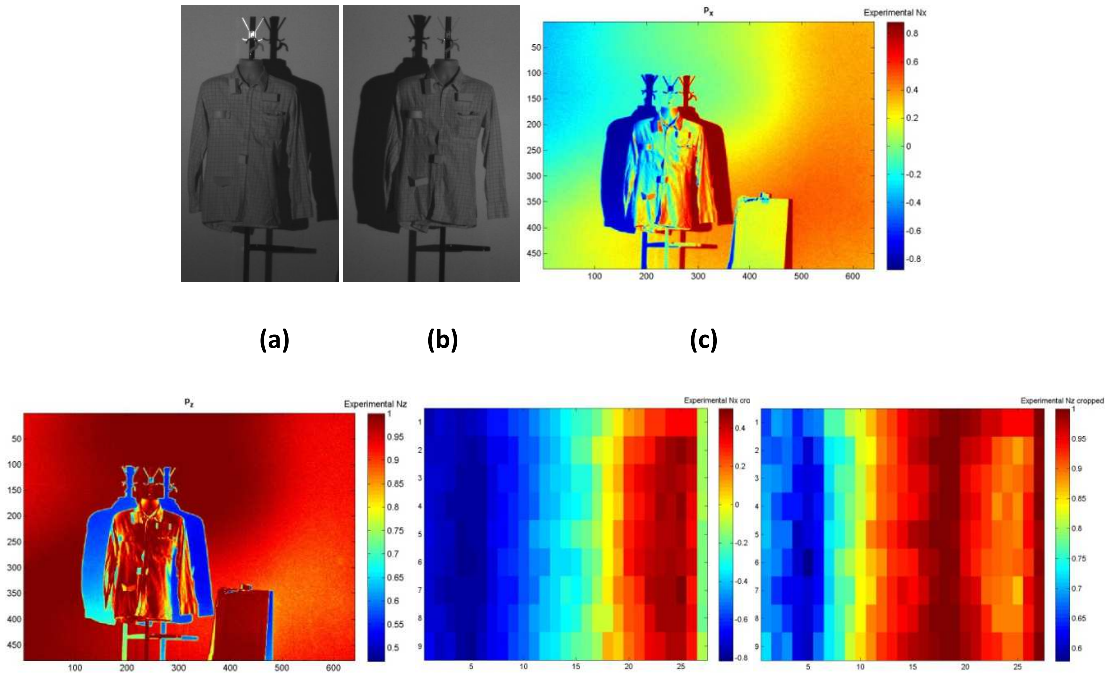
Figure 119 : (a)(b) images under two different lighting conditions with the half-cylinder at the central right of the cardboard, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

These figures show that both x and z component of the surface normals remain constant even when the shape is in various location. The results obtain from a long-range are then very promising as they demonstrate the efficiency of the algorithm to obtain similar surface normals for the same object wherever it is situated. However, this is true for its location but needs to be verified in the case of change in the orientation of the object of interest. More realistic experiments are presented in the next section that shows preliminary observations about the effect of the orientation of the object for the surface normals recovery.

- Torso with various shapes:

In this section, the shapes have been placed on a torso for more realistic experiments.

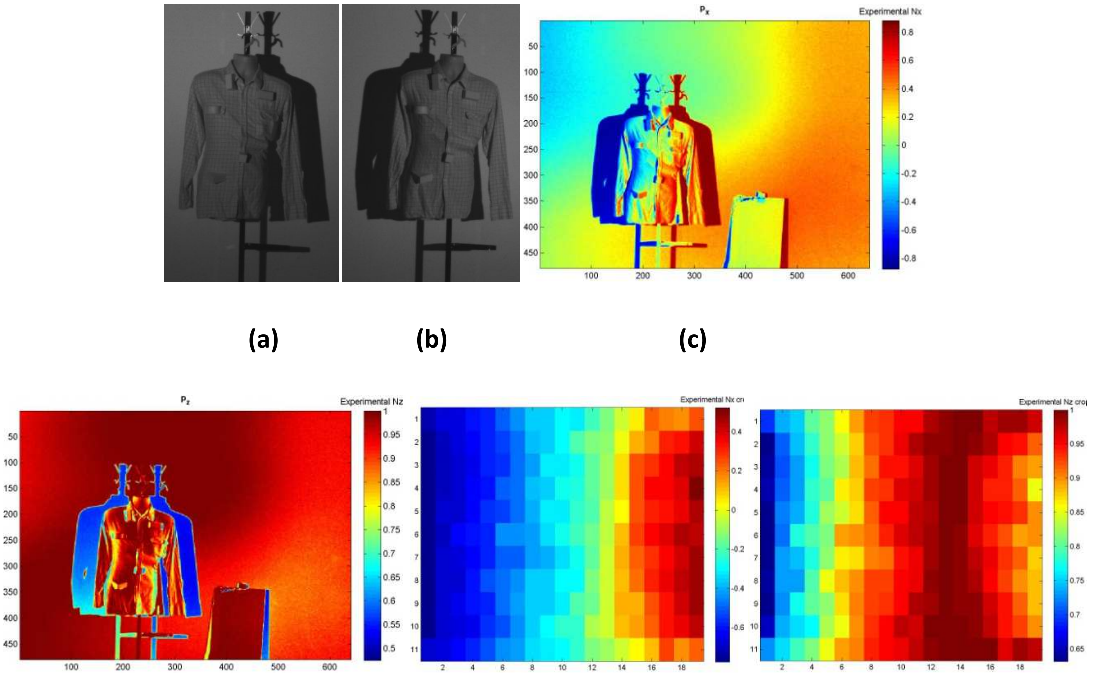
Distance from Light1 to Light2 =7.9 m, distance Camera to model=10 m:



(a) (b) (c)

(d) (e) (f)

Figure 120 : (a)(b) images under two different lighting conditions with the half-cylinder on the left side of the shirt, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

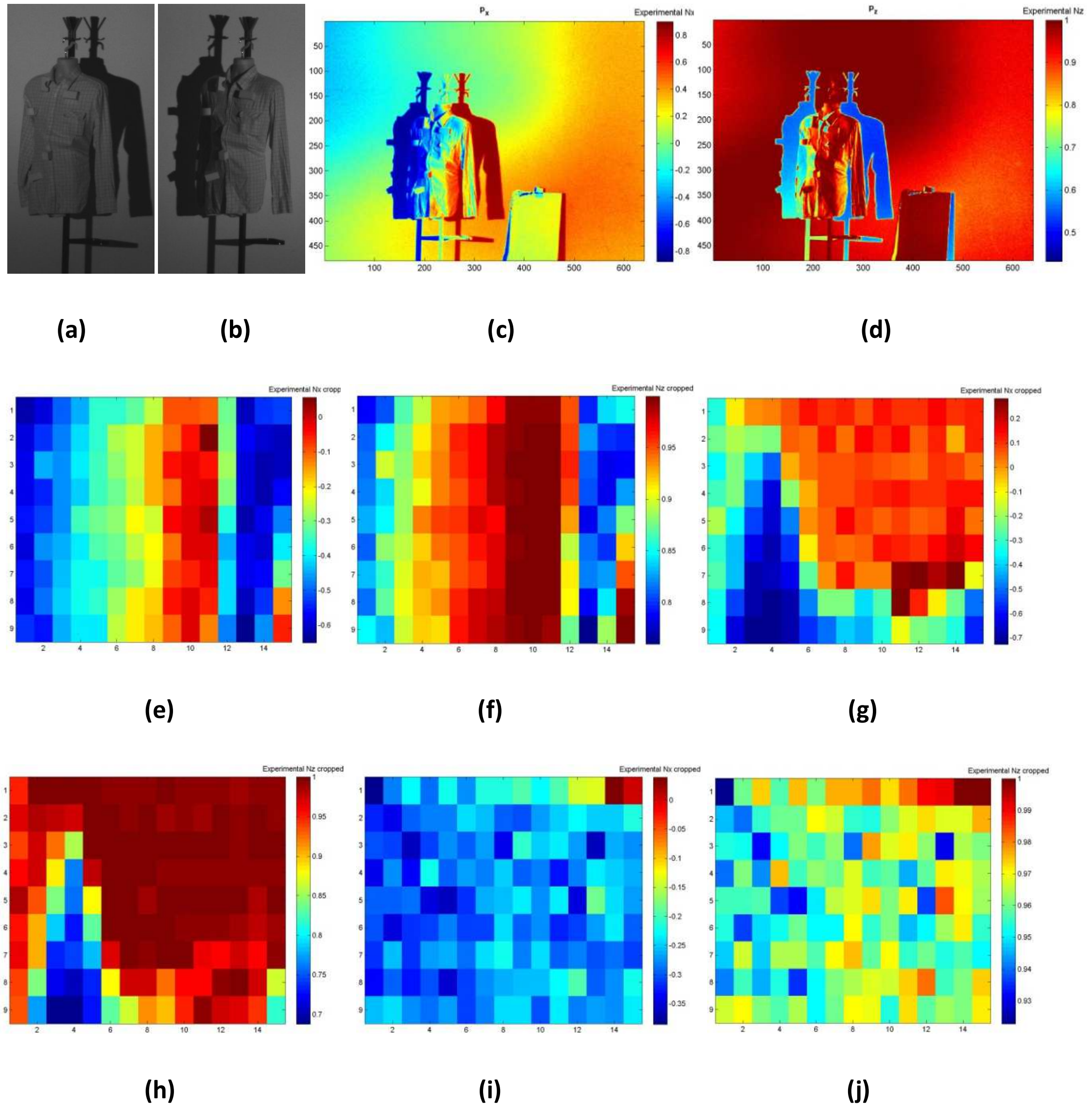


(a) (b) (c)

(d) (e) (f)

Figure 121 : (a)(b) images under two different lighting conditions with the half-cylinder on the left side of the shirt tight in the back, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder.

The two previous figures show that shape's surface normals are not as precise when attached to the shirt as on the cardboard. Indeed, while the x component varies as expected, the z component does not vary as symmetrically as it should. This must be due to the fact that the cardboard is a flat surface whereas the shirt is flowing which creates variation in the surface where the shape is placed and then affects the surface normals of the shape considered.



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

(e) (f) (g)

(h) (i) (j)

Figure 122 : (a)(b) images under two different lighting conditions with the half-cylinder on the left side of the shirt tight in the back and rotated towards the left, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder, (g) Extracted Nx for the prism, (h) Extracted Nz for the prism, (i) Extracted Nx for the cuboid, (j) Extracted Nz for the cuboid.

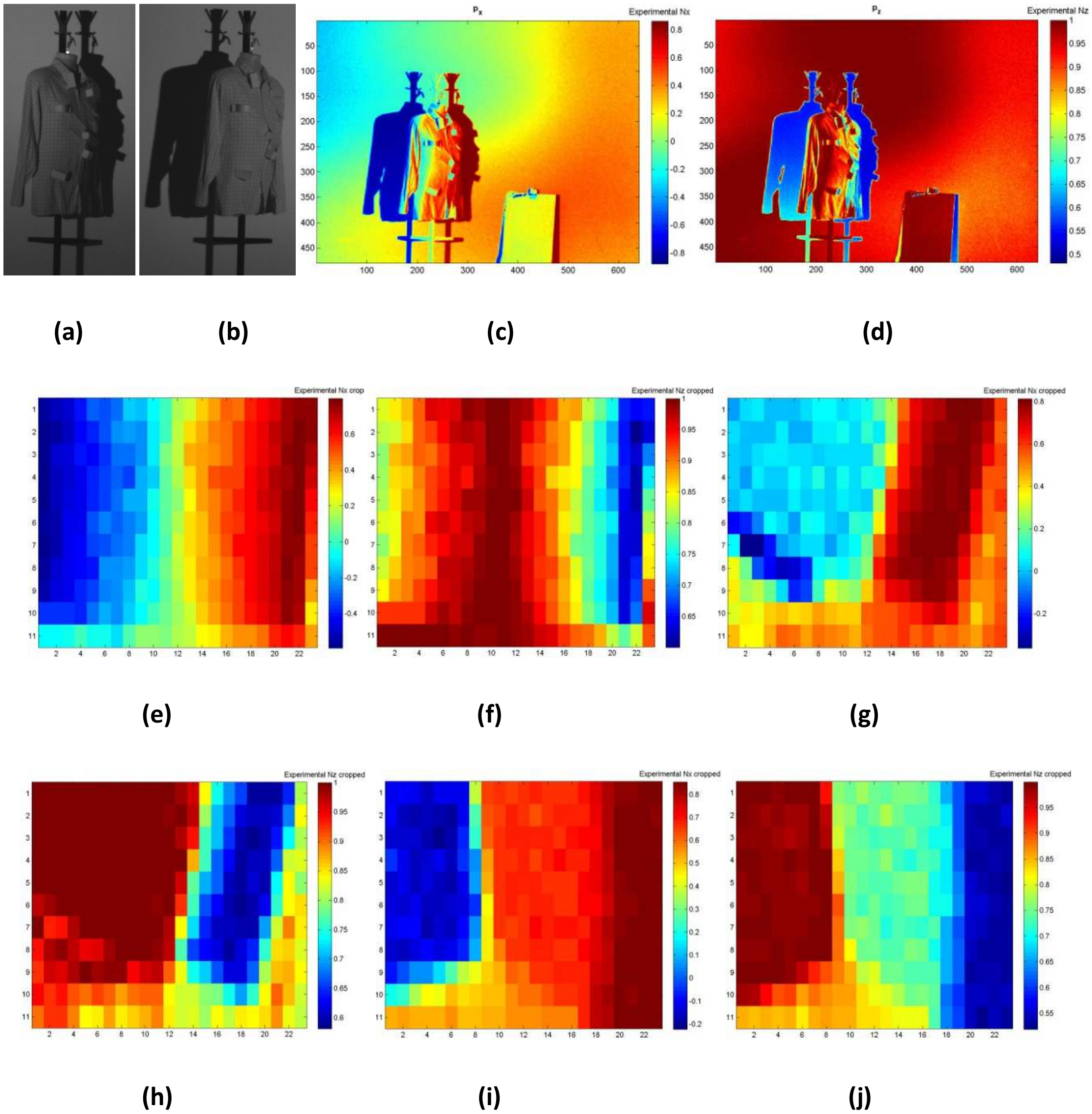
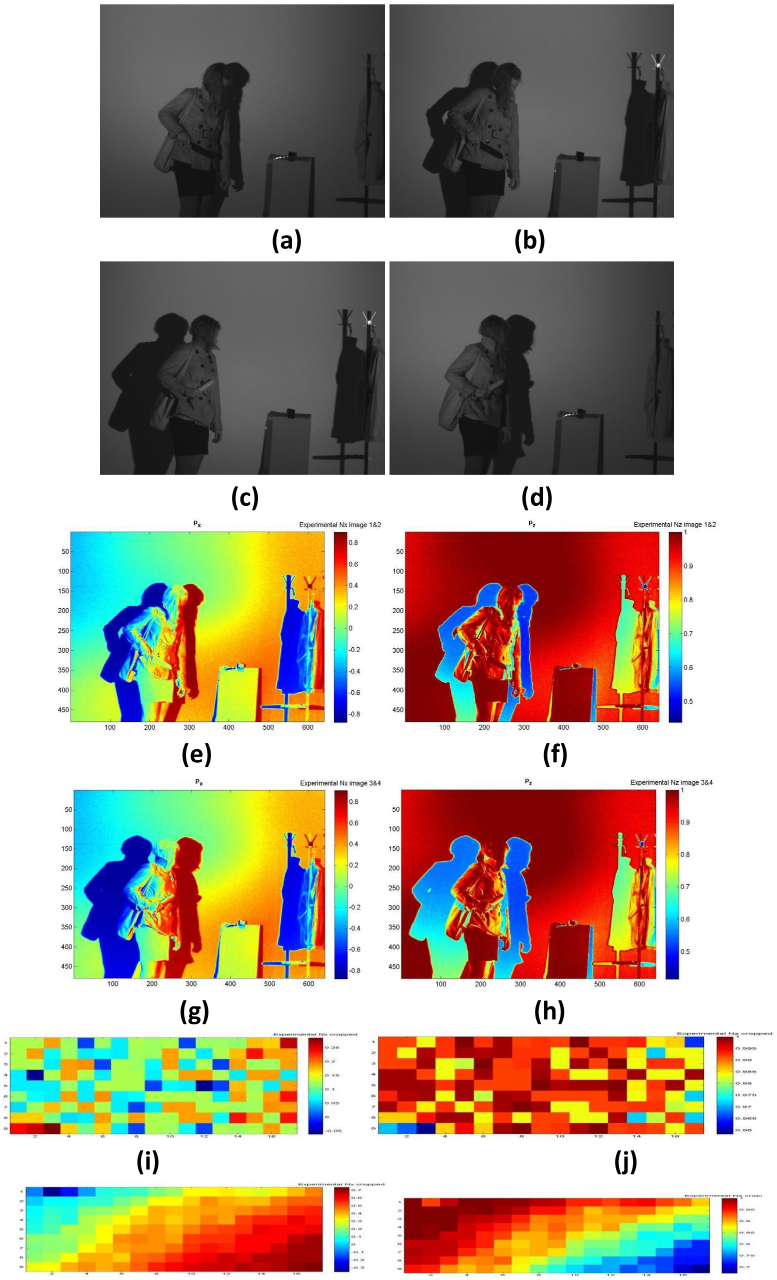


