Real Time Object Recognition and Tracking Using 2D/3D Images

Vom Fachbereich Elektrotechnik und Informatik der
Universität Siegen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
(Dr.-Ing.)

genehmigte Dissertation

von

M.Sc. Seyed Eghbal Ghobadi

1. Gutachter: Prof. Dr.-Ing. habil. Otmar Loffeld
2. Gutachter: Prof. Dr. Bernd Radig
Vorsitzender: Prof. Dr. Roland Wismüller

Tag der mündlichen Prüfung: 16.09.2010

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier.

Abstract

Real Time Object Recognition and Tracking Using 2D/3D Images

Object recognition and tracking are the main tasks in computer vision applications such as safety, surveillance, human-robot-interaction, driving assistance system, traffic monitoring, remote surgery, medical reasoning and many more. In all these applications the aim is to bring the visual perception capabilities of the human being into the machines and computers.

In this context many significant researches have recently been conducted to open new horizons in computer vision by using both 2D and 3D visual aspects of the scene. While the 2D visual aspect represents some data about the color or intensity of the objects in the scene, the 3D denotes some information about the position of the object surfaces. In fact, these aspects are two different modalities of vision which should be necessarily fused in many computer vision applications to comprehend our three-dimensional colorful world efficiently.

Nowadays, the 3D vision systems based on Time of Flight (TOF), which fuse range measurements with the imaging aspect at the hardware level, have become very attractive to be used in the aforementioned applications. However, the main limitation of current TOF sensors is their low lateral resolution which makes these types of sensors inefficient for accurate image processing tasks in real world problems. On the other hand, they do not provide any color information which is a significant property of the visual data. Therefore, some efforts have currently been made to combine TOF cameras with standard cameras in a binocular setup. Although, this solves the problem to some extent, it still deals with some issues, such as complex camera synchronization, complicated and time consuming 2D/3D image calibration and registration, which make the final solution practically complex or even infeasible for some applications.

On the other hand, the novel 2D/3D vision system, the so-called MultiCam, which has recently been developed at Center for Sensor Systems (ZESS), combines a TOF-PMD sensor with a CMOS chip in a monocular setup to provide high resolution intensity or color data with range information.

This dissertation investigates different aspects of employing the MultiCam for a real time object recognition and tracking to find advantages and limitations of this new camera system. The core contribution of this work is threefold:

In the first part of this work, the MultiCam is presented and some important issues such as synchronization, calibration and registration are discussed. Likewise, TOF range data obtained from the PMD sensor are analyzed to find the main sources of noise contributions and some techniques are presented to enhance the quality of the range data. In this section, it is seen that due to the monocular setup of the MultiCam, the calibration and registration of 2D/3D images obtained from the two sensors is simply attainable [12]. Also, thanks to a common FPGA processing unit used in the MultiCam, sensor synchronization, which is a crucial point in the multi-sensor systems, is possible. These are, in fact, the vital points which make the MultiCam suitable for a vision based object recognition and tracking.

In the second part, the key point of this work is presented. In fact, by having both 2D and 3D image modalities, obtained from the MultiCam, one can fuse the information from one modality with the other one easily and fast. Therefore, one can take the advantages of both in order to make a fast, reliable and robust object classification and tracking system. As an example, we observe that in the real world problems, where the lighting conditions might not be adequate or the background is cluttered, 3D range data are more reliable than 2D color images. On the other hand, in the cases where

many small color features are required to detect an object, like in gesture recognition, the high resolution color data can be used to extract good features. Thus, we have found that a fast fusion of 2D/3D data obtained from the MultiCam, at pixel level, feature level and decision level, provides promising results for real time object recognition and tracking. This is validated in different parts of this work ranging from object segmentation to object tracking.

In the last part, the results of our work are utilized in two practical applications. In the first application, the MultiCam is used to observe the defined zones to guarantee the safety of the personnel in a close cooperation with a robot. In the second application, an intuitive and natural interaction system between the human and a robot is implemented. This is done by a 2D/3D hand gesture tracker and classifier which is used as an interface to command the robot. These results validate the adequacy of the MultiCam for real time object recognition and tracking at the indoor conditions.

Kurzfassung

Objekterkennung und -verfolgung in Echtzeit mit Hilfe von 2D/3D Bildern

In vielen Anwendungen der Computervision besteht die Hauptaufgabe aus dem Erkennen und Verfolgen von Objekten. Dazu zählen z.B. Anwendungen aus dem Bereich der Sicherheitsüberwachung, der Mensch-Maschine-Interaktion sowie Fahrerassistenz- und Verkehrsüberwachungssysteme oder auch Anwendungen aus dem medizinischen Bereich. Allen diesen Anwendungen ist das Ziel gemein, die visuellen Fähigkeiten des Menschen auf Maschinen und Computer zu übertragen.

In diesem Zusammenhang wurden in der Vergangenheit bis heute viele Forschungsansätze verfolgt, um neue Horizonte im Bereich der Computervision zu eröffnen, indem sowohl 2D- als auch 3D-Aspekte der Szene berücksichtigt werden. Während die 2D-Informationen sich auf die Farbe oder Intensität der Objekte in der Szene beziehen, geben die 3D-Daten Aufschluss über die Positionen der Objektoberflächen. Diese beiden Aspekte repräsentieren verschiedene Modalitäten, die notwendigerweise fusioniert werden müssen, um die farbige 3D-Welt effizient zu interpretieren.

Heutzutage sind die optischen 3D-Messsysteme, die auf der Phasenlaufzeitmessung beruhen und die eine örtlich aufgelöste Abstandsmessung auf Hardwarebasis ermöglichen, für die oben genannten Anwendungsbereiche sehr attraktiv geworden. Jedoch haben die derzeitigen 3D-Sensoren nur eine sehr geringe laterale Auflösung, was für Bildverarbeitungsaufgaben bei realen Szenen sehr hinderlich ist. Zudem übertragen sie keine Informationen über die Farbe, eine wichtige Eigenschaft der visuellen Daten. Aus diesem Grund wurde in letzter Zeit einiger Aufwand getrieben, um die 3D-Kameras mit Standardkameras in einem binokularen Aufbau miteinander zu verbinden. Obwohl dadurch das Problem zu einem gewissen Ausmaß gelöst wird, entstehen neue Probleme wie die genaue Synchronisierung, Kalibrierung und Registrierung der Daten, wodurch die finale Lösung sehr komplex oder teilweise unmöglich wird. Auf der anderen Seite wurde am Zentrum für Sensorsysteme eine 2D/3D-Kamera entwickelt („MultiCam“), die einen 3D-PMD-Sensor mit einem gewöhnlichen 2D-CMOS-Sensor in einem monokularen Aufbau verbindet und somit gleichzeitig hochaufgelöste Farbbilder und Distanzdaten zur Verfügung stellt.

Diese Dissertation untersucht verschiedene Aspekte der MultiCam für eine Objekterkennung und -verfolgung in Echtzeit und stellt die Vorzüge und Einschränkungen dieser Technik heraus. Der Kernbeitrag dieser Arbeit ist in drei Punkten zu sehen:

Im ersten Teil der Arbeit wird die MultiCam vorgestellt und auf einige wichtige Eigenschaften wie die Synchronisierung, Kalibrierung und Registrierung der Daten eingegangen. Außerdem werden die Abstandsdaten der Kamera untersucht und einige Techniken zur Rauschunterdrückung werden vorgestellt. Auf Grund des monokularen Aufbaus der MultiCam kann die Kalibrierung und Registrierung der 2D/3D Bilder sehr einfach erhalten werden [12]. Die Synchronisierung der Daten ist dank einer gemeinsamen FPGA-Verarbeitung möglich, was ein entscheidender Punkt in Multisensorsystemen darstellt. Dieses sind die wichtigsten Eigenschaften, die die MultiCam für ein optisches Objekterkennungs- und verfolgungssystem sehr effizient machen.

Im zweiten Teil wird der Hauptpunkt dieser Arbeit präsentiert. Dadurch, dass 2D- und 3D-Bilder durch eine Kamera akquiriert werden, kann man die Informationen der einen Modalität mit der anderen sehr einfach fusionieren. Somit können beide Modalitäten genutz werden, um ein schnelles, zuverlässiges und robustes Objektklassifizierungs- und verfolgungssystem zu entwickeln. Zum Beispiel können bei in der Realität häufig auftretenden schlechten Lichtverhältnissen die 3D-Daten benutzt werden, um

Objekte zuverlässiger zu detektieren, als dies mit den Farbinformationen möglich wäre. Auf der anderen Seite ist zur Erkennung von Gesten eine hohe laterale Auflösung nötig, so dass hierfür das 2D-Farbbild sehr gut verwendet werden kann. Aus diesem Grund bietet die schnelle Fusion der 2D/3D-Daten der MultiCam auf einem Bildpunkte-, Merkmals- oder Entscheidungs-orientierten Level vielversprechende Ergebnisse für eine Objekterkennung und -verfolgung in Echtzeit. Dies wird in dieser Arbeit in verschiedenen Abschnitten validiert, angefangen bei der Objektsegmentierung bis hin zur Verfolgung.

Im letzten Teil werden die Ergebnisse unserer Arbeit in zwei praktischen Anwendungen realisiert. In der ersten Anwendung wird die MultiCam zur Überwachung definierter Zonen benutzt, um die Sicherheit des Bedienpersonals eines Roboters zu gewährleisten. In der zweiten Anwendung wird ein intuitives und natürliches Interaktionssystem zwischen Mensch und Roboter implementiert. Dies wird durch eine Handverfolgung und Gestendetektion erreicht, die als Schnittstelle zur Roboterbedienung dienen. Diese Resultate bestätigen die Effizienz und Eignung der MultiCam für die Objektdetektion und -verfolgung in Echtzeit bei Innenraumbedingungen.

Acknowledgments

Many people have shared their love, inspiration, friendship, time, insight and advice with me, without them it would not have been possible to write this doctoral thesis. I would like to take this opportunity to express my profound gratitude and appreciation to only some of whom it is possible to give particular mention here.

First and foremost, my special thanks are due to my family, especially to my parents who are the pillars and source of love and inspiration in my life. I owe a large debt of gratitude to my loving wife who is my best company, friend and partner and has sacrificed her time with me so that I could accomplish this dissertation.

I am deeply grateful toward my supervisor, Prof. Dr. Otmar Loffeld, for his consistent help, guidance and attention during the whole work. I owe many thanks to Dr. Klaus Hartmann, the team leader of our research group, not only for his valuable advice regarding my research work, but also for orienting this dissertation with his admirable experiences.

I would like to give thanks and appreciation to Prof. Dr. Bernd Radig for his role as the second supervisor. He helped me to find how to present this work in a scientific manner.

I am truly indebted to many who made my time here at the Center for Sensor Systems (ZESS) a wonderful experience and helped me in different ways. Omar Edmond Loepprich for the close collaboration in the projects, nice and intensive discussions and providing me with some nice codes for 2D/3D image acquisition and visualization. Thank you for being as a close friend the whole time and supporting me. Oliver Lottner for his liberal and gentle attitude, for his collaboration in writing publications and sharing his useful knowledge in 2D/3D camera system with me. Dr. Wolfgang Weihs for very helpful discussions, especially at an early stage of my work, and providing us with the latest version of required softwares. The former colleagues, Arun Prasad, Arnd Sluiter and Sigurd Kaiser, for their hard work in the implementation of the novel 2D/3D camera system.

Working days are much more pleasant with nice colleagues around. I greatly appreciate my roommate and my dissertation reader, Dr. Stefan Lammers. Thanks for many intensive discussions and critical suggestions in making this dissertation more valuable than it could be on its own.

I would also like to give a big thank to Sven Stark for helping me in doing so many measurements as well as implementation of necessary hardware components and generating ground truth images for this work. My sincere thanks are given to Wolf Twelsiek and Rolf Wurmbach for overall support and helping me at the lab. Thanks are due to Mrs. Renate Szabo for her very friendly face, kind words and help with administrative issues.

It was also a great pleasure to have Dr. Stefan Kndelik and Mrs. Silvia Niet-Wunram in the IPP administration to give any necessary help.

Contents

List of Abbreviations

2D	Two Dimensions
3D	Three Dimensions
AdaBoost	Adaptive Boosting
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CONDENSATION	Conditional Density Propagation
CPU	Central Processing Unit
CT	Computed Tomography
DOF	Degree of Freedom
Dyn3D	Dynamic 3D
EM	Expectation Maximization
FOV	Field of View
FPGA	Field Programmable Gate Array
GSVD	Generalized Singular Vector Decomposition
GUI	Graphical User Interface
HRI	Human Robot Interaction
KEM	K-Means Expectation Maximization
LDA	Linear Discriminant Analysis
MoG	Mixture of Gaussian
PCA	Principal Component Analysis
PDA	Pairwise Discriminant Analysis
PMD	Photonic Mixer Device
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SAR	Synthetic Aperture Radar
SLA	Structured Light Approach
SVM	Support Vector Machines
TCP/IP	Transmission Control Protocol / Internet Protocol
TOF	Time of Flight
VOI	Volume of Interest
VGA	Video Graphics Array

1

Introduction

Research is to see what everybody else has seen, and to think what nobody else has thought.
Albert von Szent-Gyoergyi (1893-1986)

This thesis aims at the study and analysis of different aspects of real time object recognition and tracking using a monocular 2D/3D imaging system and to validate the results in practical applications. In this chapter, an introduction is given which first states the motivation behind our work, following with the problem description and finally highlighting the key contribution of the thesis. The last section outlines the theoretical and practical framework on which this work is based.

1.1 Motivation

Object detection and tracking is of utmost importance for different kinds of applications such as safety, surveillance, man-machine interaction, driving assistance system, traffic monitoring and many more. In each of these applications, the aim is to detect the desired object and find its position at each time instance. While in the safety application the personnel, as the desired objects, should be detected and tracked in hazardous environments to keep them safe from the machinery, in the surveillance application, they are detected to analyze their motion behavior for conformity to a desired norm for social control and security. Man-Machine Interaction, on the other hand, has become an important topic for the robotic community. A powerful intuitive interaction between man and machine requires the robot to detect the presence of the user and interpret his gesture motions. This requirement can be fulfilled efficiently just by having a robust gesture recognition and tracking system. Likewise, a driving assistance system detects and tracks the obstacles, vehicles and pedestrians in order to avoid any collision in the moving path. The goal of traffic monitoring in an intelligent transportation system

is to improve the efficiency and reliability of the transport system to make it safe and convenient for the people. Vehicle tracking and accident detection in the roads are the main tasks of traffic monitoring. There are still so many significant applications in our daily life in which object recognition and tracking play an important role.

However, in all such applications the first question which arises is what kind of information should be acquired from the environment to make a fast, robust and reliable object detection and tracking system. Since the biological vision is the most significant source of information which is used by the humans for this purpose, it has apparently inspired the idea to use the same artificially for an object recognition and tracking system. Thus, advanced image sensors are employed to provide visual data for a vision based object detection and tracking system.

Visual object detection and tracking is nevertheless a complex problem due to the noise in the data, huge size of visual information, sensitivity of the visual data by changing the lighting condition, complexity in the object shape and its motion, non-rigidity of the object, occlusion, abrupt motion, real time requirements and so many other issues. In this context, visual object detection and tracking has emerged as one of the active research areas within the computer vision and artificial intelligence communities.

In order to build a typical visual object recognition and tracking system, the following main questions should first be addressed:

- How to acquire the visual data?
- How to represent the object of interest in the large visual data?
- How to detect and classify the objects in the image?
- How to track the object of interest and find its trajectory?

The first issue is addressed using different kinds of visual sensors whereas the last three questions can be answered differently based on numerous object detection and tracking algorithms. The selection of the suitable sensor and algorithm is heavily dependent on the application requirements.

In the recent years, many research has been conducted, both in the software[1] and the hardware[2] domain, either to propose a new solution for the problem of visual object recognition and tracking or to improve the performance of current approaches. In fact, different advanced algorithmic solutions besides the progress in sensor technology have opened new horizons in the computer vision field.

Most of the approaches in computer vision utilize different types of solid state imaging sensors like CCD[3] and CMOS[4] which can observe only greyscale or color information. The world is, however, three dimensional and therefore the spatial information about the objects in the scene as well as their 3D shape are vital factors in many computer vision problems. For this reason the range sensors, which can provide depth information, have gained a lot of attention to be used in different applications in last years.

On the other hand, when we, as human beings, see, we observe both the 2D geometrical features from greyscale or color data and 3D aspects from range information. The same is absolutely necessary for most of computer vision applications ranging from gaming to safety and security. Therefore, the new approaches and novel technologies are making numerous efforts to merge both 2D and 3D visual aspects to improve the performance of the application.

This thesis on the one hand will present a new multimodal 2D/3D imaging system, which has been implemented at ZESS, and on the other hand will show some solutions to the object recognition and tracking problems in real time applications by fusion of 2D and 3D images.

1 Software designates the algorithmic aspects.
2 Hardware denotes the visual sensor technology.
3 Charge Coupled Device.
4 Complementary Metal Oxide Semiconductor.

1.2 Problem Description

Employing static 2D imaging sensors, like CCD or CMOS sensors, has some difficulties for object recognition in real world problems where the lighting conditions might change as well as having shadows in the scene. Likewise, generating of 3D range data from static 2D images is a computationally expensive process. In other words, as the size of image and video information increases, the problem of implementation of a real time system to detect and classify the objects will become more complex. To overcome these problems, many approaches have been followed in recent years such as enhancement of processing speed attained by means of parallel processing techniques and implementation of new and fast algorithms.

Range images[5], on the other hand have gained a lot of attention recently for such applications as detection and classification of moving objects. The range data are usually provided by different 3D range sensor systems such as Laser Range Finder, Stereo Vision System, Structured Light Approach (SLA) and 3D Time of Flight (TOF) Camera. Since each of these sensors have their own strength and weakness points, the selection of suitable range system depends on the application requirements. For example, although the common 2D laser range finder can provide accurate range data, it usually scans the environment radially in a plane parallel to the ground. In other words, it provides the range information only in 2D slices of the environment. Also, the time it takes to scan the whole surface of the object is one of its main drawbacks in real time applications. SLA, same as laser range finder has a low acquisition rate and it is not so appropriate for most of real time applications. Likewise, it is sensitive to the illumination which creates limitations for real world applications. Stereo vision systems like CCD or CMOS cameras have difficulties to provide reliable information under varying lighting conditions. In addition, the range images of stereo vision are texture dependent and without presence of the texture on the object, the range measurement gets completely wrong data. Due to high computational requirements for the calculation of the disparity map from right and left images, the frame rate of the stereo vision is also an issue which should be considered for some applications. In comparison to all these range sensor systems, 3D TOF cameras, which fuse range measurement with the imaging aspects at the hardware level, can provide range and intensity information at video frame rates which is promising for real time applications. However, one of the main weaknesses of the current TOF sensors is their limitation in lateral resolution which makes them inefficient for many applications in which the high resolution image data is required. On the other hand, they do not provide any color information which is a significant property of the visual data. These drawbacks can be overcome by combining a TOF camera with a conventional RGB camera in which the advantages of high resolution imaging aspects is utilized in a 2D/3D scenario. In recent research works there is a tendency for such a combination because even with regard to the emerging new generation of TOF sensors with high resolution[6], an additional 2D sensor still results in a higher resolution and provides additional color information.

Regarding a simple combination of a TOF camera with a standard one in a binocular setup, in which two cameras are put close to each other, and correlate their generated images, the following issues, however, should be addressed:

- **Binocular parallax effect:** Since in such a binocular setup there are two objective lenses, parallax errors occur. In other words, the 2D and 3D images are not directly coregistered and therefore some techniques should be applied to register two different images.
- **Lens distortion:** The two cameras have different lens distortions and therefore for an error-free operation a calibration technique should be applied.
- **Frame rate:** As the two cameras usually have different maximal frame rates, a precise

5 The terms range image, 3D image and depth image are used interchangeably in this work.

6 At the time of writing this thesis the resolution of such cameras is limited up to some thousand pixels. For example: PMD-41K-S (204×204 Pixels) [31], Zcam-prototype (320×480 Pixels) [30] and SwissRanger 4000 (176×144 Pixels) [29].

synchronization for two cameras is necessary.

Thus, the combination of such two camera systems in a binocular setup demands complicated and time consuming calibration, registration and synchronization approaches which makes the final solution practically complex or even infeasible for some real time applications.

1.3 Key Contribution

The key contribution of this work is to fuse Time of Flight (TOF) range data with high resolution 2D images for real time object recognition and tracking. This is performed in a monocular setup which implies no need for any complex registration, calibration and synchronization techniques.

The multimodal image acquisition device, used in this work, is a monocular 2D/3D vision system, called the MultiCam[7]. This camera consists of two imaging sensors, a near infrared lighting system, a FPGA based processing unit, a beam splitter and USB 2.0 communication interface. A conventional 10-bit CMOS sensor with VGA resolution of 640×480 pixels and a Photonic Mixer Device (PMD) [31] with a resolution of 64×48 pixels are employed to provide high resolution 2D information and 3D range data respectively. The PMD is an implementation of an optical TOF sensor, able to deliver range data at quite high frame rates[8]. The principles of this sensor will be presented briefly in the next chapter. The dichroic beam splitter behind the camera lens is used in order to divide the incident light into two spectral ranges: The visible part, which is forwarded to the CMOS chip and the near infrared part to the TOF sensor. Thus, the MultiCam is, indeed, a multi-spectral device.

Since the MultiCam is a monocular camera with one unique objective lens and as the 2D and 3D sensors have 1:10 proportional resolution, the range image and the 2D image correlate directly using a trivial mapping technique. This makes the fusion of 2D and 3D features much easier and faster which consequently has a big positive influence on the performance concerning real time aspects.

In fact, this dissertation states that by having both 2D and 3D image modalities, obtained from the MultiCam, one can fuse the information, at pixel level, feature level or decision level, from one modality with the other one easily and fast. Therefore, one can take the advantages of both in order to make a fast, reliable and robust object classification and tracking system.

1.4 Thesis Outline

In chapter 2, we will review the 3D vision system in general and the 2D/3D vision system, especially the MultiCam, in particular. In this chapter some main aspects of the used 2D/3D camera system will be discussed and it will be shown how 2D and 3D images captured by the MultiCam can be registered easily. We will highlight this point because it is the key factor which makes this thesis different from other similar works in which the 2D and 3D image data are fused.

In chapter 3, object recognition using 2D/3D imaging data will be studied. This consists of feature extraction, multimodal data fusion, segmentation and classification. We have considered two types of feature extraction in our work. The features which are derived based on some heuristics which are called human generated features and the features which are derived using some mathematical methods which are called machine generated features. In fact, selecting the type of the features depends on the problem. While in some problems an object can be represented easily using a set of heuristic features, in some others it is not the case and therefore a mathematical approach should be applied to highlight the features with highest similarities and differences in a huge data set. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two feature extraction techniques which will be

7 The term MultiCam will be used in this work to denote the monocular 2D/3D vision system which has recently been developed at the Center for Sensor Systems (ZESS).

8 The frame rate of a TOF sensor depends on some parameters which will be discussed in the next chapter.

used as the machine generated type in this work. Also, we will discuss the heuristic approaches with an example to show how the knowledge based features can make the problem easier.

Next in this chapter, we will discuss image segmentation as an important preprocessing step in computer vision solutions. The segmentation technique based on TOF range images and using clustering techniques will be discussed. Likewise, we will study some aspects of multimodal image segmentation and represent some results.

In the last part of this chapter, the classification approach based on supervised learning techniques is reviewed. Two important classifiers including Support Vector Machines (SVM) and AdaBoost are studied and we will test these classifiers for moving object classification tasks using multimodal 2D/3D image data.

Chapter 4 addresses some solutions to the object tracking using 2D/3D images. In this chapter, we first study dynamic scene analysis aspects like background subtraction and real time issues. The tracking techniques used in this work are divided in two main categories. Object representation and identification is the first type in which the objects of interest are detected in each frame and then the correspondences are matched based on a distance function, which is derived from 2D/3D features. In this part of work we will show how the fusion of 2D and 3D features can improve the performance of tracking. Probabilistic approach is the second category of object tracking techniques, used in this work. The Kalman filter and the CONDENSATION algorithm are two main probabilistic techniques which will be discussed in this chapter. We will review these two approaches briefly and then apply them to track the people. Finally, we will compare the results of these two trackers and conclude some points.

In chapter 5, we validate the results of the reviewed techniques in two practical applications. In the first application the safety of the personnel in the close cooperation with a robot is analyzed. We will show how the different zones around an industrial robot can be monitored dynamically using 2D/3D camera system to guarantee the safety of the personnel. In the second application, which complements the first one, an interaction system between the robot and the user will be presented. In fact, we will implement an intuitive, natural commanding system to control the robot. This is done based on hand gesture recognition and tracking system using multimodal 2D/3D images.

Finally, in chapter 6 we will conclude our work and remark some points.

2

Analysis of 2D/3D Image Data

The scientist is not a person who gives the right answers, he's one who asks the right questions.
Claude Lévi-Strauss (1908-2009)

Nowadays, in order to facilitate our daily activities by employing computers, tools and machines, different types of visual data like greyscale, color, range, X-ray, CT and SAR images are used. In this chapter, we analyze 2D/3D imaging data which are combinations of Time of Flight range data with greyscale or color information. For this purpose, first we review the main techniques of range imagery and discuss some of their important issues. Next, we will study the principle of the 2D/3D camera system which is used in this work and analyze the main aspects of the 2D/3D visual data.

2.1 3D Range Measurement

As our state world is at least three dimensional, there is an increasing demand on depth perception in different applications of computer vision. In fact, in many practical applications, range data, which contains 3D information about a scene, is used to perceive the world in three dimensions. Range images, in contrast to 2D intensity or color images, can explicitly represent three dimensional information about the surface of objects in a scene. In other words, a range image is a digital image in which each pixel expresses the distance between a known reference and a visible point on the object surface in the scene. Range images should provide geometric information about an object independent of its position, direction, and intensity of light sources illuminating the scene, or even of the reflectance properties of that object. 3D range images are also referred to as depth images, depth maps, xyz maps, surface profiles and 2.5D images [15]. In this section we will review the main significant approaches which are used for depth perception.

Range sensors, in general, can be classified into passive and active devices. While active range sensors project energy like light into the scene and detect the distance by determining some properties of the reflected energy[9] back from the scene, the passive techniques reconstruct the range without emitting any energy into the scene and only based on feature matching in 2D images. However, feature matching is a time consuming process and it can fail if no features are present. On the other hand, active approaches overcome this problem and simplify many tasks in range measurement at the cost of using advanced sensitive elements to the reflection properties and consequently cost of 3D range sensor.

Some typical examples of 3D range measurement techniques are Stereo Vision Technique, Time of Flight (TOF) and Structured Light Approach (SLA). The principle of these techniques, which are schematically illustrated in Fig. 2.1, will be reviewed in the following subsections.

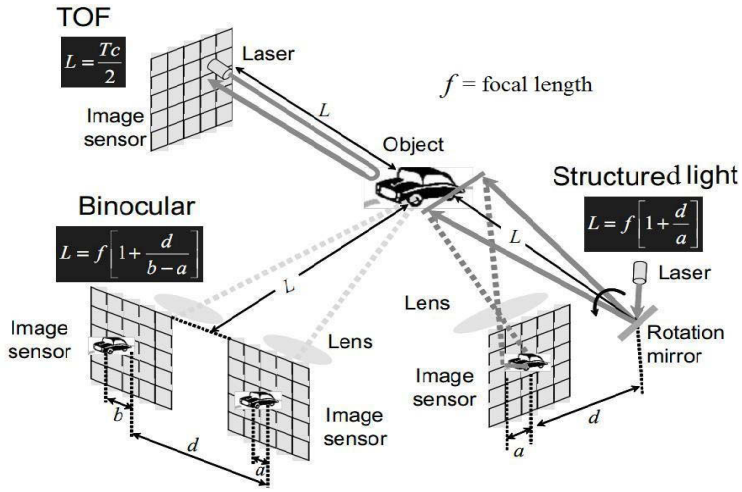

Figure 2.1: Conceptual schematic of some typical 3D range measurement techniques [4].

2.1.1 Stereoscopic Imaging

Stereoscopic imaging is a passive triangulation method in which depth information about the scene is measured from multiple static 2D images, each acquired from a different viewpoint in space. In a classical stereoscopic vision technique, called stereo vision, two cameras are employed in a binocular vision system, analogous to the two eyes in the human visual system, to capture two images. Stereo vision which is a passive method has the advantage that it does not require any light sources and only two 2D sensors are used.

In binocular stereo vision in which two cameras are displaced from each other, by knowing the camera focal lengths and using geometry, the depth of objects in an imaged scene can be estimated in a canonical stereoscopic vision system as follows [1], [4], [36]

$$L = f\,(1+\frac{d}{b-a}) \tag{2.1}$$

where f is the focal length of camera lenses, b-a is the disparity and d is the parallax or inter ocular separation (see Fig. 2.1).

9 For example, angle of return is the property which is measured in the triangulation technique and time, phase or frequency delay are the properties which are considered in the Time of Flight approach.

In any stereo vision technique the range image of the scene is obtained in two following process steps:

- **Correspondence Process:** In the first step a specific searching and matching technique is applied to find pairs of matched points in two images. These points are found such that each point in the pair is the projection of the same 3D point in the scene. The input of correspondence process are two 2D images taken from two cameras and the output will be a disparity map which is the difference of the matched points on the horizontal coordinates.
- **Reconstruction Process:** By having the disparity map, which is derived from previous step, and given the stereo geometry, the 3D image of the scene can be reconstructed.

The main drawback of stereoscopic imaging approach is that no range data can be obtained in uniform regions, like a white wall, where there are no features present for the correspondence process [18]. The shadowing effect is also a typical problem for stereo vision systems which can be minimized by using multi view triangulation systems at the price of an enormous increase of data processing as well as increasing the number of cameras [17].

2.1.2 Structured Light Approach

Structured Light Approach (SLA) is an active triangulation method for 3D range measurement. It is based on the same principle of passive stereo vision. In the SLA, a light projector and a 2D intensity camera, which are placed at a certain distance from each other, are used. The projector illuminates the scene with a light pattern, so-called structured light. The most common patterns are planes and single beams. At the same time, the 2D camera acquires an image of the scene which is called pattern image. The intersection of the projected light plane with the scene surface is a planar curve called the strip. By having the geometry of light source and the camera as well as the orientation of the project plane, the depth information of all points under the strip can be calculated. In fact, identification of the light planes in the pattern image is the characteristic problem of the SLA, comparable to the correspondence problem in stereo vision systems [77]. However, the identification or decoding problem in the SLA, contrary to the correspondence problem in the stereo case, is easier because the laser spots are normally brighter than the other points in the pattern image which can be identified obviously [56].

As it can be seen in Fig. 2.1, in the SLA technique, the light projector and the 2D camera with one point on the object surface build a triangle in which the distance L can be calculated through triangulation method as follows [4]

$$L = f\,(1+\frac{d}{a}) \tag{2.2}$$

where f is the focal length of the camera lens, d is the distance between camera and light projector and a is the lateral displacement of the light spot on the pattern image.

The main important task of depth perception in the SLA is to identify the light planes in the pattern image if more than one light plane is projected at a time. This problem can be solved by employing a light projector which encodes the light planes with different IDs, for example by assigning each light plane a specific color, the light planes can then be decoded in the pattern image [82].

The main drawbacks of the active triangulation technology are its low acquisition rate and missing range data at parts of the scene which are visible to the 2D camera and not visible to the light projector or vice versa [17], [56].

Likewise, most of the SLA systems impose strong constraints on the scene by requiring the following conditions [82]:

- Controlling the illumination in the scene
- Static scene

- Natural scene reflectivity
- Low contrast of the textures in the scene.

There have been some works in the enhancement of the SLA system and reducing the constraints. For a much more in depth understanding the reader is referred to [82], [44].

2.1.3 Laser Pulse Range Finder

Laser range finder is an active Time of Flight (TOF) based approach for measuring the distance of objects in the scene. In this technique a laser is used to emit a pulse in the scene and the distance L is determined by measuring the time the pulse takes to hit the object, be reflected and reach back to the detector as follows

$$L = \frac{T \cdot c}{2} \tag{2.3}$$

where c denotes the speed of light and T is the echo time.

It should be noted that for an unambiguous range measurement the pulse width t_p should be smaller than time T [104].

The main problem in laser range finder is the realization of an exact time measuring process. It is because the accuracy of laser range finder depends on the speed of detector and timing circuit which is used in this device. The main advantage of this technique is its large unambiguous distance measurement which requires a high dynamic receiver with a large bandwidth [17].

TOF laser range finders, which are the most commonly used systems for 3D digitization, are usually available as 2D or 3D scanners [104]. In a 2D laser scanner, the laser beam is usually swept by a rotating mirror and the laser range finder provides depth of the points which lie in the plane in which the laser beam is swept. One of the main drawbacks of 2D laser scanners is that they can only scan in horizontal direction, i.e., they provide the range only in planes. On the other hand, a 3D laser scanner which is usually built based on 2D laser scanner can provide range information in 3D volumes by rotating the scanning module in vertical direction at regular time intervals [104], [81].

The main drawback of laser range finders is their long acquisition time which is due to the scanning process. The output of laser range finders are point clouds which are not directly usable in most of 3D applications and therefore they should be converted to 3D models or range images which is itself a time consuming process.

2.1.4 Time of Flight Camera

Range imaging in a 3D-Time of Flight camera is the fusion of the distance measurement technique with the imaging aspect. The principle of the range measurement in a TOF camera, similar to the laser range finder, is based on the measurement of the time the light needs to travel from one point to another. This time which is so-called Time of Flight is directly proportional to the distance the light travels (see equation 2.3). However, in a TOF camera, the round-trip time is not measured directly, but the phase difference between the sent and received signals is measured. In the following we will review the principle of our TOF camera which is based on Photonic Mixer Device (PMD) [31].

ZESS-Time of Flight Camera[10]

Our 3D non-scanning Time of Flight (TOF) camera system consists of an infrared lighting source,

10 This is a Time of Flight camera, based on PMD-3KS, which has been implemented at Center for Sensor Systems (ZESS).

Photonic Mixer Device (PMD) sensor [31], FPGA based processing and communication unit including FireWire, USB and Ethernet.

The lighting source illuminates the scene with the modulated near infrared light signal which is generated using a MOSFET based driver and a bank of high speed infrared emitting diodes at the frequency of 20 M*Hz*. The illuminated scene is observed by a smart pixel array (PMD) via an optical lens for focusing, where each pixel on the PMD sensor can individually determine the turnaround time of the modulated light [31]. Typically this is done by using continuous modulation and measuring of the phase delays in each pixel [70]. The conceptual schematic of the TOF camera, which has been developed at ZESS is illustrated in Fig. 2.2.

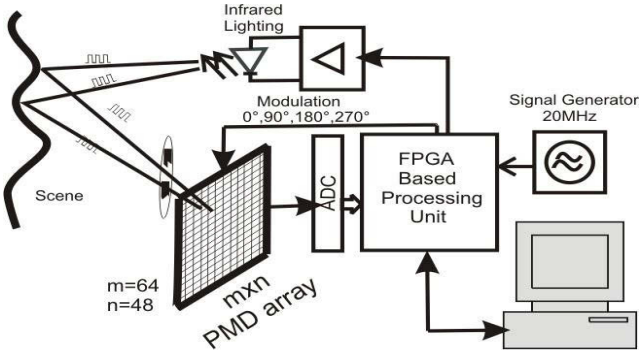

Figure 2.2: Conceptual schematic of ZESS-TOF camera based on PMD-3KS.