Figure 123 : (a)(b) images under two different lighting conditions with the half-cylinder on the left side of the shirt tight in the back and rotated towards the right, (c) Nx, (d) Nz, (e) Extracted Nx for the half-cylinder, (f) Extracted Nz for the half-cylinder, (g) Extracted Nx for the prism, (h) Extracted Nz for the prism, (i) Extracted Nx for the cuboid, (j) Extracted Nz for the cuboid.

The two previous figures confirm that not only shapes' surface normals obtained from non-flat surfaces are less precise but also that the orientation on which the object is observed is highly significant. Indeed, as can be seen on figure 122 the distinction between various shapes is not easy as those that are on the less visible side due to the rotation of the torso are partly occlude and do not allow having sufficient information about their geometry. On the contrary, on figure 123, the half-cylinder has been recognised more easily by the algorithm as this time it is not occluded due to rotation of the shirt on the contrary to the other shapes. Moreover, the prism and the cuboid have similar surface normals which do not help to differentiate them.

- Simulation of a real scene:



(a) (b)

(c) (d)

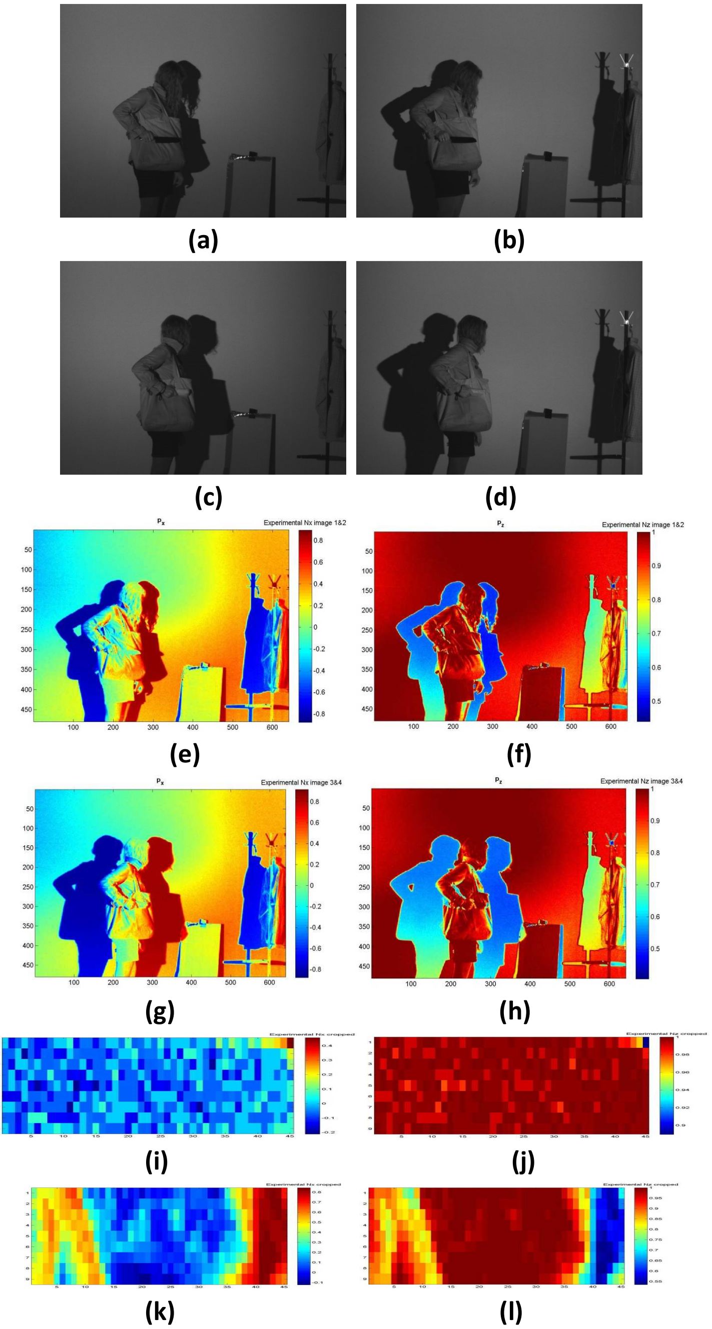
(e) (f)

(g) (h)

(i) (j)

(k) (l)

Figure 124 : (a)(b) images under two different lighting conditions of a person holding a knife, (c)(d) images under two different lighting conditions of a person holding an umbrella, (e) Nx for the scene with a knife, (f) Nz for the scene with a knife, (g) Nx for the scene with an umbrella, (h) Nz for the scene with an umbrella,, (i) Extracted Nx of the knife, (j) Extracted Nz of the knife, (k) Extracted Nx of the umbrella, (l) Extracted Nz of the umbrella.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

(k) (l)

Figure 125 : (a)(b) images under two different lighting conditions of a person holding a knife, (c)(d) images under two different lighting conditions of a person holding an umbrella, (e) Nx for the scene with a knife, (f) Nz for the scene with a knife, (g) Nx for the scene with an umbrella, (h) Nz for the scene with an umbrella,, (i) Extracted Nx of the knife, (j) Extracted Nz of the knife, (k) Extracted Nx of the umbrella, (l) Extracted Nz of the umbrella.

The figure 124 above gives an overview of what surface normals would look like in the case of a realistic scenario such as a person holding a knife or an umbrella. In this case, it is essential to make the distinction between the two objects holding by the person in order to be able to infer the function and functionality of the object to prevent from crime or prosecute criminal. Moreover, the variation of the surface normals for the knife differs from the umbrella in the sense that the knife is the flat object and the umbrella is not. This can be seen on figure 95 above where the x component of the surface normals for the knife is 0 while it varies for the umbrella and the z component of the surface normals for the knife is 1 while it also varies for the umbrella. This clearly shows this difference in variation is the key to identify both objects' shape in this case. The results obtained on figure 125 are another example of a similar experiment where it failed. This can be explained by the way the person is holding the umbrella by pressing it into the bag. So, the fact the umbrella is not exactly perpendicular to the camera, affects its surface normals and makes the system fail. This is one example of a case where the object of interest has a different angle that the GT object considered for these experiments, i.e. object perpendicular to the field of view of the camera, and will be explained in more detail in the next section.

However, one can wonder how it is possible once the shape of an object has been recognised to infer its function and functionality allowing preventing from crime and helping prosecute criminals. This will be explained in details in section 6-2-From object's shape to object classification.

First, the next section provides a quantitative analysis of the results obtained to complement the qualitative analysis described in this section and evaluates strengths and limitations of the proposed approach.

6-1-4-Discussion

This section presents a quantitative analysis of the results obtained from experiments described in the previous sections. While previous sections provided an illustration of experiments performed and surface normals recovered, this section shows the efficiency of the proposed method to recognise shapes according to a percentage of success.

The table 24 below presents the percentage of shape recognised and false recognition obtained on the experiment where the three simple shapes placed on the cardboard have had their location the interchanged from left to right and top to bottom.

	Recognised			**False recognition**		
	Half-Cylinder	Prism	Cuboid	Half-Cylinder	Prism	Cuboid
Total	5	1	9	7	11	3
%	41.7	8.33	75	58.33	91.7	25

Table 24: Success rate of results obtained from a short-range

This table shows the success rate of the shape recognition for the cuboid is the best rate with 75% whereas for the half-cylinder, it has only been recognised at 41.7 % and for the prism at 8.33%. This analysis in addition to the qualitative analysis proposed in the previous section confirms results obtain for a short-range do not allow recognising well the shape of all considered objects. As seen in the previous section, the distribution of the light has been identified as the cause of such rates; indeed, a shorter range does not allow the light to be distributed evenly on the target as a longer range does.

The next table 25 below represents results obtained with the cardboard and the interchanged position of the three shapes but this time for a long-range.

	Recognised			False recognition		
	Half-Cylinder	Prism	Cuboid	Half-Cylinder	Prism	Cuboid
Total	10	6	12	2	6	0
%	83.33	50	100	16.7	50	0

Table 25 : Success rate of results obtained from a long-range

On the contrary to the success rates obtained for a short-range, the cuboid is now recognised all the time (100%), whereas the half-cylinder is recognised most of the time (83.33%) and the prism half of the time (50%). These results confirm especially that the distance between the camera and the object affects them considerably. Thus, it is important to use this system within a long-range rather than a short-range or to use less powerful or smaller light sources in the case of short-range applications. Moreover, the mean value of the Euclidean distance was used to recognise object's shape and another solution to improve these results could be to use another method than the mean that would be more representative of the variation of the surface normals along the surface.

The next table presents the results obtained from the shapes attached to the shirt with and without rotation. Note that these results are preliminary and intended as a proof-of-concept only (a detailed study is reserved for future work). Here, it has been more difficult to present a global success rate as the experiments differ from each other. However, as a further analysis, the objects' shape recognised and false recognition are summarized in the following table for the following experiment: the three shapes placed on the shirt, same experiment except the shirt is tight in the back, same experiment as the previous with rotation of the shirt from one side and the other.

	Recognised			False recognition		
	Half-Cylinder	Prism	Cuboid	Half-Cylinder	Prism	Cuboid
SHIRT	1	0	1	0	1	0
SHIRT TIGHT	0	0	1	1	1	0
SHIRT ROTATE1	0	0	0	1	1	1
SHIRT ROTATE2	1	0	0	0	1	1

(a)

	Recognised	False Recognition
Knife	1	0
Umbrella	1	0

(b)

Table 26: (a) Results obtained from a long-range for the four experiments with the shirt (b) Results obtained from a long-range for the knife and umbrella

This table shows the shape have been recognised more easily when the shape are placed on the shirt than when it was tight in the back and rotated as well. This can be explained by the fact that when the shirt is rotated from one side or another, the shapes are not completely in the field of view of the camera anymore. Then, part of the shapes is hidden and the part that is visible is not sufficient to recognise the object's shape as surface normals are not sufficient anymore to address this task. Indeed, this approach has only been tested and designed for horizontal objects that are in the field of view of the camera. Due to time constraint, the capture of data and analysis of objects with an orientation different from horizontal has not been done. However, it is possible to extend this approach to other object's orientation such as when they are vertical or rotated with a random angle or even occluded by other objects or partly in the field of view of the camera. This can be done by using the same approach to analyse the surface normals of the objects using additional information about objects and the scene such as boundaries between objects based on feature extracted, i.e. VPs and lines to determine the objects geometry and boundaries, colour or texture thanks to segmentation, dimensions of the area of interest in terms of pixels etc. Moreover, the creation of additional GT shape will be required to allow the comparison of GT and Experimental data for objects with various orientations. In the case where experimental data correspond to an object with an orientation different from the orientation of the object considered as GT data, the amount of data will be different. Then, it will be easy to conclude as for they are not the same object. In the case where the orientation of the objects considered for the GT and experimental data will be the same, the comparison will be more complex in some cases such as to differentiate a cube and cuboid as in some orientation they can both appear to be cubes which can also be the case for real objects such a knife and an umbrella in case of occlusions. A solution to address this issue seems to be the use of other viewpoint of the object and in absence of other viewpoint of the object, only the object characteristics will allow to make a difference between one another. It is obvious that higher the number of known or extracted characteristics, better will be the accuracy to determine the type of object, the object of interest is. This is because even if objects share similar characteristics, some properties such as colour, texture or even functionality, will allow differentiating them. So, even in absence of sufficient information to deal with occluded objects, additional and meaningful information can always be extracted to assist object's shape recognition. Thus, the combination of 2D and 3D information is a plus for scene understanding and can help video surveillance systems to be more intelligent and efficient even in presence of noise, occlusions and change in exterior conditions.

An example of failure detection due to different orientation of considered object concerns the two shape recognition experiments have been performed to compare the knife and the umbrella. One was successful as illustrated on table 26 (b) whereas the other detected the knife twice. As explained in the previous section, this is a failure case due to a variation in the orientation of the umbrella compared to the knife that remained horizontal and perpendicular to the field of view of the camera. More experiments as well as a complete analysis related to various orientations of objects of interest will be explored in the future.