Assuming continuous sinusoidal or rectangular modulation, the distance is calculated as follows [70]

$$d = \frac{c \cdot \Delta\varphi}{4\pi \cdot f_{mod}} \tag{2.4}$$

where f_{mod} denotes the modulation frequency and $\Delta\varphi = 2\pi \cdot f_{mod} \cdot t$ represents the phase delay.

To calculate the phase delay, the autocorrelation function of electrical and optical signal is analyzed by a phase-shift algorithm. Using four samples A_1, A_2, A_3 and A_4 each shifted by 90 degrees, the phase delay $\Delta\varphi$ can be calculated using the following equation [70]

$$\Delta\varphi = \arctan\left(\frac{A_1 - A_3}{A_2 - A_4}\right). \tag{2.5}$$

In addition to the phase shift of the signal, the strength of the received signal a, which is also termed as modulation amplitude, and the gray scale value b are formulated respectively as follows [70]

$$a = \frac{\sqrt{(A_1 - A_3)^2 + (A_2 - A_4)^2}}{2} \tag{2.6}$$

$$b = \frac{A_1 + A_2 + A_3 + A_4}{4}. \tag{2.7}$$

Likewise, the theoretical response of a pixel can be expressed by [67]

$$r_c = \sum_{n=1}^{N} B_n \cdot e^{j4\pi \frac{r_n}{\lambda}} \tag{2.8}$$

where $\lambda = c / f_{mod}$ denotes the wave length of the modulated signal, B_n is the backscatter coefficient of

the point n and $r_n=[x_n, y_n, z_n]$ represents the distance vector to all visible object points with $n=1,...,N$.

At the modulation frequency of $20\,MHz$ the unambiguous distance is equal to $15\,m$, i.e., the maximum distance for the target is $7.5\,m$. This is because the illumination has to cover the distance twice: from the sender to the target and back to the sensor chip.

The environment lighting conditions in the background should be considered in all TOF optical sensors. This effect can be handled by various techniques such as using an optical filter which only passes the band around the active light [73], or applying some algorithmic techniques that remove the noise of ambient light [69]. In our case, PMD has an in-pixel SBI-circuitry (Suppression of Background Illumination) which increases the sensor dynamics under strong light conditions [66].

TOF cameras, unlike the stereo vision camera, are texture independent and since the range is calculated directly at the hardware level for each pixel with minimal processing, a very high frame rate, dependent on the exposure time, can be obtained[11].

One of the main limitations of current TOF cameras is their low lateral resolution which will be addressed in this chapter by implementing a 2D/3D vision system.

Finally, in order to have a general overview about the discussed 3D range measurement techniques, the main important advantages and drawbacks of each of them are summarized in Table 2.1.

Table 2.1: Advantages and disadvantages of main 3D range measurement techniques.

	Advantages	**Disadvantages**
Stereo Vision	High resolution 3D data with intensity or color information No need for energy source (passive)	No range data on the surfaces without texture Sensitive to lighting conditions Disturbed by shadows Calibration needed Time consuming correspondence problem
Structured Light Approach	High resolution 3D data Low cost	Low acquisition rate Sensitive to illumination changes Limited to static scenes
Laser Range Finder	High accuracy Long distance measurement To some extent insensitive to weather conditions	Long acquisition time due to scanning Need for mechanical components High cost
Time of Flight Camera	High acquisition rate Insensitive to lighting changes in indoor environments Providing range image in one-shot Reasonable cost	Low lateral resolution Noisy range data from poorly reflecting surfaces Not reliable for outdoor applications

11 As an example, by setting the exposure time to $5\,ms$ and using a USB 2.0 communication protocol, the frame rate of 50 range images per second, with the resolution of 64×48 pixels, is possible.

As it can be seen from Table 2.1, each of the range sensors has its own weaknesses and strengths. This is the reason why many current approaches have combined different 3D range sensors in order to take advantages of more sensors and reduce their drawbacks. However, the sensor combination or fusion should be performed in an optimal way in order not to increase the complexity or cost of the final sensor system or to become infeasible for a practical application.

2.2 2D/3D Vision System

A 2D/3D vision system usually denotes the combination of a 3D range sensor with a 2D vision sensor in order to take the advantages of both types of sensors. In this context, some investigations have been conducted either to integrate a 3D range measurement system with a conventional camera or to combine two different types of range sensors to obtain both range and intensity in a 2D/3D vision system. Some of main important sensor combinations are reviewed in the following:

- **Combination of laser range finder with a standard 2D camera system:** The integration of laser scanners and vision sensors is performed in order to compensate the limitations of each of them by using the other one. For example, the main advantage of laser range finders over vision sensors is their high accuracy of range measurement in large angular fields. However, most of current laser scanners can only provide 2D representations of the 3D space and therefore they have disadvantages in data completeness as well as lacking of color information. Fusion of these two sensors can provide reliable range data with high resolution intensity or color data [21], [68], [34], [76].
- **Combination of a stereo vision system with a laser range finder:** 2D laser range finders measure distance to the objects only in a 2D plain form, and stereo vision is not able to determine distances to surfaces with little or no texture at all. While the potential solution of 3D range finders is a quite expensive one, the integration of a 2D laser range finder with stereo vision not only solves such problems but also provides color information taken from the vision system [72], [43], [65].
- **Combination of a TOF camera with a high resolution 2D camera:** Although Time of Flight cameras are becoming more attractive as very fast range measurement sensors in different fields, they still suffer from low lateral resolution. Many applications, however, require high resolution range and color data. The combination of a TOF camera with a conventional high resolution 2D camera is a typical solution for this problem. Some of current research works in which a TOF camera is combined with a standard one have been referenced in [102], [64], [33] and [55]. In all these works, a TOF camera is fixed close to a normal camera in a prototype binocular setup.
- **Combination of TOF cameras and stereo vision systems:** The combination of TOF camera and stereo vision can improve the final range data dramatically. TOF cameras can provide real time distance data in real world conditions, where a stereo vision does not work well. For example, while stereo vision does not generate proper range data of uniform surfaces without texture or under varying lighting conditions, a TOF camera does. However, TOF sensors have low lateral resolution which makes them inefficient in providing detailed intensity and range data. Likewise, the details and discontinuities in intensity and range decrease the performance of a TOF sensor, whereas they increase the performance of stereo vision. Thus, the key idea in this integration is to fuse the range information captured by both TOF camera and stereo vision in order to solve some ambiguities of the range data and increase the performance of range measurement [42], [54], [53], [63].

Selecting the type of sensor integration in a 2D/3D vision system depends heavily on the application requirements and it should be done in such a way to achieve an optimum solution with respect to some criteria such as accuracy, reliability, real time aspects and cost.

Although, fusion of data (depth and intensity) from multiple sensors can improve the reliability of a system, if the fusion is not performed in a proper way, the whole effort of sensor integration is in vain.

Most of the aforementioned sensor combinations, however, need complex and time consuming fusion techniques, such as calibration and registration of data which might make them inefficient for many real time applications. Apart from this point, the hardware limitation of such combinations, like time synchronization, is also a big issue which should be addressed.

2.3 The MultiCam

Among all sensor combinations for making a 2D/3D vision system, integrating the TOF sensor with a conventional 2D sensor in a monocular setup which realizes 3D range measurements with high resolution intensity or color data, is one of the promising techniques in 2D/3D imaging.

In fact, although a TOF camera can determine both 2D intensity and 3D depth at pixel level, the low lateral resolution of the current TOF sensors makes them inefficient for many applications in which a high resolution visual data analysis is required. As a solution to this problem, the Center for Sensor Systems (ZESS) has recently proposed and implemented a 2D/3D vision system, the so-called MultiCam[12], in which a TOF sensor with a high resolution CMOS chip are integrated in a monocular setup. In this section, we will introduce this 2D/3D camera system and study some of its main aspects. A detailed analysis of the MultiCam in the scope of our Dyn3D-DFG project [32] is still ongoing and for a much more in depth understanding the reader is referred to [12].

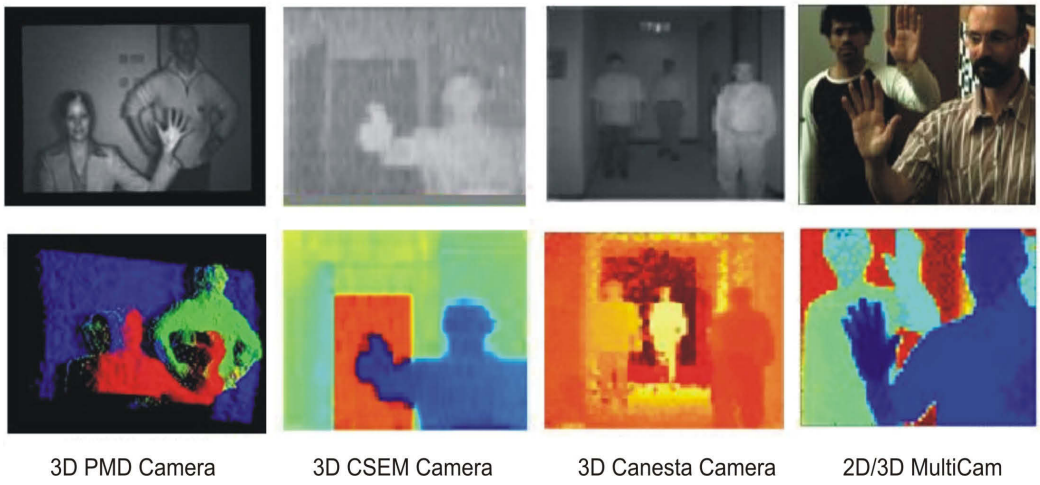

Figure 2.3: Some Time of Flight 2D/3D images. First three columns: Low resolution intensity and range images taken by three different TOF cameras: PMD [31], SwissRanger [29] and Canesta [28] respectively. Last column: 2D/3D image taken by the MultiCam.

In order to understand the main problem of current TOF sensors, some 2D/3D image samples taken by three main TOF sensor providers are illustrated in Fig. 2.3. As it can be seen, on the one hand these images have low lateral resolution and on the other hand they are lacking color information. Only the MultiCam, as a solution to such problems, provides range data with high resolution color information as it has been shown in the last column of Fig. 2.3.

The MultiCam consists of two imaging sensors (a conventional 10-bit CMOS sensor with VGA resolution and a PMD sensor with 3K resolution[13]), a dichroic beam splitter, a near-infrared

12 The concept of the MultiCam originates from [16].

13 The design of the MultiCam is such that it allows to replace the 2D/3D sensors with their alternatives at the cost of some FPGA modifications. For example, PMD-3K-S can be replaced by the new version PMD-19K-S or PMD-41K-S [31].

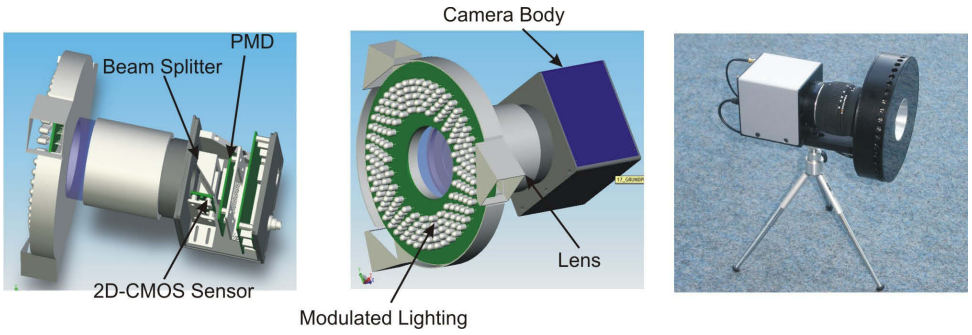

Figure 2.4: The MultiCam 2D/3D vision system developed at ZESS.

lighting system, FPGA based processing unit and USB 2.0 communication interface. As it is shown in Fig. 2.4, the MultiCam has a monocular setup which allows a simple image registration for 2D/3D images. The lighting source has a MOSFET based driver circuit which can drive the high speed near-infrared emitting diodes at different frequencies. The typical frequency used in our work is 20 *MHz* which leads to an unambiguous range measurement of 7.5 *m*. A single lens is used to gather the light for both sensors. While the 3D sensor needs to acquire the modulated near-infrared light (in our case 870 *nm*) back from the scene, the 2D sensor is used to capture the images in the visible spectrum (approximately 400 *nm* to 800 *nm*). To do this, a dichroic beam splitter[14] behind the lens has been used which divides the acquired light into two spectral ranges: the visible light which is forwarded to the 2D sensor and the near-infrared spectrum which is directed to the 3D sensor [12].

The MultiCam with two different optical designs are available: F-mount and C-mount which are illustrated in Fig. 2.5. The F-mount optical design has a simple setup due to its large flange focal distance[15] which makes positioning of the chips as well as their adjustment in the setup simple. In this case, a beam splitter which is a commercial cold mirror is fixed at the angle of 45° with the rear surface being anti-reflection-coated for the near-infrared spectrum. In fact, such a coating is the crucial part of optical design.

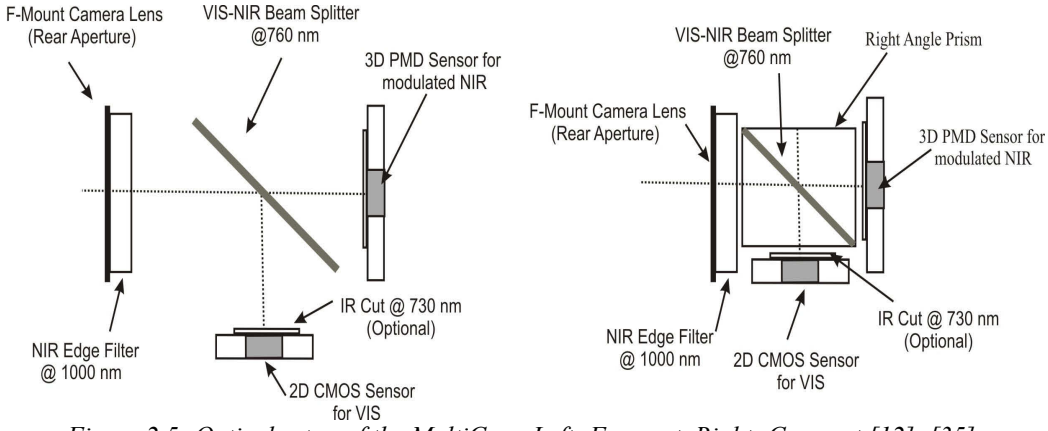

Figure 2.5: Optical setup of the MultiCam. Left: F-mount, Right: C-mount [12], [35].

However, F-mount is not suitable for 1/2" chip formats like PMD sensor because the large focal distance of F-mount with a small chip size of 1/2" yield a narrow angle of view[16] which is not suitable for some applications.

On the other hand, C-mount lenses are good options for the chips with 1/2" format. However, in a C-

14 Dichroic beam splitters are used to combine or separate beams of two different wavelengths.
15 The flange focal distance is 46.500 *mm* for a F-mount, whereas it is 17.526 *mm* for a C-mount lens.
16 The angle of view is calculated as $\alpha = 2\arctan(d/(2f))$ where d represents the size of chip and f is the focal length of lens.

mount design the flange focal distance is shorter than in a F-mount which consequently makes the mechanical design and adjustment of the sensors in the setup more complicated.

One solution to this problem, which is used in the design of the C-mount MultiCam, is to use a prism beam splitter, as illustrated in Fig. 2.5. In fact, in this case the beam splitter is placed between two prisms made out of glass. As the glass has a higher refractive index, the optical path length gets bigger which consequently increases the focal length. In other words, by using the prisms made of glass one can lengthen the distance between the lens and the sensors which makes the arranging as well as adjusting the chips in the optical setup easier.

2.3.1 Time of Flight Range Analysis

In general, the range data provided by the TOF sensors are noisy. In order to enhance the range data obtained by the MultiCam, the main noise sources in the range images are reviewed briefly and some solutions will be presented in the following:

- **Random Noise:** The range data of a TOF chip (in our case PMD) have some random noise with a changing pattern. In fact, the random noise appearing in the range data of PMD is a part of systematic error of the chip. In order to filter this noise and smooth the range data, a median filter is applied. Fig. 2.6 outlines the range measurement of a cut through the range image in the distance of about $2m$ before and after filtering. As it is seen, the range data gets smoother by applying the median filter [41].

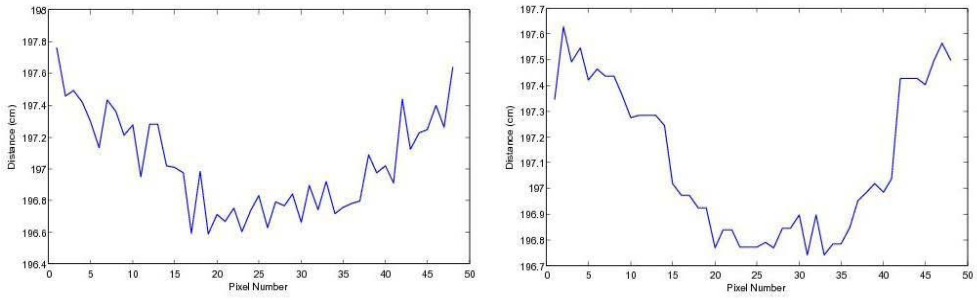

Figure 2.6: Left: Range measurement before filtering. Right: Range measurement after filtering.

- **Fixed Pattern Noise:** The second type of noises in the range data is fixed pattern noise which generally appears with a unique unchanging distribution. In [12] this kind of noise in the MultiCam range data has been determined by using a homogeneous illumination and a gray chart as a uniform target. By acquiring a fixed pattern noise matrix, the offset between the real distance and the measured distance can be minimized. This type of noise has also been addressed in analysis of range images of a SwissRanger camera in [62].
- **Motion Artifacts:** The motion artifact in a 3D range image is somehow equivalent to the motion blur in a normal 2D picture. When the object moves fast, this problem will occur, especially on the edges of the object. In fact, as it was already mentioned in section 2.1.4, the distance data in our TOF camera is calculated based on four phase images [52], [61]. Since in a dynamic scenario an object might move between the phase images, each phase image receives infrared light from a different distance. Therefore the range image which is calculated from these phase images gets the motion artifact. One general solution to this problem is to increase the frame rate of the camera which will be the case in the new generation of the MultiCam by changing the communication interface from USB 2.0 to Gigabit Ethernet. However, one can correct such noises to a great extent by applying morphological operations consisting of erosion and dilation. Fig. 2.7 shows a top distance view of a cubic box taken

while it moves at the velocity of 20 *cm/s* in horizontal direction. The motion artifacts are observed on the edges. To eliminate the motion artifacts in the range image, it is first binarized and then an eroded image is derived from it. Based on the binarized eroded image, the range image is eroded as well. Then the dilated image is obtained from the eroded image by applying a gray level dilation operation. Finally the output image is constructed from that which clears the motion artifacts as illustrated in Fig. 2.7.

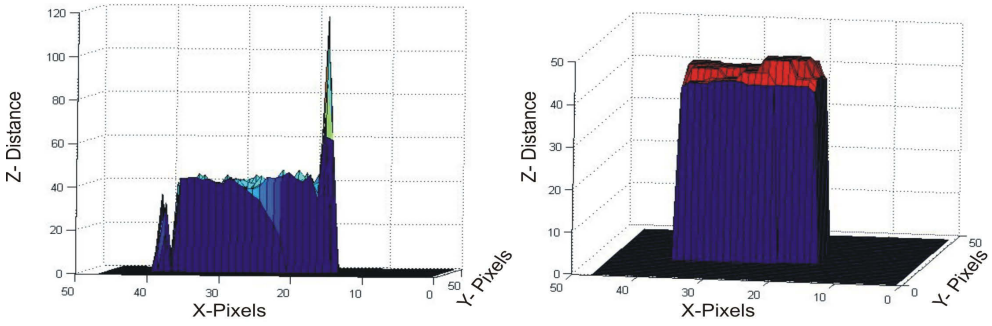

Figure 2.7: Left: 3D image from a box with motion artifact on the edges . Right: Motion artifacts are eliminated by morphological operations.

- **Out-of-Range Noise:** If the infrared lighting system does not illuminate the whole Field of View (FOV) observed by the PMD chip, the non-illuminated regions get wrong distance information which is called out-of-range noise in our work. This is shown in Fig. 2.8 where the corners of a plain background are not properly illuminated and therefore the pixels corresponding to these areas get out-of-range data. These pixels can be filtered out by checking their modulation amplitude in order not to affect the accuracy of the image processing. Likewise, employing a lighting system with an illuminating angle bigger or equal to the angle of view of the camera can solve this problem in a simple way.

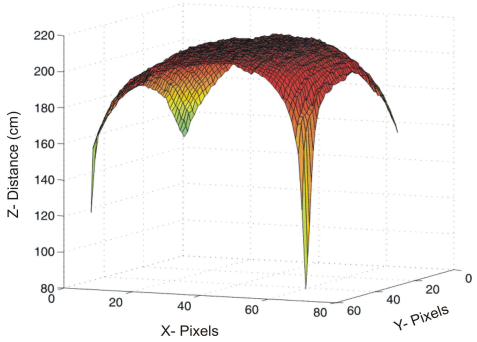

Figure 2.8: Out-of-range errors appear in the corners of a plain surface which are out of the illumination zone of the lighting system.

It should also be mentioned that there is another type of error in the range data of the PMD, even in the static case. This kind of range error which usually appears on the edges of an object is due to the fact that the resolution of the PMD sensor is low and therefore range discontinuities are observed in one pixel. In other words, the pixels corresponding to the contour of an object acquire the reflected near-infrared light from two different distance points, once from the edge point and once from the background, and therefore they get wrong range data.

The dependency of the range data on the exposure time as well as the temperature are also important and interesting issues in analysis of the MultiCam. These issues have been studied in detail in [12] and therefore we will skip them in this work and refer the reader to this reference.

2.3.2 Range Calibration

In order to investigate the range measurement accuracy of the PMD sensor in the MultiCam and calibrate it, a very precise positioning unit [101] with the resolution of $\Delta x = 10\,\mu m$ is used. In fact, we employ a setup in which the MultiCam is mounted on the positioning unit and it is moved automatically to acquire distance data from a fixed target in the range of $60\,cm$ to $450\,cm$ with a fixed distance of $5\,cm$ between each measurement step.

From the acquired distance data, the center pixel is taken as a representative point in order to build the initial data set with $X = \{(p_1, x_1), (p_2, x_2), \ldots, (p_N, x_N)\}$ with p_i as the i^{th} position and x_i as its corresponding range measurement. The deviation from an ideal range measurement device with $x_i = p_i$ is retrieved by simply subtracting X from the ideal set $I = \{(p_1, p_1), (p_2, p_2), \ldots, (p_N, p_N)\}$ delivering the difference $D = \{(p_i, p_i - x_i)\},\ i \in \{1, \ldots, N\}$.

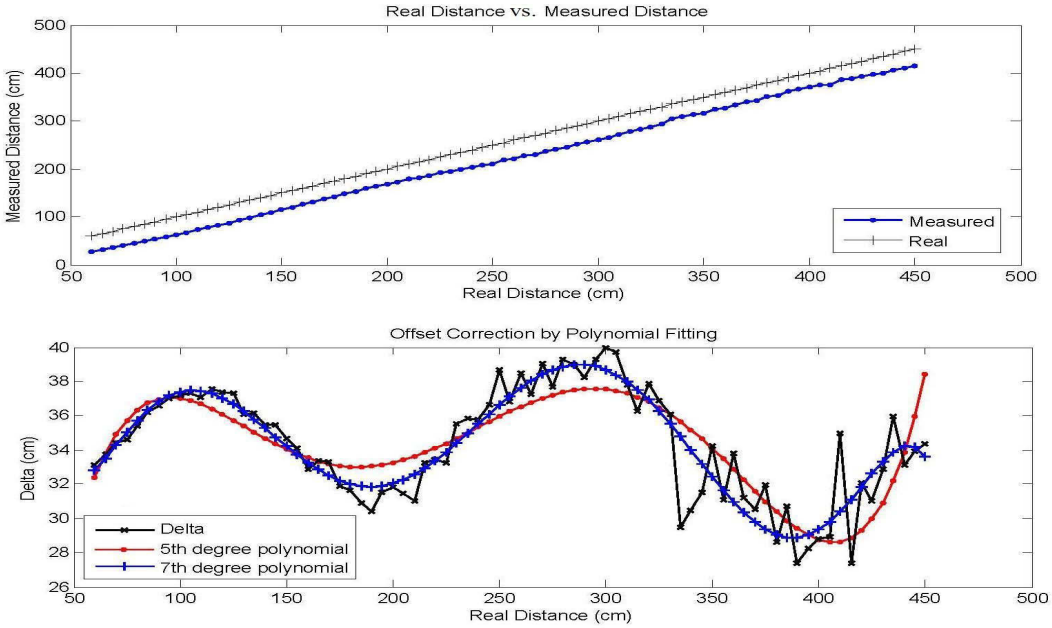

Figure 2.9: Range calibration. Top: Real distance versus measured distance. Bottom: Offset fitting and correction using polynomials.

The upper graph in Fig. 2.9 shows the plot of the ideal range measurement along with the plot of the acquired range measurement for the exposure time of $5\,ms$. In order to obtain the offset b between these two plots the set D is linearly fitted by using the data from $60\,cm$ up to $450\,cm$. Subtracting this from each element of D yields the first approximation: $D' = (p_i, p_i - x_i - b),\ i \in \{1, \ldots, N\}$. By fitting D' with an n-order polynomial, we get the set of coefficients $K = \{k_0, \ldots, k_n\}$ which are used for the final approximation process. Hence, for each measured pixel we compute [41]

$$x_{calibrated} = \sum_{i=0}^{n} (x_{measured} - b)^i \cdot k_i \,. \tag{2.9}$$

The lower graph in Fig. 2.9 illustrates the deviation plot along with 5th and 7th degree polynomial fits which are used in this work.

It should be mentioned that the offset b along with the polynomial coefficients K have been computed for each integration time from $1\,ms$ to $20\,ms$ (step size $\Delta t = 1\,ms$) independently in this work.

For more information regarding the error form in Fig. 2.9, the reader is referred to [12] and [62].

2.3.3 2D/3D Synchronization

Synchronization is one of the main issues in using multiple visual sensors to observe dynamic environments. It is, indeed, one of the main problems in the binocular setups where two different cameras with different frame rates are combined. For example, in [102], where a TOF camera is combined with a standard one, to overcome the problem of synchronization the frame rates of both cameras are set down to 10 frames per second and the spatial offset in 2D and 3D images at different velocities is calculated. In fact, in such setups there is no direct control to trigger both cameras to acquire the images at an exact time; and even if there would be such a common trigger signal, due to different electronic characteristics of each camera, the temporal synchronization cannot be exact. On the other hand, since the MultiCam has a common FPGA based processing unit for both 2D and 3D sensors, it can trigger both sensors to acquire images at the same time. This makes the problem of synchronization in the MultiCam much easier.

As already stated, a TOF range image is constructed from four phase images in the MultiCam. In this regard, three types of synchronization can be considered for the MultiCam which is shown in Fig. 2.10. In the first possibility, the first phase image and 2D image are synchronized such that they are both captured at the same time. In this case, the last three remaining phase images are acquired after finishing the acquisition of a 2D image. In the second possibility, contrary to the first, the last phase image is synchronized with 2D image. In fact, after acquiring the first three phase images, acquisition of both 2D and fourth phase image starts. The last possibility of synchronization which is the most common configuration is to acquire one 2D image per four phase images. In this case the starting time of 2D image acquisition can be set in the interval of range acquisition arbitrary.

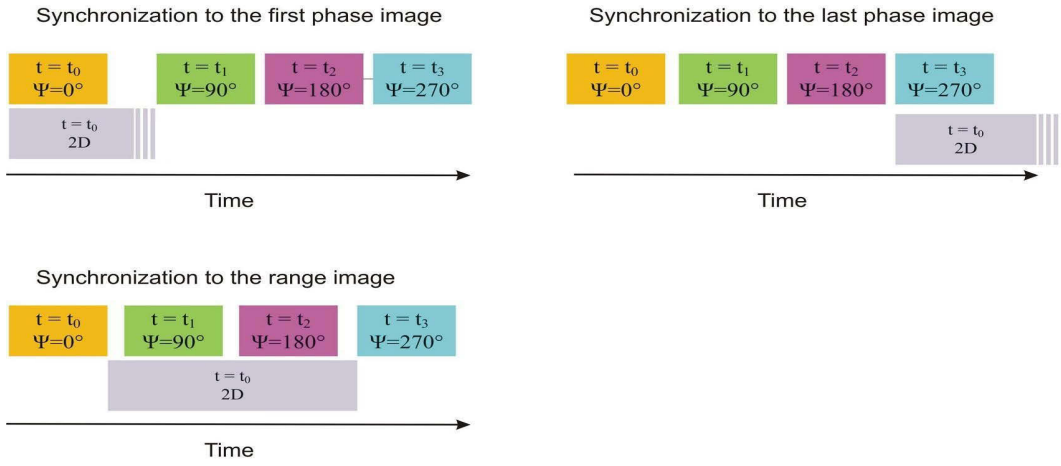

Figure 2.10: Different possibilities of 2D/3D Synchronizing in the MultiCam [12].

As it can be seen from Fig. 2.10, in the first case of synchronization the time between the first and second phase image might not be as equal as the time between the other phase images. In the second case all phase images are captured in the equal time interval and therefore it is preferred. However, in both cases there might be a quick change in the scene which is observed by one of the sensor, whereas it is not acquired by the other one. The solution to this problem is to use the last configuration in which on the one hand all four phase images have the same time interval and on the other hand a 2D image is synchronized with a full range image. In fact, in this case the total acquisition time is kept to a minimum value [12].

2.3.4 2D/3D Image Calibration and Registration

In this section we will review calibration and registration aspects for the MultiCam briefly which is done in [12]. For a much deeper understanding the reader is referred to this reference.

For the MultiCam, the pin-hole camera model like the one used in OpenCV [3] is used. In this model due to the monocular optical setup, both sensors share the same set of extrinsic camera parameters. These parameters describe the transformation function between world and camera coordinate system. On the other hand, an individual set of intrinsic camera parameters, describing the perspective projection of the scene onto the sensor, have to be determined for each sensor. This is because in spite of using a common lens, the beam splitter, the IR cut filter in front of the 2D chip, the sensor window in front of the PMD chip and the fact that the sensors are operated in different spectral ranges potentially influence the projection process. These camera parameters can be acquired by using standard software such as OpenCV, which also takes the radial and tangential distortions up to a certain degree into account [41].

In [12] the results of calibration for the MultiCam have been stated which show that the pin hole camera model can be used for the MultiCam with sufficient accuracy.

After the calibration, we need to identify a spatial transformation function which maps the image coordinates of the 2D sensor to the corresponding coordinates of the 3D sensor as follows [35]

$$[x_1, y_1, z_1]^T = \vec{f}([x_2, y_2, z_2]^T). \tag{2.10}$$

With regard to the MultiCam setup, the same field of view is observed by a conventional 2D sensor with the resolution of 640×480 pixels on the one hand, and by a 3D-TOF sensor (PMD) with the resolution of 64×48 pixels on the other hand [41]. While each pixel of the used 2D sensor has the size of $10\times10\,\mu m^2$, each pixel of the PMD is $100\times100\,\mu m^2$ [31].

To do the registration for the MultiCam, first the uncorrected view after sensor alignment in the setup should be detected. This is due to the angle error which occurs if the beam splitter in the setup is not mounted exactly at the angle of 45° [12]. Given that there are no angle errors in the sensors' alignment, the monocular design of the MultiCam has the advantage that the mapping between the images is constant. This fact is verified by mounting the MultiCam on a precise linear axis and positioning a predefined target on one end of the axis. The camera is then moved and positioned at a number of known distances to the target and both sensors' images are acquired. The target is a test pattern consisting of a grid of circles. It is also assumed that the reflectivity of the test pattern in the visible spectrum is same as its reflectivity in the near infrared spectrum [12]. After rescaling the PMD image to VGA resolution by means of a near neighborhood interpolation, the circles' centers of the target are determined in both images and stored in two sets: P_{PMD} and P_{2D}. The average distances between 2D and PMD circle points in these two sets are calculated in both x and y directions (displacements). Fig. 2.11 shows the average of the displacement between these two sets in units of 2D pixels as a function of the distance between the camera and the pattern for a constant focal length.

It can be observed that the displacement averages are stable over distance, which means that there is virtually no angle error in the sensor alignment in the observed range [41]. Likewise, by examining the distribution of the displacement in the whole image, it turns out that these displacements do not depend on the location in the image. This implies that the mapping neither depends on the resolution of the range data nor on the location of the feature in the image, which constitutes an important difference to the binocular 2D/3D combinations. In other words, in the MultiCam the registration of 2D/3D can be done simply by a two dimensional translation function which maps a 10×10 2D pixel to one single PMD pixel.

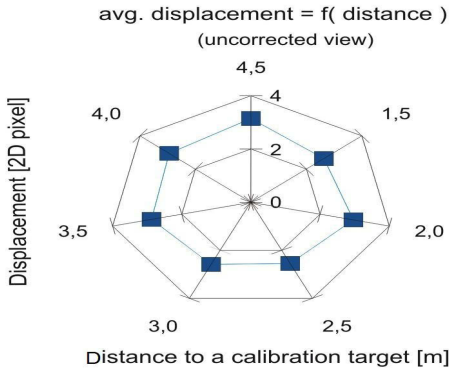

Figure 2.11: Uncorrected dislocation of PMD and 2D sensor in the MultiCam [12].

2.4 Summary

Stereoscopic imaging, Structured Light Approach and Time of Flight are the main range measurement techniques which have been reviewed in this chapter. As discussed, each range measurement technique has its own advantages and disadvantages. Thus, sensor fusion is a general solution to take the advantages of multiple sensors. For example, a 2D/3D vision system which combines the imaging aspects with the depth perception is an interesting sensor combination for computer vision applications.

The MultiCam is a 2D/3D vision system which has recently been developed at ZESS. It is a novel camera system which employs a PMD-TOF sensor with a CMOS chip in a monocular optical setup. While the PMD sensor observes the third dimension of the scene, the CMOS sensor provides high resolution intensity or color information.

Some important aspects of the MultiCam such as range analysis, 2D/3D synchronization, calibration and registration have been investigated in this chapter. As it was seen, the MultiCam, as opposed to the binocular 2D/3D vision systems, does not require any complicated and time consuming calibration and registration techniques. Likewise, the MultiCam can synchronize 2D and 3D sensors via a common FPGA processing unit efficiently. These are, in fact, the vital points in this work because they make the MultiCam suitable for the real time object recognition and tracking problems which will be discussed in the rest of this work.

3

2D/3D Object Recognition

The whole of science is nothing more than a refinement of everyday thinking.
Albert Einstein (1879-1955)

Object recognition is the visual perception of familiar objects despite of the changes in color, texture, shape, size, form, etc. The problem of object recognition can be described as the task of how to detect the desired object in the scene and classify it based on the visual information. In general, an object recognition mechanism consists of a visual sensor which acquires the visual information from the environment where the object may exist; a preprocessing unit which makes the object recognition task easier and faster; and an object classification which distinguishes between different detected objects and put the same objects in a class.

In this context, many investigations have been conducted using different kinds of visual sensors as well as applying different preprocessing techniques and classification algorithms. However, in all of these studies the performance of the real time recognition system depends on the accuracy, recognition time and reliability. In fact, these are the main criteria which can be used to validate the visual sensors as well as the algorithmic approaches in an object recognition mechanism.