Other experiments will be part of the future work where random noise will be added to the image, the scene itself (occlusions, shadow etc.) and/or surface normals to determine the accuracy and robustness of the algorithm in extreme conditions. This way, it will be even more representative of data extracted from real-world scenario. This will also allow analysing the effect of noise on data according to where it originated from, i.e. within the scene, within the image, within the surface normals, in the aim to find a method to overcome it. All these additional experiments will be done in the future to make the proposed approach more complete and closer to a commercialisation prototype.

While this section has concluded as for the results obtained and strengths and limitations, the next section describes case scenarios where this method can be applied.

6-2- From objects' shape to object classification

This section describes how the shape of objects can assist for object classification and scene understanding.

As seen previously, surface normals can be used to recognise object's shape and give promising results when used within a long-range. The study case considered here is railway and underground stations that are monitored by video surveillance cameras. Thus, the proposed approach could be easily added to existing cameras by only adding two light sources near each camera if possible. Moreover, these cameras are already monitoring the scene within a long-range which justifies the use of the proposed approach in the context of video surveillance.

Moreover, there is only little that can be done to help refocus the camera or use a camera with another viewpoint to have a better image of an object. However, the use of light-source in addition to existing cameras could help to recognise object's shape within a long-range by applying photometric stereo to recover surface normals of an object of interest.

Then, as shown in the previous sections, once the object's shape is recognised, a question remains: how is it possible to recognise objects from their shape?

For comprehension purposes, an example of type of objects that can be found in a railway station is given in the table 27 below as well as their characteristics.

Characteristics	Shape	Size	Example colour	Material/Texture	Function	Functionality
Bench	Plane, cuboid or circle	medium	Grey or brown	Wood and/or metal	Seat	People Sitting
Shelter	Cuboid	large		Glass and metal	Waiting area	People standing or sitting
Dustbin	Cuboid or cylinder	small	Black	Plastic, metal or wood	Container	Throw papers etc. away

Table 27 : Specific objects to railway and underground stations and their characteristics

These objects are all composed of planar surfaces apart from the dustbin that can either be square or cylindrical. Thus, surface normals might not be sufficient in the case of various objects with similar shapes to infer their function once the shape has been recognised. That is where additional characteristics as those mentioned in the table 27 above can assist the proposed approach to infer what objects are as benches, shelters and bins usually differ from the material, colour, texture they are made of or their dimension. Thus, an example of application for the proposed approach is given below for comprehension purposes.

While the security personnel spend days looking at screens in the control room to identify potential crimes, threats, abnormal behaviour or dangerous objects, their task remains excessive due to the amount of data recorded in addition to the complexity and context of the scenes monitored that makes it difficult to detect a specific event or object. The goal of the proposed approach is then to help prevent crime and aid criminal prosecution. This can be done if it is integrated within an existing video surveillance system that contains a human detection method. Then, once a potential threat has been detected by the system by a human operator, the relevant two light sources are illuminated in turn, focussed to the target using the results of (for example) VP-based calibration and the algorithm is applied to recover the surface normals. The security guard can then either use the recovered surface normal map directly to assess the danger, or apply the shape recognition algorithm above.

Using a classifier can help infer the object function according to the characteristics extracted; i.e. if the characteristics extracted: colour, texture, size and shape corresponds to those of the bench as in the table above, the object is classified as a bench and so on. Many classifiers exist but the tree classifier is of interest as it is based on the principle of a family tree. Thus, objects from a specific environment can be identified as member of a family and the tree classifier allow classifying each object of interest to a specific member based on a metric, i.e. probability here, on the similarity between its characteristics and those of the specific member to match. Once the object has been classified, it implies the knowledge of its function and functionality. The functionality of objects is here defined as the human behaviour implied by the function of an object, e.g. a bench is a seat so someone is expected to sit on the bench as normal behaviour and any other type of behaviour would be considered as abnormal and dangerous. An example of functionality for the bench is illustrated by the table 28 below.

	Functionality	
	Normal behaviour	Abnormal behaviour
Object of interest, i.e. here a bench.	• Someone seats on the bench. • A bag can be placed on the bench.	• Someone is lying on the bench. • Someone stands up on the bench. • A bag is abandoned on the bench.

Table 28: Example of functionality for a specific object, here a bench

Moreover, this can either apply to an isolated object of interest or can be applied to the whole scene with the aim to create a map of functionality where each object present within the scene will be classified and associated with functionality. This map of functionality can then assists security personnel and improve current video surveillance efficiency for crime prevention and criminal prosecution. It can also be applied in the case of someone holding a knife to make the difference between a knife and an umbrella which can be confused by an operator due to the poor quality of video surveillance sequences but could be differentiated thanks to the proposed approach. More details about possible applications of the proposed approach will be described in chapter 8 : conclusion. The application described in this section goes beyond the scope of the approach proposed and will not be developed for the purpose of this thesis due to time constraint. It will be developed as future work to make the proposed approach more complete.

6-3-Summary on 3D analysis

The achievements of this chapter concern the extraction of 2.5D feature such as surface normals and its combination to 2D feature extraction and segmentation proposed in the previous chapters.

A subjective summary (extensive numerical analyses have already been presented) of the proposed approach to recognise object's shape is illustrated by the following table:

	Computation	Accuracy	Robustness to change in exterior conditions	Robustness to noise	Practicality
Object's shape recognition	Inexpensive	Accurate for short-range but higher accuracy for long-range	Robust due to the use of controlled light-sources in addition to exterior conditions	Robust	No specific requirements

Table 29 : Evaluation of the proposed approach to be used as a future product. NB. Practicality refers to the ease by which the methods can be commercially deployed.

This table evaluates strengths and limitations of the proposed object's shape recognition approach. Please note however, that these are relative and subjective only to give an idea as to where they are considered to be close to commercial use ("Good") and still fall far short of requirements ("Bad").

The results obtained with the proposed approach have been proven to provide very accurate results for long-range experiments that are most of interest in the case of video surveillance scenes. Moreover, several applications to use this approach have been described and are feasible in terms of real deployment. The use of 2.5D data has been proven to be richer than 2D data as it allows classifying objects from shape information. Furthermore, it can either be applied as a whole or be combined with 2D feature extraction and segmentation methods proposed to provide a more complete system and make current video systems more intelligent.

One of the limitations of this approach is that it has only been tested for horizontal objects and realistic but very simplistic scenes. Indeed, someone holding a knife would not probably hold it in his hand for it to be visible by everyone to avoid being arrested and stopped before he succeeded to commit a crime. Furthermore, abnormal behaviour usually involves motion and this approach requires further investigation for its application to dynamic objects.

To conclude, the proposed approach has a real potential to assist for scene understanding and represent a novel contribution for the research community as well as industry. However, the public and organisations such as the police expect different output or have a different perception of CCTV than industry and academia. Before concluding on the global proposed approach, the next section provides details of the expectation and beliefs about CCTV from the police and the public.

Chapter 7: Conclusion

This thesis has demonstrated the significance of combining 2D with 2.5D/3D data to tackle the common challenges of 3D scene understanding in video surveillance. Moreover, the combination of Photometric-Stereo and improved existing methods to perform 2D feature extraction and segmentation, has been proven to increase the robustness and accuracy of the algorithm to perform 3D scene understanding in the context of video surveillance as these methods complement one another. This thesis has shown that Photometric-stereo allows for object shape recognition even in the presence of noise and changeable exterior conditions. It has been shown to be a valuable tool for 3D scene understanding as a whole or combined with 2D features when required for better efficiency and robustness. Compared to other methods, the main advantages of this approach is that it can easily be applied to more complex environments such as real world scenarios and make CCTV systems more intelligent and robust to the type of environment considered. Thus, it gives the possibility to prevent and predict abnormal behaviour based on a specific map of functionality to the considered environment. The main issues of this approach relate to the absence of a complete view or in presence of a distorted view of an object of interest. However, solutions to adapt the approach to these specific cases are proposed and will be investigated in the near future.

In this chapter, the main findings and key contributions of the proposed approach are summarized. Limitations of the proposed approach are evaluated and potential solutions for improvements are proposed. Finally, possibilities for future work and various applications of this approach are given.

The next section of this chapter summarizes the main findings and contributions of each chapter.

7-1-Thesis Summary and limitations

The main findings and key contributions of each chapter of this thesis are summarized below:

- Chapter 4 showed common 2D feature extraction methods can be applied to a more complex environment to accurately detect VPs only if pre-processing and a specific clustering and voting scheme to type of considered scene are involved. It also demonstrated 2D features such as lines, VPs and colour information, are very limited to perform accurate image analysis in environments where noise and change in exterior conditions are common.
- To solve the issues of 2D, Chapter 5 investigated two methods to acquire 3D data on which a proposed plane extraction approach has been performed. However, the result of this investigation showed the poor quality of extracted 3D information, due to noise and change in exterior conditions and the capturing of such 3D features in real railway and underground stations would be difficult.
- Chapter 6 explored the use of Photometric-stereo to assist 3D scene understanding in the context of video surveillance scenes. Moreover, it shows the role of surface normals for object shape recognition to infer object’s function and functionality. It has demonstrated that the combination of 2.5D/3D and 2D data is the key to tackle issues due to noise and change in exterior conditions.

- Chapter 7 describes the public and police perception and expectations of the use of CCTV in public places and the current requirements. It allows the proposed approach to fall into place between the research community and industrials for public's safety.

This thesis therefore offers a key step towards better and more intelligent video surveillance systems to perform 3D scene understanding. However, it needs to be adapted further to be more general and be able to recognise an object's shape even if only a partial or distorted view of an object of interest is available. This issue represents a real challenge to be solved, especially as it often happens that CCTV footage provides only partial or distorted view of objects due to occlusions, image quality, noise, change in exterior conditions and other factors.

The next section proposes solutions to tackle the limitations of the proposed approach and describes potential applications.

7-2-Future work and possible applications

Several avenues for future work are described below:

- The development of a camera calibration method based on VPs to recover the distance between an object of interest and the camera as well as the size of objects.
- A classification method to infer object's function from their shape;
- The development of a map of functionality of the considered environment based on object's characteristics such as shape, colour, size, texture, function and functionality;
- The adaptation of the proposed approach to objects with various orientations using additional characteristics and a machine learning. Moreover, in some cases, a tracking algorithm would allow to differentiate an object from another when occluded and sharing similar properties if one of the other or the two have been previously tracked and detected in other frames of the video sequence analysed;
- The adaptation of the proposed approach to other types of environment such as airports, shopping centres, car parks etc.
- The deployment of the proposed approach to commercial use.

The idea of the map of functionality has been inspired by the work proposed by Winston et al. [174] in 1983 and Gupta et al. [173] more recently.

The former introduces the notion of functionality by looking at the functional representation of an object rather than their shapes, colour etc. They define functional by the function of the object, e.g. a mug can be blue, red or green, small, medium or large, with or without handles but these characteristics make the object recognition task difficult. However a common characteristic of every mug is their functionality as they are all used to drink tea or coffee etc.

On the contrary, the Gupta et al. paper examined the scene geometry, i.e. to predict possible human actions in a specific environment based on the concept of the "workspace of a robot". This is made possible by looking at the objects' function instead of their identity (e.g. identity: a chair, function: sitting) to lead to a joint space human-scene interactions.

Both approaches inspired the proposed approach to lean towards the use of functionality to address 3D scene understanding issues. Thus, it led to a variety of applications described below.

These applications all are related to contextualisation or scene understanding in the sense that it puts objects or humans into context from information extracted from the scene. Methods chosen to proceed to contextualisation in the context of video surveillance and in particular railway and underground stations were VP detection, colour segmentation, plane extraction and surface normal analysis.

For any applications, the common operation scenario of the proposed approach is the one that follows:

1. VP extraction.
2. Camera calibration using VPs extracted in order to estimate the distance between a target and the camera as well as the size of the target.
3. User intervention: security personnel switch on the light sources according to where the object of interest is located, determined in step 2.
4. Recovery and analysis of the surface normals of the object of interest.
5. Object's shape recognition using a tree classifier; this can be done for objects with various orientations thanks to the classifier that will allow differentiating an object from another thanks to additional characteristics such as colour, texture, boundaries, areas of the object in term of pixels etc.
6. Inference of the object's function and then functionality.
7. Creation of a map of functionality in case the previous cases have been applied to the whole scene instead of an object of interest.
8. This step can lead to various possibilities such as:
    a. Differentiation between dangerous from non-dangerous objects using the Map of Functionality, this implies an alarm is raised when a dangerous object has been detected and can help prosecute a criminal in public or private places such as airports, schools, hospitals, factories etc.;
    b. Differentiation between abnormal from normal behaviours based on the combination of the map of functionality and an existing human detection framework; this implies an alarm is raised when abnormal behaviour has been detected and can help prosecute a criminal in public or private places such as airports, schools, hospitals, factories etc.;
    c. Detection of missing item such as in airports, railway station, shops, homes etc.
    d. Detection of specific items or features on wanted criminals such as tattoo, hats, clothes etc.

Thus, this approach can be applied to the following scenarios in the case of railway and underground stations:

- Left luggage detection;
- Dangerous object detection such as knife, bombs etc;
- Event detection such as threats, murder, assault, fall etc.;

- Abnormal behaviour detection: if combined to a human detection system, the map of functionality allows determining if someone has violated a functionality of an object and then be pursued and prosecuted;
- The scenarios above thanks to a portable device that would allow the police to recover all of the information related to an object and look for a specific event related to this object in the database.

To conclude, in spite of limitations, the proposed approach allows us to solve some challenges of 3D scene understanding in the context of video surveillance thanks to:

- An investigation into computer vision methods' literature and state-of-the-art in terms of current CCTV systems;
- A collaboration between academia and industry;
- Insights from security personnel, the police and the public towards CCTV;
- The combination of 2D and 3D data;
- A novel concept to formalise the perception of objects within a specific environment.