As already mentioned in section 1.2, the 2D imaging sensors, like CCD or CMOS sensors, have some difficulties in real world problems where the lighting conditions might change. Likewise, they do not provide range information which is a crucial factor in 3D object recognition. Therefore, they cannot fulfill the accuracy and reliability criteria in real world problems. Generating of 3D data from static 2D images is also a computationally expensive process, i.e., it does not satisfy the recognition time criterion.

Although 3D range sensors can provide depth information, they have still some limitations which were already discussed for each sensor in the previous chapter. On the other hand, employing both 2D and 3D visual sensors in a monocular setup can satisfy the accuracy, recognition time and reliability

requirements for the object recognition to a great extent in many computer vision applications.

We have already discussed the first part of object recognition mechanism in this work by presenting the 2D/3D vision system in the previous chapter. In fact, the presented 2D/3D vision system is used because it can fulfill the aforementioned criteria for an object recognition task in the indoor applications.

Now, in this chapter, we will study and discuss the two remaining parts of an object recognition mechanism, consisting of preprocessing and classification approaches, using Time of Flight range data as well as 2D/3D images.

The original visual data is usually too large and too redundant to be directly used as the input information, esp. for real time applications. The preprocessing, consisting of different techniques, such as feature extraction and segmentation, is usually performed on the one hand to extract the informative part of the data and on the other hand to reduce the dimension of the input data which consequently speeds up the computational processing. With regard to the preprocessing in this work, we will discuss some techniques to extract numeric or symbolic features from 2D/3D images. Furthermore, we will present some aspects of multimodal data fusion and object segmentation and validate them by presenting some results.

After applying preprocessing techniques, which simplify the problem and extract the features, in the next step, a classification approach is applied to distinguish between different detected objects in the scene. The classification in this work will be performed based on two advanced supervised learning techniques which have shown promising results in many machine learning tasks recently.

3.1 Feature Extraction

Each observation taken by a visual sensor consists of the data which might include intensity, color or range information. With the increase in the dimension of the data, the processing time increases significantly and becomes a big issue in real time applications. In order to address this problem, a set of information derived from the original data, so-called feature vectors, are used. Feature vectors should effectively represent the information contents of an observation while reducing the dimensionality. Therefore, a well-defined feature extraction technique makes the recognition process more effective and efficient. In this work, the features have been grouped in two types: Machine generated features and Human generated features. Machine generated features are the similarity and differences in the large data set which cannot be visualized easily by the human being, but they can be highlighted by means of some mathematical approaches such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). On the other hand, human generated features are knowledge based data which can be extracted through some heuristic approaches. The knowledge based information can either be derived from parametric visual data in an image like edge, corner, line, etc. or from statistical properties of the data.

3.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is a way of simplifying the data set by expressing the data in such a way as to highlight their similarities and dissimilarities. It is a linear transformation of the data to a new coordinate system which represents the data set in a better way. This transformation is a rotation of the original axes to the new orientations that are orthogonal to each other. The first axis of the new coordinate system contains the maximum amount of variation which represents the first Principal Component (PC), the second axis with the second greatest variance, orthogonal to the first axis, represents the second PC and so forth until the last axis which has the least variation which represents the last PC. The PCA can be used to reduce the dimensionality by eliminating the later PCs.

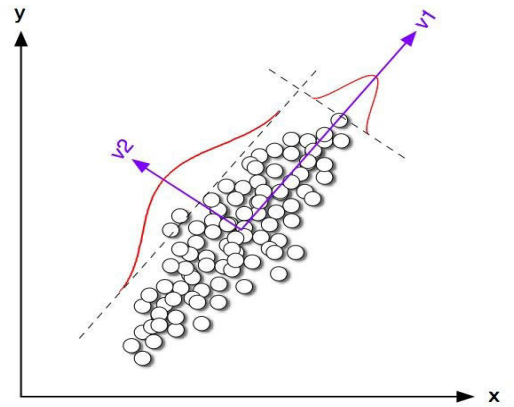

Figure 3.1: Graphical representation of PCA in two dimensions [24].

Fig. 3.1 shows the graphical representation of PCA in two dimensions.

Principal components are found by extracting the eigenvalues and eigenvectors of the covariance matrix of the data which are calculated efficiently via Singular Value Decomposition (SVD). In the following the stepwise approach to perform Principal Component Analysis on n images is discussed:

- **Data preparation:** All pixel values of each image (original features) are arranged in the column vectors of the matrix A with the size of $m \times n$ where m represents the dimension of data (number of original features including range, intensity, modulation amplitude and color data) and n represents the number of images (observations).

$$A = \begin{matrix} \text{Im=1,...} & & & \text{... ,Im=n} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \end{matrix}$$

m features, n observations

- **Calculation of adjusted matrix:** In the next step, the mean vector along the features is calculated, and then it is subtracted from matrix A to derive the adjusted matrix M as follows

$$\varphi_i = \frac{1}{n} \sum_{j=1}^{n} a_{ij} \tag{3.1}$$

$$M = A - \varphi u \tag{3.2}$$

where u is a $1 \times n$ vector of 1's and the dyadic product φu has the size of $m \times n$.

- **Calculation of the covariance of adjusted matrix:** The covariance matrix S is calculated as follows

$$S = \frac{M M^T}{n-1} . \tag{3.3}$$

The covariance matrix S gets the size of $m \times m$, where m is the dimension of the data.

- **Calculation of the eigenvalues and eigenvectors of the covariance matrix:** The covariance matrix S is diagonalized using eigenvalue decomposition to find the eigenvalues and eigenvectors such that

$$S = Q \Lambda Q^T \tag{3.4}$$

where Q is the eigenvector matrix and Λ is the corresponding diagonal matrix of its eigenvalues as follows

$$Q = [eigenvec_1, eigenvec_2, \ldots, eigenvec_m], \tag{3.5}$$

$$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_m). \tag{3.6}$$

- **Rearrangement of eigenvectors based on eigenvalues:** In the next step, the eigenvectors are rearranged in descending order of their corresponding eigenvalues starting with the largest λ. The first p largest eigenvectors[17] are then selected, where $p < m$, to reduce the dimension of the data such that

$$v = (\lambda_1, \lambda_2, \ldots, \lambda_p), \tag{3.7}$$

$$f = [eigenvec_1, eigenvec_2, eigenvec_3, \ldots, eigenvec_p]. \tag{3.8}$$

- **Calculation of new data set:** In the last step, the new data set is derived by projecting the original data to the new coordinate system (with the dimension of p) through the feature vector f as follows

$$final\ data = f^T M. \tag{3.9}$$

Computational Cost of PCA

Computational cost is one of the most important criteria which is usually considered in the selection of an algorithm. The cost can be evaluated based on the time complexity function of the algorithm. For a PCA algorithm, the main part of computation is the eigenvalue and eigenvector calculation. The typical algorithms which are used to find the eigenvectors of a $m \times m$ matrix have a time complexity function of $O(m^3)$. For our applications in this work, each pixel of a 2D/3D image has range, intensity, modulation amplitude or even color information. Therefore, each observation corresponds to a vector in a space of thousands and eventually more dimensions (m) and so the calculation of eigenvectors becomes computationally expensive which consequently makes the application of PCA for real time application computationally infeasible. In addition to this point, in our applications mostly the number of observations n is smaller than the dimension of the data m. For example we have some hundred images as training data set where each image has at least some thousands pixel data. In this case $n < m$ and therefore we will get m-n+1 zero eigenvalues. A zero eigenvalue corresponds to an eigenvector along its directions the data set gets zero variance and therefore such an eigenvector does not give any information about the similarities and differences in the data set. In order not to calculate the zero eigenvalues as well as resolve the problem of computational expense of PCA, a trick is applied which is described in the following [2]:

First, we define a new matrix H as follows

17 The criteria for selection of the first p largest eigenvalues will be discussed later in this section.

$$H = \frac{M^T M}{n-1} . \tag{3.10}$$

H gets the size of $n \times n$ which is much smaller than the covariance matrix S with the size of $m \times m$. In the next step, the eigenvalues and eigenvectors of the lower dimension matrix H are calculated. The n-1 eigenvalues of matrix H are the same as the non-zero eigenvalues of covariance matrix S which has m-n+1 additional zero eigenvalues. Now, the eigenvectors in the original data set can be calculated by

$$v = M\, v' \tag{3.11}$$

where v' represents the eigenvectors of matrix H and M represents the adjusted matrix.

Eigenimages and PCA Reconstruction

As already discussed in this chapter, using the images directly as the input data makes the classification task cumbersome. Having n images of different objects, the task of recognition is to discriminate them in several classes. PCA is used on the one hand to reduce the dimension of the data and on the other hand to derive some characteristic properties of the image data. In fact, using PCA each image in the data set can be represented as a weighted sum of the basis vectors. These vectors are the eigenvectors which are derived in the PCA process, described in the previous section, and called eigenfaces[18]. In principle, PCA searches for directions in the image data with the largest variance and projects the images from image space onto those directions. In this way, a lower dimensional representation of the image data is obtained, which neglects some of the noisy directions. Thus, instead of directly working with high dimensional images, we project them using the eigenimages and their weights in the new coordinate system.

Figure 3.2: Some examples of 2D color images from a person data set taken by MultiCam, converted to gray scale and resized.

In Fig. 3.2 some example of 2D images from a person data set have been shown. In the person data set, we have collected 2D/3D images of three seated people with different poses. These images are taken by the MultiCam. The 2D color images have been converted to gray scale images and then resized to

18 The term eigenface originates from facial recognition community because this technique was developed and used successfully for the first time by Turk et al. [96] for the face recognition problem. However, as this technique has been used later for different object recognition tasks, some researchers prefer to use the term eigenimage instead of eigenface.

32×24 pixels. These images are highly correlated in image space. Projecting them through the eigenimages make them uncorrelated in the new subspace with lower dimension.

The first four eigenimages derived from 2D images are shown in Fig. 3.3. Each eigenimage represents only certain features which might be present in the original image. For example, an image might have 40% of the first eigenimage, 25% of the second eigenimage, 15% of the third eigenimage, 10% of the fourth eigenimage and 10% of the rest of eigenimages. Using the weights (eigenvalues in PCA) and eigenimages (eigenvectors in PCA), an image is projected to the new coordinate system with lower dimension.

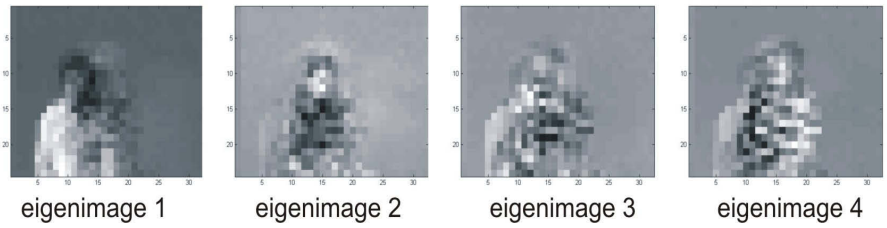

Figure 3.3: The first four eigenimages of 2D images from person data set.

Fig. 3.4 shows some example of range images from the person data set. In these images, the range information has been coded in the gray values such that the darker pixels represent the smaller distance to the camera.

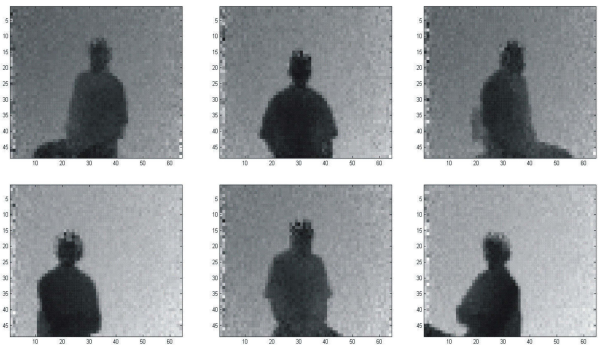

Figure 3.4: Some examples of range images from the person data set taken by MultiCam. The range data have been coded in gray values such that the darker the pixel, the smaller the distance to the camera.

The first four eigenimages derived from range images are illustrated in Fig. 3.5. One can observe that the eigenimages cover the shape of the persons with different poses.

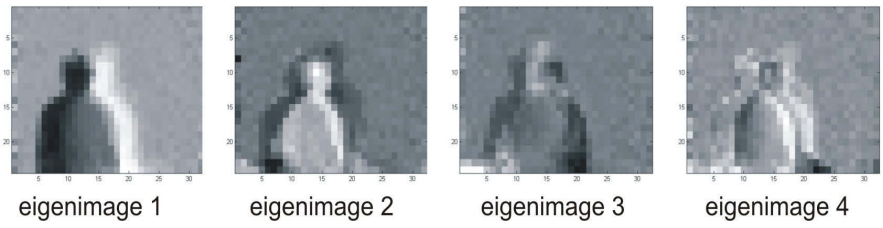

Figure 3.5: The first four eigenimages of range images from person data set.

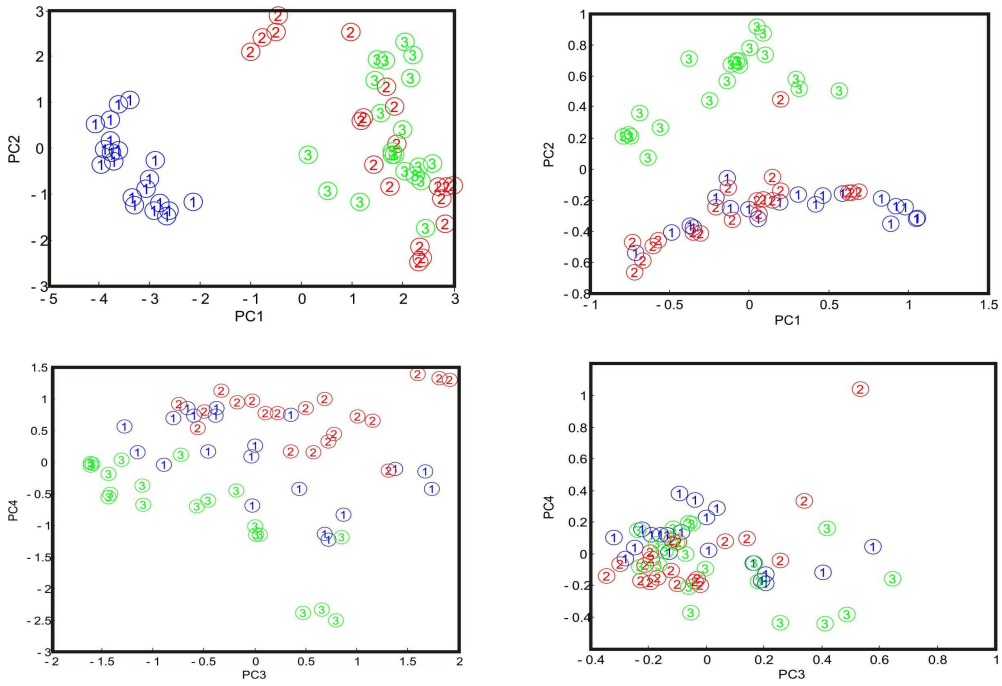

Figure 3.6: Distribution of the projected data on the first four principal components. Left: Derived from the 2D images. Right: Derived from the range images.

In Fig. 3.6, the projection of 60 images (2D and range) of three people (20 images per person) from the person data set on the first four principal components is shown. One can observe that the first two principal components are more informative than the other last two components and therefore the projected data are better separated concerning the first two principal components than the last two components. The distribution of the projected data on the first two PCs shows that two classes are well separated in 2D image space whereas the same two classes are mixed in 3D range space and vice versa. Thus, fusing the features derived from 2D images with those extracted from 3D range images yields a feature space where all three classes are well separated. Another approach would be to make a classifier for each feature set (derived from 2D and range images) and combine them to make a strong classifier. The number of chosen principal components determines the new dimension of the data. One of the key questions is how many eigenimages should be selected. In order to answer this question, one can look at the spectrum of the eigenvalues. The eigenvalues with low slope can be neglected because their corresponding eigenimages do not give any useful information about the data.

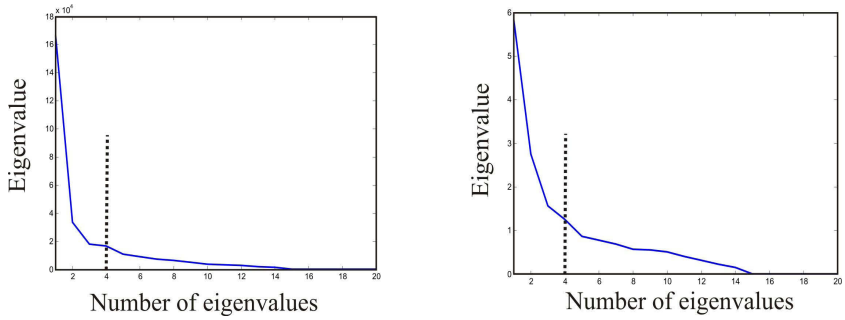

Figure 3.7: Eigenvalue spectrum, Left: Derived from range images. Right: Derived from 2D images. The first four eigenvalues have the highest slope.

In Fig. 3.7, the spectrum of the first 20 eigenvalues, sorted in descending order, for 2D and range

images of person data set are plotted. As we can see, for this case the first four eigenvalues have the highest slope and therefore they have the maximum impact in giving the information about the data.

After projecting the images to the new subspace, they can be reconstructed in the cases where needed. The reconstruction can be done by retaining p principal components. As p increases the reconstructed image becomes more accurate and would be as original when the number of principal components is equal to the resolution of the original image. An example of image reconstruction using the first p components can be seen in Fig. 3.8.

Figure 3.8: An example of image reconstruction. First row: The original 2D image together with its PCA reconstruction. Second row: The original range image together with its PCA reconstruction.

3.1.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is another technique in machine learning for feature extraction and dimension reduction which is usually applied to the data before classification. LDA, unlike Principal Component Analysis is a supervised technique. In other words, LDA includes the label information of the data (to which class the data belongs) in the projection process.

A classical LDA as the normal linear function projects the m dimensional data x onto a lower p dimensional y space as follows

$$y = w^T x \tag{3.12}$$

where w is the projection matrix with the size of $m \times p$ and y is the projected data in the new dimension. Fig. 3.9 shows the projection of two dimensional data features of three classes to one dimensional subspace using LDA technique.

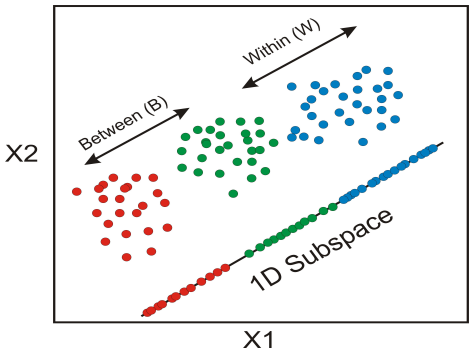

Figure 3.9: Projection of the 3-class feature data to a 1D subspace using LDA. X_1 and X_2 represent two arbitrary features.

Now, the question is how we can use the label information to find the best projection of the data. Fisher [2] proposed the answer by finding the direction along which the classes are best separated. This is done by maximizing a function *J* that will give a large separation between the projected class means while giving a small variance within each class as follows

$$J(w)=\frac{w^T S_b w}{w^T S_w w} \tag{3.13}$$

where S_b is the class separation which is called between classes scatter matrix and is given by

$$S_b=\sum_c N_c(\mu_c-\bar{x})(\mu_c-\bar{x})^T \tag{3.14}$$

where $\mu_c=1/N_c \sum_{x_i \in C} x_i$ is the mean of the data in the class *c* and $\bar{x}=1/N \sum_c N_c \mu_c$ is the mean of the whole data in which N_c is the number of data in class *c* and $N=\sum_c N_c$ is the total number of data points.

In equation 3.13, S_w is the within classes scatter matrix and can be formulated as follows

$$S_w=\sum_c \sum_{i \in c} (x_i-\mu_c)(x_i-\mu_c)^T . \tag{3.15}$$

Maximization of *J* function can be done by differentiating 3.13 with respect to *w* as follows

$$\begin{aligned} \frac{d}{dw}[J(w)] &= \frac{d}{dw}\left[\frac{w^T S_b w}{w^T S_w w}\right] \\ &= \left[w^T S_w w\right]\frac{d\left[w^T S_b w\right]}{dw}-\left[w^T S_b w\right]\frac{d\left[w^T S_w w\right]}{dw} \\ &= \left[w^T S_w w\right]2S_b w-\left[w^T S_b w\right]2S_w w \quad = \quad 0 \quad . \end{aligned} \tag{3.16}$$

By dividing the both sides of 3.16 by $w^T S_w w$ and multiplying it by S_w^{-1} we will obtain

$$(S_w^{-1} S_b) w = J w \tag{3.17}$$

which looks like an eigenvalue equation. The first *p* largest eigenvectors of $S_w^{-1} S_b$ corresponding to the *p* largest eigenvalues determine the projection vector *w*.

Since LDA as a supervised algorithm takes the label information into consideration, it usually outperforms PCA when there exist a large sample data set for each class. However, when the data are undersampled, i.e., there is a small sample size, LDA fails because a classical LDA requires that all scatter matrices to be nonsingular [85], [92]. This is not the case in many applications like in our person data set example where the sample size (number of observations) does not exceed the dimension of data. This limitation of LDA is known as the singularity or undersampled problem.

Recently many approaches have been proposed to solve this problem such as PCA+LDA, regularized LDA, Pairwise Discriminant Analysis (PDA), Penalized LDA and LDA/GSVD [88], [51], [85], [92].

The PCA+LDA approach which has been used in face recognition problem proposes to project the data first to an intermediate space using PCA and then apply the classical LDA to project them to the

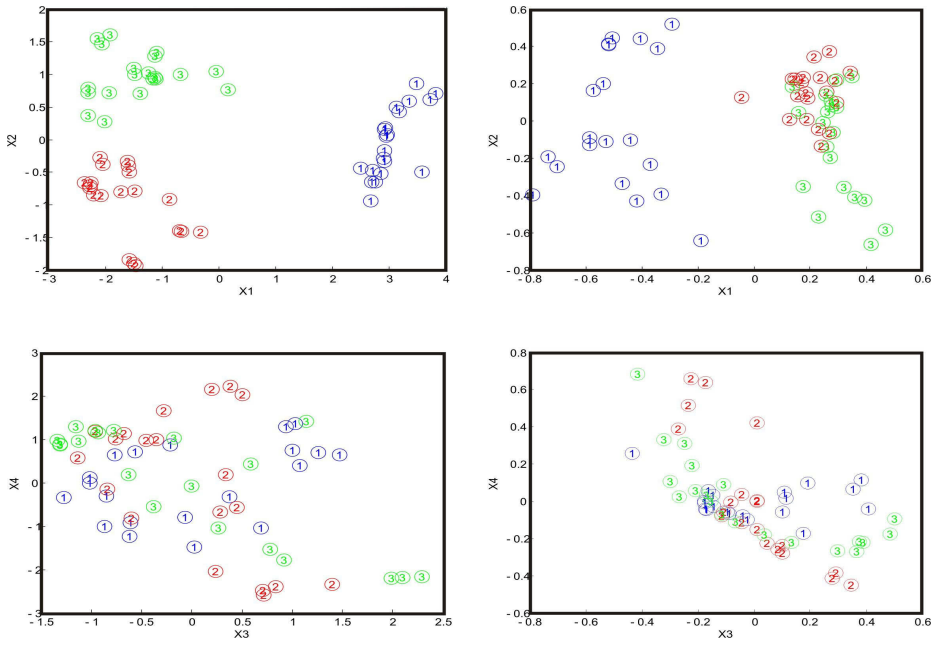

Figure 3.10: Distribution of the projected data using PCA+LDA in the four dimension.

final space.

In Fig. 3.10 we have shown the results of projecting the person data set onto a four dimensional subspace using PCA+LDA technique.

Comparing these data with the results of the PCA in Fig. 3.6 shows that PCA+LDA outperforms PCA in this case. However, the computational cost of PCA+LDA is higher which should be considered in the real time applications.

3.1.3 Knowledge Based Features

Knowledge based features belong to the human generated features. These kinds of features refer to the information which is derived from the 2D/3D images through some heuristics. For this reason, it has also been termed as heuristic features in some literature. In our work, knowledge based features are either derived from parametric visual data which give the information about the shape, size, color and position of the desired object or they are calculated from statistical properties which give the information about the distribution of the data. In fact, knowledge based features are application dependent which give abstract information about the object of interest in the image. The abstract information can be used as the lower dimension features for detection, classification and tracking algorithms. In the following, two main types of these features are described:

- **Parametric Visual Data:** They can be derived directly from the 2D/3D images based on some image processing techniques. Some of these features are listed as follows:
  i. Number of edges in the Region of Interest (ROI)
  ii. Detected corners in the ROI
  iii. Lines in the ROI
  iv. The circularity ratio of the detected object in the ROI
  v. Color
  vi. Size of the object derived from 3D image

vii. Distance of the object to the camera

- **Statistical Properties:** The statistical properties can be defined by the user based on the knowledge about the desired object in the application. For example, the number of non-zero pixels after the segmentation, the minimum of the range data, standard deviation of the data in ROI or histogram properties can be used as good features for some object classification problems.

However, the knowledge based features cannot be generalized for all types of problems. While they can be used as good features for a specific problem (for example face detection), in another type of problem (for instance torso detection) they cannot yield good results. On the other hand, integrating the knowledge based data with machine generated features, which is so-called hybrid features, will provide more accurate information for classification and tracking problems. We will show some results of hybrid features used in object classification problem in section 3.3.2.

An Example of Knowledge Based Features using 2D/3D Images

In this section, it is shown how a heuristic technique can be applied to 2D/3D images to extract knowledge based features which are used to classify two poses of the hand (palm and fist). We assume that the hand is detected using a kind of machine learning technique (in our case we have used AdaBoost which will be described in section 3.3.4). Now, in the next step, the pose of the hand should be classified. In fact, we consider a binary classification problem to distinguish palm from fist. As we have some prior knowledge about the shape of the object of interest, a heuristic technique is implemented to extract this knowledge and convert it to a function feature which is used to classify the hand. The heuristic technique which is similar to [60], is summarized as follows:

Heuristic Algorithm:

Assumption: Given the Region of Interest *D(x,y)* in the 2D image *I(x,y)*, in where the hand is detected.

1) Extract the all *n* pixels with the skin color in *D* and save them in new data set *E(x,y).*

2) Find the mass center *O* of the hand in *E(x,y)* as follows

$$x_O=\frac{\sum X_i}{n}\ ,\ y_O=\frac{\sum Y_i}{n}\ .$$

where X_i and Y_i represent the position of the pixels with skin color in *E* derived in step 1.

3) Extract the distance value *d* of the center point *O* from 3D range image.

4) Draw a circle at the center point of *O* with the radius of *r* as

$$\mathbb{C}:(x-x_O)^2+(y-y_O)^2=r^2$$

where $r=k\cdot d$, and $k=cte$, i.e., the size of circle is adjusted using the range data from 3D image. In other words, the closer the hand to the camera is, the bigger the circle is.

5) Trace the circle and construct a binary intersection function of the circle with the hand such that

$$f(x,y)=\begin{cases}1 & if\ (x,y)\in\mathbb{C}\\ 0 & if\ (x,y)\notin\mathbb{C}\end{cases}$$

6) Count the number of transitions from 1 to 0 and save it as a feature.

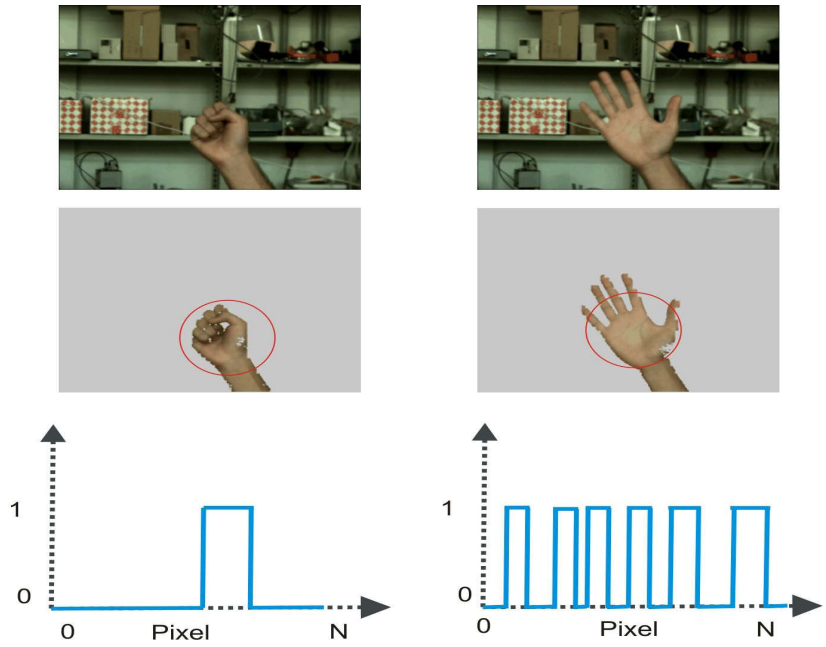

Figure 3.11: Knowledge based feature extraction using heuristics for hand pose classification. First row: 2D images. Second row: Hand detected images from 2D/3D data. Third row: The binary intersection function derived from tracing the circle.

Fig. 3.11 shows an example of the applied heuristic technique to the hand images. While the feature number for palm posture is usually 6 (five fingers plus wrist), it is always 1 for fist posture because in this case the hand lies inside the circle and only the wrist intersects with the circle.

3.1.4 Haar-like Features

Haar-like features have been used successfully in object recognition problems such as face detection and hand recognition. Haar-like features, which originate from Haar-wavelets, encode the knowledge about the desired object which is difficult to be extracted from the raw pixel data. The value of a Haar-like feature in an image is calculated by subtracting the sum of pixel values in the black subregion from the same in the white subregion as follows

$$f(x) = \sum_{black} (pixel\ value) - \sum_{white} (pixel\ value) \quad . \tag{3.18}$$

The standard Haar-like features are illustrated in Fig. 3.12.

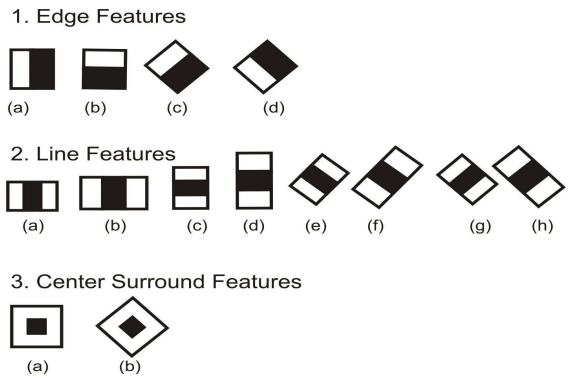

Figure 3.12: Standard Haar-like Features [80].

To detect the object, the image is scanned by a search window containing a Haar-like feature. The presence of a Haar-like feature is determined by comparing feature *f(x)* with a threshold θ which is found in the training phase. If *f(x)* is above the threshold, that feature is said to be present and accordingly a binary classifier *h(x)* for the Haar-like feature *t* is generated as follows

$$h_t(x)=\begin{cases}1 & if\ f(x)>\theta \\ 0 & if\ f(x)<\theta\end{cases}. \tag{3.19}$$

Determination of the presence or absence of all Haar-like features at every location of the image with different scales is computationally too expensive. Viola and Jones [80] proposed a solution which is extremely fast and can be used for real time applications. This method will be reviewed in section 3.3.4.

3.2 Multimodal Image Segmentation

Segmentation is the first challenge in vision based object recognition problems. It is the way of partitioning the image into a set of meaningful regions. This simplifies the processing task of object recognition because instead of dealing with a large number of pixel data, the complex scene is segmented to the separated areas which provide a simpler description of the desired object in the image. Therefore, segmentation is usually considered as a preprocessing step in many computer vision applications. Since the results of the segmentation directly affect the performance of the subsequent processing techniques like feature extraction and classification, it is one of the most important steps in computer vision problems.

There are many approaches which have been used for object segmentation using 2D images (color and intensity) as well as 3D range data [22], [14], [19], [87]. In general, these methods can be categorized as pixel based, edge based, region based and hybrid segmentation approaches:

- **Pixel Based Segmentation:** They are local segmentation methods which group the similar pixels in the image in one segment with respect to some characteristics such as intensity, color, texture or range data. Some of the approaches used in this category are as thresholding, clustering, histogramming and fuzzy clustering.
- **Edge Based Segmentation:** They consist of local and global segmentation methods. The difference between local and global methods is the way they define an edge point which is characterized by a vector with size, position and direction. In a local technique, the edge pixel is determined with the information in the neighborhood of that pixel, whereas in a global technique it is identified after many optimizations in a large area. Canny edge detector, differential edge detection and Markov random field are some of such segmentation techniques in this group.
- **Region Based Segmentation:** They are global segmentation methods which use uniformity criteria calculated in the regions of image domain. These techniques are divided into two groups: region growing and split and merge. While the region growing starts with some uniform regions (seeds) and applies some strategies to add the surrounding neighbors to grow the region, split and merge method starts from nonuniform regions and split them to obtain the uniform ones, and then applies some merging techniques to get the maximum uniform region.
- **Hybrid Segmentation:** It is a method which combines the previous mentioned techniques to provide more accurate segmentation results.

There are many publications in the computer vision literature which have studied the image segmentation approaches ranging from simple ad hoc schemes to more sophisticated ones using object

models [22], [56]. There are also some good survey articles which have compared these techniques [22], [100], [95]. However, there is no universally applicable segmentation technique which can be used for all of the applications, and therefore the appropriate segmentation technique should be selected and validated based on the particular application criteria. For example, there might be some segmentation techniques which are highly accurate for a specific object recognition problem, but they are computationally expensive which makes them inefficient for real time tasks. Thus, there should be a trade-off between the required criteria to find the optimum technique.

In the image segmentation area, due to the lack of benchmarks the evaluation of segmentation performance is very difficult. In fact, in a general purpose problem, "correct" segmentation is not well defined and there is no unique ground truth which can be compared against the result of segmentation. Likewise, the evaluation results of different segmentation techniques in one application cannot be applied to other applications. However, there are still some good benchmarks which produce a score for the algorithms and therefore they can be compared with each other. Currently the most important benchmarking used for evaluation of 2D gray scale and color image segmentation is "The Berkeley Segmentation Dataset and Benchmark" [27] with 12000 hand-labeled segmentations of 1000 Corel dataset images from 30 human subjects. Also, Hoover et al. [95] has created a publicly available tool to measure the results of 3D range image segmentation.

Studying the current segmentation techniques as well as creating a benchmark to compare the results is out of scope of this work. In this section the object segmentation using multimodal data including range, modulation amplitude and intensity is studied. These data are provided either by a 3D Time of Flight camera (low resolution range and intensity data) or by the novel monocular 2D/3D imaging system (low resolution range and high resolution intensity or color data). In fact, the segmentation in this work, as a preprocessing step, aims first at distinguishing foreground objects from background and then classify them using multimodal data in order to make the object recognition problem easier and faster.

3.2.1 Range Segmentation

A range image which reveals direct distance information about the object's surface can explicitly represent the surface geometry of the scene. In many applications like autonomous navigation where the objects should be identified based on their distance to the sensor, range segmentation plays a key role. As the range image acquisition techniques are different, there are different types of range images. For this reason, unlike intensity image segmentation, there is no generic algorithm for range image segmentation.

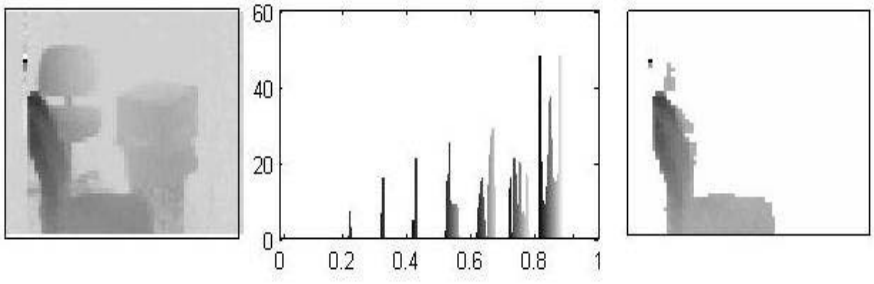

Figure 3.13: Range Segmentation. Left: TOF range image subtracted from the background. Middle: Normalized histogram of range image. Right: Range segmented object.

A simple example of TOF range image segmentation is shown in Fig. 3.13. The range data has been coded in the intensity values such that the darker a pixel is, the closer it is to the sensor. Three objects can be seen in this image. We assume that the closest chair in the foreground is the object of interest

which should be segmented from the background as well as other objects in the foreground. The thresholding technique is used for this purpose. The concept of this technique is based on subtracting the range image I_r from the background image I_b. The background image is already averaged (median) from many range images from the scene in the absence of the objects to reduce the statistical noise of TOF sensor to the minimum value. The background can also be modeled from range images like what is usually done in 2D background subtraction techniques. Since the variation of range data is not so high by changing the lighting conditions in the environment, background averaging is sufficient and therefore background modeling is not necessary.

Each pixel element in the subtraction matrix is compared with the threshold T and outputs the label background or object by

$$\begin{cases} I_s(m,n) = I_r(m,n) & if \quad |I_r(m,n) - I_b(m,n)| \geq T \\ I_s(m,n) = 0 & else \end{cases} \tag{3.20}$$

where $I_s(m,n)$ corresponds to the value of segmented image at row m and column n. The threshold level T can be selected by examining the normalized histogram of the range image. In the example shown in Fig. 3.13, the threshold of 0.58, corresponds to the distance of 2 m from the sensor, is selected to segment the foreground chair from the rest of objects in the scene. Likewise, the multiple thresholding technique can be used to segment a mid-ground object in the range image data.

Since TOF range images contain explicit 3D information about the objects, the range segmentation is comparably easy and computationally inexpensive. However, TOF range images have problems with transparent or reflecting materials like glasses or metal objects. Also, the range data in a TOF image is affected by the color of the objects, i.e. two surfaces with different colors at the same distance might get different range data in a TOF image (see Fig. 3.14). Therefore, in a real world problem with a complex scene, the problem of range segmentation cannot be solved so easily just by using range data. This problem can be solved to some extent by fusing the range data with modulation amplitude and intensity data as we will describe in the next section.

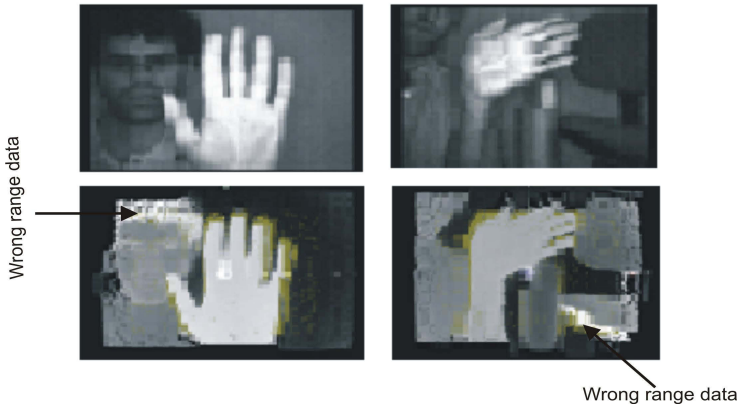

Figure 3.14: Two examples of TOF range mis-measurements. First row: TOF intensity images (64×48 pixels). Second row: Corresponding range images coded in gray values. Left: The hair gets wrong range data because of its black color. Right: The metal part of the chair has wrong distance data.

3.2.2 Multimodal Data Fusion

The TOF camera delivers three data items for each pixel at each time step consist of intensity, range and modulation amplitude which is the amplitude of the received modulated light back from the scene. Therefore, a modulation image is like a quality index image for the range data. The intensity image of

the TOF camera, comparable to the intensity images in CCD or CMOS cameras, relies on the environment lighting conditions, whereas the range image and the amplitude of the received modulated light are mutually dependent. None of these individual data can be used solely to make a robust segmentation under challenging conditions in a real world problem. Fusing these data provides more reliable information which is used to improve the performance of the segmentation technique.

In this work we have used the basic technique for the fusing of the range and intensity data which has already been used in other fields like SAR imaging. We observed that the range data in our TOF sensor is dependent on the reflection factor of the object surface. Therefore, there is a correlation between the intensity and range vector sets in a TOF image. These two vector sets are fused to derive a new data set, so-called "phase", which indicates the angle between two intensity and range vector sets and is derived as follows:

First, using the intensity and range data in each image a new resulting set of complex number C is derived such that

$$C_{rc} = g_{rc} + jd_{rc} \quad , \quad j = \sqrt{-1} \tag{3.21}$$

where g_{rc} corresponds to the normalized gray value and d_{rc} represents the normalized range information for each pixel in row r and column c of the intensity and range images respectively.

Next, the phase of each complex number φ is calculated in the polar coordinate system for the whole array of the pixels as follows

$$\varphi_{rc} = \arctan\left(\frac{d_{rc}}{g_{rc}}\right) . \tag{3.22}$$

The flow of the range and intensity fusion is depicted in Fig. 3.15.

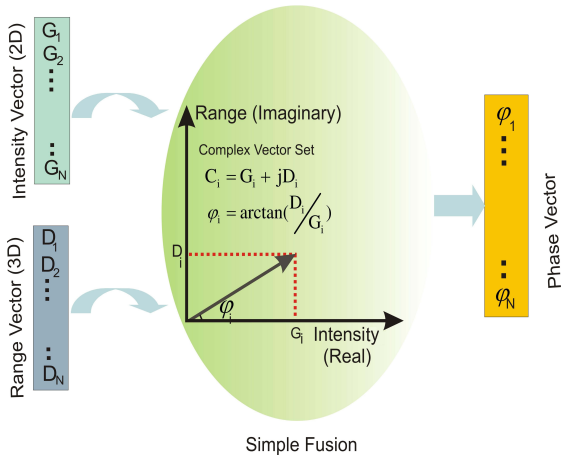

Figure 3.15: Flow of the range and intensity fusion.

The phase of the complex value and range data are then combined into a 2D feature space where each pixel is described by a feature vector f_{rc}, containing range and phase information as follows

$$f_{rc} = (d_{rc}, \varphi_{rc}) . \tag{3.23}$$

Here, range denotes the position of the object in z direction in the world coordinate system which is

aligned to the optical axis.

Another type of fusion which has also been used in our work is to weight the value of the range for each pixel using the modulation amplitude factor and then calculate the phase vector. In this case, the phase vector is calculated from the following complex set

$$C_{rc} = g_{rc} + j(d_{rc} \cdot m_{rc}) \quad , \quad j = \sqrt{-1} \tag{3.24}$$

where m_{rc} represents the normalized modulation amplitude for the pixel in row r and column c. This adjusts the range level in those regions where the range data might get wrong.

3.2.3 Unsupervised Clustering

Unsupervised clustering techniques aim at partitioning a data set into K clusters. They are popular methods in image segmentation area. K-means and Expectation Maximization (EM) are two widely used approaches which we have employed in our work because on the one hand they are easy to implement and on the other hand they are fast enough to be used for real time object recognition tasks.

K-Means

K-means is one of the simplest unsupervised learning algorithms that solves the well known clustering problem by partitioning the data set $\{x_1, x_2, ..., x_n\}$ into some number K of clusters. This is done by minimizing the objective function, given by [2]

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2 \tag{3.25}$$

where r_{nk} is a binary membership function which is defined for each data point x_n as follows

$$r_{nk} = \begin{cases} 1 & if\ x_n\ \ assigned\ to\ cluster\ k \\ 0 & otherwise\ \ . \end{cases} \tag{3.26}$$

In equation 3.25, $\|x_n - \mu_k\|^2$ represents the square of the distance between the data point x_n and the cluster center μ_k. In fact, the goal is to find the values for the $\{r_{nk}\}$ and the $\{\mu_k\}$ so as to minimize J. This is done through an iterative procedure in which each iteration involves two successive steps corresponding to successive optimization with respect to the r_{nk} and the μ_k [2].

The main advantages of this algorithm are its simplicity and speed. The computational cost of K-means is $O(KN)$, which allows it to run on large data sets. However, K-means is a data dependent algorithm. Although it can be proved that the procedure will always terminate, the algorithm does not achieve a global minimum. Since K-means is a distance based or hard membership algorithm, every data point, at each iteration, is assigned uniquely to one, and only one, of the clusters. For the data points which lie roughly midway between cluster centers, the hard assignment to the nearest cluster might not be the most appropriate one. By adopting the probabilistic approaches, like Expectation Maximization (EM), a soft assignments of data points can be obtained.

Expectation Maximization

Expectation Maximization (EM) is a powerful method to find the maximum likelihood solution for models with incomplete or missing data (see Appendix A - Expectation Maximization). This approach

can be used for image segmentation where each segment (cluster) is mathematically represented by a parametric Gaussian distribution. The entire data set (image) is therefore modeled by a mixture of Gaussian distributions. The missing data in Gaussian mixture model, which is also known as hidden or latent variable, is the Gaussian cluster from which the observation data originate.

As mentioned, we assume that the entire image data $X=\{x_1,x_2,...,x_n\}$ is modeled using a mixture of K Gaussian distributions as follows[19]

$$f_{X/\Theta}(\xi/P)=\sum_{k=1}^{K}\alpha_k f_{X/\theta_k}(\xi/\rho_k),\quad \forall X\in\mathbb{R}^n \tag{3.27}$$

where ξ and $P=\{\rho_1,...,\rho_K\}$ are dummy vectors such that $\xi, P\in\mathbb{R}^n$. α represents the mixing weights for the Gaussian distributions where $\sum_{k=1}^{K}\alpha_k=1$.

Likewise, $\Theta=(\alpha_1,...,\alpha_K,\theta_1,...,\theta_K)$ describes the collection of all parameters in the mixture model.

In fact, in the mixture model, each component is represented by a Gaussian with the parameter θ_k consisting of mean and covariance $\theta_k=(\mu_k,\Sigma_k)$ as follows

$$f_{X/\mu_k,\Sigma_k}(\xi/\mu_o,\Sigma_o)=\frac{1}{(2\pi)^{d/2}|\Sigma_o|^{1/2}}\exp\left\{-\frac{1}{2}(\xi-\mu_o)^T\Sigma_o^{-1}(\xi-\mu_o)\right\} \tag{3.28}$$

where d represents the dimension of the data. In other words, it is the number of features which are used for image segmentation.

The EM algorithm, as a generalization of maximum likelihood estimation, consists of the iterations of Expectation-step and Maximization-step to find the parameters of the mixture model.

The first step in applying the EM algorithm is to initialize the parameters. After the initialization, the aforementioned two steps are iteratively performed till the algorithm converges and gives a maximum likelihood estimation. The implementation of EM algorithm can be summarized as follows:

- **Initialization:** The parameters we want to learn are initialized which consist of mean μ_k, covariance Σ_k and mixing coefficients α_k.
- **Expectation:** In the expectation step the current parameters are used to evaluate the posteriori probabilities, which are equivalent to the expected values of the latent variables, given the parameters $\Theta=(\alpha_1,...,\alpha_K,\theta_1,...,\theta_K)$. In fact, it is the probability that k^{th} Gaussian distribution fits to the observation data x_i and formulated as follows

$$p\{\omega:k=\kappa/x_i=\xi,\Theta=P^{old}\}=\frac{\alpha_\kappa f_{x_i/\theta}(\xi_i/\rho_\kappa)}{\sum_{m=1}^{K}\alpha_m f_{x_i/\theta}(\xi_i/\rho_m)} \tag{3.29}$$

  where $w\in\Omega$ is a point value in the sample space $\Omega=\{1,...,K\}$ and $\kappa\in\mathbb{R}$ is the dummy variable.

19 The notations are taken from [10] and [11].

- **Maximization:** Once the expected values have been calculated, the parameters of the mixture model are re-estimated as follows [2], [8], [97], [107].

$$\alpha_k^{new} = \frac{1}{n}\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\} \tag{3.30}$$

$$\mu_k^{new} = \frac{\sum_{i=1}^{n} \xi_i \, p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}}{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}} \tag{3.31}$$

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}(\xi_i - \mu_k^{new})(\xi_i - \mu_k^{new})^T}{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}} \tag{3.32}$$

- **Evaluation:** In this step the log of the likelihood function 3.27 is evaluated and the convergence of the parameters or the log likelihood is checked. If the convergence criterion is not satisfied the algorithm returns to the expectation step.

K-means Expectation Maximization (KEM)

As it was already mentioned the K-means algorithm has a hard membership function and a small shift of a data point can flip it to a different cluster. The solution to this problem is to replace hard clustering of K-means with soft probabilistic assignments [2]. This is done by EM algorithm because EM has no strict boundary among clusters and a data point is assigned to each cluster with a certain probability. However, the techniques such as EM might yield poor clusters if the parameters are not initialized properly. To solve this problem we propose a technique which combines K-means with EM, so-called KEM. This technique is similar to that presented in [59]. It employs K-means as the initial clustering to find the initial cluster centers. This reduces the sensitivity of the initial points and gives the centers which are widely spread within the data. These centers are used as the initial parameters for EM and it starts iterating to find the local maximum.

3.2.4 Experiments and Results

In this section, the results of object segmentation using multimodal data from a TOF-PMD camera and MultiCam are discussed. As stated before, the evaluation of segmentation results is still an open issue in this field. This is due to the lack of benchmarks in this area. In our work, it is even more difficult, because at the time of writing this thesis there is no TOF range image database which can be used to test the performance of the segmentation algorithms. Creation of a database for the TOF images in general and 2D/3D images in particular has high potential and will surely have big impact on new research work in the future.

The results in this section are presented in three subsections: i) The results of hand segmentation using clustering techniques based on TOF images taken by a PMD-3K camera, ii) The results of object segmentation using clustering techniques based on 2D/3D images taken by MultiCam, and iii) The

results of segmentation by using the integration of two techniques and based on 2D/3D images taken by MultiCam.

Low Resolution Hand Segmentation Results

All experiments have been done under real time conditions. The range and intensity images are taken directly in each snap shot of a TOF camera based on a PMD sensor. The resolution of the camera we have used is 3k (64×48 pixels). The modulation frequency and the exposure time have been set to 20*MHz* and 5*ms* respectively. Under these conditions, the frame rate of the camera is about 50 images per second, including the intensity, range and modulation amplitude images. Using K-means Expectation Maximization (KEM) which was discussed in the last section, each image is segmented. The frame rate of the segmented images is above video frame rates which is suitable for the real time gesture recognition and tracking. We evaluate our segmentation technique for the three following cases:

- **Case 1 - Hand gesture is posed in the foreground of a simple scene:** Fig. 3.16 shows some images for this case. In these images the hand is posed in the distance of over 10*cm* from the background, torso or face. The first row in Fig. 3.16 shows the intensity images for 8 different poses. The second row shows the coded range images such that the pixels of the background are darker than the pixels of the hand gesture in the foreground. The third row shows the results of segmentation using the KEM technique for six clusters. The images 1 to 3 show the hand gesture in a plain background while the images 4 to 8 show it in the scene where the user's body, face or arm are observed as well. As it is seen from the segmented images, the pixels related to the hand gesture have been grouped in one cluster very well without any error.

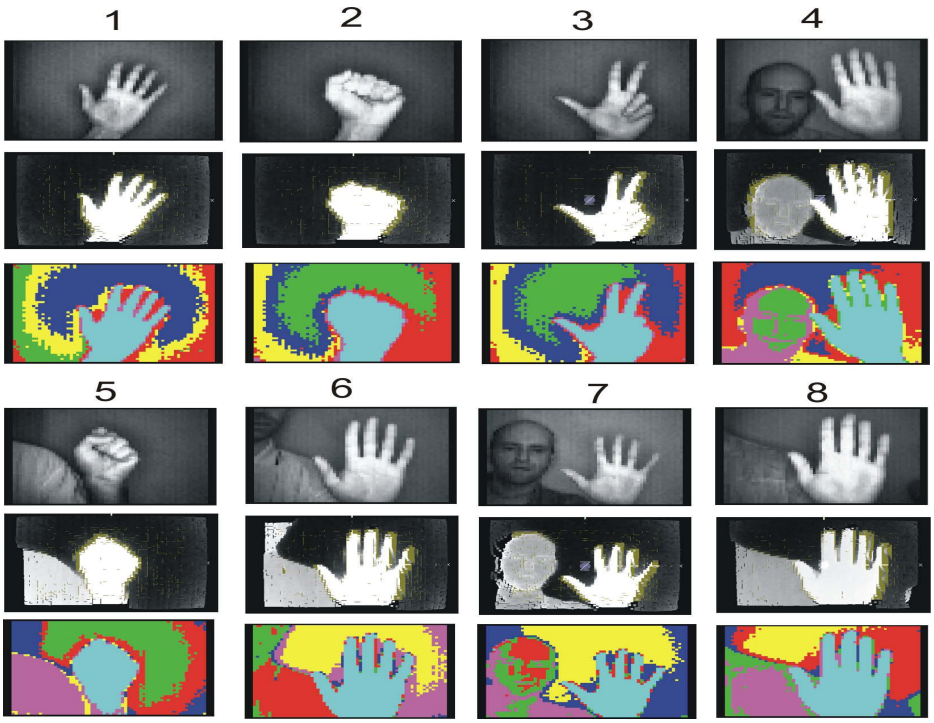

Figure 3.16: Results of hand gesture segmentation in a simple scene . First row: Intensity images. Second row: Range images. Third row: Segmented images [50].

In this case, since the hand distance from the background, torso or face (10 *cm*) is larger than the statistical noise of TOF range images (about 4 *cm*), the range information, without fusing with the intensity data, can be used as a single feature for the segmentation algorithm and it yields the same results as when the fusion of range and phase is employed.

- **Case 2 - Hand gesture is posed in the foreground in a cluttered and complex scene:** Some of these images are shown in Fig. 3.17. In this case the hand gesture is posed in a cluttered and complex scene where the lighting condition as well as the color of the objects affect the TOF images and make the problem more complicated. In this case, we have segmented the images once using just the range data as a single feature and once using the range and phase data derived from the fusion of range and intensity as discussed in section 3.2.2. The first and the second row of each hand gesture in Fig. 3.17. show the intensity and the range images respectively. The third row shows the results of the segmentation using the range data while the last row shows the segmented images using the fused data and range information.

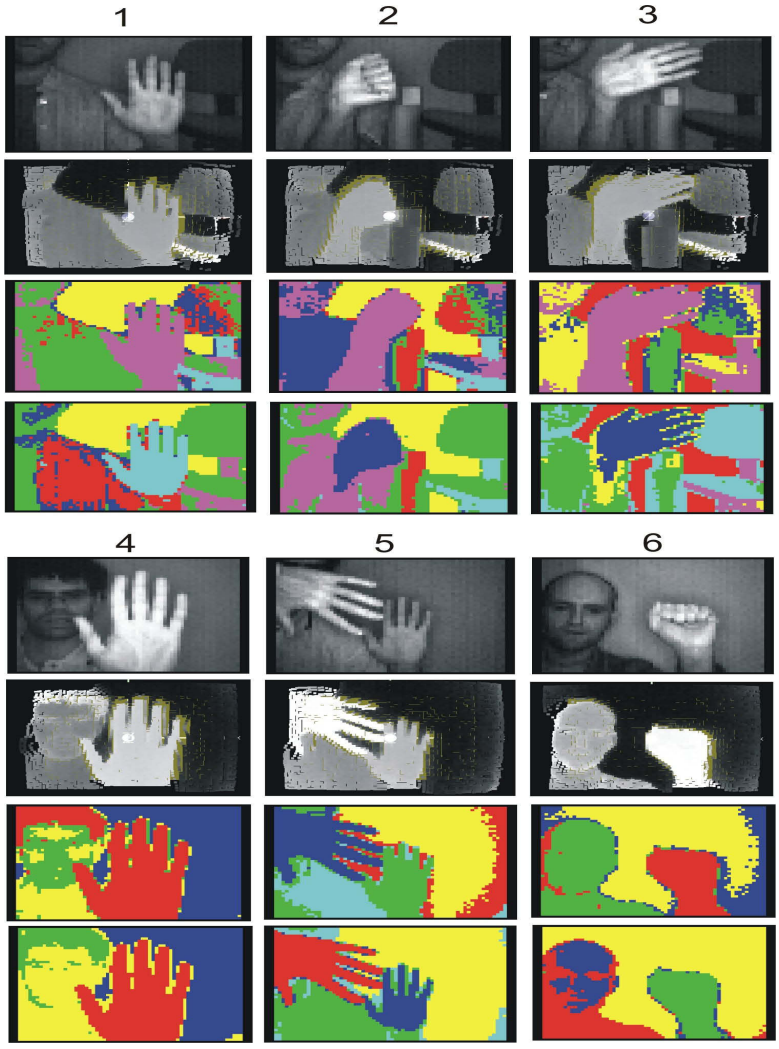

Figure 3.17: Results of hand gesture segmentation in a complex scene. First row: Intensity images. Second row: Range images. Third row: Segmented images using range feature. Fourth row: Segmented images using the fusion of range and phase features [50].

As we can see from the results, the segmented images using just range information get too much error and the pixels related to the hand gesture do not get separated from the pixels of

the other objects very well. In the images 1 and 4 the range data are affected by the color, i.e., the black color[20] does not reflect too much infrared light and therefore the range data get wrong values for these objects. This is one of the problems of a TOF camera which we have already discussed in this chapter. In the range images 2 and 6 since the face is not illuminated very well by the lighting system, it gets some errors in the range data. Likewise, the range data in images 3 and 5 are noisy because the hand gesture distance to the torso, face, arm or other objects in these images is smaller than the statistical error rate of TOF image (about 4 *cm*). However, these examples show that the TOF range images cannot be used solely to build a robust segmentation under these conditions. Fusing the range data with the intensity images can solve this problem to a great extent. As the results of the segmentation based on fused data in the last row of Fig. 3.17 show, the pixels related to the hand have been grouped in one cluster and the hand gesture has been segmented very well from the face, torso or other objects in the complex scene.

- **Case 3 - A sequence of moving gesture from foreground to the background:** Fig. 3.18 shows a sequence of moving hand from foreground to the background in the steps of 15 *cm*. As the previous figures, the first and second rows show the intensity and range images respectively while the third row shows the segmented images using KEM technique. The hand is segmented from the user's body, face and arm very well in all of the sequences except image sequence number 4 where the hand gesture and face are posed in the same distance from the camera and they both have the same intensity (or color) values. This is actually the case that the segmentation fails. However, this problem can still be solved by applying post-processing techniques like connected component algorithm to segment face and hand.

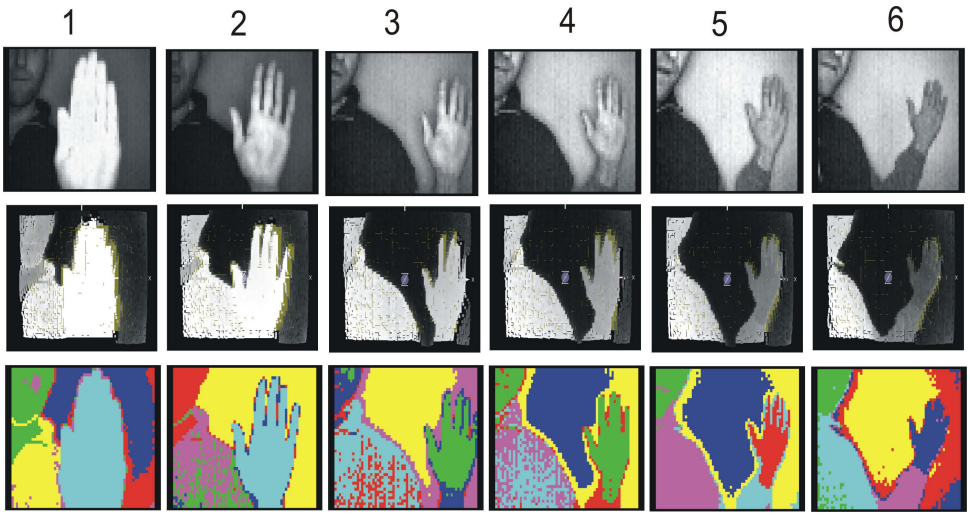

Figure 3.18: Results of gesture segmentation in a sequence of movement from foreground to the background. First row: Intensity images. Second row: Range images. Third row: Segmented images [50].

2D/3D Object Segmentation Results

Using MultiCam, we can acquire low resolution TOF images consisting of range, modulation amplitude and intensity data; and high resolution 2D Images. Same as the previous section, first we apply a clustering technique to segment the low resolution TOF images. Next, we map the 3D segmented image to 2D image. Due to the monocular setup of MultiCam, as discussed in chapter 2,

20 The color of the shirt in image number 1 and the color of the hair in image number 4 in Fig. 3.17.

mapping the 3D range image to the 2D image is a trivial and fast task which consequently makes the segmentation of high resolution 2D image computationally cheap.

This kind of segmentation has two main advantages over 2D segmentation. On the one hand 3D range segmentation is more reliable and robust in a natural environment where lighting conditions might change and on the other hand due to the low resolution of 3D image, segmentation is faster.

In order to present the results, two examples are considered. In the first example which is illustrated in Fig. 3.19, a person stands in mid-ground, between a closet as foreground and a door as background, in the distance of about $4\,m$ which is in the range of unambiguity of TOF sensor. First, the human is segmented based on range information by applying the K-means clustering technique. Next, we resize the segmented image to the resolution of 640×480 pixels which corresponds to the resolution of 2D image and map it directly to the 2D image. As can be seen from the results in Fig. 3.19, the person has been segmented very well except the boundary pixels correspond to the contour of the body. This is because the resolution of 3D image is low and range discontinuities between the contour of the body and background are usually observed in one pixel which results the wrong range information.

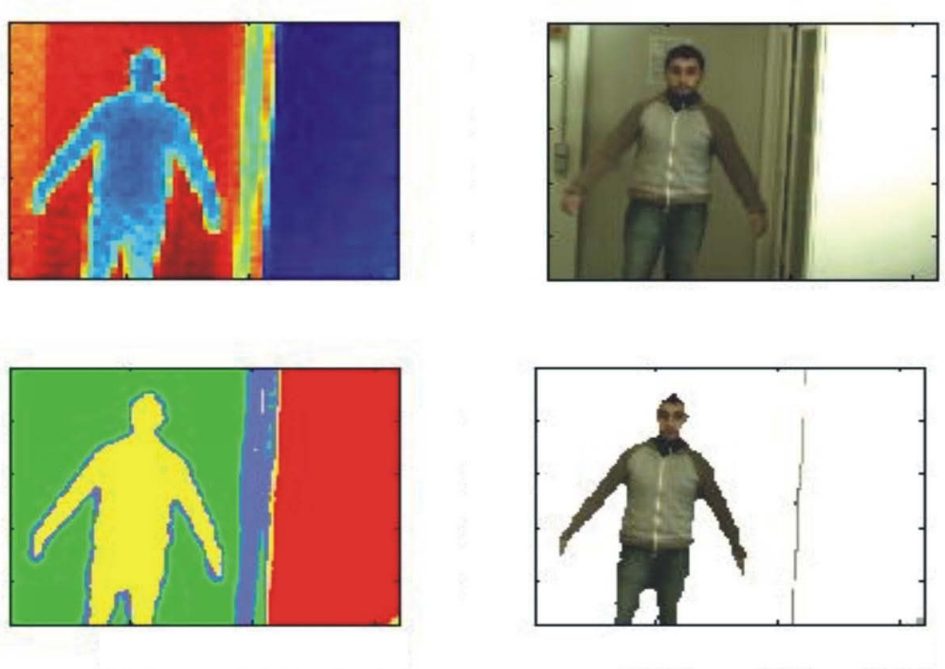

Figure 3.19: Human segmentation in 2D/3D images. Top Left: Low resolution range image from PMD-3K camera. Top Right: High resolution (640×480) 2D color image. Bottom Left: 3D segmented image using K-means and rescaled in high resolution. Bottom Right: 2D segmented image result from mapping.

In the second example shown in Fig. 3.20 we consider a more complicated case. In this example, although the person stands in the unambiguity range of TOF sensor, the area behind him is out of the range of 3D sensor (more than $7.5\,m$ at the modulation frequency of $20\,MHz$). Therefore, the pixels corresponding to this area get wrong range data. For example, the objects in the real distance of $9\,m$ get the distance of $1.5\,m$ wrongly and therefore range segmentation gets much error (see Fig. 3.20). In order to solve this problem, we take the modulation amplitude image into consideration. We know the objects lie in the further distance reflect less infrared light in comparison to the objects in the closer distance[21]. Thus, we multiply the normalized modulation amplitude with range data and use the result as the input data for clustering technique. This solves the problem to some extent as the pixels with wrong range data get corrected through their modulation amplitude values as the weighting factors. As can be seen in Fig. 3.20, by applying this method the person has been segmented from the background.

21 Unless the objects are extremely different in reflecting the near infrared light.

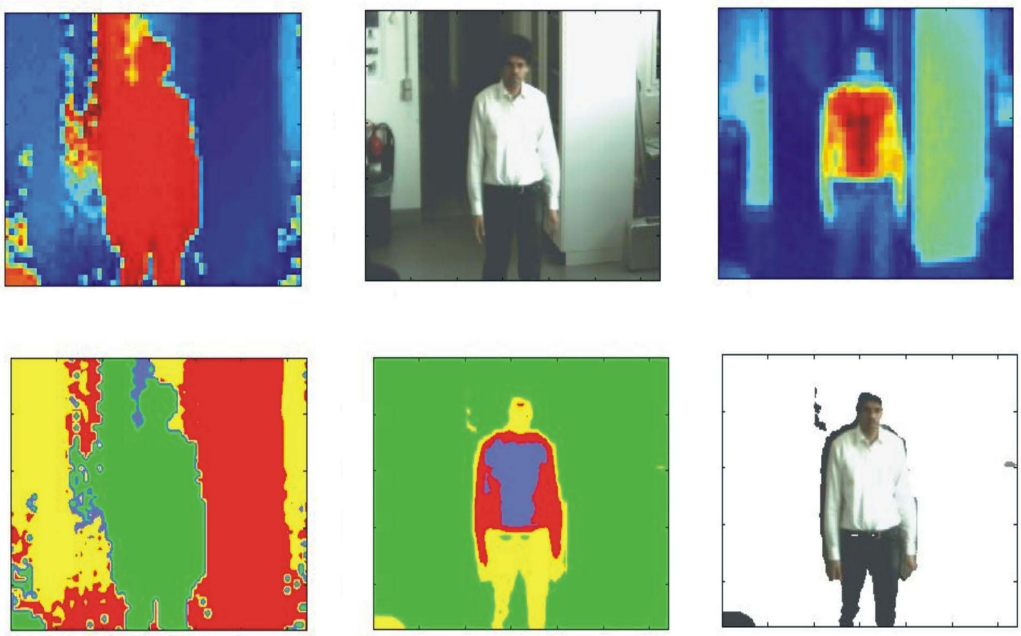

Figure 3.20: Human segmentation in 2D/3D images. Top Left: Low resolution range image from TOF sensor. Top Middle: High resolution 2D image. Top Right: Modulation amplitude image. Bottom Left: 3D rescaled segmented image. Bottom Middle: Rescaled segmented image using fusion of range and modulation amplitude data. Bottom Right: 2D segmented image result from mapping.

In addition to some mis-segmentation in this example, it is noted that the hair has been excluded in the segmentation result. This is because the hair in this example has black color which does not reflect as much as light like other parts of the body such as face and hands.

Integration of Edge Detection and Clustering in 2D/3D Image Segmentation

In the last section of the segmentation results an enhancement to the segmentation in 2D/3D images will be presented. The proposed method is based on the combination of edge detection and the unsupervised clustering technique. While the former is applied to the high resolution 2D images, the latter is done based on the low resolution TOF images as we have already discussed in the previous section.

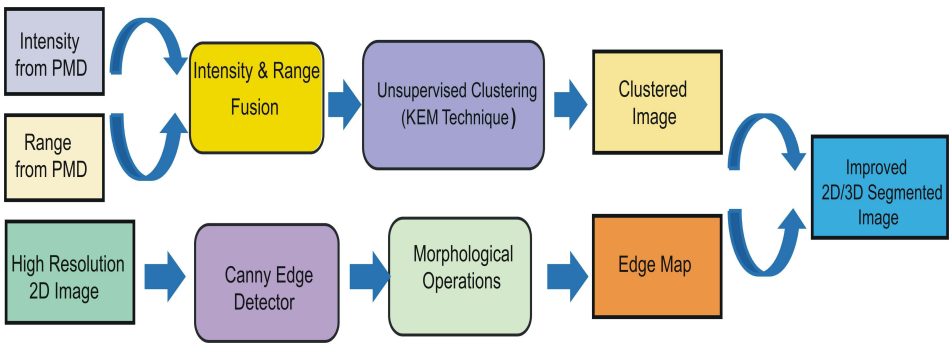

Figure 3.21: Block diagram of the improved segmentation technique.

The block diagram of the improved segmentation method is shown in Fig. 3.21. The range and

intensity images taken from PMD sensor are fused as discussed before. The fused result, so-called phase information, with range data are employed as the input for the KEM clustering. On the other hand, the edges in the high resolution 2D image are detected by applying a Canny edge detector and improved by applying some morphological operations to get an edge map. Finally the clustered image derived from 3D-TOF images is integrated with the edge map to derive an improved 2D/3D segmented image.

The Canny edge detection operator which is applied to the 2D intensity image tries to satisfy three goals simultaneously: i) Minimize the number of wrong edge detections, ii) Place the edges accurately (near to the real edges in the image), and iii) Mark each edge only once. To achieve this, the algorithm first computes the gradient of the image intensity, and later thins and thresholds the edge map in order to obtain a binary edge map [40].

By means of morphological operations (erosion, dilation), small edge elements are filtered. Finally, the boundary edges are traced in order to have individual contour information on the objects. We consider two examples of improved object segmentation using integration of clustering with the edge map.

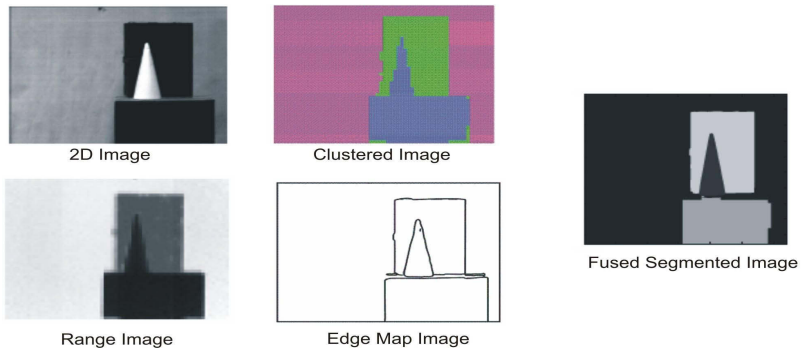

Figure 3.22: Object segmentation using the integration of clustering and edge detection - simple case.

In the first example, shown in Fig. 3.22, two objects (cone and beneath box) are posed in the same distance to the camera to make the object segmentation based on range data infeasible. On the other hand, this case is simple for segmentation using the 2D image, because one can see that the individual objects in the scene are segmented quite well by only using the edge information. This is because there is no specific texture on the objects that could be misinterpreted as edges and because the objects have clearly distinct intensity distributions. After tracing the boundary edges, three distinct objects are revealed and knowledge on a pixel level is obtained, i.e., one knows exactly which pixels belong to which object. Therefore the mis-segmentation problem in the clustered image is solved by using the edge information.