References

[1] http://people.csail.mit.edu/torralba/courses/6.870/slides/lecture5.pdf, Last accessed (17.07.2014)

[2] http://en.wikipedia.org/wiki/Computer_vision, Last accessed (20.03.2014)

[3]Hartley, Richard, and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[4] Comaniciu, D.; Meer, P.; , "Mean shift: a robust approach toward feature space analysis," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.5, pp.603-619, May 2002

[5] Kulis, B.; Saenko, K.; Darrell, T.; , "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.1785-1792, 20-25 June 2011

[6] Barroso, J.; Dagless, E.L.; Rafael, A.; Bulas-Cruz, J.; , "Number plate reading using computer vision," Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on , vol., no., pp.761-766 vol.3, 7-11 Jul 1997

[7] Belbachir, A.N.; Schraml, S.; Nowakowska, A.; , "Event-driven stereo vision for fall detection," Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on , vol., no., pp.78-83, 20-25 June 2011

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, James Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age”, ACM Comput. Surv., Vol. 40, No. 2. (May 2008), pp. 1-60

[9] Erik Murphy-Chutorian, Mohan Manubhai Trivedi, "Head Pose Estimation in Computer Vision: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 607-626, April, 2009.

[10] Wenyi Zhao, Rama Chellappa and Peter O. Stubler, "Face Processing: Advanced Modeling and Methods", J. Electron. Imaging 15, 049901 (27 November 2006)

[11] D. M. Gavrila, “The Visual Analysis of Human Movement: A Survey”, Computer Vision and Image Understanding Vol. 73, No. 1, January, pp. 82–98, 1999

[12] Thomas B. Moeslund, Adrian Hilton, and Volker Kr\&\#252;ger. 2006. “A survey of advances in vision-based human motion capture and analysis”. Comput. Vis. Image Underst. 104, 2 (November 2006), 90-126

[13] P. Thévenaz, “Pattern Recognition and Image Processing in Physics”, NATO Advanced Study Institute, Proceedings of the Thirty-Seventh Scottish Universities Summer School in Physics, Dundee, UK, July 29-August 18, 1990, R.A. Vaughan, Ed., Adam Hilger, Bristol, pp. 129-166.

[14] Florian Raudies, Heiko Neumann, “A review and evaluation of methods estimating ego-motion”, Computer Vision and Image Understanding, Volume 116, Issue 5, May 2012, Pages 606-633

[15] Tian, T.; Tomasi, C.; Heeger, D., "Comparison of Approaches to Egomotion Computation", 1996, IEEE Computer Society Conference on Computer Vision and Pattern Recognition: 315.

[16] Hanxuan Yang , Ling Shao , Feng Zheng , Liang Wang, Zhan Song, “Recent advances and trends in visual tracking: A review”, Neurocomputing 74 (2011) 3823–3831

[17] Berthold K.P. Horn and Brian G. Rhunck, “Determining Optical Flow”, Artificial Intelligence, 1981, vol. 17, pages 185-203.

[18] Geiger, A.; Lauer, M.; Urtasun, R.; , "A generative model for 3D urban scene understanding from movable platforms," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.1945-1952, 20-25 June 2011

[20]: http://www.araliasystems.com/index.htm, Last accessed (20.03.2014)

[21.] http://www.vcatechnology.com/, Last accessed (20.03.2014)

[22.] http://www.ipsotek.com/, Last accessed (20.03.2014)

[23.] http://www.nice.com/video, Last accessed (20.03.2014)

[24.] http://www.indigovision.com/, Last accessed (20.03.2014)

[25.] http://www.genetec.com/, Last accessed (20.03.2014)

[26.] http://www.avi-infosys.com/CCTV.html, Last accessed (20.03.2014)

[27.] http://www.synecticsuk.com/, Last accessed (20.03.2014)

[28] Dee, H. M., & Velastin, S. A. (2008). How close are we to solving the problem of automated visual surveillance?. *Machine Vision and Applications*, *19*(5-6), 329-343.

[29] http://cds.cern.ch/record/400313/files/p21.pdf, Last accessed (20.04.2014)

[30] http://hippie.nu/~nocte/tutorial-currentchapter/xhtml-chunked/ch01s02.html, Last accessed (20.04.2014)

[31] R.M. Haralick, “Using perspective transformation in scene analysis”, Computer Graphics Image Processing, 13:191-221, 1980

[32] S.T. Barnard, “Methods for interpreting perspective images2”, In Proc. Image Understanding Workshop, pages 193-203, Palo Alto, California, Sept 1982

[33] M.J. Magee and J.K. Aggarwal, “Determining vanishing points from perspective images”, Computer Vision, Graphics and Image Processing, 26:256-267, 1984

[34] L. Quan and R. Mohr,” Determining perspective structures using hierarchical Hough transform”, Pattern Recognition Letters, 9(4):279-286, 1989

[35] R.T. Collins and R.S. Weiss, “Vanishing point calculation as a statistical inference on the unit sphere”, Computer Vision, Graphics and Image Processing, pages 400-403, 1990

[36] A Tai, J Kittler, M Petrou, T Windeatt, "Vanishing point detection", Image and Vision Computing, Volume 11, Issue 4, May 1993, Pages 240–245

[37] Lutton, E.; Maitre, H.; Lopez-Krahe, J., "Contribution to the determination of vanishing points using Hough transform," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.16, no.4, pp.430,438, Apr 1994

[38] McLean, G.F.; Kotturi, D., "Vanishing point detection by line clustering," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.17, no.11, pp.1090,1095, Nov 1995

[39] Tuytelaars, T.; Van Gool, L.; Proesmans, M.; Moons, T., "The cascaded Hough transform as an aid in aerial image interpretation," Computer Vision, 1998. Sixth International Conference on, vol., no., pp.67,72, 4-7 Jan 1998

[40] Frank A. Van Den Heuvel, "Vanishing Point Detection For Architectural Photogrammetry", Proc. ECCV '02, 1998,652—659

[41] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes", in Proc. IEEE Conference on Computer Vision and Pattern Recognition, June 1998, pages 482-488

[42] F. Schaffalitzky, A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points",Image and Vision Computing, Volume 18, Issue 9, June 2000, Pages 647–658

[43] Carsten Rother, "A new approach to vanishing point detection in architectural environments", Image and Vision Computing, Volume 20, Issues 9–10, Pages 647–655, 1 August 2002

[44] Qiang He; Chee-Hung Henry Chu, "An Efficient Vanishing Point Detection by Clustering on the Normalized Unit Sphere," Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on , vol., no., pp.203,207, 18-20 Aug. 2006

[45] Frank Schmitt, Lutz Priese, "Vanishing Point Detection with an Intersection Point Neighborhood", Discrete Geometry for Computer Imagery Lecture Notes in Computer Science, Volume 5810, 2009, pp 132-143

[46] Almansa, A., Desolneux, A., Vamech, S., "Vanishing point detection without any a priori information", IEEE Trans. Pattern Anal. Mach. Intell. 25(4) (2003) 502–507

[47] Seo, K.S., Lee, J.H., Choi, H.M.: An efficient detection of vanishing points using inverted coordinates image space. Pattern Recogn. Lett. 27(2) (2006) 102–108

[48] Tardif, J.-P., "Non-iterative approach for fast and accurate vanishing point detection," Computer Vision, 2009 IEEE 12th International Conference on , vol., no., pp.1250,1257, Sept. 29 2009-Oct. 2 2009

[49] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-Linkage", In European Conference on Computer Vision, pages 537–547, 2008

[50] Xuehui Chen; Ruiqing Jia; Hui Ren; Yinbin Zhang, "A New Vanishing Point Detection Algorithm Based on Hough Transform," Computational Science and Optimization (CSO), 2010 Third International Joint Conference on , vol.2, no., pp.440,443, 28-31 May 2010

[51] Marcos Nieto, Luis Salgado, "Simultaneous estimation of vanishing points and their converging lines using the EM algorithm", Pattern Recognition Letters, Volume 32, Issue 14, 15 October 2011, Pages 1691–1700

[52] Bazin, J. -C; Yongduek Seo; Demonceaux, C.; Vasseur, P.; Ikeuchi, K.; Inso Kweon; Pollefeys, M., "Globally optimal line clustering and vanishing point estimation in Manhattan world," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , vol., no., pp.638,645, 16-21 June 2012

[53] Ebrahimpour, R.; Rasoolinezhad, R.; Hajiabolhasani, Z.; Ebrahimi, M., "Vanishing point detection in corridors: using hough transform and K-means clustering," Computer Vision, IET, vol.6, no.1, pp.40,51, January 2012

[54] Gerogiannis, D.; Nikou, C.; Likas, A., "Fast and efficient vanishing point detection in indoor images," Pattern Recognition (ICPR), 2012 21st International Conference on , vol., no., pp.3244,3247, 11-15 Nov. 2012

[55] S. Liou and R. Jain, “Road following using vanishing points”, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 41-46, 1986

[56] B. Caprile, V. Torre, "Using vanishing points for camera calibration", International Journal of Computer Vision March 1990, Volume 4, Issue 2, pp 127-139

[57] L. Wang and W. Tsai. Computing camera parameters using vanishing line information from a rectangular parallelepiped. Machine Vision and Applications, 3:129-141,1990.

[58] Wang, L.L., Tsai, W.H., “Camera calibration by vanishing lines for 3-d computer vision”, IEEE Transactions on Pattern Analysis and Machine Intelligence 13(4), 370{376 (1991)

[59] B. O'Mahony, “A Probabilistic Approach to 3D Interpretation of Monocular Images”, PhD thesis, City University, London, 1992.

[60] L. Shigang, S. Tsuji, and M. Imai, “Determining of camera rotation from vanishing points of lines on horizontal planes”, Computer Vision, Graphics and Image Processing, pages 499-502, 1990

[61] Paul Beardsley, David Murray, "Camera Calibration using Vanishing Points", BMVC92, 1992, pp 416-425

[62] Jonathan Deutscher, Michael Isard, John Maccormick, “Automatic camera calibration from a single manhattan image”, Eur. Conf. on Computer Vision (ECCV), 2002

[63] Lazaros Grammatikopoulosa, George Karras, Elli Petsa, "An automatic approach for camera calibration from vanishing points", ISPRS Journal of Photogrammetry and Remote Sensing, Volume 62, Issue 1, May 2007, Pages 64–76

[64] N. Avinash, S. Murali, "Perspective Geometry Based Single Image Camera Calibration", Journal of Mathematical Imaging and Vision March 2008, Volume 30, Issue 3, pp 221-230

[65] Sung Chun Lee; Nevatia, R., "Robust camera calibration tool for video surveillance camera in urban environment," Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on , vol., no., pp.62,67, 20-25 June 2011

[66] D. Svedberg, S. Carlsson, “Calibration, Pose and Novel Views from Single Images of Constrained Scenes”, Proceedings of the 11th Scandinavian Conference on Image Analysis (SCIA’99), June 1999, Kangerlussuaq, Greenland, 1999, pp. 111–117

[67] R. Cipolla, T. Drummond, and D. Robertson, “Camera calibration from vanishing points in images of architectural scenes”, In BMVC’99

[68] E Guillou, D. Meneveaux, E. Maisel, K. Bouatouch, "Using vanishing points for camera calibration and coarse 3D reconstruction from a single image", The Visual Computer, November 2000, Volume 16, Issue 7, pp 396-410

[69] A. Criminisi, I. Reid, A. Zisserman, "Single View Metrology", International Journal of Computer Vision, November 2000, Volume 40, Issue 2, pp 123-148

[70]: http://en.wikipedia.org/wiki/Image_segmentation, Last accessed (28.05.2014)

[71] Cheng Heng-Da, et al. "Color image segmentation: advances and prospects." PR 34.12 (2001): 2259-2281.

[72] Władysław Skarbek and Andreas Koschan, Colour Image Segmentation: A Survey,1994

[73] Dey, V., Y. Zhang, and M. Zhong. "A review on image segmentation techniques with remote sensing perspective." Proceedings of the International Society for Photogrammetry and Remote Sensing Symposium, Vol. 38. 2010.

[74] J. Freixenet and X. Muñoz and D. Raba and J. Martí and X. Cufí, Yet another survey on image segmentation: Region and boundary information integration, In ECCV, 2002, 408--422

[75] Robert M. Haralick ; Linda G. Shapiro; Image Segmentation Techniques. Proc. SPIE 0548, Applications of Artificial Intelligence II, 2, April 5, 1985;

[76] Tripathi, Shraddha, et al. "Image segmentation: A review." International Journal of Computer Science and Management Research 1.4 (2012).

[77] Narkhede, H. P. "Review of Image Segmentation Techniques." International Journal of Science and Modern Engineering 1.8 (2013): 54-61.

[78]http://sensors-actuators-info.blogspot.co.uk/2009/08/laser-triangulation-sensor.html, Last accessed (27.03.2014)

[79] http://en.wikipedia.org/wiki/Triangulation, Last accessed (27.03.2014)

[80] Hsieh, Jun-Wei, et al. "Video-based human movement analysis and its application to surveillance systems." Multimedia, IEEE Transactions on 10.3 (2008): 372-384.

[81] Scott, William, Gerhard Roth, and Jean-François Rivest. "View planning for automated 3D object reconstruction inspection." ACM Computing Surveys 35.1 (2003).

[82] http://www.metrilus.de/range-imaging/time-of-flight-cameras/, Last accessed (27.03.2014)

[83] http://en.wikipedia.org/wiki/Time-of-flight_camera, Last accessed (27.03.2014)

[84] Amann M, Bosch T, Myllyla¨ R, Rioux M, Lescure M; Laser ranging: a critical review of usual techniques for distance measurement. Opt. Eng. 0001;40(1):10-19, 2000.

[85] Bevilacqua, Alessandro, Luigi Di Stefano, and Pietro Azzari. "People Tracking Using a Time-of-Flight Depth Sensor." AVSS. Vol. 6. 2006.

[86] Caraian, Sonja, and Nathan Kirchner. "Robust manipulability-centric object detection in time-of-flight camera point clouds." Proceedings of the Australasian conference on robotics and automation. 2010.

[87] Wei, Xue, Son Lam Phung, and Abdesselam Bouzerdoum. "Pedestrian sensing using time-of-flight range camera." Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on. IEEE, 2011.

[88] http://en.wikipedia.org/wiki/Lidar, Last accessed (27.03.2014)

[89] http://www.csc.noaa.gov/digitalcoast/_/pdf/lidar101.pdf, Last accessed (27.03.2014)

[90] http://www.lidar-uk.com/how-lidar-works/, Last accessed (27.03.2014)

[91] Juan C. Suárez, Carlos Ontiveros, Steve Smith, Stewart Snape, "Use of airborne LiDAR and aerial photography in the estimation of individual tree heights in forestry", Computers & Geosciences, Volume 31, Issue 2, March 2005, Pages 253–262, Geospatial Research in Europe: AGILE 2003

[92] Himmelsbach, Michael, et al. "Lidar-based 3d object perception." Proceedings of 1st International Workshop on Cognition for Technical Systems. Vol. 1. 2008.