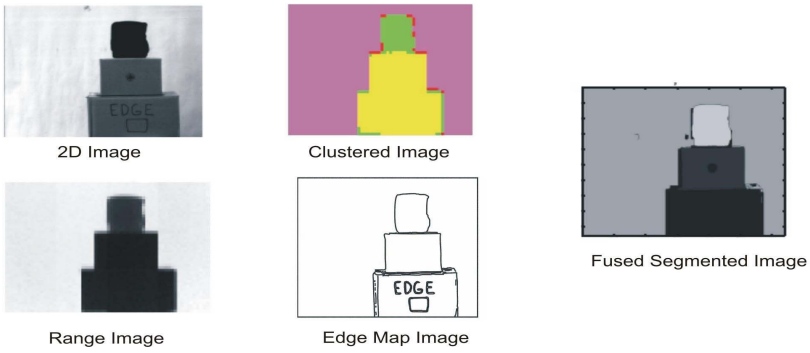

Figure 3.23: Object segmentation using the integration of clustering and edge detection - complicated case with texture.

In the second example which is illustrated in Fig. 3.23 we have made the segmentation task more complicated by putting some textures on one of the objects and posing two boxes at the same distance. The edge detection mechanism missegments the texture, which consists of the letters "EDGE" in this example, as boundary edges, and it is difficult to identify the texture and the object without prior knowledge of the scene. Also, the clustering technique missegments the boxes which are posed at the same distance. In this example neither the clustering nor the edge detector yield good segmentation result.

3.3 Object Classification

Object classification, which is a subtopic of pattern recognition, occurs in a wide range of our daily activities. When we observe, we usually see a complex scene which consists of a collection of objects. Due to the strong perception power of human beings, a specific object in the complex scene can be easily detected and classified very fast. The same, a fast and accurate recognition system, has been wished for real world computer vision problems, where it aims to make computers have such capabilities (seeing and recognition) owned by the human being.

Object classification, as the last stage of an object recognition problem, is a procedure, in which some decisions or forecasts are made on the basis of already acquired observations. The decisions are then applied to a new set of observations to assign each observation in a predefined group, called class. Each observation consists of a set of attributes or features in an image which are derived by feature extraction techniques already discussed in this chapter.

A wide variety of approaches has been taken towards the object classification problem grouped in three categories as statistical models, neural networks and machine learning techniques [9]. In this work, the problem of object classification is addressed using supervised machine learning techniques, which will be discussed in this sections.

A supervised learning classification is one of the machine learning techniques in which a decision function is built from a set of labeled training data, in contrast to an unsupervised learning, where instances are unlabeled. The decision function, called classifier, is used to assign a new observation to a predefined class.

In this context a classification system is implemented in three steps as follows:

- **Creation of training data set**: Training data set *S* is created from the extracted features of *n* observations as follows

  $$S=\{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$$

  where $X_j=(x_1, x_2, \ldots, x_m)$ is an observation with *m* features and $y=(1,2,\ldots,k)$ represents the output domain with *k* classes.
- **Training:** A supervised training algorithm, dependent on the type of problem, is chosen to create a classifier from training data set.
- **Test and evaluation:** The derived classifier is tested with a test data set. The test data set should have the same features which are used in the training data set.

In the implementation of a classifier, some issues of concern to the would-be classifier are listed as follows:

- **Generalization:** The ability of a classifier to correctly classify new observations which are not in the training set. In other words, it is the accuracy of a classifier in labeling the test data set.
- **Speed:** In real time applications, the speed of a classifier is a very important issue. For example a classifier which is 90% accurate might be preferred over a classifier with 95% accuracy if it is 100 times faster [9].

- **Complexity:** A classifier which is too complex may fit the noise and finally leading to overfitting problem. The best way to avoid overfitting is to use lots of training data.
- **Size of training data:** The size of the training data set is also an important issue. Some supervised learning techniques require a huge training data set in order to yield a highly accurate classifier. In some applications, collecting so many training data sets is infeasible. This problem can be solved by applying online learning techniques where a decision function is updated in response to each new observation. Likewise, semi-supervised techniques such as co-training can be applied to solve this problem.
- **Time to learn:** If a classifier should be learned real time in rapidly changing conditions, time to learn will play an important role which must be considered. This might imply that we need to use a small training data set to learn classifier.

In this work, we use two recent most popular supervised learning techniques to classify moving objects in 2D/3D images: Support Vector Machines and the AdaBoost technique.

3.3.1 Support Vector Machines

Support Vector Machines (SVMs) provide one of the most recently proposed machine learning techniques which has been very effective for general purpose pattern recognition problems. SVM is selected in our work because:

- It outperforms conventional classifiers, especially when the size of training data is small and the number of features is large.
- It can lead to high performance in practical applications.
- SVM has no local minima, i.e., the found solution is always the optimum solution derived from the given training data set.
- Since a few vectors out of the training set (support vectors) are selected to build the decision function the computational cost is reduced which is an important criteria in real time applications.
- It is based on some simple ideas which makes its implementation easy.

The basic idea behind SVM is that for a given finite amount of training data, the best generalization performance will be achieved if the right balance is struck between the capacity of learning and the accuracy on that particular training set [106]. Successfully governing the relation between the capacity and performance of a learning machine requires a sophisticated theory of generalization. Several theories exist that can be applied to this problem. The theory of Vapnik and Chervonenkis (VC) is most appropriate and classically used to describe SVMs and motivate them [7].

In the following we briefly review the theory of SVM and for a much more in depth understanding the reader is referred to [2], [5], [6], [7] and [106]. In our work the term SVM will refer to the classification with support vector methods while regression has been excluded.

Consider the problem of classifying a data set S with n observation vectors $x_1, x_2, \ldots, x_n$ and corresponding class labels $y_1, y_2, \ldots, y_n$. For the moment, we assume we have a binary classification problem, therefore the data set S can be described as

$$S: (x_1, y_1), \ldots, (x_n, y_n) \in \Re^n \times \{\pm 1\}. \tag{3.33}$$

The goal is to create a function f from the training data set S such that f will correctly classify new examples. The procedure of creating such a function with the support vector method is explained stepwise in the following:

Linear Support Vector Machines - Separable Case (Hard-Margin Classifier)

In this case there exists a normal vector $w \in \mathbb{R}^n$ and bias $b \in \mathbb{R}$ such that

$$f(x_i) = \begin{cases} w^T x_i + b \geq +1 & for \quad y_i = +1 \\ w^T x_i + b \leq -1 & for \quad y_i = -1 \end{cases} \tag{3.34}$$

where (w,b) defines a set of hyperplane functions as $w^T x + b = 0$. There may, of course, exist many such functions that separate the classes exactly (see Fig. 3.24.) Therefore, we should try to find the one that will give the smallest generalization error. The support vector machine addresses this problem through the so-called maximal margin solution, i.e., SVM tries to find the Optimal Separating Hyperplane (OSH) for which the margin is maximized. An example of OSH is illustrated in Fig. 3.24.

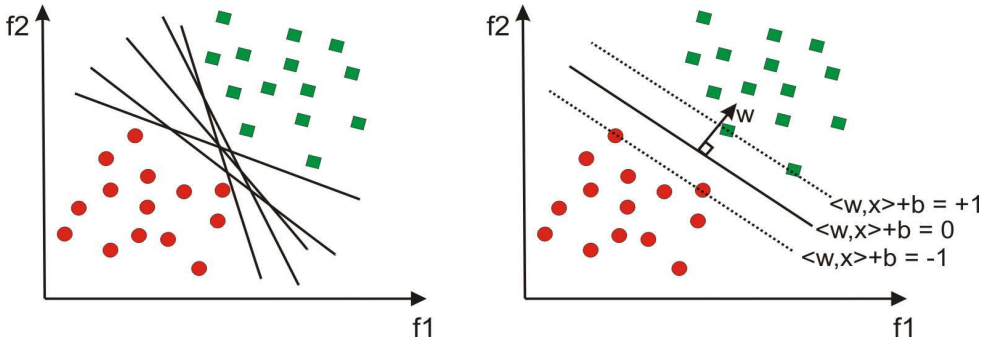

Figure 3.24: Left: Set of separating hyperplanes. Right: Optimal Separating Hyperplane (OSH). The dashed lines identify the margin. f1 and f2 represent two arbitrary features.

Equation 3.34 can be written into one set of inequality as follows

$$y_i (w^T x_i + b) \geq 1 \quad \forall i \,. \tag{3.35}$$

The orthogonal distance of point x_i from the separating hyperplane can be formulated by

$$d_i = \frac{w^T x_i + b}{\|w\|} \,. \tag{3.36}$$

By combining inequality 3.35 and equation 3.36 we will get

$$y_i d_i \|w\| \geq 1 \quad \Rightarrow \quad y_i d_i \geq \frac{1}{\|w\|} \,. \tag{3.37}$$

Hence $1/\|w\|$ is the lower bound on the distance between the point x_i and the separating hyperplane (w,b) and the margin is simply $2/\|w\|$. Now, we can find the maximum margin by maximizing $1/\|w\|$, which is equivalent to minimizing $\|w\|^2$, subject to the constraint in equation 3.35. Therefore the OSH can be regarded as the solution to the following optimization problem

$$P_1 : \begin{cases} Minimize: & \frac{1}{2}\|w\|^2 \\ wrt & : \; y_i(w^T x_i + b) \geq 1 \quad \forall i \; . \end{cases} \tag{3.38}$$

Note that the factor $1/2$ is included for later convenience. This is a quadratic programming problem in which we try to minimize a quadratic function subject to a set of linear inequality constraints.

To solve this problem we switch to the Lagrangian form by introducing positive Lagrangian multipliers $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$.

In fact, the solution to the problem P_1 is equivalent to determining the saddle point of the Lagrangian function as follows

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \{ y_i (w^T x_i + b) - 1 \} \; . \tag{3.39}$$

We must now minimize L with respect to b and w such that

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=0}^{N} \alpha_i y_i = 0 \, , \tag{3.40}$$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \quad \Rightarrow \quad w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i x_i \, . \tag{3.41}$$

Rewriting the Lagrangian function L using the obtained above conditions gives the dual representation of the maximum margin problem in which we should maximize

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.42}$$

with respect to

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad , \quad \alpha \geq 0 \quad \forall i \, . \tag{3.43}$$

Thus, the dual problem can be formulated by

$$P_2 : \begin{cases} Maximize: & L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \alpha^T D \alpha \\ wrt & : \; \sum_{i=1}^{N} \alpha_i y_i = 0 \quad , \quad \alpha \geq 0 \quad \forall i \end{cases} \tag{3.44}$$

where $(D_{ij}) \equiv y_i y_j x_i^T x_j$ is a $N \times N$ Hessian matrix.

The optimization problem P_2 can be solved by introducing an additional Lagrangian multiplier λ and applying the Karush-Kuhn-Tucker (KKT) conditions as follows

$$L(\alpha,\lambda)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\alpha^T D\alpha+\lambda\sum_{i=1}^{N}\alpha_i y_i \,. \tag{3.45}$$

By setting $\partial L(\alpha,\lambda)/\partial\alpha=0$, the Lagrangian coefficients α_i's can be determined. For every training point there is a Lagrangian multiplier α_i. Those points for which $\alpha_i>0$ are called support vectors and they correspond to the points that lie on the maximum margin hyperplane in the feature space as illustrated in Fig. 3.24. Having solved the quadratic programming problem and found a value for each α, we can then determine the value of *w* as follows

$$w=\sum_{i=1}^{N}\alpha_i y_i x_i \,. \tag{3.46}$$

Once the model is trained, a significant proportion of the data can be discarded because they have $\alpha=0$ and therefore only the support vectors are retained.

As it was seen while *w* is explicitly determined by the training procedure, the threshold *b* is not because it does not appear in the dual form and so *b* must be found by applying the Karush-Kuhn-Tucker (KKT) conditions to primal problem P_1 as follows

$$\alpha_i(y_i(w^T x_i+b)-1)=0 \quad \forall i \,. \tag{3.47}$$

For SV points: $\alpha\neq0$ and therefore for each SV point we calculate *b* by

$$b=y_i-w^T x_i \,. \tag{3.48}$$

From the standpoint of precision of calculations, it is better to take the mean value of *b* resulting from all support vectors such that

$$b=\frac{1}{N_S}\sum_{i\in S}(y_i-w^T x_i) \tag{3.49}$$

where *S* is the set of support vector indices and N_S is the total number of support vector points.

Thus, after the calculation of *w* and *b*, the decision function can be written as follows

$$D(x)=\sum_{i\in S}\alpha_i y_i x_i^T x+b \quad . \tag{3.50}$$

And finally the problem of classifying a new data point *x* is solved by

$$\begin{cases} class\,1 & if\ D(x)>0 \\ class\,2 & if\ D(x)<0 \end{cases} \,. \tag{3.51}$$

Linear Support Vector Machines - Inseparable Case (Soft-Margin Classifier)

Since in many real world problems there is no linear separation in the data, the hard margin classifier which was discussed in the previous subsection cannot be used directly. Therefore we need to modify it so as to allow some of the training data to be misclassified. To do this, we introduce

nonnegative slack variables $\xi_i > 0$ such that

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \forall i \,. \tag{3.52}$$

In this case, the goal is to maximize the margin while penalizing the misclassified points. Therefore we can write the OSH as follows

$$\begin{cases} Minimize: & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \\ wrt \quad : & y_i(w^T x_i + b) \geq 1 - \xi_i \;\;,\;\; \xi_i \geq 0 \quad \forall i \end{cases} \tag{3.53}$$

where C is called the generalization parameter which controls the trade-off between the slack variable penalty and the margin.

By rewriting the above optimization problem in Lagrangian form we obtain

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i [\, y_i(w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{N} \beta_i \xi_i \tag{3.54}$$

where α and β are the Lagrangian multipliers.

Similar to the previous case we apply the KKT conditions as follows

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = 0 \quad \Rightarrow \quad w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.55}$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=0}^{N} \alpha_i y_i = 0 \tag{3.56}$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi} = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C \quad \forall i \tag{3.57}$$

$$\alpha_i [\, y_i(w^T x_i + b) - 1 + \xi_i] = 0 \quad \forall i \tag{3.58}$$

$$\beta_i \xi_i = 0 \quad \forall i \tag{3.59}$$

By substituting equations 3.55, 3.56 and 3.57 into equation 3.54, we obtain the dual form of the problem as follows

$$\begin{cases} Maximize: & L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \alpha^T D \alpha \\ wrt \quad : & \sum_{i=1}^{N} \alpha_i y_i = 0 \;\;, \qquad 0 \leq \alpha_i < C \quad \forall i \end{cases} \tag{3.60}$$

which is identical to the separable case, except that the α_i cannot exceed C.

Again the points, for which $\alpha_i = 0$ do not contribute in making the decision functions. These points are

not support vectors and are classified correctly. The remaining data points are support vectors for them we have either $\alpha_i < C$ or $\alpha_i = C$. In the first case, equation 3.57 implies $\beta_i > 0$, which yields $\xi_i = 0$ from equation 3.59, and therefore such points lie on the margin. On the other hand, SV points with $\alpha_i = C$ lie inside the margin. In this case, if $0 < \xi_i \leq 1$ the points are correctly classified, i.e., they lie inside the margin but on the correct side of decision function and for the points which $\xi_i > 1$ they lie on the wrong side of the decision function and are misclassified.

The decision function is the same as that for the hard margin case given in equations 3.50 and 3.51.

Nonlinear Support Vector Machines - Kernel Based

If the training data are not linearly separable, the obtained decision function may not have high generalization ability. In order to solve this problem and enhance linear separability, the original input data is mapped into a higher dimensional space called the feature space where the data are linearly separable (see Fig. 3.25 [23]).

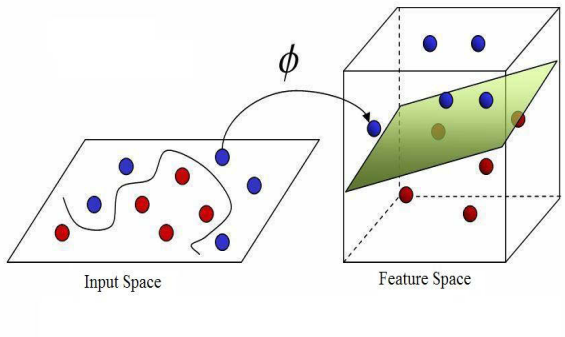

Figure 3.25: Mapping the nonlinear input training data into a higher dimensional feature space via map function ϕ [23].

For the nonlinear case, the decision function in equation 3.50 is formulated in the feature space as follows

$$D(x) = \sum_{i \in S} \alpha_i y_i \phi(x_i)^T \phi(x) + b \tag{3.61}$$

where $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m, (m > n)$ is a mapping function.

Although mapping the data to the higher dimensional feature space solves the problem of nonlinearity, working with high dimensional data is computationally expensive and the curse of dimensionality problem might occur. This problem is solved by the kernel trick in which the inner product of $\phi(x_i)^T \phi(x_j)$ is directly computed in the feature space as a function of the original data in the input space. In other words, a kernel is a function such that

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) . \tag{3.62}$$

Now we just need to replace $x_i^T x_j$ by $K(x_i, x_j)$ in the training data and by reusing the whole considerations of the previous sections the algorithm will produce a support vector machine in the high dimension feature space with roughly the same amount of time it would take on the unmapped input data.

In fact, by using a kernel *K* we do not even need to know the mapping function ϕ . Hence, the decision

function can be written as follows

$$D(x)=\sum_{i\in S}\alpha_i y_i K(x_i, x)+b \quad . \tag{3.63}$$

The Gaussian Radial Basis Function (RBF) is one of the commonly used kernel which has the following form

$$K(x_i, x_j)=\exp\left(\frac{-\|x_i-x_j\|^2}{2\sigma^2}\right). \tag{3.64}$$

For the sake of simplicity up to now we just discussed the binary classification problem. For the multi class (*n* classes), the problem can be solved by using two basic approaches:

- **One-vs.-All approach:** *n* SVMs are trained; each of the SVMs separates a single class from all remaining classes.
- **One-vs.-One approach:** *n(n-1)/*2 SVMs are trained; each of the SVMs separates a pair of classes.

The One-vs.-All approach with Radial Basis Function (RBF) kernel is used for our experiments, which will be described in the next section, because it has shown better performance in our case and only *n* SVMs have to be trained.

Here we focused only on the key concepts of SVM and for more detailed discussion the reader is referred to [2], [5], [6], [7] and [106].

3.3.2 Moving Object Classification Using Support Vector Machines

In this section we describe a classification system for moving objects based on Support Vector Machines and using 3D range images. Two kinds of camera systems are used to provide the classification system with 3D range images: a PMD Time-of-Flight camera and a stereo vision system.

The current approaches for the classification of moving objects can be categorized in shape-based and motion-based methods. In this work we focus on shape-based methods where the features extracted from 3D appearance of the objects are used as the input data for the classification system. The motion based techniques are excluded because in our experiments the objects have similar motion form while their 3D shape differ from each other.

Set-Up and Image Acquisition

Each camera system is mounted in a fixed structure, pointing down and oriented in such a way to have the same Field of View (FOV). The 3D range images are taken from the moving objects during their motion in the field of view of the camera via an object detection program. The object detection results from the continuous comparison of the acquired range image with that of the background, previously recorded in the absence of any object. The comparison criteria are based on some statistical characteristics of the range data which are considered as threshold in the detection program. The background is averaged from 100 range images taken by each camera in order to reduce the statistical noise, especially in the range images of PMD [61].

Image sampling is one of the significant points in detection and classification of moving objects. The number of acquired images using the TOF camera is higher than that using a stereo vision system. Because the distance data in the TOF camera is determined directly inside the hardware using smart

pixel array of the PMD, whereas the stereo vision system provides the 3D data from stereo imaging through some computational techniques which are time consuming. For the TOF camera this number is a function of the velocity v of the moving object, exposure time t_e, transfer time t_t and processing time t_p as follows

$$N = f(v, t_e, t_t, t_p). \tag{3.65}$$

The exposure time is the time which the TOF camera needs to illuminate the scene in order to get accurate range data.

For stereo vision, the sampling number is a function of velocity v, computational time t_c, transfer time t_t and processing time t_p as follows

$$N = f(v, t_c, t_t, t_p). \tag{3.66}$$

While the exposure time is neglected for the stereo system, the computational time is the time which stereo system needs to calculate the disparity map from the right and left images.

The velocity is tunable in the setup from 5 *cm/sec* to 20 *cm/sec*. As both cameras transfer the images via a FireWire interface and use the similar PC, the transfer time as well as processing time are same for the both. The above functions yield a frame rate of 20 range images per second for the TOF camera and 5 range images per second for the stereo vision system. At the velocity of 5 *cm/sec*, the TOF camera captures 160 range images from the object during its motion in the FOV (40 *cm* with the focal length of 16 *mm*) while the stereo system takes 40 range images. At the maximum velocity of 20 *cm/sec,* the number of sample images are 40 and 10 for the TOF and the stereo system, respectively.

Classification Algorithm

An overview of the algorithm is shown in Fig. 3.26. The input data are range images which are taken by both the TOF and the stereo system and saved in two image sets. The stereo range images

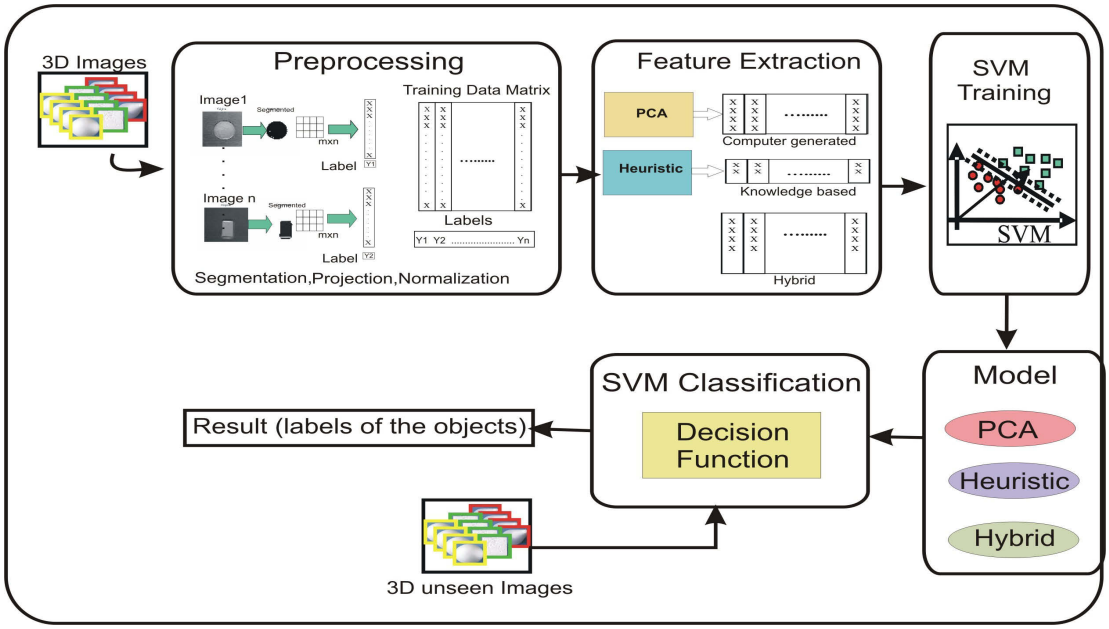

Figure 3.26: Flow of the moving object classification algorithm.

with the resolution of 640×480 pixels are resized to have the same size as TOF range images with the resolution of 64×48 pixels. The range images are then used as the input data for the classification system. They are first segmented from the background and then normalized and projected to compute the 3D coordinates of the points on the object surface with respect to the camera coordinate system. Then, the features are extracted, using two different kinds of approaches, consisting of computer generated features and human generated features. While the former are extracted using Principal Component Analysis (PCA), the latter are knowledge-based data obtained by applying some heuristic methods. The derived features are saved in the vectors of the training dataset matrix for each object. These features constitute three training datasets as a human-generated, a computer generated and a hybrid dataset which is combined from both features. For each configuration a support vector machine classifier is trained and tested in a setup under real time conditions.

The range images of the TOF camera are to a great extent independent of the lighting conditions, whereas the range images of the stereo vision system have the following difficulties:

- No 3D information over the plain surface of an object without texture,
- Strongly dependent on the lighting conditions,
- Disturbed by shadows.

Considering these difficulties we have taken three image sets for each object:

- **TOF**: Range images taken by the TOF camera under varying lighting conditions and objects are moving at different velocities.
- **Stereo 1**: Range images taken by the stereo system under the same conditions as TOF.
- **Stereo 2**: Range images taken by the stereo system under artificial conditions.

Artificial conditions consist of stable and non-varying lighting conditions and painting textures over the surface of the objects to get 3D information.

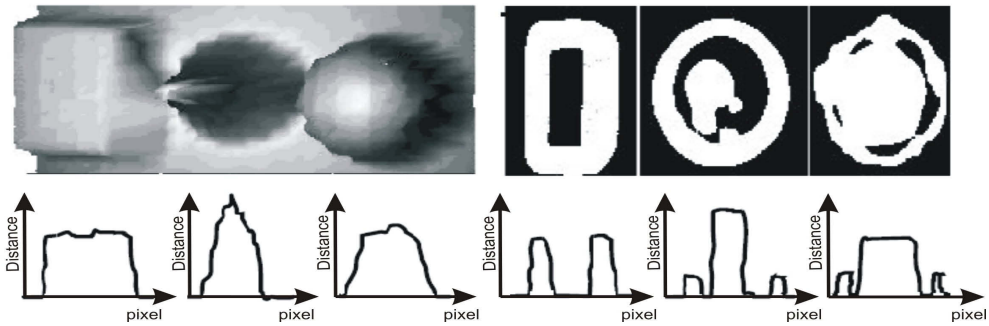

Figure 3.27: Range images of the multi object set including box, ball and cone. Top Left: TOF range images. Top Right: The stereo range images and the lower row shows the longitudinal sections through the middle of each range image.

Fig. 3.27 shows some range images of the multi object set including box, ball and cone, which were taken by the TOF camera and the stereo system under natural conditions. As it is observed the range images of the stereo system cannot provide any reliable 3D data on the plain surface of the object and therefore the value is set to the value of the background which is shown as black in the picture.

Classification Results Using PMD TOF Images

For the PMD set the range images have been collected for each object set at two different velocities of 5 *cm/s* and 10 *cm/s* under varying lighting conditions. For each range image collection per day, a

corresponding background image has been taken and used for segmentation technique to minimize the statistical noise at a constant threshold. The other parameters, such as modulation frequency, exposure time and the aperture of the lens have been always set the same at 15 *MHz*, 1 *ms* and 1.4 (for 16 *mm* lens) respectively.

For the TOF case, three sets of the objects have been considered in our experiments to test the proposed classification system:

- **Set 1**: Multi-set with three different shape objects, including cubic box, spherical ball and cone which has been illustrated in Fig. 3.27.
- **Set 2**: Binary set with two exact shape objects from the point of view (POV) of the camera including sphere and hemisphere.
- **Set 3**: Binary set with two exact shape, same surface but different height from POV of the camera including cubic boxes.

While the set 1 has been selected to test the classification system in general, the set 2 and set 3 challenge the proposed system for classifying the objects with the same appearance and the same color as the background. Detection and classification of such objects by using just gray values in 2D sensor based classification system is a big issue which requires special tricks.

For each training dataset, three different classifiers have been trained based on heuristic, PCA and hybrid features. The RBF kernel with the kernel argument of 1 and regularization constant of 10 have been used in the SVM training algorithm.

The training dataset includes 171 range images for object set 1 and 112 range images for object set 2 and 3 respectively. The system has been tested with the testing dataset including 129 images for object set 1, 280 for object set 2 and 84 test images for object set 3.

Table 3.1 shows the results of the classification for the general object set 1. While the PCA feature based classifier outperforms the heuristic feature based one, the hybrid feature based classifier which employs the combination of both features yields the best result with the minimum error rate of 2.32%.

Table 3.1: Number of misclassifications and error rate for object set 1 (General Case).

	Heuristic	PCA	Hybrid
Cubic Box	4	0	0
Spherical Ball	1	3	1
Cone	0	1	2
Error (%)	3.87	3.1	2.32

Table 3.2 shows the results of the classification of the object set 2. PCA based classifier improves the error rate from 5.71% in the heuristic case to 3.21%. Using the hybrid features the classifier outputs the best result with the full accuracy of 100%.

In Table 3.3 the results of the classification for the object set 3 are shown. In this case, the heuristic feature based classifier contrary to that in the previous cases gives the best performance. It is observed that the heuristic features representing the shape of the object are quite appropriate in the training for the classification of the cubic boxes.

Table 3.2: Number of misclassifications and error rate for object set 2 (Sphere and Hemisphere).

	Heuristic	PCA	Hybrid
Sphere	10	3	0
Hemisphere	6	6	0
Error (%)	5.71	3.21	0

Table 3.3: Number of misclassifications and error rate for object set 3 (Boxes).

	Heuristic	PCA	Hybrid
Cubic Box 1	0	4	3
Cubic Box 2	0	8	0
Error (%)	0	14.28	3.57

The other advantage of the heuristic based classifier in this case is that since the calculation of features is trivial and the numbers of features are limited, the classification process is not time consuming and can be interesting for the real time object classification. The PCA based classifier gives a poor result with the error rate of 14.28% for this case. This error is expected because at the edges perpendicular to the direction of the movement of the boxes some mis-measurements occur which affect the result of PCA method strongly, whereas they do not affect the result of the heuristic techniques. These mis-measurements are called motion artifacts which appear in TOF images when an object moves. They are comparable to the motion blur in 2D images. As we have already discussed it in chapter 2, the motion artifacts can be removed by applying some morphological operations. Likewise, employing the hybrid features improves the accuracy and reduces this error to the rate of 3.57%.

It is noticed that in the cases where the heuristic features cannot represent a good distinction factor like in object set 1 and object set 2, the classifier which takes the most relevant principal components improves the accuracy. However, the combination of features which are obtained from heuristic and PCA outputs the best results.

Classification Results Using Stereo Range Images

In the previous experiments we compared three classifiers using different features derived from TOF images. Now, in this section we will compare the results of object classification based on PCA and heuristic features derived from TOF and stereo images. For each range image set (Stereo 1 and Stereo 2), two training data sets have been derived using PCA and heuristic features. Then for each training data set a multi class SVM classifier has been trained. As in the previous case, we have selected the RBF kernel with the kernel argument of 1 and regularization constant of 10. The SVM classifier is trained with 90 range images for each image set which are taken at the velocity of 5 *cm/sec*. The classification system has been tested with 90 range images for each set. The results for the multi object case (general case) are shown in Table 3.4. While the TOF range based classifier outperforms the stereo range based one, it is observed that the classifier which employs the range images of Stereo 2 (artificially textured with fixed lighting conditions) yields the best results with the full accuracy of 100%.

Table 3.4: Error rate for multi object set classification (%).

	Heuristic	PCA
Stereo 1	30.00	30.00
Stereo 2	1.11	0
TOF	3.87	3.1

As we observe the Stereo 1 based classifier shows a high error rate. This is because the range data in Stereo1 are too inaccurate. Fig. 3.28 shows the distribution of the features on the first two principal components for Stereo1 and TOF. It is observed that the features of two classes in Stereo1 are very mixed and difficult to distinguish. Training the data in this case yields a classifier with a high number of support vector points (88 support vectors out of 90 training data points) which indicates a poor classifier.

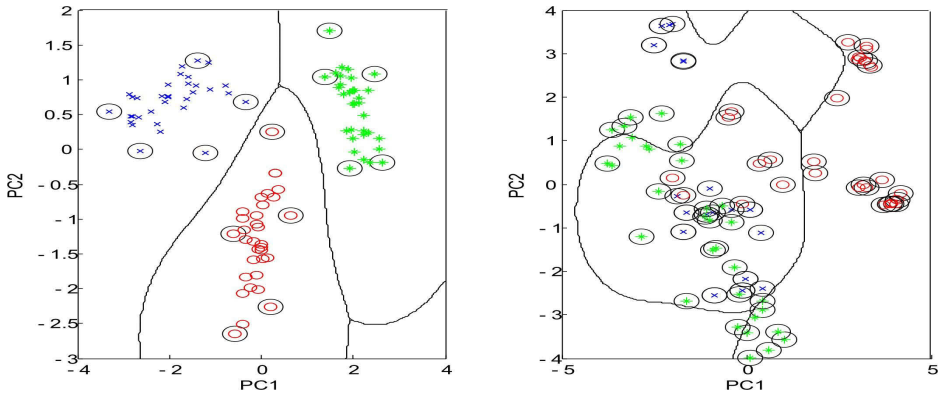

Figure 3.28: Distribution of the features on the first two principal components with the separating SVM based hyperplane. Left: TOF. Right: Stereo 1 (The support vectors are indicated by circles).

By applying the artificial techniques which we mentioned before, these data are separated and the margin between the classes increases. Therefore, we get the best results in the case of Stereo 2. Also, it can be noticed that in all of the cases the PCA feature based classifier gives a better result than the heuristic feature based one.

For the binary object set (e.g., two cubic boxes with the same shape form but different heights from the camera's point of view), same as for multi object set, the results for the stereo image set under natural conditions (Stereo 1) are very poor. Changing the conditions as discussed before improves the results in the image set of Stereo 2 to the lowest error rate of 1.67% using PCA features.

3.3.3 AdaBoost Classification

AdaBoost, adaptive boosting, is one of the most widely used form of boosting techniques which combines multiple base classifiers in order to produce a strong classifier. This technique which was developed by Freund and Schapire [2], [80] proved that the training error of the strong classifier approaches zero exponentially in the number of boosting rounds even if the base classifiers are weak

with a performance that is only slightly better than random.

In this section we describe the original form of the AdaBoost technique. In the next section, the modified AdaBoost technique, proposed by Viola and Jones which is sometimes called Viola-Jones method [80], will be explained and we will show how this method has been utilized using 2D/3D images for a fast and accurate real time hand detection.

Having a binary classification problem, with the training data set of $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$, AdaBoost constructs a strong classifier *H(x)* from a linear combination of weak (base) classifiers *h(x)* through an iterative T rounds of boosting as follows

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{3.67}$$

where α_t are coefficients found during the boosting process.

Each data point in the training set is given an associated weighting parameter w_i which is initially set to $1/n$ for all data points. Also, we shall assume that there is a training procedure to derive the weak classifier $h_t(x)$ from the weighted training data in each iteration. After each round the weighting coefficients are updated such that the weights of the misclassified points in the round before get greater. This is because the next weak classifier should focus much more on the hard examples which are not correctly classified in the round before. The details of the AdaBoost technique are as follows:

AdaBoost Algorithm:

Assumption: Given the training data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in \mathbb{R}^n \times \{\pm 1\}$

1. **Initialization:** Initialize the weighting factors for all data points to get a uniform distribution of the data: $w_i^{(1)} = 1/n$, i.e., at the beginning all the data points have the same importance for the learning algorithm.
2. **Loop:** for $t = 1, \ldots, T$ do:
   A) Train a weak classifier by minimizing the weighted error function as follows

$$J_t = \sum_{i=1}^{n} w_i^{(t)} I(h_t(x_i) \neq y_i), \qquad I(h_t(x_i) \neq y_i) = \begin{cases} 1 & if \quad h_t(x_i) \neq y_i \\ 0 & otherwise \end{cases}$$

   B) Evaluate the error rate and calculate α_t as follows

$$\epsilon_t = \frac{\sum_{i=1}^{n} w_i^{(t)} I(h_t(x_i) \neq y_i)}{\sum_{i=1}^{n} w_i^{(t)}} \qquad \text{and} \qquad \alpha_t = \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

   C) Update the weighting factors such that the misclassified data get greater weights as follows

$$w_i^{(t+1)} = w_i^{(t)} \exp\{\alpha_t I(h_t(x_i) \neq y_i)\} \ .$$

3. **Create strong classifier:** after finding the weak classifier, the final strong model is constructed by

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \ .$$

3.3.4 2D/3D Object Detection Using the Viola-Jones Method

In this section we discuss another example of object recognition using 2D/3D images. The proposed technique is a very fast and accurate method for object detection. The detection technique is based on the AdaBoost approach which was proposed by Viola and Jones [80]. Therefore it is also called Voila-Jones method. We will first review their method and then discuss, how it has been used for object detection using 2D/3D images and finally represent some results.