[93] Ko, T., "A survey on behavior analysis in video surveillance for homeland security applications," Applied Imagery Pattern Recognition Workshop, 2008. AIPR '08. 37th IEEE , vol., no., pp.1,8, 15-17 Oct. 2008

[94] http://en.wikipedia.org/wiki/Computer_stereo_vision, Last accessed (27.03.2014)

[95] http://www.cse.unr.edu/~bebis/CS791E/Notes/StereoCamera.pdf, Last accessed (27.03.2014)

[96] Xing-zhe, Xie, et al. "3D terrain reconstruction for patrol robot using point grey research stereo vision cameras", Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on. Vol. 1. IEEE, 2010.

[97] Aguilar, Juan-José, F. Torres, and M. A. Lope. "Stereo vision for 3D measurement: accuracy analysis, calibration and industrial applications." Measurement 18.4 (1996): 193-200.

[98] Georgoulas, Christos, and Ioannis Andreadis. "Real-Time Stereo Vision Techniques." Proceedings of 16th IFIP/IEEE International Conference on Very Large Scale Integration, Rhodes, Greece. 2008.

[99] Muñoz-Salinas, Rafael, Eugenio Aguirre, and Miguel García-Silvente. "People detection and tracking using stereo vision and color." Image and Vision Computing 25.6 (2007): 995-1007.

[100] http://en.wikipedia.org/wiki/Structured_Light_3D_Scanner, Last accessed (27.03.2014)

[101] http://en.wikipedia.org/wiki/Structured_light, Last accessed (27.03.2014)

[102] Mouaddib, E.; Batlle, J.; Salvi, J., "Recent progress in structured light in order to solve the correspondence problem in stereovision," Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on , vol.1, no., pp.130,136 vol.1, 20-25 Apr 1997

[103] http://www.3dmd.com/3dMDface/, Last accessed (27.03.2014)

[104] Kawasaki, H.; Furukawa, R.; Sagawa, R.; Yagi, Y., "Dynamic scene shape reconstruction using a single structured light pattern," Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on , vol., no., pp.1,8, 23-28 June 2008

[105] Izadi, Shahram, et al. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011.

[106] Geng, Jason. "Structured-light 3D surface imaging: a tutorial." Advances in Optics and Photonics 3.2 (2011): 128-160.

[107] Shin, Yong-Deuk, et al. "Moving objects detection using freely moving depth sensing camera." Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.

[108] Guan, C., L. Hassebrook, and D. Lau. "Composite structured light pattern for three-dimensional video." Optics Express 11.5 (2003): 406-417.

[109] http://www.3d-shape.com/presse/docs_e/sfshading_05_e.pdf, Last accessed (27.03.2014)

[110] Horn, Berthold KP. "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view." (1970).

[111] Horn, B.K.P., "Obtaining Shape from Shading Information," chapter 4 in The Psychology of Computer Vision, Winston, P. H. (Ed.), McGraw-Hill, New York, April 1975, pp. 115-155.

[112] Ikeuchi, Katsushi, and Berthold KP Horn. "Numerical shape from shading and occluding boundaries." Artificial intelligence 17.1 (1981): 141-184.

[113] Atkinson, Gary A., and Melvyn L. Smith. "Using Photometric Stereo for Face Recognition." International Journal of Bio-Science & Bio-Technology 3.3 (2011).

[114] Sun, Jiuai. "Counter camouflage through the removal of reflectance." (2010).

[115] T. Rindfleisch. Photometric method for lunar topography. Photogram. Eng., 32:262–277,

1966.

[116] Goldman, Dan B., et al. "Shape and spatially-varying brdfs from photometric stereo." Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.6 (2010): 1060-1071.

[117] Vlasic, Daniel, et al. "Dynamic shape capture using multi-view photometric stereo." ACM Transactions on Graphics (TOG) 28.5 (2009): 174.

[118] Hartley, Richard I., and Peter Sturm. "Triangulation." Computer vision and image understanding 68.2 (1997): 146-157.

[119] Sequeira, Vitor, et al. "Automated reconstruction of 3D models from real environments." ISPRS Journal of Photogrammetry and Remote Sensing 54.1 (1999): 1-22.

[120] Chou, George Tao-Shun. Large-scale 3D reconstruction: A triangulation-based approach. Diss. Massachusetts Institute of Technology, 2000.

[121] Gokturk, S. Burak, Hakan Yalcin, and Cyrus Bamji. "A time-of-flight depth sensor-system description, issues and solutions." Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. IEEE, 2004.

[122] Mure-Dubois, James, and Heinz Hügli. "Fusion of time of flight camera point clouds." Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008. 2008.

[123] Blais, François. "Review of 20 years of range sensor development." Journal of Electronic Imaging 13.1 (2004).

[124] Vincent, R., and Michael Ecker. Light detection and ranging (LiDAR) technology evaluation. No. OR11-007. 2010.

[125] Mouaddib, E.; Batlle, J.; Salvi, J., "Recent progress in structured light in order to solve the correspondence problem in stereovision," Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on , vol.1, no., pp.130,136 vol.1, 20-25 Apr 1997

[126] Brown, Myron Z., Darius Burschka, and Gregory D. Hager. "Advances in computational stereo." Pattern Analysis and Machine Intelligence, IEEE Transactions on 25.8 (2003): 993-1008.

[127] Woodham, Robert J. "Photometric stereo: A reflectance map technique for determining surface orientation from image intensity." 22nd Annual Technical Symposium. International Society for Optics and Photonics, 1979.

[128] Dhond, Umesh R., and Jake K. Aggarwal. "Structure from stereo-a review." IEEE Transactions on Systems Man and Cybernetics 19.6 (1989): 1489-1510.

[129] Jungong Han; Ling Shao; Dong Xu; Shotton, J., "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," Cybernetics, IEEE Transactions on , vol.43, no.5, pp.1318,1334, Oct. 2013

[130] Horn, Eli, and Nahum Kiryati. "Toward optimal structured light patterns." Image and Vision Computing 17.2 (1999): 87-97.

[131] Salvi, Joaquim, Jordi Pages, and Joan Batlle. "Pattern codification strategies in structured light systems." Pattern Recognition 37.4 (2004): 827-849.

[132] Salvi, Joaquim, et al. "A state of the art in structured light patterns for surface profilometry." Pattern recognition 43.8 (2010): 2666-2680.

[133] Zhang, Ruo, et al. "Shape-from-shading: a survey." Pattern Analysis and Machine Intelligence, IEEE Transactions on 21.8 (1999): 690-706.

[134] Basri, Ronen, David Jacobs, and Ira Kemelmacher. "Photometric stereo with general, unknown lighting." International Journal of Computer Vision 72.3 (2007): 239-257.

[135] Hernandez, C.; Vogiatzis, G.; Cipolla, R., "Multiview Photometric Stereo," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.30, no.3, pp.548,554, March 2008

[136] http://en.wikipedia.org/wiki/Image_analysis, Last accessed (12.07.2014)

[137]J. F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 679-698, 1986.

[138] http://en.wikipedia.org/wiki/Canny_edge_detector, Last accessed (12.07.2014)

[139] http://homepage.cs.uiowa.edu/~cwyman/classes/spring08-22C251/homework/canny.pdf, Last accessed (12.07.2014)

[140] http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf, Last accessed (12.07.2014)

[141] P. V. C. Hough, “Method and means for recognizing complex patterns,” U.S. Patent 3 069 654, Dec. 18, 1962.

[142] A. Rosenfeld, *Picture Processing by Computer*. New York: Academic, 1969, pp. 335–336.

[143] R. O. Duda, P. E. Hart (1971) Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Technical Note 36, Artificial Intelligence Center, SRIInternational*.

[144]D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes”, Pattern Recognition, vol. 13, no. 2, pp. 111-122, 1981.

[145]Hart, P.E.; , "How the Hough transform was invented [DSP History]," *Signal Processing Magazine, IEEE* , vol.26, no.6, pp.18-22, November 2009

[146] L. Xu, E. Oja, and P. Kultanan, "A new curve detection method: Randomized Hough transform (RHT)", *Pattern Recog. Lett.* 11, 1990, 331-338.

[147] N. Kiryati, Y. Eldar, A. M. Bruckstein, A probabilistic Hough transform, Pattern recognition, 1991

[148] J. Matas and C. Galambos and J. Kittler, Progressive Probabilistic Hough Transform, 1998

[148] Vandewalle,P., S. Süsstrunk and M. Vetterli, “A frequency domain approach to registration of aliased images with application to super-resolution, Eurasip Journal on Applied Signal Processing, pages 1-14, 2006

[149] Lazaros Grammatikopoulosa, George Karras, Elli Petsa, "An automatic approach for camera calibration from vanishing points", ISPRS Journal of Photogrammetry and Remote Sensing, Volume 62, Issue 1, May 2007, Pages 64–76

[150] Maini, Raman, and Himanshu Aggarwal. "Study and comparison of various image edge detection techniques." *International Journal of Image Processing (IJIP)* 3.1 (2009): 1-11.

[151] Otsu, N., “A threshold selection method from gray-level Histograms”. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62{66 (1979)

[152] Sun, Haojun, Shengrui Wang, and Qingshan Jiang. "FCM-based model selection algorithms for determining the number of clusters." *Pattern recognition*37.10 (2004): 2027-2037.

[153] Gonzalez, Rafael Ceferino, and Richard E. Woods. *Instructor's Manual for Digital Image Processing*. Addison-Wesley, 1992.

[154] http://fiji.sc/Auto_Threshold, Last accessed (12.07.2014)

[155] http://en.wikipedia.org/wiki/Kinect, Last accessed (15.04.2014)

[156] Cruz, L.; Lucio, D.; Velho, L., "Kinect and RGBD Images: Challenges and Applications," Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on , vol., no., pp.36,49, 22-25 Aug. 2012

[157] http://msdn.microsoft.com/en-us/library/jj131033.aspx, Last accessed (28.11.2014)

[158] http://msdn.microsoft.com/en-us/library/hh973078.aspx, Last accessed (28.11.2014)

[159]Khoshelham, Kourosh, and Sander Oude Elberink. "Accuracy and resolution of kinect depth data for indoor mapping applications." Sensors 12.2 (2012): 1437-1454.

[160] http://uk.ign.com/wikis/xbox-one, Last accessed (28.11.2014)

[161]http://www.wired.com/2014/05/xbox-one-kinect/, Last accessed (28.11.2014)

[162] http://ww2.ptgrey.com/stereo-vision/bumblebee-2, Last accessed (28.11.2014)

[163]M. Attene, B. Falcidieno, M. Spagnuolo (2006) “Hierarchical mesh segmentation based on fitting primitives”, *The Visual Computer*. **22**:181–193

[164] J. Illingworth, J. Kittler, A Survey of the Hough Transform *Computer Vision, Graphics and Image Processing*, Vol. 44, pp. 87-116, 1988.

[165] Martin A. Fischler, Robert C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Commun. ACM, Vol. 24, No. 6., pp. 381-395, June 1981

[165] Poppinga, J.; Vaskevicius, N.; Birk, A.; Pathak, K., "Fast plane detection and polygonalization in noisy 3D range images," Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, vol., no., pp.3378,3383, 22-26 Sept. 2008

[166] Dube, D.; Zell, A., "Real-time plane extraction from depth images with the Randomized Hough Transform," Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on , vol., no., pp.1084,1091, 6-13 Nov. 2011

[167] Dorit Borrmann, Jan Elseberg, Kai Lingemann, Andreas Nüchter, "The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design", 3D Research, November 2011, 2:3

[168] A. Censi, S. Carpin (2009) "HSM3D: Feature-Less Global 6DOF Scan-Matching in the Hough/Radon Domain", *In Proceedings of the IEEE International Conference on Robotics and Automation*.

[169] F. Tarsha-Kurdi*, T. Landes, P. Grussenmeyer, "HOUGH-TRANSFORM AND EXTENDED RANSAC ALGORITHMS FOR AUTOMATIC DETECTION OF 3D BUILDING ROOF PLANES FROM LIDAR DATA", ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, IAPRS Volume XXXVI, Part 3 / W52, 2007

[170] Bretar, F. and Roux, M., 2005. Hybrid image segmentation using LiDAR 3D planar primitives. *ISPRS Proceedings*. Workshop Laser scanning. Enschede, the Netherlands, September 12-14, 2005.

[171] http://www.csse.uwa.edu.au/~pk/research/matlabfns/#spatial, Last accessed (15.04.2014)

[172] M. F. Hansen, G. A. Atkinson, L. N. Smith and M. L. Smith. 3D face reconstructions from photometric stereo using near infrared and visible light. *Computer Vision and Image Understanding*. 114:942-951, August 2010.