Viola-Jones Method

The Viola-Jones object detection framework which employs the Haar-like features, which was already described in section 3.1.4, consists of three main contributions as follows:

- **Integral Image:** Although Haar-like features might be good features to detect an object in the image, the exhaustive set of these features in a search window is very large and therefore the calculation of all these features in an image is very time consuming. The integral image is an intermediate representation of the image which makes the computation of Haar-like features extremely rapid.

Given an image *i(x,y)*, the integral image *ii(x,y)* can be calculated as follows [80]

$$ii(x,y)=\sum_{x'\le x,\,y'\le y} i(x',y') \quad . \tag{3.68}$$

In fact, the integral image at location *x,y* contains the sum of the pixels above and to the left of *x, y.* The integral image can be calculated in a very simple way in one pass through the original image. Having the integral image any rectangular sum (like Haar-like features) can be computed very fast as it is illustrated in Fig. 3.29.

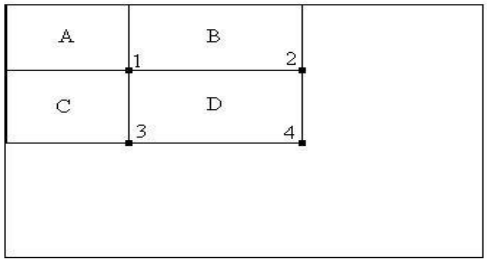

Figure 3.29: The sum of the pixels in the rectangle D can be calculated simply as: D=(4+1)-(2+3) where 1=A, 2=A+B, 3=A+C and 4=A+B+C+D [80].

- **AdaBoost:** The next main contribution is to use AdaBoost in order to select a small set of Haar-like features and train a strong classifier from these features.

The number of possible Haar-like features in a search window is far larger than the number of pixels in that window. Although by applying the integral image one can calculate a Haar-like feature very fast, as the number of these features is vast, considering all of them in a rapid object detection problem is infeasible. On the other hand, most of the features in a search window are useless and do not represent any information about the detected object in the image. By applying AdaBoost to a given training data set the best features with the minimum error rate (smallest number of misclassification) are iteratively selected. After picking the first best features from the feature pool, the training samples are re-

weighted such that the misclassified data points get a larger weight and correctly classified samples obtain smaller weights. In the next iteration, the now best feature is picked and then the samples re-weighted and so on. In each iteration a weak classifier, which corresponds to the best feature in that round, is selected. The final strong classifier is then constructed from these weak classifiers as it was explained in the AdaBoost algorithm.

For each feature f_j a weak classifier h_j is defined by

$$h_j(x)=\begin{cases}1 & if \quad p_j f_j(x) < p_j \theta_j \\ 0 & otherwise\end{cases} \tag{3.69}$$

where x is the search window, f_j is the absolute value of the feature, p_j is the parity indicating the direction of the inequality and θ_j is the threshold. The value of θ_j is determined in the AdaBoost procedure such that the classification error on the training data (positive and negative classes) is minimized.

- **The Attentional Cascade:** The last contribution of the Viola-Jones method is a technique for combining successively more strong classifiers in a cascade structure which dramatically increase the detection performance.

The performance of a strong classifier which was created through AdaBoost is not good enough for many real world object classification problems. In order to achieve increased detection performance while radically reducing computation time several classifiers are arranged in a cascade form as illustrated in Fig. 3.30. Using the cascade structure many negative samples (sub-windows) are rejected at the earliest stage possible. The first stage has a high detection rate (nearly 100%) and the false positive rate (about 50%). It means in this stage all of positive samples are detected correctly while about half of the negative samples are rejected. Each stage in the cascade reduces the false positive rate and decreases the detection rate. As the detection rate of each stage is close to one, their multiplication results a final detection rate of close to one, while the multiplication of the false positive rates approaches zero.

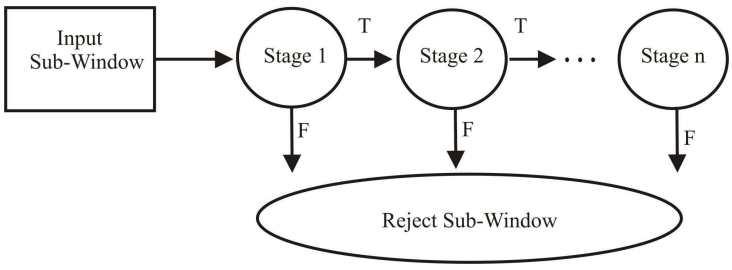

Figure 3.30: Cascade of classifiers [80].

Overview of Object Detection Algorithm

There are two main issues in real time object detection using the AdaBoost technique. The first issue is that background noise in the training images degrades detection accuracy significantly, especially when we have a cluttered background under varying lighting conditions which is the case in many real world problems. The second issue is that the computation of all sub-windows in an image for every scale is too costly if the real time constraints are to be met [49].

The fundamental idea of our algorithm is to address the solution to these problems using the fusion of 3D range data with 2D images. In order to extinguish the background issue from object recognition problem, the procedure of object detection is divided into two levels. In the low level we use range

data in order to: i) Define a 3D volume where the object of interest is appearing and eliminate the background to achieve robustness against cluttered backgrounds and ii) Segment the foreground image into different clusters using unsupervised learning techniques which were already discussed in section 3.2.3. In the high level, we map the 3D segmented image to its corresponding 2D color image and apply the Viola-Jones method (searching with Haar-like features) to find the object in the image. Fig. 3.31 shows some examples of these two levels.

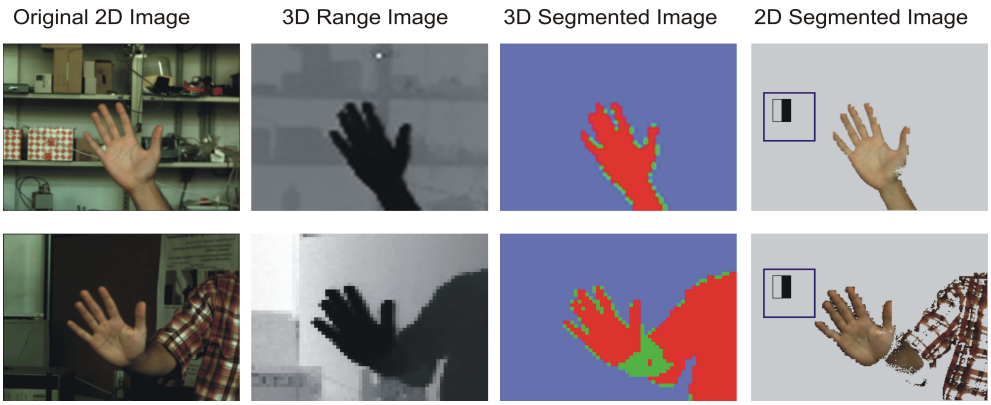

Figure 3.31: Solution to the background issue in object detection using the Viola-Jones method. Using range data, the cluttered background is removed and the foreground image is segmented and mapped to the 2D image. The Viola-Jones technique is applied to the 2D segmented image to find the object of interest.

The second issue (computation of all search windows in an image for every scale is too costly) can be addressed by using the range information directly. After segmentation, the distance of the segmented object from the camera can be easily derived from the 3D range image. By having the information about the distance of the object from the camera, its size can be roughly estimated and a set of search windows which could fit to the size of the object is selected. This reduces the computational cost of the Viola-Jones technique to a great extent which is a significant point in real time applications. An example of selecting the search windows for hand detection has been illustrated in Fig. 3.32.

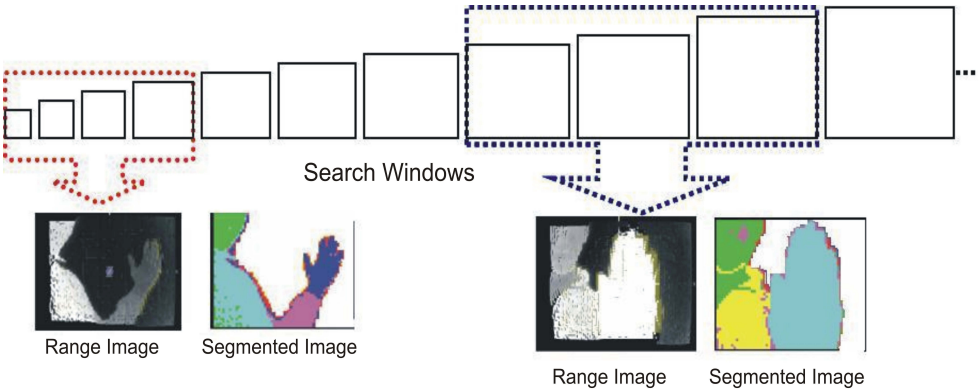

Figure 3.32: Selection of search windows using range information for hand detection. Left: Hand is far from the camera and therefore the image is searched with small search windows. Right: Hand is close to the camera and therefore the image is scanned with large search windows to find the hand in the image.

3.4 Summary

An object recognition mechanism consists of a sensor system, a preprocessing unit and an object classification. In the preceding chapter, we discussed the first part by introducing a 2D/3D camera system. In this chapter, some different aspects of preprocessing and classification techniques using 2D and 3D image data were studied.

Preprocessing, in our work, consists of feature extraction and image segmentation which are performed before classification in order to make the problem simpler.

We have discussed machine generated and human generated features by using Principal Component Analysis, Linear Discriminant Analysis and some heuristic approaches.

Multimodal image segmentation is the next issue which has been studied in this chapter. It was seen that even the low resolution range data can be used directly to segment the objects in the scene. Likewise, we have studied some aspects of image segmentation in which range, intensity and modulation amplitude are fused to increase the performance of the segmentation. On the other hand, some improvements in image segmentation have been realized by integration of two different approaches: Clustering and Edge detection. While the former is applied to 3D range data, the latter is performed on high resolution 2D image. Integration of the clustered range image with 2D edge map yields a highly accurate segmented image.

In the last section of this chapter, two advanced classification techniques, which have recently gained a lot of attention, have been presented: Support Vector Machines (SVM) and AdaBoost. Both of these approaches are supervised learning techniques. They have been employed for classification of moving object in the real world problems in this chapter. It was seen that the SVM performs very well in 3D object classification using TOF range data even with small training data set. On the other hand, AdaBoost is a very fast classifier which has shown promising results for real time applications. Using 3D range data, the background noise as well as the computational issue have been addressed in the AdaBoost approach.

4

2D/3D Object Tracking

When I want to understand what is happening today or try to decide what will happen tomorrow, I look back.
Omar Khayyam (1048-1131)

In the previous chapter we discussed some techniques for object detection using 2D/3D images. The next step in many real time computer vision applications is to locate the position of the detected object(s) in each frame in order to find its (their) trajectory in a video. In general, object tracking in a real world environment is a very challenging problem due to different issues such as complex object motion, scene illumination changes, scene appearance changes, noise in images, nonrigid nature of objects, occlusions and real time processing requirements. There are numerous techniques which have been proposed to solve object tracking problems recently. Yilmaz et al. [86] have given a good overview and comparison of many state-of-art tracking methods. The reader is referred to this reference in order to have a general overview about object tracking techniques. Almost all object tracking techniques apply some constraints to the problem to simplify it. For example, in many techniques it is assumed that the motion of the object is smooth without any abrupt changes. In some other techniques, different constraints are applied either to the appearance of the object(s) or to the background. Likewise, some prior knowledge about the object of interest such as shape, size or color is used as a prior assumption to simplify the problem. Although these kinds of constraints and assumptions reduce the complexity of object tracking task, they, in fact, constrain real world problems.

Object detection, in fact, is one of the main mechanisms in object tracking which is either done in each frame or at least in the first frame where the object of interest should be detected to trigger the tracking system. In this chapter, some object tracking techniques based on 2D/3D images will be

studied and we will show, how multimodal 2D/3D data can be used to simplify the problem.

Two main aspects of this simplicity in our work are as follows:

- Using additional range information, the number of constraints which are usually applied to the tracking problem can be reduced. Thus, the proposed solutions based on 2D/3D image data are more applicable to real world problems. For example, 3D range data are more reliable than 2D images under varying lighting conditions and therefore there is no need to keep the lighting conditions unchanged. Also, by having 3D data, some assumed information, like size and shape of object can be exactly calculated in each frame without any prior assumptions.
- Computational time is a critical issue in an object tracking problem, especially in those cases where the object moves fast. 2D/3D data in our vision system can complement each other, without any time consuming procedures, such that the good features for tracking can be extracted very fast. As an example, in a gesture based robot control application, where the hand of the user is to be detected and tracked, using 3D range data the operating volume can be defined and extracted easily. This volume is then mapped to the corresponding 2D image where the useful features are calculated. This reduces the computational demand to a great extent because just the 2D/3D data in the volume is processed and there is no need to process the whole image data.

A typical object tracker can generally be built in two ways: i) A bottom-up approach in which an object detection technique is first employed to find the object(s) of interest in each frame and then it is followed by a technique to find corresponding objects across the frames, ii) A top-down process, which deals with the dynamics of the moving object as well as prior observations of the scene. In this case the position of the object is estimated by iteratively updating object location from previous frames [90].

In this chapter both techniques are reviewed in general, and we apply them by using some recent advanced approaches in each technique and based on 2D/3D images. As object tracking is a very wide and complex problem, first we define the framework of our work in order to avoid any confusion for the reader and to make our results comparable with the others in the same framework. Object tracking in this chapter is performed in the following framework:

- A single static MultiCam (2D/3D Vision System) observing the scene.
- Objects move in an indoor dynamic scene in any arbitrary direction.
- Lighting conditions may change.
- Single and multiple object tracking are considered.
- Person, robot and hand are the main objects of interest in this work for applications which will be described in chapter 5.

In this context, we will mainly answer the questions such as how to represent the object of interest, how to subtract foreground objects from the background, what kind of features should be used, how to extract the features, how to detect the objects of interest, how to match detected objects in consecutive frames or how to predict the position of the object of interest in the new frame and finally how to find the trajectory of the moving object.

4.1 Dynamic Scene Analysis

In our work a dynamic scene is a scene which changes from time to time due to changes in the environment illumination, motion changes, like static object removal or intrusion as well as changes in the scene, due to occlusion. In this section some aspects of a dynamic scene based on 2D/3D images are studied.

4.1.1 Background Subtraction

In many computer vision applications where the camera is fixed, scene analysis often starts by distinguishing foreground objects from the background. In fact, background subtraction is the most common approach to identify the moving objects in the foreground from the background. In general, background subtraction can be done by making a reference image and subtracting each new frame from the reference image and threshold the result [79]. Although it looks very simple, it rarely works in the real world applications and there are many challenges in making a good background subtraction. For example a good background subtraction should be robust against illumination changes as well as avoiding the detection of non-stationary objects in the background as foreground objects. Likewise, it should be fast enough to be used in the real time applications.

A typical approach to subtract the background is to make a statistical model of the scene, called background model, and update it constantly over the time. The moving objects identification in each frame can be performed by spotting the parts of the current image that deviate from that model.

There are numerous approaches to model the background which differ in the type of background model as well as way of its updating. A standard method is to average the images over time to create a time averaged background model. Although it is an effective and simple method, it has many problems which cannot be used for a real world problem. For example, it is not robust in the cases where there are many moving objects, especially if they move slowly. Also, it requires a training period to learn the background in the absence of foreground objects. In addition to these drawbacks, gradual changes in the lighting conditions cause problem for this technique. Therefore, one of the main requirements for making a good background subtraction is to reestimate the model constantly.

One of the successful background modeling is to learn each pixel of the background using a Mixture of Gaussian (MoG) models. This approach which was proposed by Stauffer and Grimson [84] was later improved by KaewTraKulPong et al. [79] to run faster and become more accurate in the busy environments.

As in the original approach of Grimson et al. each pixel is modeled with a mixture of Gaussian over color, it performs poorly in the backgrounds with dynamic textures such as trees waving in the wind. Recently, a generalization of the MoG approach is proposed by Grimson [39] himself which handles dynamic textures as well.

The modification of MoG done by KaewTraKulPong et al. [79] has been implemented in OpenCV for real time background subtraction and it is used for 2D background subtraction in our work. On the other hand, we will implement a simple background subtraction using 3D range data which are more reliable than 2D images in the varying light conditions. The results of these two techniques are evaluated and we will select the better one for background subtraction.

Adaptive Gaussian Mixture Model

As opposed to modeling the values of all pixels with a particular type of distribution function, Stauffer and Grimson [84] proposed to model the value of a particular pixel by a mixture of K Gaussian distributions. They define the probability that a certain pixel s has the value of X at time t as

$$f_{X_t}(\xi)=\sum_{i=1}^{K} w_{i,t}\cdot f_{X_t/\mu_{i,t},\Sigma_{i,t}}(\xi/\mu_o,\Sigma_o) \tag{4.1}$$

where ξ is the dummy variable corresponding to the pixel value X, K is the number of distributions and $f_{X_t/\mu_{i,t},\Sigma_{i,t}}(\xi/\mu_o,\Sigma_o)$ is the i^{th} Gaussian probability density function which is formulated as follows

$$f_{X_t/\mu_{i,t},\Sigma_{i,t}}(\xi/\mu_o,\Sigma_o)=\frac{1}{(2\pi)^{d/2}|\Sigma_o|^{1/2}}\exp\left\{-\frac{1}{2}(\xi-\mu_o)^T\Sigma_o^{-1}(\xi-\mu_o)\right\} \tag{4.2}$$

where $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean and the covariance of i^{th} Gaussian density function in the mixture model at time t and d represents the dimension of the pixel data X. For example, if we use RGB color images, then $X\in\mathbb{R}^3$ and therefore the model will be a multivariate Gaussian with three dimensions. If we consider the range data in addition to the color for each pixel, we will have $X\in\mathbb{R}^4$ and the model will be a multivariate Gaussian with four dimensions.

In equation 4.1, $w_{i,t}$ is a weight factor for i^{th} Gaussian at time t which represents what portion of the data is accounted for by this Gaussian model.

For computational reasons, Stauffer and Grimson [84] suggested to take the covariance matrix as a diagonal one such that

$$\Sigma_{i,t}=\sigma_{i,t}^2 I\,. \tag{4.3}$$

This assumption holds that the feature values of a pixel, $X_t=\{x_{t1},...,x_{td}\}$, are independent and have the same variance.

In their approach all weights are updated at every new frame and every given new pixel value X_t is checked against the existing K Gaussian distributions. If the pixel value falls within 2.5 standard deviation of any distribution[22], the distribution is marked as a matched component and its parameters are updated as follows

$$w_{it}=(1-\alpha)w_{i,t-1}+\alpha \tag{4.4}$$

$$\mu_{i,t}=\rho X_{i,t}+(1-\rho)\mu_{i,t-1} \tag{4.5}$$

$$\sigma_{i,t}^2=(1-\rho)\sigma_{i,t-1}^2+\rho(X_{i,t}-\mu_{i,t})^T(X_{i,t}-\mu_{i,t}) \tag{4.6}$$

where α is the learning rate and $1/\alpha$ determines the speed at which the parameters are changed. ρ is the second learning factor which is defined as

$$\rho=\alpha f_{X_t/\mu_{i,t},\Sigma_{i,t}}(\xi/\mu_o,\Sigma_o)\,. \tag{4.7}$$

For unmatched distributions, the parameters μ and σ remain the same, whereas the weighting factors for those distributions are reduced as follows [38]

$$w_{i,t}=(1-\alpha)w_{i,t-1}\,. \tag{4.8}$$

If none of distributions match the current pixel value in that frame, then the distribution with the least

22 It is about 98.76% of all realizations lie within the interval [μ-2.5σ, μ+2.5σ].

weight is replaced with a new distribution with the mean of X_t, an initial large variance and a small weight.

Once every Gaussian model is updated, the distributions are ordered by the value of w/σ and then the first B distributions are chosen to create the model of background where B is calculated by

$$B = argmin_b(\sum_{j=1}^{b} w_j > T) \tag{4.9}$$

where the threshold T is the minimum fraction of the data which should be accounted for by the background model.

In their paper, those pixels which have the values greater than 2.5 standard deviation of any distribution in B are then marked as foreground pixels. In our experiments we take this value as a variable threshold which is set in order to evaluate the results in a quantitative form.

Background Subtraction Using Range Thresholding

As the range data are more robust than 2D images against lighting changes as well as other variations in the scene, it is not absolutely necessary to apply complex approaches to get a good background subtraction result. A simple range thresholding can even identify foreground objects from background without so much computational demand.

Given a range image $I_{x,y,z,t}$ in each frame, first the range image is filtered such that the out of range data are eliminated. In our case, we have defined out of range data as very small or very large range values which either belong to objects in the ambiguous range of PMD sensor[23] or consist of systematic noises of the PMD sensor described in chapter 2. To eliminate them, a bandpass filter with two cut-offs is defined. In fact, the cut-offs define the limit of a 3D scene volume in z direction. This volume represents the 3D volume where the objects of interest may move in and therefore we call it the volume of interest in our work. Next, the filtered range data are smoothed using a Gaussian kernel succeeding with morphological operations consisting of erosion and dilation with a 3x3 rectangular as structuring element, in order to remove irrelevant information.

In the last step a fixed level thresholding is applied to the processed range data which discriminates the foreground objects from the background and finally the remaining foreground objects in the volume of interest are separated using a connected component technique.

Evaluation of Results

In order to evaluate the results of the discussed techniques for background subtraction using 2D and 3D images, we have considered two scenarios: i) A person walks in normal lighting conditions with a cluttered background, ii) A person walks in a challenging scene where the lighting conditions vary extremely and we have a clutter background with illumination disturbances and shadows. In both cases the 2D and 3D range data are recorded in a video format and they are used as the index videos for comparison of the background subtraction techniques which will be explained in this section.

Some results of background subtraction techniques using 2D images and 3D range data under normal lighting condition are depicted in Fig. 4.1. In order to make the results comparable, 3D range images are rescaled to 640×480 pixels which is the original resolution of the 2D images.

In the calculated foreground images white color represents the foreground pixels, whereas the background pixels are marked in black. The shown results in Fig 4.1 are calculated by setting the

23 It is more than $7.5m$ at a frequency of $20MHz$.

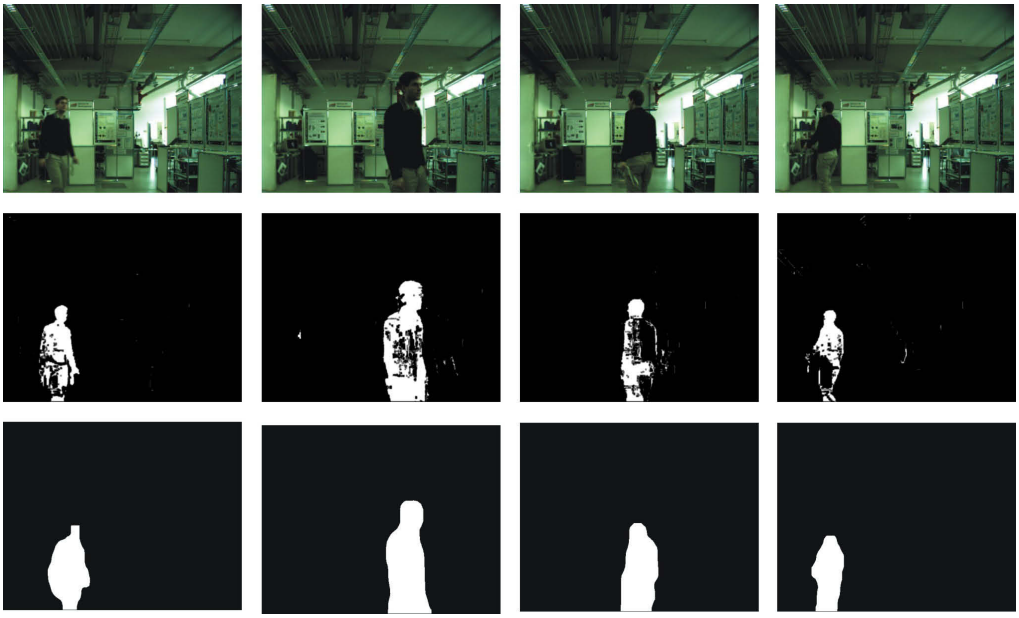

Figure 4.1: Some results of background subtraction under normal lighting conditions. First row: Original 2D images, frames number 55, 105, 160, 220 and 260. Second row: Corresponding foreground images derived using the adaptive Gaussian mixture model technique on the 2D images. Third row: Corresponding foreground images derived using range thresholding and the connected components technique from corresponding 3D images.

threshold for each technique at the most optimal point with the maximum hit rate and minimum false positive rate.

In order to compare the performance of the two discussed techniques quantitatively, we will use Receiver Operating Characteristic (ROC) curves [103] which are useful graphs for the visualization of the performance. ROC curves have long been used in signal detection theory which is later adopted to machine learning and computer vision to evaluate and compare the results of segmentation, classification and tracking techniques. ROC curves are usually two dimensional graphs which represent the trade-off between hit rate (true positive) and false positive (false alarm). However, in the cases where the time plays a key role as an evaluation factor, ROC curves can be plotted dependent on time.

To plot ROC curves we need to calculate the hit rate and false positive rate of each technique at different thresholds. The hit rate is defined as the ratio of the number of correctly detected foreground pixels to the number of all foreground pixels in the ground truth. The false positive rate is determined as the ratio of the number of wrongly detected foreground pixels to the total number of background pixels in the ground truth. Since the video sequence consists of too many images, we have selected some index images within the video at constant time stamps.

The foreground pixels are manually marked in each index image to create its corresponding ground truth image. Having ground truth images and the results of background subtraction (calculated foreground images) for each threshold T, the hit rate and false alarm are calculated for each index image at that threshold. The calculated hit rates and false alarms are then averaged to derive the final hit rate and false positive rate for the video at threshold T. Likewise, the processing time to calculate the foreground image for each threshold in each technique is recorded.

The ROC curves for the normal lighting conditions are shown in Fig. 4.2. The hit rate and false alarm depends heavily on the threshold. Setting threshold to a very low level implies high true and false positive rates which means it takes all the pixels which are slightly different from the background as

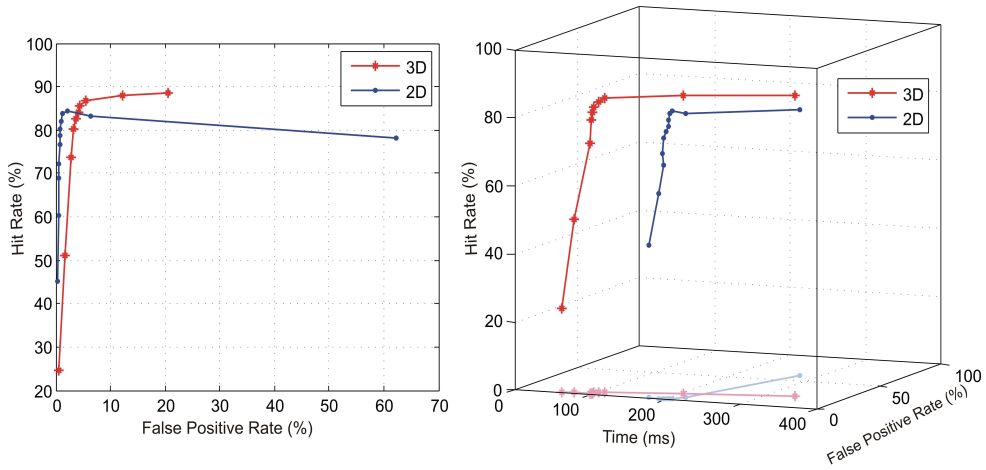

Figure 4.2: ROC curves for the image sequences under normal lighting conditions. Left: Trade-off between hit rate and false positive rate. Adaptive Gaussian mixture model using 2D images outperforms 3D range background subtraction. Right: Time dependent ROC curve. Considering time as the third evaluation parameter, 3D background subtraction outperforms 2D background subtraction.

foreground (top right point on ROC curve). In this case, although the hit rate is high, there is also a high false alarm which reduces the performance of the technique. On the other hand by setting the threshold to a high level, the rates of hit and false positive reduce dramatically which means just pixels which are very different from background are considered as foreground pixels (bottom left point on ROC curve). The best threshold would be somewhere between these two points which has the highest hit rate with the lowest false alarm.

The left graph in Fig. 4.2 shows the trade-off between the hit rate and false alarm which is like a usual ROC graph. To compare these two techniques, we do not calculate and consider the area under ROCs as a performance criteria, but rather we compare the best operating point for each ROC. As it can be seen from this graph, the best operating point for the Gaussian mixture model is the threshold which yields the hit rate of 84.4% with false alarm of 2.2%, whereas for the range thresholding the best point is the threshold which gives the hit rate of 85.5% which is slightly more than the hit rate of the former technique, but results in the false alarm of 4.4% which is two times bigger than the false alarm for the other technique. Therefore, in this manner the adaptive Gaussian mixture model technique using 2D images outperforms the simple 3D range thresholding.

By taking the time as the third evaluation parameter into consideration, we have plotted the ROC curves depending on the time which are depicted in the right part of Fig. 4.2. As it is clearly recognizable from these graphs, adaptive Gaussian mixture model technique is computationally more expensive than the simple range thresholding. Therefore, in such a manner the range thresholding technique outperforms the adaptive Gaussian mixture model. As an example, for the above mentioned best operating points the Gaussian mixture model takes 205 *ms* to subtract the background, whereas for range thresholding it just takes 103 *ms*. (both techniques have been run under the same processing conditions). Thus, the selection of one of these techniques depends on the priority of the evaluation criteria. While the hit rates of both techniques at the best threshold are nearly the same, 3D range thresholding has double false alarm rate with half time expense to the 2D Gaussian technique. The high rate of false alarms in the 3D case is mainly because the ground truth images have been derived manually from 2D images which have higher resolution than the range images (1 to 100 in this case).

In the second case, we make the problem of background subtraction more challenging. In this case, the lighting condition change extremely to disturb the background subtraction. Due to the light variation over the time, some shadows appear in some image sequences and disappear afterwards. Some examples of such images taken from the video are shown in Fig. 4.3. Same as before, background subtraction is performed using the two aforementioned techniques and the results are compared. Some

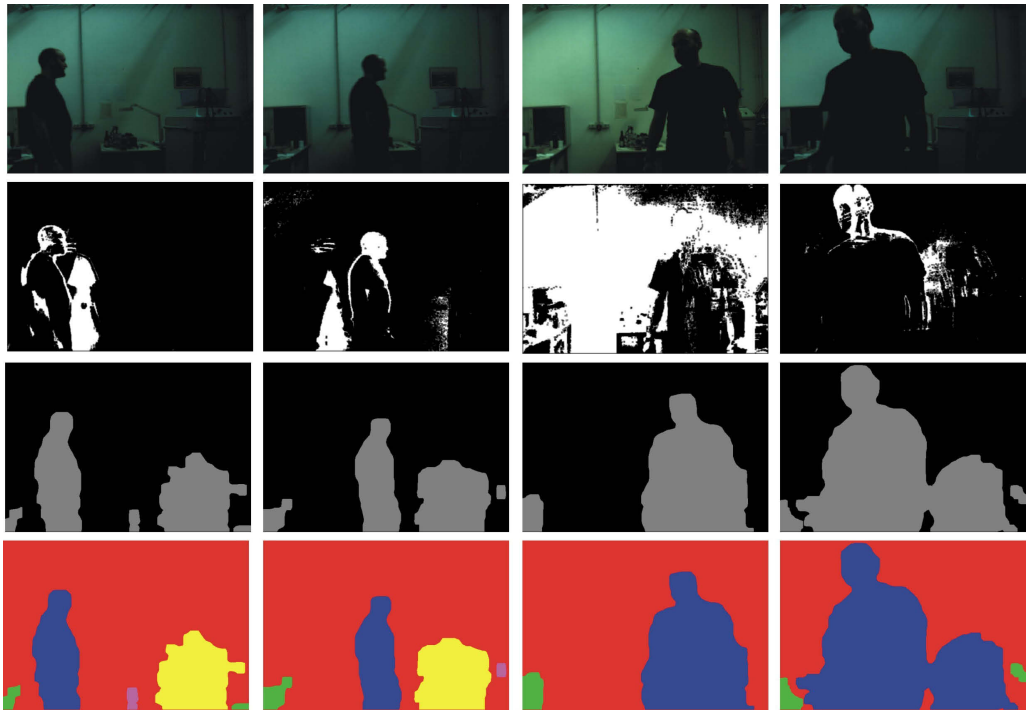

Figure 4.3: Some results of background subtraction under varying lighting conditions. First row: Original 2D images, frames number 130, 180, 310 and 415. Second row: Corresponding foreground images using the adaptive Gaussian mixture model on 2D images. Third row: Corresponding foreground images using range thresholding. Forth row: Connected components on 3D foreground images.

index images are selected from the video such that they can present the whole video under varying lighting condition.

Since the lighting has a big influence on 2D images, the results are not so satisfactory, as it can be seen in sample images in Fig. 4.3. On the other hand, range data are quite reliable in such conditions and changing lighting and/or having shadows in the scene does not influence range information. Same as the previous case the ROC curves are provided which are illustrated in Fig. 4.4. As we can see from the ROC curves, 3D range thresholding succeeding with connected components outperforms the 2D adaptive Gaussian mixture model dramatically in both time dependent and independent cases.

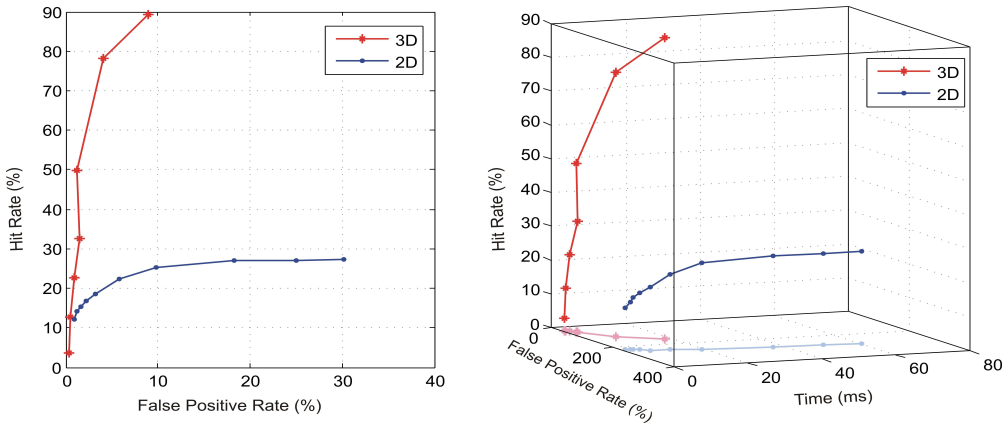

Figure 4.4: ROC curves for image sequences under varying lighting conditions. Left: Trade-off between hit and false positive rates. Right: ROC curve dependent on time. In both cases range thresholding followed by connected components outperforms the adaptive Gaussian mixture model using 2D images.

4.1.2 Real Time Aspects

Real time computing is one of the key issues in object tracking which implies selecting the algorithms and hardware such that they meet the given real time constraints. Real time constraints are the deadlines from event to system response. In other words, the time which a system needs to analyze or detect an event at exactly the time as the event is taking place in the reality. For example, in a real time traffic video analysis a car should be detected and tracked as fast as its motion in reality. The term real time is sometimes misunderstood as analyze and calculation of an event extremely quick. This is totally wrong because the event itself might take place slowly and therefore it is not absolute necessary to have a high performance computing system.

In order to implement a real time tracking system, first we need to know, how fast an object can move and how fast it can be detected and tracked. In this work, as already mentioned, we focus on tracking of people, hand gestures and robots. While an average walking speed of a person is about 4 to 5 *km/h* which is equal to 1.1 to 1.4 *m/s*, the robot which is used in this work can move at different velocities from 5 *cm/s* to 2 *m/s.* For the hand movement there is no reference value because it can be moved at different velocities. However, based on our laboratory experiences, the maximum velocity of 1 *m/s* is assumed for that. Therefore the maximum motion velocity in this work is limited to 2 *m/s*.

In the next step we need to define an event. An event is the motion of an object of interest with a fixed distance resolution. For example, a person should be detected and tracked at every 20 *cm* motion or a hand can be detected and tracked for every 2 *cm* motion in a real time hand based robot control application. Given the maximum motion velocity of the object V_{max} and detection resolution $R_{detection}$, the time of event can be calculated by

$$t_{event} = \frac{R_{detection}}{V_{max}} . \tag{4.10}$$

In a real time system the hardware and software should be selected such that the total time for taking the data (2D/3D images) and analyze it (detection and tracking) should be less or equal than t_{event}.

Now, we will have a look at the hardware which is the camera system. As already discussed in the previous chapter the time it needs to take an observation is a function of exposure time t_e, transfer time t_t and processing time t_p as follows

$$t_{observation} = f(t_e, t_t, t_p) . \tag{4.11}$$

Exposure time is the time which the TOF camera needs to illuminate the scene to get accurate range data. Transfer time depends on the communication protocol requirements (in our work USB 2.0) and processing time is the time to calculate range and modulation amplitude data from PMD phase images.

In the software part we term response time $t_{response}$ as the time which the whole algorithms and techniques, such as data preprocessing, background subtraction, segmentation, feature extraction, classification and tracking, need to detect the object of interest and locate its position.

Finally, to make an object detection and tracking system “real time”, the following condition must be met

$$t_{observation} + t_{response} \leq t_{event} . \tag{4.12}$$

This condition will be discussed more in details for the real time applications in chapter 5.

4.2 Object Representation and Identification

Object representation and identification is the most common approach to track the objects. It is a bottom-up approach which is performed by first detecting the objects of interest in each frame and then identify them[24] from frame to frame. In general, this approach consists of three steps which have been illustrated in Fig. 4.5. In the first step, background subtraction is done to obtain the possible regions which can represent the objects of interest in the scene. In our work, background subtraction is done using range images and based on the technique which was already discussed in this chapter. In the next step, a contour detection technique is applied to the foreground image to find all contours. Very small contours belong to noise in the images are filtered out. Each detected contour is then labeled and mapped to the corresponding 2D image. Due to the monocular setup of 2D/3D camera, mapping is trivial and fast. In the last step, each object is represented by some useful features which are extracted within detected contours in 2D and 3D images and finally the feature sets derived from the contours in two consecutive frames are used for correspondence matching which will be discussed in the following section.

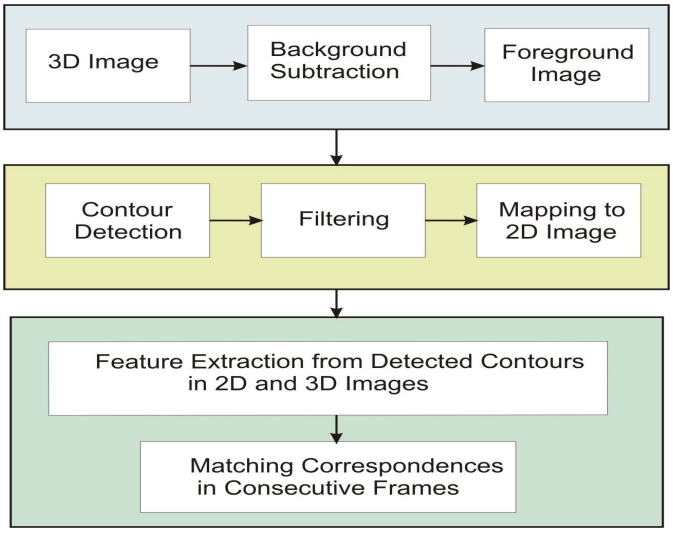

Figure 4.5: Block diagram of bottom-up tracking approach in general.

4.2.1 Feature Extraction and Correspondence Matching

The main part of object representation and identification technique is to correspond detected objects across successive frames. To do that, first we need to represent the objects using a model. For example, an object can be represented using geometric shapes or (and) appearance models. While simple geometric shapes like rectangle and ellipse are appropriate to represent a simple rigid object, the appearance models based on contour and silhouette are suitable to represent nonrigid objects. Since in the applications of this work we are involved in detecting and tracking nonrigid objects like persons (full body) and hand, we have used the contour which provides an accurate object representation. The contours in the image are obtained by applying contour detection to the foreground image. The found contours corresponding to the noises in the foreground image are eliminated by filtering the contours through the area size criteria. The remaining contours are then mapped to the 2D images. Some examples are shown in Fig. 4.6.

24 Identification means to find out which object is which.

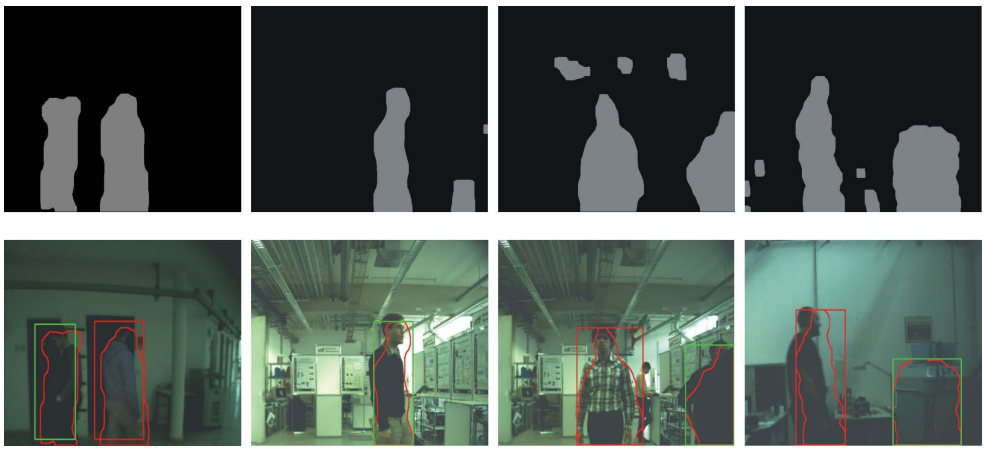

Figure 4.6: Top: Foreground images derived using background subtraction technique. Bottom: Corresponding 2D images with detected contours as object of interest. It is noticeable that due to the low resolution of 3D images, the contours which are derived from range data do not completely fit to the boundary of their corresponding objects in the 2D images.

As we can see from these images, the found contours do not fit exactly to the boundary of the objects in the 2D images. This is because the resolution of the range image (64×48 pixels) is much smaller than the resolution of the 2D image (640×480 pixels) and therefore mapping the found contours from the 3D foreground image to the corresponding 2D image gets some inaccuracies. Likewise, due to the systematic range error in the boundary pixels of the object in 3D image[25], the derived foreground image has some error which consequently induces inaccuracies in the result of the contour derivation. For the applications where the boundary of the object must be extracted precisely, the contours can be improved by applying some image post-processing techniques at the expense of time. Since in our work we rather detect and track the full objects than the parts of the objects, this error is neglected.

The flowchart of feature extraction and correspondence matching is shown in Fig. 4.7. In the first frame, after detecting the contours in the image, each contour is summarized by a bounding box with a color which represents the number of that object in the scene. For example, green represents object number one, red describes object number two, blue identifies object number three and so on. The colors do not imply any specific meaning or information about the objects. They are just defined arbitrarily to identify the objects in order. The next step is to match similar objects and label them with the same numbers (colors). Each object is represented by a feature set including the number of object, associated to it in the first frame, center of mass of the object in the world coordinate system, the average and standard deviation of color of the object in RGB space, something similar to Gaussian probability distribution function for the object, and the size of the object.

In a new frame at time t, first the number of detected objects n in the foreground image is compared to the number of detected objects m in frame t-1. As we can see from the flowchart in Fig. 4.7, based on the comparison result of m to n, three cases can be considered as follows:

1. $n = m$ implies that all detected objects in frame t-1 exist in the new frame t. In this case, a correspondence matching technique is applied to identify these detected objects in the new frame. In our work, a heuristic graph matching approach is used to determine the correspondences between the objects. The principle of this technique is to calculate the similarity-based probability between an object O_i^t in the new frame t with all existing objects $O_{1,2,\ldots,m}^{t-1}$ in the previous frame. This procedure is iteratively done for all objects in frame t and the results of probability calculation are stored in a $m \times n$ matrix such that each column i shows the similarity-based probabilities between object i in frame t and all m objects in frame

25 The range error of the boundary pixels of an object was already discussed in chapter 2.

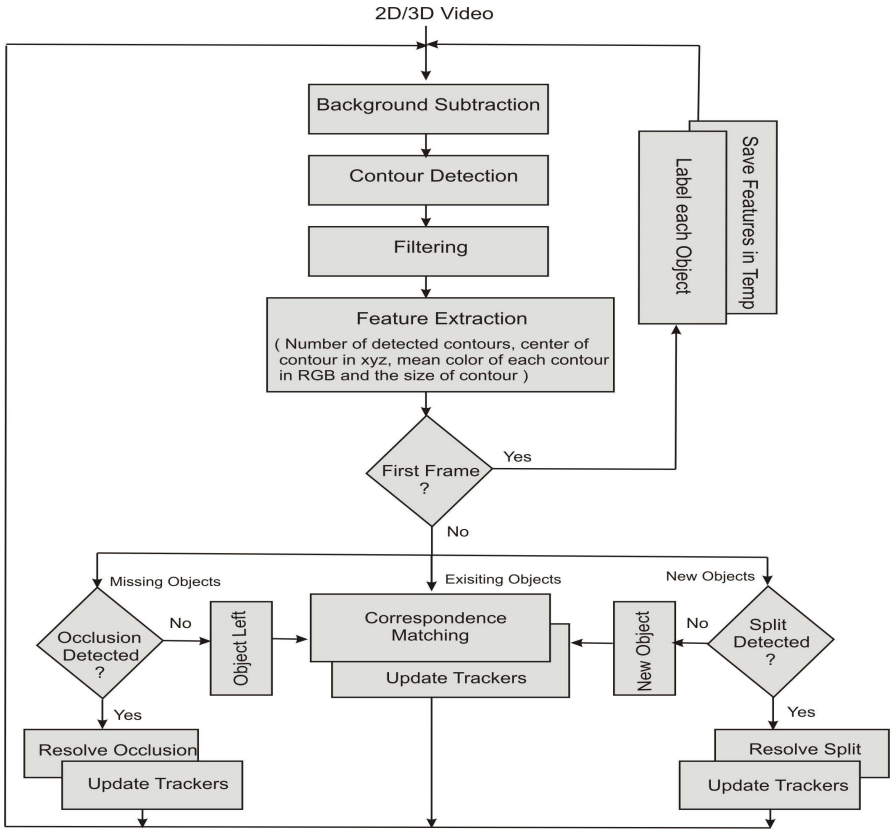

Figure 4.7: Flowchart of feature extraction and correspondence matching technique.

t-1. The maximum over the columns of the matrix is then calculated. The result of the maximum is a row vector containing the maximum element from each column with its corresponding index. Object *i* is then matched to the object *j*, where *j* represents the index of i^{th} maximum element in the row vector. If two or more objects in frame *t* are associated to one unique object in frame *t*-1, there will be similar indexes in the row vector. In such a case, the similarity-based probabilities of the objects are compared to keep the highest probability and set the other(s) to zero. Again, the graph matching technique is fulfilled to determine the correspondences and this procedure continues iteratively until every object in frame *t* takes one and only one correspondence in frame *t*-1.

2. $n < m$ means that either one or more objects have merged and made occlusion (partially or full) or one or more objects have already left the scene or they are beyond the detection zone of the 3D sensor which cannot be recognized in the 3D foreground image[26]. In the case of occlusion, an occlusion handling is performed which will be discussed in detail in the next section. In the case of object disappearance, the correspondence matching technique, same as the previous case, is directly applied to find the correspondences of the remaining objects in the new frame. After finding the matches of the objects from frame *t*-1, the disappeared object(s) and their corresponding feature set(s) are determined and removed from detected object set.

3. If $n > m$, either one or more new objects have entered the scene or two or more objects, which had already merged and made occlusion, have split. In the former case, by applying a graph matching technique two or more objects in the new frame *t* correspond to one object in frame

26 Over $7.5m$ at a frequency of $20MHz$. In other words, the objects which lie in the distance over $7.5m$ cannot be detected in the range images, although they might be seen in the 2D images. See Fig. 4.6 third image from left.

t-1. This happens because the number of objects in the new frame *t* is bigger than the number of objects in frame *t*-1. To solve this problem, the similarity-based probabilities of such objects with the unique correspondence are compared. While the object with the highest probability is selected as the match to the object in frame *t*-1, the other one(s) are labeled as new object(s) with new number(s) and they will be added to the detected object set. In the latter case, where two or more objects split, first the object split occurrence is confirmed and then the correspondence matching is performed. This will be explained more in the next section where we address the solution to the occlusion problem.

The similarity-based probability between two objects in the consecutive frames is calculated by

$$P(X_i^t | X_j^{t-1}) = \sum_{k=1}^{K} \alpha_k (1 - f(d_{ij}^k)) \tag{4.13}$$

where X_i^t is the feature set of object *i* in the frame *t*, X_j^{t-1} is the feature set of object *j* in frame *t*-1, $f(d_{ij}^k)$ is the normalized distance function between two similar features *k* for objects *i* and *j*. *K* represents the total number of features in the feature set and α_k is the weighting factor of the probability for feature *k,* which defines the importance of feature *k* in making similarity-based probability between two objects.

The values of α_k should be selected such that

$$\sum_{k=1}^{K} \alpha_k = 1, \quad 0 \le \alpha_k \le 1 \ . \tag{4.14}$$

The equation 4.13 can be understood by assuming two identical objects where the distance between their features is zero and therefore the normalized similarity-based probability between these two objects is 1 or the other way around, by taking two dissimilar objects with the maximum normalized distance of 1 which yields the normalized similarity-based probability of 0.

We use multiple features, extracted from 2D and 3D images to reduce the error of correspondence matching. For example, if we just consider the probability density function of the color of the object as a feature, there might be cases where two different objects with similar color probability density functions cannot be distinguished. This is also true for other single features like distance or size of the object. Thus, the fusion of all these features in making the similarity-based probability makes the error of matching to the lowest rate. On the other hand, the number of different features should not be so big such that their calculation as well as their fusion become computationally expensive.

The normalized distance function between two objects *i* and *j* based on all the features used in our work is formulated by

$$f(d_{ij}) = d_{ij}^z + d_{ij}^{xy} + d_{ij}^{RGB} + d_{ij}^a \tag{4.15}$$

where d_{ij}^z is the distance between objects *i* and *j* in *z* direction and it is derived directly from range images.

d_{ij}^{xy} is the Euclidean distance between the center of mass of two objects *i* and *j* which is formulated as follows

$$d_{ij}^{xy} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.16}$$

in which (x_i, y_i) and (x_j, y_j) are the center of mass points of two objects *i* and *j* respectively derived

from bounding box of contours.

In equation 4.15, d_{ij}^{RGB} stands for the metric distance between the color of objects *i* and *j* in RGB space which can be described by

$$d_{ij}^{RGB} = \sqrt{(\Delta R)^2 + (\Delta G)^2 + (\Delta B)^2} \tag{4.17}$$

where ΔR , ΔG and ΔB represent the difference between red, green and blue channels of two objects *i* and *j* in the consecutive frames.

Finally d_{ij}^{a} in equation 4.15 represents the distance between the area size of two objects.

As already mentioned, for each feature a weighting factor is defined which can take the value between 0 and 1. For the aforementioned discussed features, the weighting factors α_1, α_2, α_3 and α_4 are defined respectively and they should be set such that the correspondence matching gets the minimum error rate.

To analyze the results of the correspondence matching technique over the weighting factors, we have recorded four video sequences including about 3000 images. These videos have been recorded in different environments with different lighting conditions, with people walking around (mostly two persons) and appearing at close, medium and far distances to the camera. In these video sequences, there are a variety of cases where two persons appear at the same distance to the camera, they have same area size, or even they appear in very similar colored clothes. Also, all three general cases discussed before ($n = m$ existing objects, $n < m$ occlusion or object disappearance and $n > m$ object split or new object intrusion) have been considered in these videos.

In order to evaluate the results, we have defined two evaluation parameters: correspondence matching accuracy and correspondence matching error rate. While the former refers to the number of correctly matched frames to the total number of frames in the video sequences, the latter defines the number of wrongly matched frames in relation to the total number of frames. For each video set, we have manually determined the correspondences between detected objects over the frames in order to create ground truth data. By having the ground truth images and changing the values for α_1 (corresponding to the distance feature), α_2 (corresponding to the center of mass feature), α_3 (corresponding to the color feature) and α_4 (corresponding to the area size feature) we can calculate the correspondence matching accuracy and error rate for each set of α. Since setting α's for all possible values and calculating the results is an extreme costly task, we have limited the values of α to seven sets between the main three following *P* sets:

1. $P_1=\{\alpha_1=\alpha_2=\alpha_4=0,\ \alpha_3=1\}$, in this case the similarity-based probability is made by only using the color feature. In other words, correspondence matching is done just based on 2D features.
2. $P_4=\{\alpha_1=\alpha_2=\alpha_3=\alpha_4=0.25\}$, this α set represents contribution of 2D/3D features with equal weights in making a similarity-based probability. Note that for occlusion handling the matching is done by just distance feature (see section 4.2.2).
3. $P_7=\{\alpha_2=\alpha_3=\alpha_4=0,\ \alpha_1=1\}$, in this set the distance feature extracted from 3D range image is the only contributor in similarity-based probability function.

The results of the correspondence matching over mentioned *P*'s are shown in Fig. 4.8. As it can be seen from these graphs, setting the α set to P_4, where both 2D and 3D features contribute in probability function, yields the best result. While setting the α set to P_1, where just 2D color feature is used, has the lowest performance, adjusting α set to P_7, where distance feature is only contributor in probability function, has better result. Here, we just present how the fusion of features can improve the results of correspondence matching dramatically and the results can neither be generalized for all applications nor compared as the final results of tracking. In fact, setting the weighting factors is heavily dependent on the application domain. Likewise, the presented results might even be improved by finding a better α set which can be derived by a kind of automatic program.

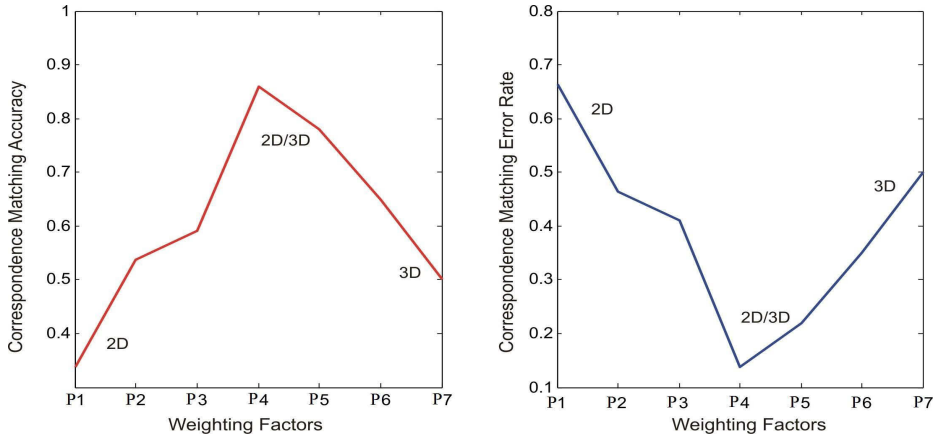

Figure 4.8: Performance of the correspondence matching technique over different weighting factors (P sets). P_1 represents the weighting factor set where just 2D features contribute in making the similarity-based probability. P_4 contains the weighting factors, where 2D/3D features contribute to the correspondence matching and P_7 represents the weighting factor set where 3D features contribute in making the similarity-based probability.

4.2.2 Occlusion Handling

Occlusion is one of the most crucial problems in object tracking. It is one of the constraints in object tracking which in some works is not considered at all, or in some others it is minimized by applying some assumptions. However, there are still some novel approaches which address the solution to this problem cited in [93], [83], [78], [71], [75] and [89]. One of the principal techniques to handle occlusion is to use multiple cameras where the depth information can be used to overcome the occlusion problem in a multi-object tracking problem.

In this section we will discuss a heuristic technique for handling occlusion using 2D/3D data. The other standard techniques will not be discussed in this work and the reader is referred to aforementioned references.

The proposed technique consists of three steps to handle occlusion: i) Object occlusion detection step to detect the start of occlusion during the tracking process, ii) Object split detection step to detect the end of occlusion and iii) Correspondence matching to determine the correspondences after occlusion.

Object Occlusion Detection

An occlusion normally starts partially when two (or more) objects merge and can continue to become a full occlusion problem. When one of the detected objects is occluded by the other one, that object and its corresponding features get lost in the scene. Thus, the first step in handling an occlusion problem is to detect whether it has been occurred or not. To do that, in each frame, after background subtraction and detecting the objects of interest, the number of objects of interest and their corresponding size are calculated and recorded. A partial occlusion in frame *t* happens if the following conditions are met:

- The number of detected objects *n* in the new frame *t* is smaller than the number of detected objects *m* in frame *t*-1. This is because two merged objects in the new frame are detected as one.
- The size of one of detected objects changes, i.e., the size increase is bigger than a predefined threshold.

Figure 4.9: Left: Before occlusion, the number of detected objects= 3. Middle: Partial occlusion happening, number of detected objects= 2 and the change of area size of one of the objects (occluded objects) is bigger than the predefined threshold. Right: After occlusion, two objects split, the number of detected objects increases to 3 and the size of detected objects (split ones) reduce dramatically.

An example of occlusion occurring in low resolution range data is illustrated in Fig. 4.9, where two persons pass by and occlude. In this work, when an occlusion is detected, the two merged objects are identified with a default black bounding box and the box is labeled with "Occlusion" word.

There are some techniques to recover the missing object regions during occlusion like the shape prior technique proposed by Yilmaz et al. [89], and the appearance models approach presented by Senior et al. [78]. In this work, we do not focus on this specific part of the occlusion problem in detail. However, one of the possible solutions is to use 3D range data of a merged object and apply clustering technique to classify the pixels of the merged object into two clusters which represent two merged objects. For this purpose, we have used the K-means technique which was already discussed in section 3.2.3. Some results of segmentation of merged objects during occlusion in order to recover the missing parts are shown in Fig. 4.10.

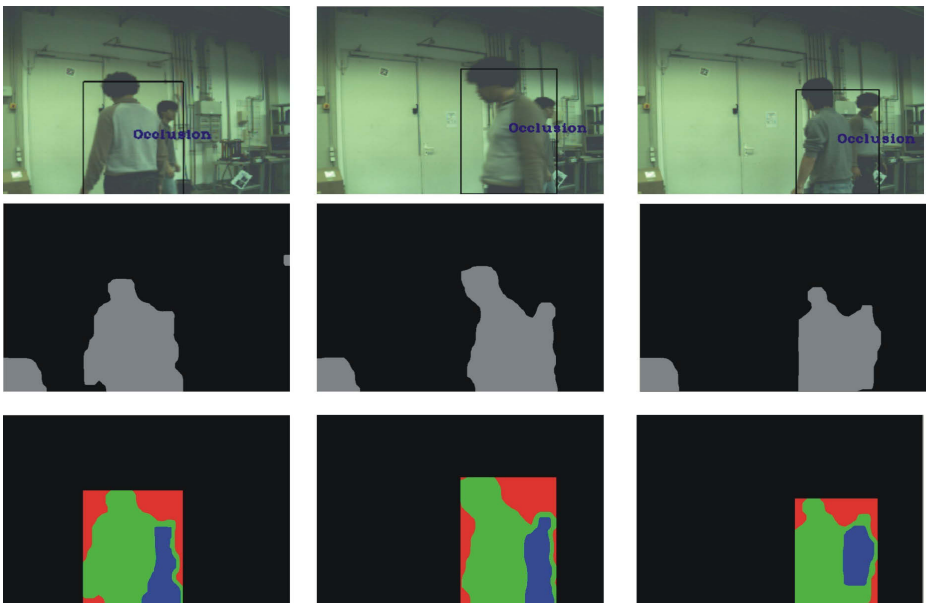

Figure 4.10: Segmentation of merged objects during partial occlusion using range data and based on the K-means clustering technique. Top: 2D images, occlusion has been detected and labeled. Middle: Foreground range images. Bottom: Segmentation of merged objects.

Object Split Detection

When an occlusion is detected, the occluded objects are labeled and the tracker takes the two occluded object as one object with a default specific number. The occlusion is resolved when two occluded objects split. Therefore, after occlusion the tracker starts checking in every new frame if an object split happens. An object split occurs if the following conditions are met:

- The number of detected objects *n* in the new frame *t* is bigger than the number of detected objects *m* in frame *t*-1.
- The size of one of the objects changes, i.e., the reduction in size is bigger than a predefined threshold.

An example of object split after occlusion is shown in Fig. 4.9. When an object split is detected, the occlusion flag in the tracking is switched off and object matching function is called to determine the correspondences.

Object Matching

The last step of occlusion handling is to match the split objects after occlusion to their corresponding objects before occlusion. In order to do that, the similarity-based probability between the objects, as discussed in the previous section, is used. The conceptual graph of object matching for the three cases, consists of before, within and after occlusion, is illustrated in Fig. 4.11.

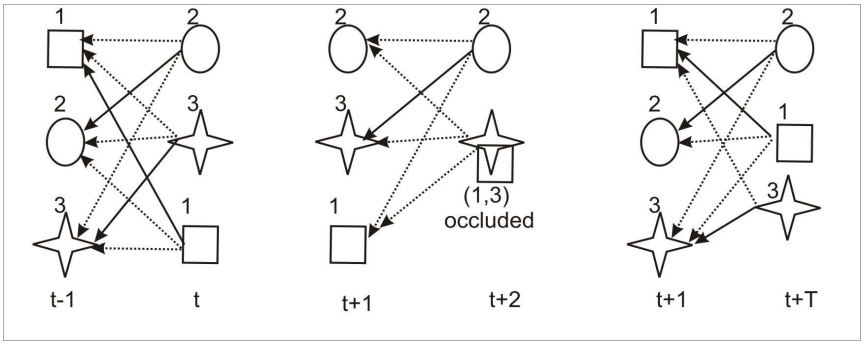

Figure 4.11: Correspondence matching. Left: Existing objects (n = m). Middle: Object occlusion (n < m). Right: Object split (n > m). Note that after two objects split in frame t+T, they are matched to the objects before occlusion in frame t+1.

Since in the first frame immediately after the object split, two objects lie in absolutely different distances to the camera (*z* direction), the range information which is directly derived from 3D image is a very strong feature to create a good similarity-based probability. Therefore, for object matching in

Figure 4.12: Occlusion handling. Left: Before occlusion. Middle: Occlusion detected and labeled. Right: After occlusion.

the first frame after occlusion we use just the range feature as the contributor for making the probability function. An example of occlusion handling is shown in Fig. 4.12.

4.2.3 Tracking with Classifiers

Another approach for object identification in a tracking process is to use a classifier directly to distinguish between different detected objects. In fact, if the classification method is fast enough to operate at the image acquisition frame rate, it can be directly used for tracking as well. For example, supervised learning techniques such as Support Vector Machines (SVM) and AdaBoost can be directly employed to classify the objects in each frame because they are fast techniques which can work at real time rates for many applications.

Object identification is performed by applying a supervised classification function (model) to each frame which associates each of the detected objects in the scene to a class. The classification model is generated from training data. Therefore, a tracking system which employs a classifier for object identification requires storing many images of different views of the objects in advance. A feature extraction technique is then applied on the one hand to derive a representative feature set for each object and on the other hand to reduce the dimension of the data. In the last step, each derived feature set, corresponding to an object, is manually associated to a class label and they altogether compose the training data set. Finally, the training data set is used to learn a function which can map each new object to a class label.

In this section, we describe tracking with classifier more in detail by applying a supervised classifier based on AdaBoost to 2D/3D videos in order to detect and track the hand. The details of the classifier was already discussed in section 3.3.4.

Hand Detection and Tracking in 2D/3D Videos

A general overview of the algorithm is shown in Fig. 4.13. The inputs of the algorithm are the low resolution range and modulation amplitude data from the TOF sensor and high resolution 2D color image from the CMOS sensor, taken by the MultiCam. In the first step, the Volume of Interest (VOI) is extracted from the range image. VOI, which has already been specified by the user is the volume where the user assumes to stand in and operate. For example, in human-robot interaction (HRI) it is the volume in where the user communicates with the robot. This volume which is specified in x,y and z direction in the world coordinate system is projected onto the 3D image. The pixels out of the volume in the range and modulation amplitude images are then filtered out. This makes the detection of moving objects in the cluttered background much simpler because the objects outside VOI do not appear in the image. In the next step, the filtered range and modulation data are fused as the input features for a supervised clustering technique to segment the objects in the volume of interest. As 3D image has a low resolution, the segmentation is done very fast. The segmented range image is then mapped to the 2D color image. Due to the monocular setup of the MultiCam, mapping from 3D range data to the corresponding 2D color data is trivial, and it does not need any extra calibration or registration techniques. This consequently makes the segmentation of 2D color image fast enough for our application. In the next step, the mapped color image is plugged into the cascade of AdaBoost to find the region of the hand in the image. Since AdaBoost sometimes finds only a part of the hand, the found region is post-processed to extract the complete hand and eliminate the non-connected components in that region. The centroid of the extracted hand region is recorded as the position of the hand in that frame, and the posture of the hand (palm or fist) is classified using a fast heuristic method which was already discussed in section 3.1.3.

To search for the object in the whole image one can move the search window across the image and check every location using the classifier. As already discussed the AdaBoost classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which

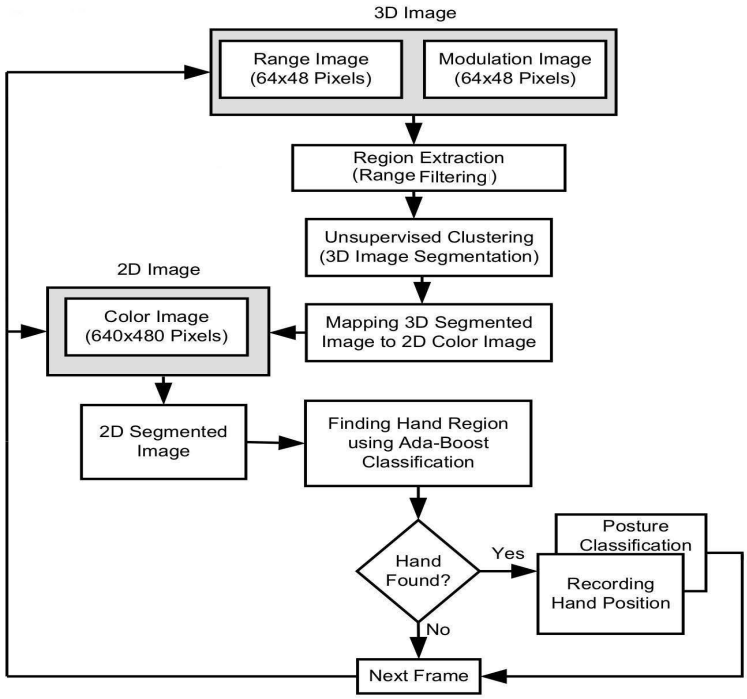

Figure 4.13: Block diagram of the hand detection algorithm.

is more efficient than resizing the image itself [2]. However, in our work, as we have the range information from 3D image, the size of the hand (desired object) can be estimated and therefore we can set the initial size of search window without starting from a very small kernel size. This reduces the computational time for finding the hand in the image [37].

Evaluation of Hand Detection Results

We have tested the performance of our algorithm in a robot control application which will be explained in chapter 5. For training of the classifier, we took 1037 positive hand images from 7 people, and 1269 negative images from non-hand objects in our lab environment. Using OpenCV [3] we trained our classifier with 20 stages and a window size of 32×32 pixels.

In order to analyze the performance of the system, we recorded the results of the hand detection from our GUI in video format while different users were commanding the robot. Likewise, we moved the camera and took the videos from the environment where non-hand objects could be observed. These videos are labeled as "positive" and "negative" data. While positive stands for the hand, the negative represents the non-hand objects from the background which can confuse the classifier. The data were acquired using a PC with dual core 2.4 GHz CPU. The exposure time for 3D sensor was set at $2\,ms$ while for 2D sensor it was about $10\,ms$. The confusion matrix derived from these videos with 2857 hand images and 2717 non-hand images is shown in Table 4.1. As we can calculate from this table, the system has a hit rate of 0.921, false positive rate of 0.032 and the recognition accuracy of 94.4%. Some examples of hand detection results are shown in Fig. 4.14.

Table 4.1: Confusion matrix for hand detection system.

	Hand	Non-Hand
Hand	2633	87
Non-Hand	224	2630
Sum	2857	2717

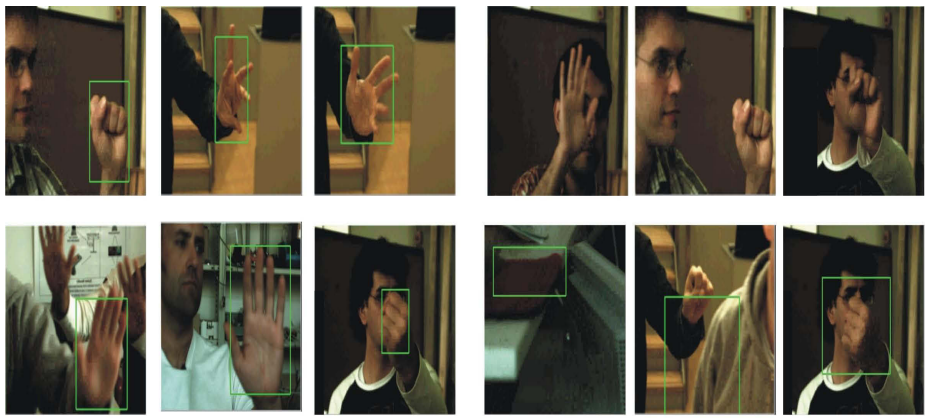

Figure 4.14: Some results of hand detection. Left: Example of correctly detected images (True Positive). Right: Example of wrongly detected images (First row: missed hand - False Negative. Second row: misclassified-False Positive).