[173] Gupta, A.; Satkin, S.; Efros, A.A.; Hebert, M., "From 3D scene geometry to human workspace," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.1961,1968, 20-25 June 2011

[174] Patrick H. Winston, Thomas O. Binford T, Boris Katz, Michael Lowryt, "Learning Physical Descriptions from Functional Definitions, Examples and Precedents", 1983

[175] Jianzhuang Liu, Liangliang Cao ; Zhenguo Li ; Xiaoou Tang, "Plane-Based Optimization for 3D Object Reconstruction from Single Line Drawings" , Feb. 2008, Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 30, Issue: 2, Page(s): 315 – 327

[176.]: http://www.legislation.gov.uk/ukpga/1998/29/contents, Last accessed (20.03.2014)

[177]: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/204775/Surveillance_Camera_Code_of_Practice_WEB.pdf, Last accessed (20.03.2014)

[178]: http://ico.org.uk/~/media/documents/library/Data_Protection/Detailed_specialist_guides/ICO_CCTVFINAL_2301.pdf, Last accessed (20.03.2014)

[179] : http://ico.org.uk/for_organisations/data_protection/topic_guides/cctv, Last accessed (20.03.2014)

[180]: http://www.ifsecglobal.com/surveillance-camera-code-of-practice-faqs/, Last accessed (20.03.2014)

[181]: http://www.atoc.org/clientfiles/files/publicationsdocuments/National%20Rail%20%20Underground%20CCTV%20Guidance%20Document%20%20FULL%20November%202010.pdf, Last accessed (20.03.2014)

[182]: http://www.siraview.com/wp-content/uploads/2012/04/cctvsystems-uk-police-requirements.pdf, Last accessed (20.03.2014)

Timeline references

http://www.srmti.com/news/the-history-of-cctv-in-the-uk-10079/, Last accessed (18.03.2014)

http://www.maxtag.com/history-of-CCTV.html, Last accessed (18.03.2014)

http://www.channel4.com/learning/microsites/I/ideasfactory/webit/teachers/sites/cctv/history_of_cctv.htm, Last accessed (18.03.2014)

http://srsaldanhablog.wordpress.com/2012/12/09/a-brief-history-of-cctv-systems/, Last accessed (18.03.2014)

http://www.notbored.org/england-history.html, Last accessed (18.03.2014)

http://www.cctvconsult.net/page.aspx?pid=8, Last accessed (18.03.2014)

http://www.wecusurveillance.com/cctvhistory, Last accessed (18.03.2014)

[picture1] http://commons.wikimedia.org/wiki/File:Emily_Davison_portrait.jpg, Last accessed (23.02.2015)

[picture 2] http://en.wikipedia.org/wiki/V-2_rocket, Last accessed (23.02.2015)

[picture 3] https://www.flickr.com/photos/digitalcollectionsum/15287947626/, Licence: https://creativecommons.org/licenses/by-nc-nd/2.0/, Last accessed (23.02.2015)

[picture 4] http://en.wikipedia.org/wiki/Guy_Fawkes, Last accessed (23.02.2015)

[picture 5] http://en.wikipedia.org/wiki/London_Victoria_station, Last accessed (23.02.2015)

[picture 6] http://commons.wikimedia.org/wiki/File:Underground.svg, Last accessed (23.02.2015)

[picture 7] http://en.wikipedia.org/wiki/Traffic_enforcement_camera, Last accessed (23.02.2015)

[picture 8] http://en.wikipedia.org/wiki/1993_World_Trade_Center_bombing, Last accessed (23.02.2015)

[picture 9] http://en.wikipedia.org/wiki/Automated_teller_machine, Last accessed (23.02.2015)

[picture 10] http://commons.wikimedia.org/wiki/File:W3C%C2%AE_Icon.svg, Last accessed (18.03.2014)

[picture 11] https://www.flickr.com/photos/911pics/7835973648/, Licence: https://creativecommons.org/licenses/by/2.0/, Last accessed (23.02.2015)

[picture 12] , Last accessed (23.02.2015)

[picture 14] http://www.geoarb.com.au/cctv/analogue-cameras/, Last accessed (18.03.2014)

[picture 16] http://archiveshub.ac.uk/features/mar04b2.shtml, Last accessed (18.03.2014) [picture 17] http://www.bbc.co.uk/news/uk-politics-26335569, Last accessed (18.03.2014)

[picture 18] http://www.autospies.com/news/Man-Vs-Camera-Florida-Representative-Fights-Back-53092/, Last accessed (18.03.2014)

[picture 20] http://henriwilliams.blogspot.co.uk/2010/08/entering-panopticon-study-of-ring-of.html, Last accessed (18.03.2014)

[picture 21] http://www.jmysecurity.com/index.php/solutions, Last accessed (18.03.2014)

[picture 22] http://www.bestcybercoach.com/web-tutorial/, Last accessed (18.03.2014)

[picture 23] commercial ??? http://www.dhgate.com/store/product/500gb-hdd-with-4ch-dvr-cctv-system-420tvl/140151309.html, Last accessed (18.03.2014)

[picture 24] http://theambitiousgy.hubpages.com/hub/How-Face-Recognition-Works-Part-1, Last accessed (18.03.2014)

[picture25] http://news.nationalgeographic.com/news/2011/09/pictures/110908-about-911-september-9-11-twin-world-trade-center-towers-indelible/, Last accessed (18.03.2014)

[picture 26] http://shelf3d.com/Search/Uploaded%20by%20vcatechnology, Last accessed (18.03.2014)

[picture 27] http://worldunderwatch.blogspot.co.uk/2011/02/aclu-wants-ban-on-chicagos-new.html, Last accessed (18.03.2014)

Appendix 1: Video Surveillance requirements and applications

In the first section of this chapter, the focus is on CCTV requirements from the Metropolitan Police point of view. Information stated below has been provided by the collaboration of the Detective Chief Inspector Mick Neville and his team during a visit to the New Scotland Yard. This visit allowed us to find a motivation and inspiration for this work as in spite of issues, CCTV is also the key to help solving crimes and make the world safer without hindering people's privacy.

1-1-Metropolitan Police requirements

In this section, the interview of DCI Mick Neville from the 12th of June 2014 is recounted. All the information below has been provided by DCI Mick Neville during the interview.

From 2011 riots and looters' incidents, CCTV has significantly contributed to successful prosecutions and remains the biggest contributor.

According to DCI Mick Neville, very few automated systems work and there are no known systems able to automatically or manually segment a scene to assess problems for awareness such as a person falling on the tracks etc. Furthermore, he stated that CCTV has only become more important and considered by the police since last year. Previously, the Police did not take the use of CCTV seriously as it was considered to be preventive. Moreover, only the use of DNA and fingerprints were considered as real performances for officers due to lack of CCTV training and performance.

Since CCTV status changed, it has been essential to prove crimes and used it to identification of suspects. For example, last year, 9571 crimes of which 509 robberies have been solved thanks to CCTV footage. However, the figures remain quite small; e.g. CCTV solves only 20% of robberies.

The main reasons for such figures are the inability to extract video of sequence of a scene, the quality of images extracted, the amount of data recorded and the time data are kept.

The first issue could be resolved by a standardization and simplification of the method to extract data recorded, especially for private places where it remains sometimes impossible to solve crimes due to the inability to extract recorded data.

The quality of the images needs to be improved as it is essential to differentiate people with good from bad behaviour, identify someone when zooming in, identify someone using a facial recognition system and prosecute. Furthermore, a more judicious camera positions and angle of view would contribute to improve quality of data recorded by avoiding occlusions from the background of the scene, trees etc; by being witness of a crime such as a break-in, riots etc; by improving facial recognition; by obtaining an overview and/or a detailed video sequence of a scene when possible. These are quite common issues encountered with CCTV footage and contribute to the quality of the footage in addition to the quality of the cameras used etc. Then, using cameras with various positions and angle of view such as various heights even at head-height, various angles would permit to deal with these issues. Furthermore, in presence of only camera for several incidents, it would be more useful to capture high-quality images of one incident rather than low-quality images of the two; as well as an overview is better than a detailed view of scene when both are not possible. However, the required quality for images depends on the goal.

The amount of data recorded can be solved by the use of automated and more intelligent CCTV systems. The main impacts of such systems are the reduction in man-power world to avoid looking for a specific event among hours recorded.

Furthermore, features such as blue lights, sounds such as police siren etc indicate presence of police and emergency vehicles and could be helpful to focus the camera on the offence rather than on uninteresting data such as police cars, officers etc. Finally, in some circumstances, an extension of the time during which data are kept would be useful.

According to DCI Neville, the real use of CCTV is mainly for detection rather than prevention as it was seen originally. Moreover, the Police have interests for the detection of events such as:

- Detection of movements of expensive items in shops;
- Detection and tracking of a particular item;
- Knife detection in schools;

As well as improvements on :

- Image resolution;
- Facial recognition as only 1 in 6000 CCTV video sequences are currently used for facial recognition;
- Logo recognition;
- Plate registration recognition;
- Tattoo recognition;
- Clothes' colour extraction;
- State of an item such as a bag or a basket which appears full of items and then empty after a few seconds etc.

In the next section, documents describing current CCTV systems requirements are presented.

1-2-Current CCTV systems requirements

Current CCTV systems requirements are described in the following documents:

- Data protection act [176];
- Surveillance camera code of Practice [177];
- Information commissioner CCTV code of practice [178].

According to the introduction of the first document, the data protection Act 1998 concerns the regulation of the processing of information relating to individual, including the obtaining, holding, use or disclosure of such information.

As CCTV surveillance systems uses camera to monitor and record scenes, any current CCTV has access to information relating to individuals and are usually analysed to prevent from abnormal behaviour, to be used as proof to prosecute criminals etc. So, the owner of any current CCTV systems has to follow this regulation in the case where the data recorded needs to be processed. This is related to ethical code and applies to most organisations or businesses [179] that use CCTV but not to individuals' private or household purposes. It has been created to help public to be confident in the responsible use of CCTV [179].

The other two documents provide guidance and advice for CCTV users on how to comply with the Data protection act [179]. These documents only apply to public bodies such as the police and local governments but it encourages private companies to follow it as a guide on how to operate CCTV systems [180]. Moreover, the surveillance camera code of practice provides the following 12 guiding principles [177] that CCTV systems should follow.

Guiding Principles

2.6 System operators should adopt the following 12 guiding principles:

1. Use of a surveillance camera system must always be for a specified purpose which is in pursuit of a legitimate aim and necessary to meet an identified pressing need.

2. The use of a surveillance camera system must take into account its effect on individuals and their privacy, with regular reviews to ensure its use remains justified.

3. There must be as much transparency in the use of a surveillance camera system as possible, including a published contact point for access to information and complaints.

4. There must be clear responsibility and accountability for all surveillance camera system activities including images and information collected, held and used.

5. Clear rules, policies and procedures must be in place before a surveillance camera system is used, and these must be communicated to all who need to comply with them.

6. No more images and information should be stored than that which is strictly required for the stated purpose of a surveillance camera system, and such images and information should be deleted once their purposes have been discharged.

7. Access to retained images and information should be restricted and there must be clearly defined rules on who can gain access and for what purpose such access is granted; the disclosure of images and information should only take place when it is necessary for such a purpose or for law enforcement purposes.

8. Surveillance camera system operators should consider any approved operational, technical and competency standards relevant to a system and its purpose and work to meet and maintain those standards.

9. Surveillance camera system images and information should be subject to appropriate security measures to safeguard against unauthorised access and use.

10. There should be effective review and audit mechanisms to ensure legal requirements, policies and standards are complied with in practice, and regular reports should be published.

11. When the use of a surveillance camera system is in pursuit of a legitimate aim, and there is a pressing need for its use, it should then be used in the most effective way to support public safety and law enforcement with the aim of processing images and information of evidential value.

12. Any information used to support a surveillance camera system which compares against a reference database for matching purposes should be accurate and kept up to date.

Furthermore, other documents such as [181] provide useful guidance information especially dedicated to the use of CCTV in railway and underground stations in the UK. The document [182] provides police requirements for CCTV systems in UK.

In this section, the need there is for CCTV to be improved from the point of view of the Metropolitan police as well as requirements it is advised to follow has been shown. In the next section, information showing people's beliefs and perception of CCTV are presented.

1-3-CCTV perception and beliefs

The information below has been provided by DCI Mick Neville from an investigation made by independent research specialists ICM in March 2014. It has been performed through Online Omnibus over 2032 GB adults from age 18+.

The investigation addressed questions related to:

- The awareness of CCTV ;
- Surveillance Camera Code of Practice ;
- Support for Public space CCTV;
- Reasons to be in favour of CCTV use in public spaces;
- Reasons not to be in favour of CCTV use in public spaces;
- Reasons increasing the support of public space CCTV;
- The purpose of CCTV;
- Information on CCTV;
- Perception of CCTV;
- And CCTV fundings.

The table 30 below summarized the main findings of this investigation:

Percentage	Main findings
86 % support	The use of CCTV in public places
74% support	CCTV because it helps prevent crime
4% (45% feel it has no impact on public safety while 44% against it due to lack of information on how and why it is used)	Against the use of CCTV
43%	Would increase their support in case of a guarantee that cameras would be monitored more closely.
80% feel (same as in 2012)	They do not have enough information on how and why CCTV is used in their area
10% knew	What was the Surveillance Camera Code of Practise
67%	Never heard of Surveillance Camera Code of Practise
70%	Welcomed the Surveillance Camera code of Practise after being told what it was
64% believe	The **current** primary purpose of CCTV is to help prevent crime and antisocial behaviour
76% believe	The primary goal of CCTV **should** be to help prevent crime and antisocial behaviour in public places
3% are influenced on their perceptions of CCTV and how it is used (of which 4% receiving information from the Police)	From Information from local authority
64% would worry	if local council was reducing CCTV to save money

Table 30 : Summary of the key findings of the investigation on CCTV perceptions and beliefs

This chapter gave an overview of the public and police perception and expectation of CCTV as well as current requirements for such system. The next chapter summarizes and evaluates the approaches proposed to address some of the main challenges of CCTV; it also proposes solutions to tackle the limitations of these approaches and give orientations for future work.

Appendix 2: Pre-written functions from OpenCV used in the proposed VP detection approach.

The following descriptions related to the Canny Edge Detector, the Hough transform and the solve functions have been directly extracted from the OpenCV library website and have in any case been written by the author.

Canny Edge Detector (cf. http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)

Goal

In this tutorial you will learn how to:

Use the OpenCV function *Canny* to implement the *Canny Edge Detector*.

Theory

1. The *Canny Edge detector* was developed by John F. Canny in 1986. Also known to many as the optimal detector, *Canny* algorithm aims to satisfy three main criteria:

    - **Low error rate such that** only existent edges are well detected.
    - **Good localization such that** the distance between edge pixels detected and real edge pixels have to be minimized.
    - **Minimal response such that** there is only one detector response per edge.

Steps

1. Filter out any noise. The Gaussian filter is used for this purpose. An example of a Gaussian kernel of $size = 5$ that might be used is shown below:

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. Find the intensity gradient of the image. For this, we follow a procedure analogous to Sobel:

    a. Apply a pair of convolution masks (in x and $y$directions:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

b. Find the gradient strength and direction with:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan(\frac{G_y}{G_x})$$

The direction is rounded to one of four possible angles (namely 0, 45, 90 or 135)

3. Non-maximum suppression is applied. This removes pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will remain.

4. *Hysteresis*: The final step. Canny does use two thresholds (*upper and lower*):

    a. If a pixel gradient is higher than the *upper* threshold, the pixel is accepted as an edge

    b. If a pixel gradient value is below the *lower* threshold, then it is rejected.

    c. If the pixel gradient is between the two thresholds, then it will be accepted only if it is connected to a pixel that is above the *upper* threshold.

    Canny recommended a *upper*:*lower* ratio between 2:1 and 3:1.

5. For more details, you can always consult your favourite Computer Vision book.

Code

1. **What does this program do?**

    - Asks the user to enter a numerical value to set the lower threshold for our Canny Edge Detector (by means of a *Trackbar*)

    - Applies the *Canny Detector* and generates a **mask** (bright lines representing the edges on a black background).

    - Applies the mask obtained on the original image and display it in a window.

2. The tutorial code's is shown lines below. You can also download it from here

```
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <stdio.h>

using namespace cv;

/// Global variables

Mat src, src_gray;
Mat dst, detected_edges;

int edgeThresh = 1;
int lowThreshold;
int const max_lowThreshold = 100;
int ratio = 3;
int kernel_size = 3;
char* window_name = "Edge Map";

/**
 * @function CannyThreshold
 * @brief Trackbar callback - Canny thresholds input with a ratio 1:3
 */
void CannyThreshold(int, void*)
{
 /// Reduce noise with a kernel 3x3
 blur( src_gray, detected_edges, Size(3,3) );

 /// Canny detector
 Canny( detected_edges, detected_edges, lowThreshold, lowThreshold*ratio, kernel_size );

 /// Using Canny's output as a mask, we display our result
 dst = Scalar::all(0);

 src.copyTo( dst, detected_edges);
 imshow( window_name, dst );
 }


/** @function main */
int main( int argc, char** argv )
{
 /// Load an image
 src = imread( argv[1] );

 if( !src.data )
 { return -1; }

 /// Create a matrix of the same type and size as src (for dst)
 dst.create( src.size(), src.type() );

 /// Convert the image to grayscale
 cvtColor( src, src_gray, CV_BGR2GRAY );
/// Create a window
namedWindow( window_name, CV_WINDOW_AUTOSIZE );

/// Create a Trackbar for user to enter threshold
createTrackbar( "Min Threshold:", window_name, &lowThreshold, max_lowThreshold, CannyThreshold );

/// Show the image
CannyThreshold(0, 0);

/// Wait until user exit program by pressing a key
waitKey(0);

return 0;
}
```

Explanation

1. Create some needed variables:

```
2.     Mat src, src_gray;
3.     Mat dst, detected_edges;
4.
5.     int edgeThresh = 1;
6.     int lowThreshold;
7.     int const max_lowThreshold = 100;
8.     int ratio = 3;
9.     int kernel_size = 3;
10.    char* window_name = "Edge Map";
11.
12.  Note the following:
13.
14.  a. We establish a ratio of lower:upper threshold of 3:1 (with the variable *ratio*)
15.  b. We set the kernel size of :math:`3` (for the Sobel operations to be performed internally by the
     Canny function)
```

```
c. We set a maximum value for the lower Threshold of :math:`100`.
```

16. Loads the source image:

```
17.  /// Load an image
18.  src = imread( argv[1] );
19.
20.  if( !src.data )
21.    { return -1; }
```

22. Create a matrix of the same type and size of *src* (to be *dst*)

```
23.  dst.create( src.size(), src.type() );
```

24. Convert the image to grayscale (using the function *cvtColor*:

25. `cvtColor( src, src_gray, CV_BGR2GRAY );`

26. Create a window to display the results

27. `namedWindow( window_name, CV_WINDOW_AUTOSIZE );`

28. Create a *Trackbar* for the user to enter the *lower threshold* for our *Canny detector*:

29. `createTrackbar( "Min Threshold:", window_name, &lowThreshold, max_lowThreshold, CannyThreshold );`

    Observe the following:

    a. The variable to be controlled by the *Trackbar* is *low Threshold* with a limit of *max_low Threshold* (which we set to 100 previously)

    b. Each time the *Trackbar* registers an action, the *callback* function Canny Threshold will be invoked.

30. Let's check the Canny Threshold function, step by step:

    a. First, we blur the image with a filter of kernel size 3:

    b. `blur( src_gray, detected_edges, Size(3,3) );`

    c. Second, we apply the OpenCV function *Canny*:

    d. `Canny( detected_edges, detected_edges, lowThreshold, lowThreshold*ratio, kernel_size );`

    where the arguments are:

    - *detected_edges*: Source image, grayscale
    - *detected_edges*: Output of the detector (can be the same as the input)
    - *lowThreshold*: The value entered by the user moving the Trackbar
    - *highThreshold*: Set in the program as three times the lower threshold (following Canny's recommendation)
    - *kernel_size*: We defined it to be 3 (the size of the Sobel kernel to be used internally)

31. We fill a *dst* image with zeros (meaning the image is completely black).

```
32. dst = Scalar::all(0);
```

33. Finally, we will use the function copyTo to map only the areas of the image that are identified as edges (on a black background).

```
34. src.copyTo( dst, detected_edges);
```

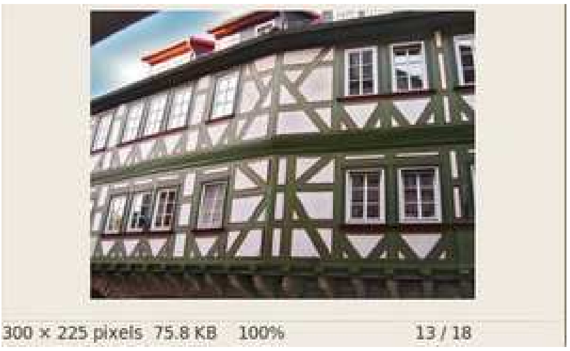
copyTo copy the *src* image onto *dst*. However, it will only copy the pixels in the locations where they have non-zero values. Since the output of the Canny detector is the edge contours on a black background, the resulting *dst* will be black in all the area but the detected edges.

35. We display our result:

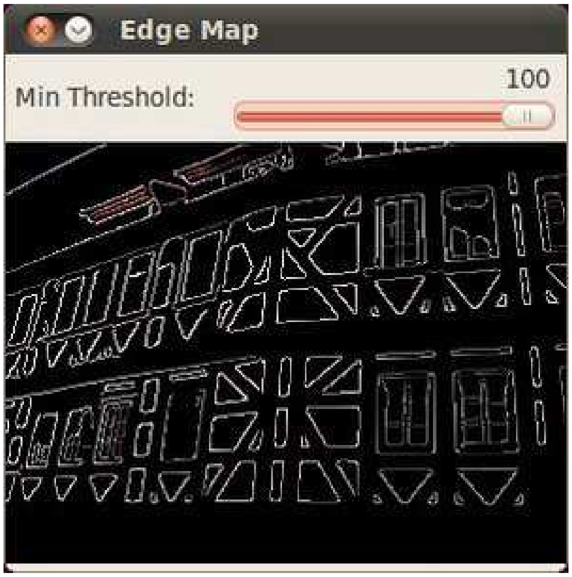
```
36. imshow( window_name, dst );
```

Result

- After compiling the code above, we can run it giving as argument the path to an image. For example, using as an input the following image:



- Moving the slider, trying different threshold, we obtain the following result:



Edge Map

Min Threshold: 100

- Notice how the image is superposed to the black background on the edge regions.

Hough Line Transform(cf. http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html)

Goal

In this tutorial you will learn how to:

- Use the OpenCV functions HoughLines and HoughLinesP to detect lines in an image.

Theory

Note

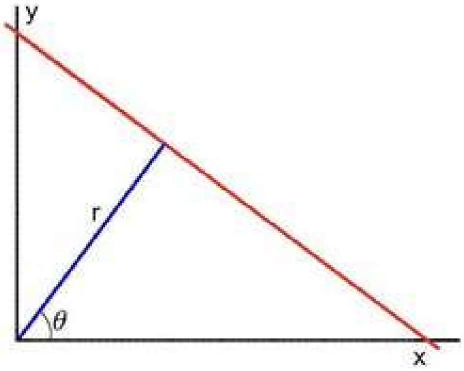
The explanation below belongs to the book **Learning OpenCV** by Bradski and Kaehler.

Hough Line Transform

1. The Hough Line Transform is a transform used to detect straight lines.
2. To apply the Transform, first an edge detection pre-processing is desirable.

How does it work?

1. As you know, a line in the image space can be expressed with two variables. For example:
   a. In the **Cartesian coordinate system:** Parameters: (m, b).
   b. In the **Polar coordinate system:** Parameters: (r, θ)



For Hough Transforms, we will express lines in the *Polar system*. Hence, a line equation can be written as:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

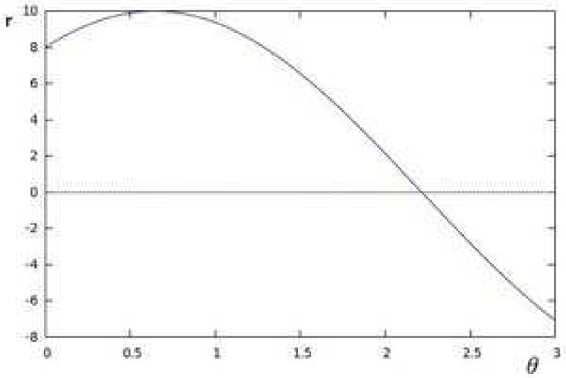
Arranging the terms: $r = x\cos\theta + y\sin\theta$

1. In general for each point (x_0, y_0), we can define the family of lines that goes through that point as:

$$r_\theta = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta$$

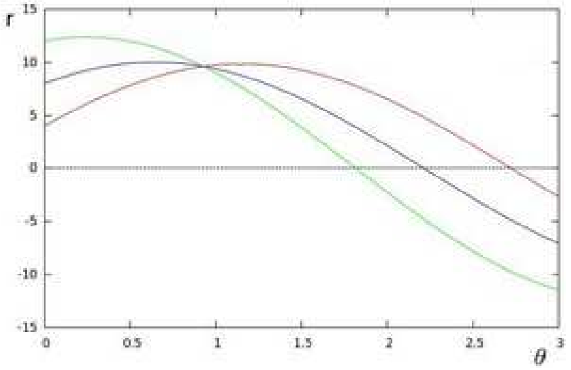
Meaning that each pair (r_θ, θ) represents each line that passes by (x_0, y_0).

2. If for a given (x_0, y_0) we plot the family of lines that goes through it, we get a sinusoid. For instance, for $x_0 = 8$ and $y_0 = 6$ we get the following plot (in a plane θ - r):



We consider only points such that $r > 0$ and $0 < \theta < 2\pi$.

3. We can do the same operation above for all the points in an image. If the curves of two different points intersect in the plane θ - r, that means that both points belong to a same line. For instance, following with the example above and drawing the plot for two more points: $x_1 = 9$, $y_1 = 4$ and $x_2 = 12$, $y_2 = 3$, we get:



The three plots intersect in one single point $(0.925, 9.6)$, these coordinates are the parameters (θ, r) or the line in which (x_0, y_0), (x_1, y_1) and (x_2, y_2) lay.

4. What does all the stuff above mean? It means that in general, a line can be *detected* by finding the number of intersections between curves.The more curves intersecting means

that the line represented by that intersection have more points. In general, we can define a*threshold* of the minimum number of intersections needed to *detect* a line.

5. This is what the Hough Line Transform does. It keeps track of the intersection between curves of every point in the image. If the number of intersections is above some *threshold*, then it declares it as a line with the parameters (θ, r_θ) of the intersection point.

Standard and Probabilistic Hough Line Transform

OpenCV implements two kind of Hough Line Transforms:

a. **The Standard Hough Transform**

- It consists in pretty much what we just explained in the previous section. It gives you as result a vector of couples (θ, r_θ)
- In OpenCV it is implemented with the function HoughLines

b. **The Probabilistic Hough Line Transform**

- A more efficient implementation of the Hough Line Transform. It gives as output the extremes of the detected lines (x_0, y_0, x_1, y_1)
- In OpenCV it is implemented with the function HoughLinesP

Code

1. **What does this program do?**

   - Loads an image
   - Applies either a *Standard Hough Line Transform* or a *Probabilistic Line Transform*.
   - Display the original image and the detected line in two windows.

2. The sample code that we will explain can be downloaded from here. A slightly fancier version (which shows both Hough standard and probabilistic with trackbars for changing the threshold values) can be found here.

```
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>

using namespace cv;
using namespace std;

void help()
```

```
{
 cout << "\nThis program demonstrates line finding with the Hough
transform.\n"
         "Usage:\n"
         "./houghlines <image_name>, Default is pic1.jpg\n" << endl;
}

int main(int argc, char** argv)
{
 const char* filename = argc >= 2 ? argv[1] : "pic1.jpg";

 Mat src = imread(filename, 0);
 if(src.empty())
 {
     help();
     cout << "can not open " << filename << endl;
     return -1;
 }

 Mat dst, cdst;
 Canny(src, dst, 50, 200, 3);
 cvtColor(dst, cdst, CV_GRAY2BGR);

 #if 0
  vector<Vec2f> lines;
  HoughLines(dst, lines, 1, CV_PI/180, 100, 0, 0 );

  for( size_t i = 0; i < lines.size(); i++ )
  {
     float rho = lines[i][0], theta = lines[i][1];
     Point pt1, pt2;
     double a = cos(theta), b = sin(theta);
     double x0 = a*rho, y0 = b*rho;
     pt1.x = cvRound(x0 + 1000*(-b));
     pt1.y = cvRound(y0 + 1000*(a));
     pt2.x = cvRound(x0 - 1000*(-b));
     pt2.y = cvRound(y0 - 1000*(a));
     line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
  }
 #else
  vector<Vec4i> lines;
  HoughLinesP(dst, lines, 1, CV_PI/180, 50, 50, 10 );
  for( size_t i = 0; i < lines.size(); i++ )
  {
    Vec4i l = lines[i];
    line( cdst, Point(l[0], l[1]), Point(l[2], l[3]), Scalar(0,0,255), 3,
CV_AA);
  }
 #endif
 imshow("source", src);
 imshow("detected lines", cdst);

 waitKey();
 return 0;
}
```

Explanation

1. Load an image

```
2. Mat src = imread(filename, 0);
3. if(src.empty())
4. {
5.   help();
6.   cout << "can not open " << filename << endl;
7.   return -1;
8. }
```

9. Detect the edges of the image by using a Canny detector

```
10.  Canny(src, dst, 50, 200, 3);
```

Now we will apply the Hough Line Transform. We will explain how to use both OpenCV functions available for this purpose:

11. **Standard Hough Line Transform**

a. First, you apply the Transform:

```
b. vector<Vec2f> lines;
c. HoughLines(dst, lines, 1, CV_PI/180, 100, 0, 0 );
```

with the following arguments:

- *dst*: Output of the edge detector. It should be a grayscale image (although in fact it is a binary one)
- *lines*: A vector that will store the parameters (r, θ) of the detected lines
- *rho* : The resolution of the parameter r in pixels. We use **1** pixel.
- *theta*: The resolution of the parameter θ in radians. We use **1 degree** (CV_PI/180)
- *threshold*: The minimum number of intersections to "*detect*" a line
- *srn* and *stn*: Default parameters to zero. Check OpenCV reference for more info.

d. And then you display the result by drawing the lines.

```
e. for( size_t i = 0; i < lines.size(); i++ )
```

```
f. {
g.   float rho = lines[i][0], theta = lines[i][1];
h.   Point pt1, pt2;
i.   double a = cos(theta), b = sin(theta);
j.   double x0 = a*rho, y0 = b*rho;
k.   pt1.x = cvRound(x0 + 1000*(-b));
l.   pt1.y = cvRound(y0 + 1000*(a));
m.   pt2.x = cvRound(x0 - 1000*(-b));
n.   pt2.y = cvRound(y0 - 1000*(a));
o.   line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
p. }
```

12. **Probabilistic Hough Line Transform**

    a. First you apply the transform:

```
b. vector<Vec4i> lines;
c. HoughLinesP(dst, lines, 1, CV_PI/180, 50, 50, 10 );
```

with the arguments:

- *dst*: Output of the edge detector. It should be a grayscale image (although in fact it is a binary one)
- *lines*: A vector that will store the parameters $(x_{start}, y_{start}, x_{end}, y_{end})$ of the detected lines
- *rho* : The resolution of the parameter r in pixels. We use **1** pixel.
- *theta*: The resolution of the parameter θ in radians. We use **1 degree** (CV_PI/180)
- *threshold*: The minimum number of intersections to "*detect*" a line
- *minLinLength*: The minimum number of points that can form a line. Lines with less than this number of points are disregarded.
- *maxLineGap*: The maximum gap between two points to be considered in the same line.

    d. And then you display the result by drawing the lines.

```
e. for( size_t i = 0; i < lines.size(); i++ )
f. {
g.   Vec4i l = lines[i];
h.   line( cdst, Point(l[0], l[1]), Point(l[2], l[3]),
   Scalar(0,0,255), 3, CV_AA);
i. }
```

13. Display the original image and the detected lines:

```
14.  imshow("source", src);
15.  imshow("detected lines", cdst);
```

16. Wait until the user exits the program

```
17.  waitKey();
```

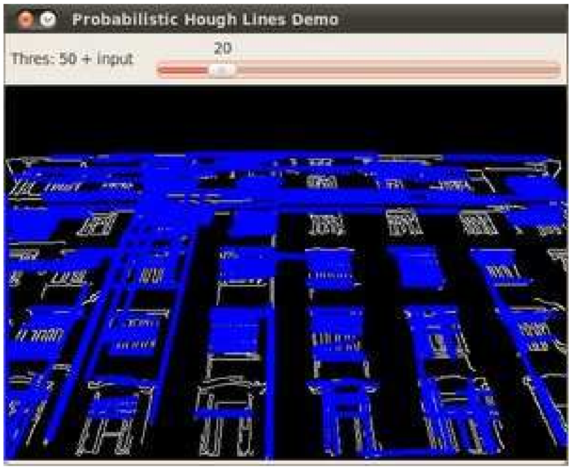
Result

Note

The results below are obtained using the slightly fancier version we mentioned in the *Code* section. It still implements the same stuff as above, only adding the Trackbar for the Threshold.

Using an input image such as:

We get the following result by using the Probabilistic Hough Line Transform:



Probabilistic Hough Lines Demo

Thres: 50 + input 20

You may observe that the number of lines detected vary while you change the *threshold*. The explanation is sort of evident: If you establish a higher threshold, fewer lines will be detected (since you will need more points to declare a line detected).

Solve (cf. http://docs.opencv.org/modules/core/doc/operations_on_arrays.html#solve)

Solves one or more linear systems or least-squares problems.

C++: bool **solve**(InputArray **src1**, InputArray **src2**, OutputArray **dst**, int **flags**=DECOMP_LU)

Python: cv2.**solve**(src1, src2[, dst[, flags]]) → retval, dst

C: int **cvSolve**(const CvArr* **src1**, const CvArr* **src2**, CvArr* **dst**, int **method**=CV_LU)

Python: cv.**Solve**(A, B, X, method=CV_LU) → None

Parameters:

- **src1** – input matrix on the left-hand side of the system.
- **src2** – input matrix on the right-hand side of the system.
- **dst** – output solution.
- **flags** –solution (matrix inversion) method.
  - **DECOMP_LU** Gaussian elimination with optimal pivot element chosen.
  - **DECOMP_CHOLESKY** Cholesky LL^T factorization; the matrix $\texttt{src1}$ must be symmetrical and positively defined.
  - **DECOMP_EIG** eigenvalue decomposition; the matrix $\texttt{src1}$ must be symmetrical.
  - **DECOMP_SVD** singular value decomposition (SVD) method; the system can be over-defined and/or the matrix $\texttt{src1}$ can be singular.
  - **DECOMP_QR** QR factorization; the system can be over-defined and/or the matrix $\texttt{src1}$ can be singular.
  - **DECOMP_NORMAL** while all the previous flags are mutually exclusive, this flag can be used together with any of the previous; it means that the normal equations $\texttt{src1}^T \cdot \texttt{src1} \cdot \texttt{dst} = \texttt{src1}^T \texttt{src2}$ are solved instead of the original system$\texttt{src1} \cdot \texttt{dst} = \texttt{src2}$.

The function $\texttt{solve}$ solves a linear system or least-squares problem (the latter is possible with SVD or QR methods, or by specifying the flag$\texttt{DECOMP_NORMAL}$):

$$\texttt{dst} = \arg \min_X \| \texttt{src1} \cdot \texttt{X} - \texttt{src2} \|$$

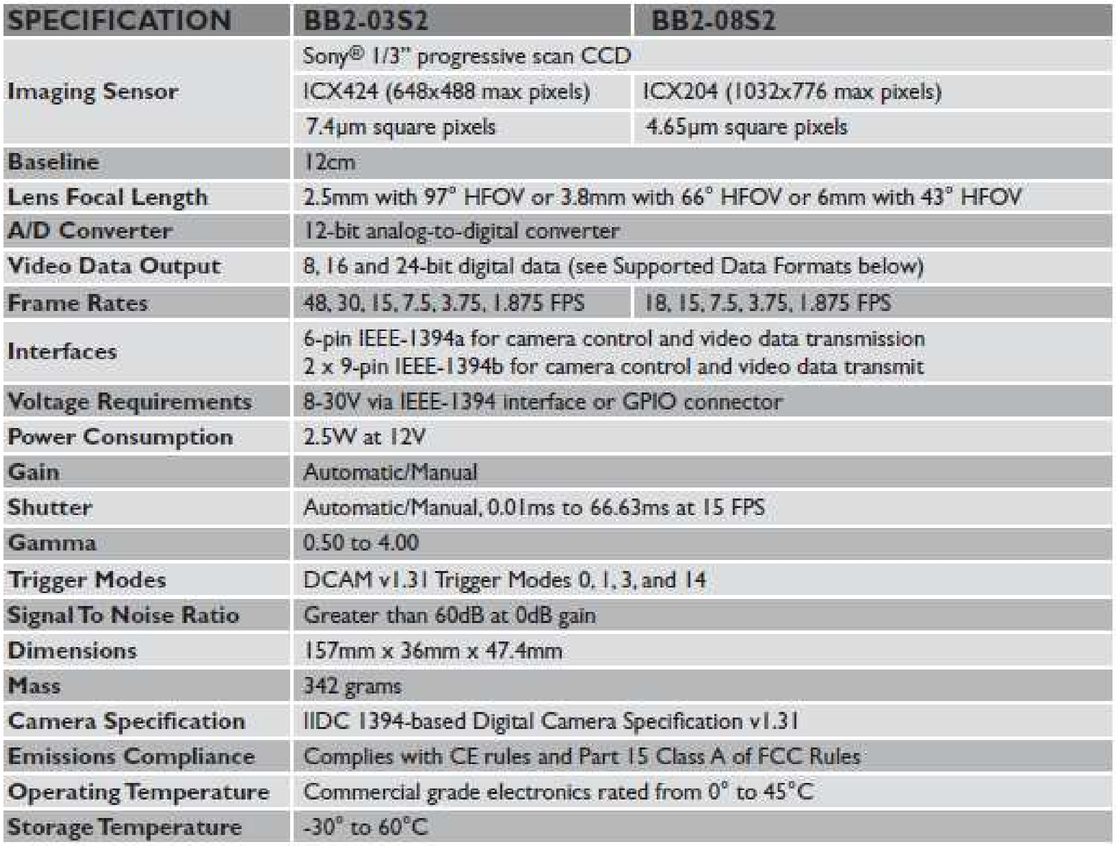
If $\texttt{DECOMP\_LU}$ or $\texttt{DECOMP\_CHOLESKY}$ method is used, the function returns 1 if $\texttt{src1}$ (or $\texttt{src1}^T \texttt{src1}$) is non-singular. Otherwise, it returns 0. In the latter case, $\texttt{dst}$ is not valid. Other methods find a pseudo-solution in case of a singular left-hand side part.

Note

If you want to find a unity-norm solution of an under-defined singular system $\texttt{src1} \cdot \texttt{dst} = 0$, the function `solve` will not do the work. Use **`SVD::solveZ()`** instead.

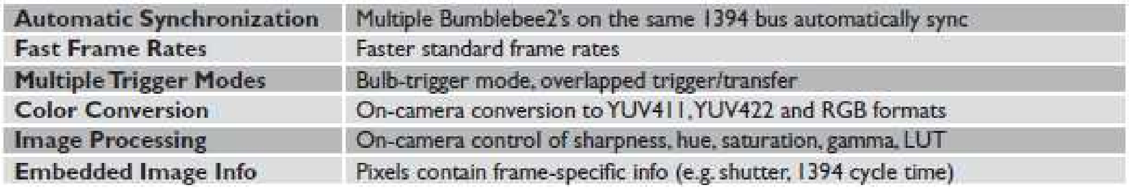
Appendix 3: Bumblebee 2 Camera Specifications provided by Point Grey

Bumblebee 2 Specifications



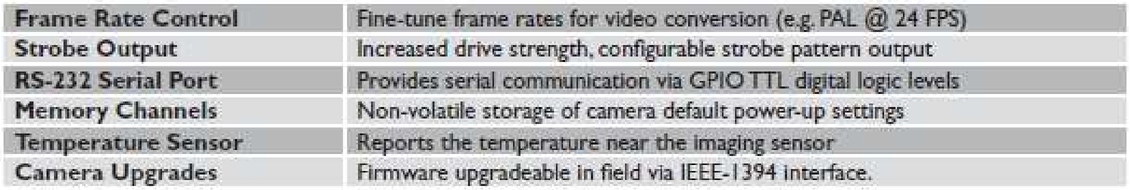
SPECIFICATION	BB2-03S2	BB2-08S2
Imaging Sensor	Sony® 1/3" progressive scan CCD	
	ICX424 (648x488 max pixels)	ICX204 (1032x776 max pixels)
	7.4µm square pixels	4.65µm square pixels
Baseline	12cm	
Lens Focal Length	2.5mm with 97° HFOV or 3.8mm with 66° HFOV or 6mm with 43° HFOV	
A/D Converter	12-bit analog-to-digital converter	
Video Data Output	8, 16 and 24-bit digital data (see Supported Data Formats below)	
Frame Rates	48, 30, 15, 7.5, 3.75, 1.875 FPS	18, 15, 7.5, 3.75, 1.875 FPS
Interfaces	6-pin IEEE-1394a for camera control and video data transmission 2 x 9-pin IEEE-1394b for camera control and video data transmit	
Voltage Requirements	8-30V via IEEE-1394 interface or GPIO connector	
Power Consumption	2.5W at 12V	
Gain	Automatic/Manual	
Shutter	Automatic/Manual, 0.01ms to 66.63ms at 15 FPS	
Gamma	0.50 to 4.00	
Trigger Modes	DCAM v1.31 Trigger Modes 0, 1, 3, and 14	
Signal To Noise Ratio	Greater than 60dB at 0dB gain	
Dimensions	157mm x 36mm x 47.4mm	
Mass	342 grams	
Camera Specification	IIDC 1394-based Digital Camera Specification v1.31	
Emissions Compliance	Complies with CE rules and Part 15 Class A of FCC Rules	
Operating Temperature	Commercial grade electronics rated from 0° to 45°C	
Storage Temperature	-30° to 60°C	

Image acquisition



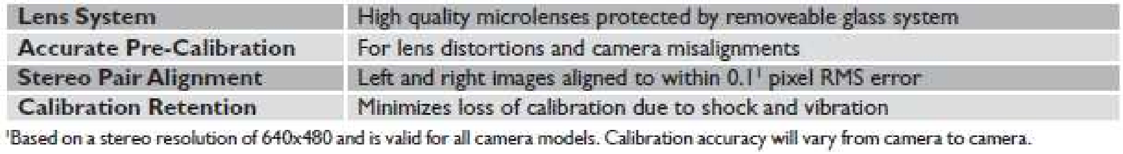
Automatic Synchronization	Multiple Bumblebee2's on the same 1394 bus automatically sync
Fast Frame Rates	Faster standard frame rates
Multiple Trigger Modes	Bulb-trigger mode, overlapped trigger/transfer
Color Conversion	On-camera conversion to YUV411, YUV422 and RGB formats
Image Processing	On-camera control of sharpness, hue, saturation, gamma, LUT
Embedded Image Info	Pixels contain frame-specific info (e.g. shutter, 1394 cycle time)

Camera and device control



Frame Rate Control	Fine-tune frame rates for video conversion (e.g. PAL @ 24 FPS)
Strobe Output	Increased drive strength, configurable strobe pattern output
RS-232 Serial Port	Provides serial communication via GPIO TTL digital logic levels
Memory Channels	Non-volatile storage of camera default power-up settings
Temperature Sensor	Reports the temperature near the imaging sensor
Camera Upgrades	Firmware upgradeable in field via IEEE-1394 interface.

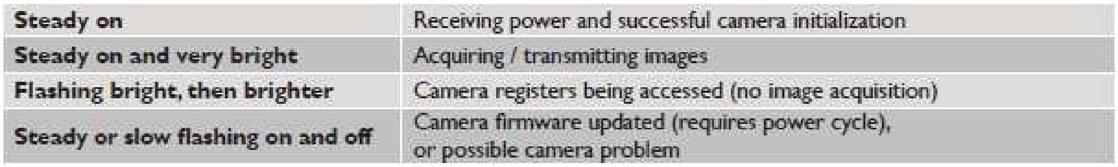
Calibration and mechanics



Lens System	High quality microlenses protected by removeable glass system
Accurate Pre-Calibration	For lens distortions and camera misalignments
Stereo Pair Alignment	Left and right images aligned to within 0.1[1] pixel RMS error
Calibration Retention	Minimizes loss of calibration due to shock and vibration

[1] Based on a stereo resolution of 640x480 and is valid for all camera models. Calibration accuracy will vary from camera to camera.

Status Led



Steady on	Receiving power and successful camera initialization
Steady on and very bright	Acquiring / transmitting images
Flashing bright, then brighter	Camera registers being accessed (no image acquisition)
Steady or slow flashing on and off	Camera firmware updated (requires power cycle), or possible camera problem