4.3 Probabilistic Object Tracking

The aim of any kind of object tracking technique is to determine the location of the object of interest across successive frames. In all non-probabilistic tracking approaches, like object representation and identification which was presented in the previous section, the determination of the location of the object is obtained directly from the observations. Each observation is acquired from the sensor's outputs. However, measurements taken from visual sensors contain noises. For example, in this work we have already discussed some of them in the 3D time of flight images in chapter 2. Although such noises in the measurements are always tried to be eliminated using some preprocessing techniques in earlier stages, they are still inevitable and therefore any observation is corrupted to some extent by noise which consequently creates inaccuracies in object tracking results. Moreover, issues arising from complex object motions like maneuvering, appearance changing, occlusion and sudden disappearance of observations[27] make the deterministic trackers inefficient.

Probabilistic object tracking methods address a solution to these problems. They, as opposed to object representation and identification, are top-down tracking approaches. In other words, they first estimate the position of the desired object jointly from the history of observations, obtained from previous frames, with the model of the system (dynamic of object). This estimation is then corrected with a new measurement in the current frame. In fact, if the dynamic of the moving object is known, which is usually assumed as known, the position of the object (state space) in time *t* can be predicted and it can

27 For example, when an object gets out of the field of view of the camera suddenly for a short while and then it appears again.

be combined with the current measurement at time *t* to get a robust and more accurate result.

In this section we briefly review two widely used probabilistic approaches for object tracking and then we will show how these approaches have been used in our work for object tracking using 2D/3D images.

4.3.1 The Kalman Filter

The Kalman filter is the most widely used technique in probabilistic tracking approaches. It is a stochastic, recursive data processing algorithm which tries to obtain an optimal estimate of state of the system from noisy data with the minimum error rate.

This section reviews the basic idea behind the Kalman filter and explains how it has been used in our work. For a much more in depth understanding about the principle of the Kalman filter, the reader is referred to [10], [108], [11] and [105].

Assume that we want to know the position of an object in a video data with the state vector of $s \in \mathbb{R}^n$ which consists of three position variables *x*, *y* and *z* and their velocities v_x , v_y and v_z respectively. The Kalman filter addresses the solution to this problem in two steps: Prediction (time update) and Correction (measurement update). In the prediction step, the current state and error covariance are projected forward to obtain the *a priori* estimates for the next time step. In the correction step, the new measurement at time *t* is incorporated into the *a priori* estimate to obtain an improved *a posteriori* estimate [108].

Estimation of the Kalman filter is basically performed under three following assumptions [11]:

- The system has a linear dynamic.
- System and measurement noises are "white".
- System and measurement noises are of Gaussian forms.

Based on these assumptions the state of the object of interest can be formulated by a linear difference equation as follows[28]

$$s_t = A\, s_{t-1} + w_t \tag{4.18}$$

where A represents the state transition matrix (transfer matrix) and w_t is associated with the white noise process with the Gaussian distribution and the known covariance Q_t .

On the other hand, the measurements z_t are obtained from an observation which might or might not be a direct measurement of the state and therefore it can be formulated by

$$z_t = H\, s_t + v_t \tag{4.19}$$

where H represents the measurement matrix and v_t is the associated measurement noise with the Gaussian distribution and the covariance matrix R_t .

Now, all we need to do is to consider the aforementioned assumptions and use the above equations to perform the Kalman filter in two mentioned steps as follows:

- **Prediction (Time Update)**

In this step, first the *a priori* estimate (prediction) of the state $\hat{s}_t^-$ [29] is computed from the state space model by

28 Note that the state of a system in general is written as $s_t = As_{t-1} + Bu_t + w_t$ where the term Bu_t is associated to the control unit in the system. Since in our work, there is no external control on the state of the system (motion of an object), this term is neglected.

29 The superscript minus sign means at the time immediately prior to the new measurement.

$$\hat{s}_t^- = A\,\hat{s}_{t-1}^+ + w_t\,. \tag{4.20}$$

Likewise, the *a priori* estimate for the error covariance Σ_t^- of the prediction is calculated as follows

$$\Sigma_t^- = A\,\Sigma_{t-1}\,A^T + Q\,. \tag{4.21}$$

In fact, these equations make a prediction about the state of the system based on prior knowledge. Since the prediction is not accurate enough, it will be corrected in the next step by a new measurement.

- **Correction (Measurement Update)**

In this step, the Kalman gain is first computed by

$$K_t = \Sigma_t^- H^T (H\,\Sigma_t^- H^T + R)^{-1}\,. \tag{4.22}$$

The Kalman gain gives some information about how to weight the new information against the prior one. By having the Kalman gain and taking a new measurement z_t, the *a posteriori* state and *a posteriori* error covariance are generated as follows

$$\hat{s}_t^+ = \hat{s}_t^- + K_t(z_t - H\,\hat{s}_t^-) \tag{4.23}$$

$$\Sigma_k = (I - K_t H)\,\Sigma_t^-\,. \tag{4.24}$$

The term $(z_t - H\,\hat{s}_t^-)$ in equation 4.23 is known as innovation or measurement residual.

After each prediction and correction, the process is repeated with the previous *a posteriori* estimates to obtain new *a priori* estimates. In fact, this procedure is performed in a recursive manner such that there is no require for all previous data to be kept and reprocessed every time which is a very important point in the practical real time applications.

Now, we will discuss, how the Kalman filter is used in this work to track an object in 2D/3D videos. First of all, we need to detect the object of interest in the video to activate the tracking system and provide it with an initial state. In fact, as soon as the desired object appears in the scene, the detection mechanism should recognize it and trigger the Kalman filter in order to track its motion. We have already discussed object recognition techniques using 2D/3D images in the previous chapter as well as in the previous section where an object detection (classifier) is performed in each frame to detect the object of interest and locate its position. For example, an AdaBoost classifier can be applied to find the object of interest, or some appearance based techniques, like contour detection can be performed to detect the desired object. However, in our work we have used a simple object detection mechanism consisting of background subtraction, foreground segmentation and contour detection to find the object of interest in the image. This is exactly the same as what we presented in non-probabilistic object tracking in the previous section. The object of interest is defined with a point corresponding to its center of mass. Therefore, the state of the object is summarized by three position variables in x, y and z directions with their corresponding velocities as v_x, v_y and v_z. In the first frame where the object is detected, the state is initialized with the position of the object (x, y, z) and with zero velocities ($v_x = v_y = v_z = 0$). However, we assume that afterward the object moves with a constant velocity. Therefore, we neglect the acceleration in the state space equations. Thus, the state vector s_t and the state transition matrix A in equation 4.18 can be written as follows

$$s_t = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.25}$$

where Δt corresponds to the time step between two frames in a video and it is determined by the frame rate of the 2D/3D camera system which is dependent on the observation time given in equation 4.11.

On the other hand, since the velocity of an object cannot always be constant in our applications, the above assumption is not true which consequently causes some errors in the prediction of the state from the dynamics of the object. Therefore, we assign process noise w_t in equation 4.18 with covariance matrix of Q_t to reflect this error.

To do a new measurement for the correction step in the Kalman filter, like what we did in the previous section, first we subtract the background and then apply contour detection to find the object of interest. In the next step, the center of mass of the found object in *x*, *y* and *z* is calculated and considered as the new measurement. Therefore, the measurement z_t and the measurement matrix *H* in equation 4.19 can be formulated as

$$z_t = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4.26}$$

This is actually the simplest case where there is just one single object moving in the scene. In the case of multi-object tracking, for each object a Kalman filter step is performed to predict its state. In the next pace, a correspondence matching approach similar to what we presented in the last section is fulfilled in order to correspond the new measurements to the predicted data derived from the Kalman filters.

For the cases with the nonlinear dynamics or measurement relations, an extended Kalman filter [108] can be used to handle these nonlinearities. Since in our applications it turned out that the Kalman filter is still useful we have excluded the extended Kalman filter from our work.

4.3.2 The CONDENSATION Algorithm

The main limitation of the Kalman filter in visual tracking is the assumption that the state's probability distribution is unimodal Gaussian. Therefore, the Kalman filter cannot represent multiple hypotheses simultaneously. For example, in cluttered scenes there are typically more competing observations which lead to a non-Gaussian state density [20], or in the cases where the object of interest might stop, reverse the direction or continue moving at the same speed, the Kalman filter is unable to represent such multimodal distributions. The Conditional Density Propagation (CONDENSATION) algorithm [20] which is based on particle filters is designed to address the solution to such problems.

The CONDENSATION algorithm is an iterative algorithm to calculate the *a posteriori* density $f_{x(t)|Z(t)}(\xi|Z_t)$, where the *x(t)* represents the state vector at time *t* with the dummy vector ξ [30] and *Z(t)*

30 The notations are taken from [10] and [11].

is the history of all observations $\{z_1, z_2, \ldots, z_t\}$ up to time *t*. An iteration step of the CONDENSATION algorithm at time *t* starts with a sample set $\{s_t^{(n)} : n = 1, \ldots, N\}$ with weights $\pi_t^{(n)}$ (sampling probability). This sample set represents the *a posteriori* density $f_{x(t-1)|Z(t-1)}(\xi_{t-1}|Z_{t-1})$ from the previous time step. The sample set *s* is propagated to obtain a new sample set *s′* according to the system model $f_{x(t)/x(t-1)}(\xi|x_{t-1})$. The sample set *s′* represents the *a priori* density $f_{x(t)|Z(t-1)}(\xi|Z_{t-1})$. Using the observation density $f_{z(t)|x(t)}(\zeta|\xi)$ and by applying the factored sampling a new sample set *s″* is derived from *s′* which represents the new *a posteriori* density $f_{x(t)/Z(t)}(\xi|Z_t)$. A more detailed explanation of this algorithm can be found in [20], [94], [91] and [48].

For the implementation of the CONDENSATION algorithm in our work, in the first frame, the regions of interest which might represent the position of the desired objects are found in 2D/3D video. This is done by applying background subtraction and contour detection as we discussed before. In the next step, we initialize 100 3D-particles, with *x*, *y* and *z* elements, randomly inside the detected contours with the equal weights, i.e., the particles (hypotheses) are associated to the contours which represent the object of interest potentially.

Based on a new measurement, the weights of all particles are updated. This is done by looking at the distance value (3D data) of that particle in the 3D segmented image. Finally, the particles are updated in a re-sampling process in which a new set of samples are generated based on the computed confidences.

4.3.3 Evaluation of Results

In this section, we evaluate some results of the probabilistic tracking approaches using the discussed Kalman filter and the CONDENSATION techniques. To do that, we have selected three scenarios which are recorded as 2D/3D videos, in each of them a person is moving in the scene. To evaluate the results, we find the trajectory of the person in *x* and *y* directions in the image coordinate system and label it as the ground truth. In fact, in the ground truth, the position of the person in each time stamp is derived manually by identifying the center of the bounding box to the person at that time. To compare the results of the tracking techniques we plot the trajectory of the object obtained from each tracker against the ground truth and calculate the error rate.

In the first scenario, a person appears in the scene from the left, follows a path to the right, stops in a distance close to the camera and then returns back the same path and finally leaves the scene. The person is tracked using both the Kalman filter and the CONDENSATION technique. Some of the results of tracking are shown in Fig. 4.15. For the Kalman filter, both the prediction and correction results are depicted as the red and yellow points in the image respectively. From these samples, it can be seen explicitly that how the predicted point in each image (red point) deviates from the real center of mass and how it has been corrected (yellow point) using a new measurement in the Kalman process. For the CONDENSATION, the person is represented with the updated particles in each frame which are shown as the green points in the image. To derive the trajectory, we calculate the mean[31] of all particles and consider it as the result of the tracker. The mean point of the particles is shown as a yellow point in the image results.

The ground truth trajectory of the person with the found trajectories from both trackers are shown in Fig. 4.16. As we can conclude from the results, while the CONDENSATION technique outperforms the Kalman filter in the tracking of the person in *x* direction (main direction of the movement), the trajectory found by the CONDENSATION technique in *y* direction is not as satisfactory as in the *x* direction. To compare the results of the trackers quantitatively we calculate the error rate of the tracking as the mean Euclidean distance between the tracked points and the ground truth points over all the images in the video. This error rate is then presented in percentage by dividing it by the width and height of the image for *x* and *y* directions respectively.

31 The mean of particles is a linear average of positions of the particles with equivalent weights.

Figure 4.15: Some results of person tracking using the Kalman filter and the CONDENSATION technique (frame number 10, 35, 60, 150, 185, 215, 225, 235). First two rows represent the Kalman filter results. Red point shows the predicted (intermediate) position and yellow shows the corrected point (final result). Last two rows represent the CONDENSATION result where the particles are depicted in green and the mean of all (final result) is in yellow.

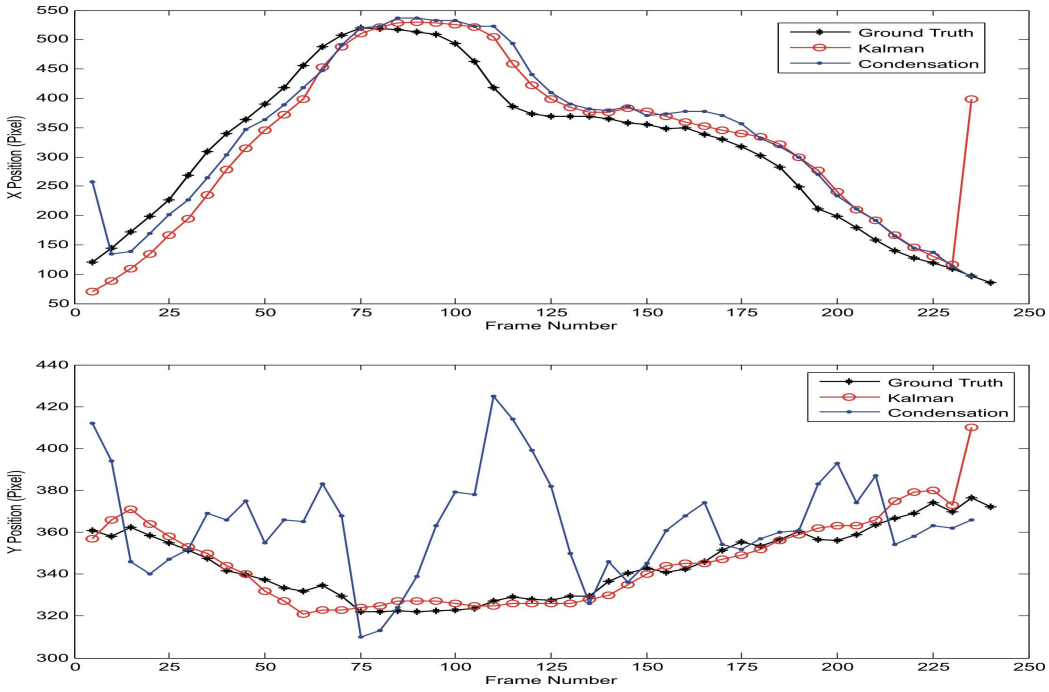

Figure 4.16: Trajectory of the person in the image coordinate system. Black: Ground truth, Red: Kalman filter, Blue: CONDENSATION. Top: Trajectory in x direction. Bottom: Trajectory in y direction.

In this case, the Kalman filter has an error rate of 6.54% in the *x* direction while the CONDENSATION gets the error of 5.3%, but on the other hand, the Kalman filter performs much better in *y* direction with the error rate of 1.01%, whereas the CONDENSATION has the error rate of 6.54%.

In the second scenario, contrary to the first scenario, a person enters the scene from the camera side and moves away from the camera to reach the maximum 3D visible distance[32]. He returns the same path back to leave the scene. Some of the images of this scenario with the tracking results are shown in Fig. 4.17.

Same as the previous case we compare the results of the trajectories found by the trackers in *x* and *y* directions in the image coordinate system which are shown in Fig. 4.18. As can be seen from these results, in the *x* direction both trackers perform well except in the couple of first frames for the CONDENSATION which is because of wrong initialization of the filter. Excluding the first two frames from the results, the CONDENSATION has the error rate of 2.71% while the Kalman filter performs with the error rate of 3.71%.

In *y* direction, the Kalman filter, same as the previous case, outperforms the CONDENSATION with the error rate of 4.52% which is about 6.87% for the CONDENSATION technique by excluding the first two frames from the results and 8.38% by including these two frames.

Figure 4.17: Some results of person tracking using the Kalman filter and the CONDENSATION technique (frame number 55, 80, 95, 150, 195, 215, 275, 295). First two rows represent the Kalman filter results. Red point shows the predicted (intermediate) position and yellow shows the corrected point (final result). Last two rows represent the CONDENSATION result where the particles are depicted in green and the mean of all (final result) is in yellow.

In the last case, we have considered a challenging scenario where a person first stands in the middle of the scene and then he starts moving towards the camera and stops in a fixed distance to the camera, then he bows down, waving the hands, jumping up and then moves again. This scenario represents a multi-modal movement (multi hypotheses) which can be difficult for a normal Kalman filter.

32 It is usually 7.5 m at frequency of 20 MHz.

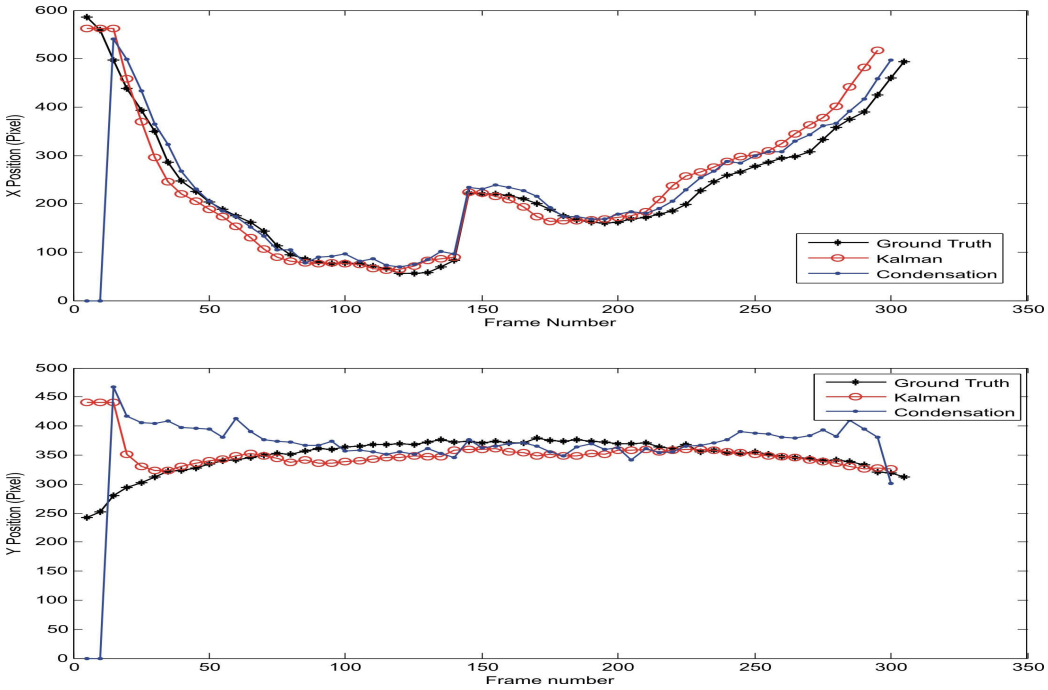

Figure 4.18: Trajectory of the person in the image coordinate system. Black: Ground truth, Red: Kalman filter, Blue: CONDENSATION. Top: Trajectory in x direction. Bottom: Trajectory in y direction.

Figure 4.19: Some results of person tracking using the Kalman filter and the CONDENSATION technique (frame number 25, 40, 70, 85, 120, 230, 280, 355). First two rows represent the Kalman filter results. Red point shows the predicted (intermediate) position and yellow shows the corrected point (final result). Last two rows represent the CONDENSATION result where the particles are depicted in green and the mean of all (final result) is in yellow.

However, we have applied both the Kalman filter and the CONDENSATION technique to track the person. Some results of this scenario are shown in Fig. 4.19. Likewise, the results of the trajectories obtained by the trackers are depicted in Fig. 4.20. As it can be seen from these results, in this case the CONDENSATION technique outperforms the Kalman filter in both *x* and *y* directions.

In *x* direction, while the Kalman filter has a higher error rate of 7.58%, the CONDENSATION technique results in just the error rate of 3.53%. In *y* direction, the CONDENSATION yields the error rate of 5.91%, whereas the Kalman filter has the error rate of 9.22%.

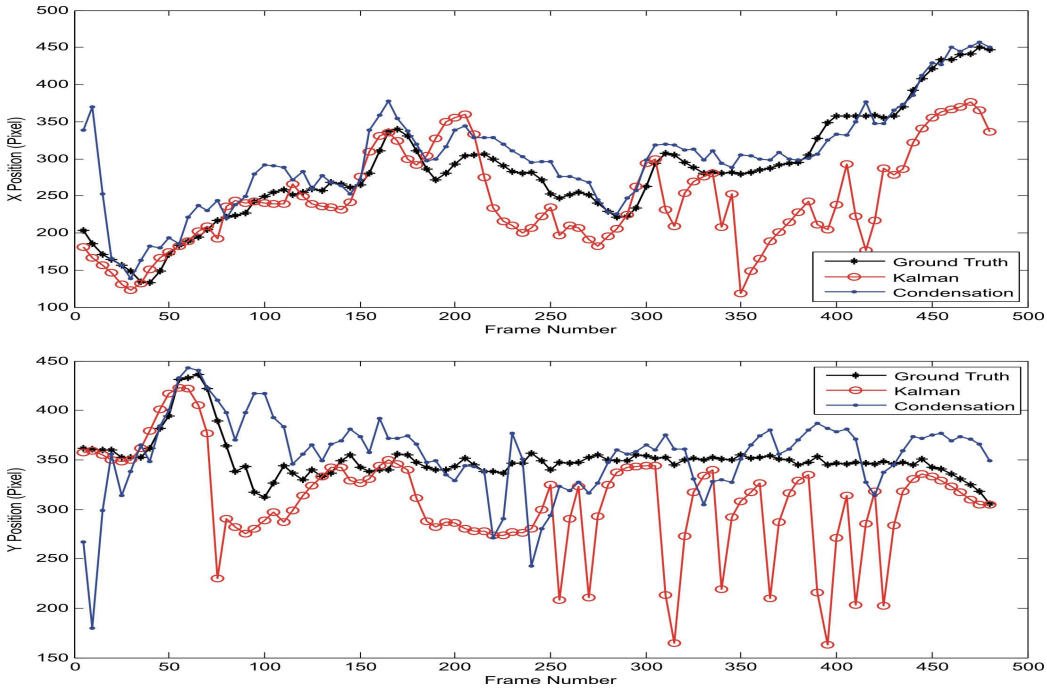

Figure 4.20: Trajectory of the person in the image coordinate system. Black: Ground truth, Red: Kalman filter, Blue: CONDENSATION. Top: Trajectory in x direction. Bottom: Trajectory in y direction.

As seen, in the first two examples the person does not have so much movement in *y* direction, like jumping up or bowing down movements. Therefore, the results of the Kalman filter in *y* direction is better than the results of the CONDENSATION. In other words, the tracking in *y* direction can be performed with a single Gaussian motion model better than by using a multimodal non-Gaussian one.

In the last example, in which the person sometimes jumps up and bows down, he has a multimodal movement in *y* direction. Therefore, the CONDENSATION which has a multimodal concept yields better result than a unimodal Kalman filter.

In all three examples, the CONDENSATION outperforms the Kalman filter in tracking the person in *x* direction.

4.4 Summary

This chapter discussed object tracking using 2D/3D images. In the first step, a dynamic scene analysis is performed, consists of background subtraction and real time analysis. We have introduced a Mixture of Gaussian (MoG) and range thresholding, as background subtraction techniques, in 2D and 3D image modalities respectively. The results showed that range thresholding performs better for the real time applications under varying lighting conditions.

In the next part, we have studied 2D/3D object representation and identification as a bottom-up object tracking approach. In this context, the features, which are extracted from both 2D and 3D images, are used to construct a similarity-based function. This function is used directly to establish correspondence between detected objects across the frames. In this section, a heuristic technique for handling occlusion problem using 2D/3D image data has been presented.

Another approach for object identification is to use a classifier. In fact, if the object classifier is fast enough, it can be used directly for tracking purpose as well. This has been verified in this chapter by using Viola-Jones method for hand tracking.

Finally, we presented two main probabilistic object tracking consisting of the Kalman filter and the CONDENSATION. These techniques have been tested for the person tracking in this chapter. As it was seen, while the Kalman filter is a good tracker for the objects with the single Gaussian motion model, the CONDENSATION can perform much better in tracking the objects with multimodal non-Gaussian motion models like a fast maneuvering person.

5

Applications

The only source of knowledge is experience.
Albert Einstein (1879-1955)

In the previous two chapters we studied some aspects of object recognition and tracking based on 2D/3D image data. In this chapter, we realize these aspects by putting them into practice in two major applications. These two applications have been selected such that to validate some results of the reviewed techniques in this work as well as to highlight the motivation behind our work in some real world problems.

The considered applications in this chapter are related to safety and control issues in the field of Human Robot Interaction (HRI). In the first application, the safety of the personnel working in close cooperation with an industrial robot is analyzed. In the second application, we will show how to make a natural interaction system to command an industrial robot, using hand gestures. These two applications can, in fact, complement each other in many robot based domains in which on the one hand the safety of the operator is the main concern and on the other hand a natural and effective interaction with the robot is required.

5.1 Personnel Safety in a Human Robot Cooperation

The safety of the personnel in advanced industrial automation, where the human and the robot share the workplace and cooperate closely, is a significant issue. Monitoring of the working environment is one of the main approaches which can improve the safety of the personnel in a close cooperation with a robot. In this section we will show an example of a dynamic visual monitoring system for the safety of the personnel in the cooperation with a 4-axis swivel arm robot based on 2D/3D imaging data.

5.1.1 Background

In a conventional robot based automation system, the main approach to ensure safety is to exclude the involvement of humans in the working area of the robot in order to protect them from any hazards [58]. Therefore, many sensors and equipments are employed to separate the working area of the robot from the personnel. The isolation of the operational area of the robot in industrial applications is usually performed based on the main standards for robot safety in the factories including of the American standard ANSI/RIA 15.06[33] [99] and the European standard En-775[34] [98]. In fact, these standards address the requirements for the safety of the personnel in a fence guarded system, where the robot's workplace is completely separated from the human. In other words, these standards prescribe that the safety is achieved by defining a region around the robot or the machine [13]. Some examples of the typical safety systems which isolate the defined region from the presence of the human are shown in Fig. 5.1 [26].

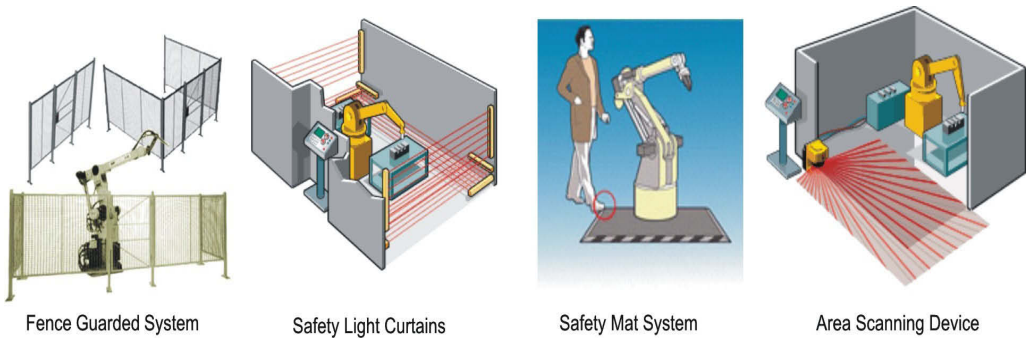

Figure 5.1: Typical safety systems.

However, in novel and advanced robot based automation, the demands for highly flexible robotic systems, in which the complementary capabilities of robots and humans are combined, are rapidly increasing [58], [13], [109], [57]. In fact, in such applications the robot and human have to share the operational space and cooperate very closely. Thus, the complete isolation of the robot from the human is impossible and consequently the aforementioned standards are no longer tenable. Likewise, as the robots take more attention to be used widely in unstructured environments like in medical, office or home, close interaction or cooperation between humans and robots becomes inevitable and therefore such standards will be precluded from performing.

In general, there are three main approaches which can be applied to mitigate the hazards during the close cooperation of the human with a robot: i) Redesign the mechanical system of the robot, ii) Warn and train the operator and iii) Control the hazard using a safeguarding system [13].

The proper mechanical design of the robot can eliminate hazards to a great extent. For example, by using viscoelastic covering or spherical and compliant joints in the mechanics of a robot, one can reduce the risk effectively [13]. Also, personnel training in order to instruct them, how to cooperate closely with a robot is a widely used option in industry. However, control the hazards during the operation is the most important part of a safety solution in a close cooperation between human and robot. In fact, by controlling the hazards during operation, one can protect the personnel from the machinery and the machines from unauthorized objects. Monitoring of the working environment, as a key component of the hazards control, can provide useful information about the potential hazards in order to avoid them from occurring and enhance the safety.

33 Standard for Industrial Robots and Robot Systems - Safety Requirements.
34 Manipulating Industrial Robots - Recommendations for Safety.

5.1.2 Dynamic Visual Monitoring

The main aim of monitoring in a robot safety system is to detect and track the humans, objects as well as the robot itself during the operation in order to avoid any collision. To achieve this aim, first the zones around the working place of the robot should be defined. Then we need to specify the risk level for each zone. The standard safety solution systems, such as optoelectronic protective devices or light grids, can monitor the predefined areas and the personnel can be protected from hazards. However, such safety systems have still some limitations which make them inefficient for using in a close man robot cooperation. Some of their limitations are as follows:

- They are unable to monitor the zones (3D volumes), they can just monitor the planes [25].
- They are not dynamically adjustable, i.e., any change to the field construction in the factory demands redesign and remounting the whole safety system.
- They always implement an emergency stop in the case of danger, independent of the level of danger. After each emergency stop, an expert must be called to return the robot to its exact position prior to the stop in order to restart. This costs time which has a negative effect, especially in the production line [25].
- They are expensive and complex in mounting.
- They usually need complementary sensors or components to provide a high safety level.

On the other hand, 3D vision systems, which can deliver range information, have the main advantage to observe the objects three dimensionally in any predefined zones and find their position precisely. This is the reason why nowadays the 3D Time of Flight cameras as well as laser scanners and stereo vision systems have gained a lot of attention to be used in this field [26], [109], [25].

In this work, using our new 2D/3D camera system, we propose a simple monitoring system, which not only provides us with the distance information but also with the high resolution intensity or color data. While the distance information is directly used to prevent any danger in terms of unauthorized entry in the predefined zones, the color or intensity data from 2D sensor are employed to detect or classify the endangered object in order to make a right decision. In other words, the 2D/3D monitoring system is used such that to fulfill the following main safety requirements:

- **Reliability:** The hazards should be detected and tracked accurately and fast.
- **Simple mount:** The monitoring system usually consists of a camera(s) and a PC. Therefore, there is no need for any complex wiring or extra physical components.
- **Dynamic adjustment:** Different zones in the field of view of the camera can be created dynamically and a risk level is specified to each zone simply in the software level without any need for mounting sensors, components or physical barriers. The zones can be eliminated, combined or take different shapes for different scenarios.
- **Non-binary functionality:** Different zones have different risk levels. Therefore, as opposed to the conventional safety systems with binary functionality[35], the output of system is a non-binary function. In other words, a decision is made based on the detected danger level in each zone which does not necessarily imply an emergency stop.

In fact, the concept of safeguarding application presented in this work is based on the observation of dynamically producible volumes with arbitrary shapes and contexts to which we refer to as risk level. An example of an observation system using two cameras with a simple context[36] is shown in Fig. 5.2. As we can see, in this case the whole area around the robot is observed with two cameras. Three

35 Binary functionality of a safety system means that either it does not detect any hazard and let the robot operate, or it detects a danger and stops the operation of the robot completely.

36 Simple context signifies simple zone labeling.

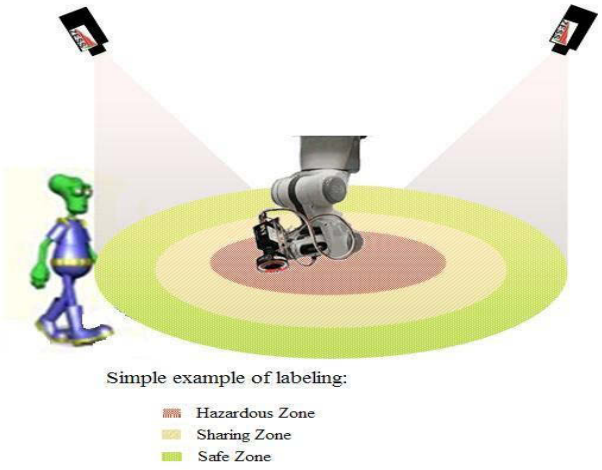

Figure 5.2: An example of dynamic safety zones [figures are taken from web-unknown source]

zones have been created around the robot with spherical shapes and for each zone a label is specified which signifies the risk level in that zone.

Safe zone is asserted to the volume where the robot and the tools attached to it do not reach to that and therefore the personnel are safe in this zone. Detection of the human or objects in this zone does not have any effect on the operation of the robot.

A shared zone is the defined volume, where the human and the robot cooperate closely. Although the presence of the personnel in this zone is allowed, this volume is the critical zone and it should be monitored precisely to avoid any collision. If neither personnel nor unauthorized objects are detected in this volume, the robot can operate at maximum velocity. If a human or an authorized object[37] is detected in the shared zone, the velocity of the robot will be reduced. In this case, while any potential danger with the low risk level[38] in this zone implies a warning signal, the one with the high risk level performs an emergency stop. Finally, detection of any unauthorized object in the shared zone implies danger with the high risk level which stops the robot from operation.

Hazardous zone is indeed a non-permissive volume around the robot in which the presence of any object or personnel is prohibited and therefore detection of anything except the robot and its attached tools in this zone implements an emergency stop.

5.1.3 Experiments and Results

In this section we present some results of our safety concept for cooperation with a Turboscara SR6, which is a 4-axis robot from Bosch Rexroth AG. The proper design of the safety system for this particular application relies upon a hazard analysis of the robot system's use, programming, and maintenance operations. According to the standard documents for the safety of the personnel who interact with the used robot, we have implemented the concept of context volume monitoring as shown in Fig. 5.3. Four zones with different context, interpreted as different levels of hazardous, have been defined and observed by the use of a 3D-TOF camera and a 2D/3D MultiCam. While the 3D camera is a C-mount camera with a large field of view, the used 2D/3D Multicam is a F-mount one with a narrow field of view. The main reason to use two camera systems in this application is to observe the whole four defined zones. As mentioned before, the zones can be defined and shaped dynamically during runtime. We label zone 1 as the hazardous volume. Zones 2 and 3 are the workplaces which can

37 An authorized object can be any known object on which the robot operates. For example, for a packaging robot a known package is an authorized object whereas a metal tool left by the personnel is an unauthorized object.

38 The risk level is identified based on the distance of the personnel or object to the robot.

be shared by the human and the robot and zone 4, since it is out of the robot's range, is a safe volume for the personnel. In fact, if there is no object or person in the hazardous zone as well as in the shared volumes, the robot works at maximum speed. If an object or person is detected in a shared zone different of that in which the robot operates in, for example object or human is in zone 2, while the robot works in zone 3, an alarm is triggered and the robot reduces its operation speed. As soon as the object or person enters the hazardous zone or is occluded by the robot in the shared zones, an emergency stop is performed and the robot motion stops immediately. In our setup, while the 3D-TOF camera monitors the four mentioned zones at high frame rates (100 images per second in our case), any motion in the shared zones are analyzed based on 2D/3D images.

It should be mentioned that the choice of four context volume is done for the sake of simplicity. One can think of an implementation using a higher amount of context zones, leading to a much more sophisticated control of the scene.

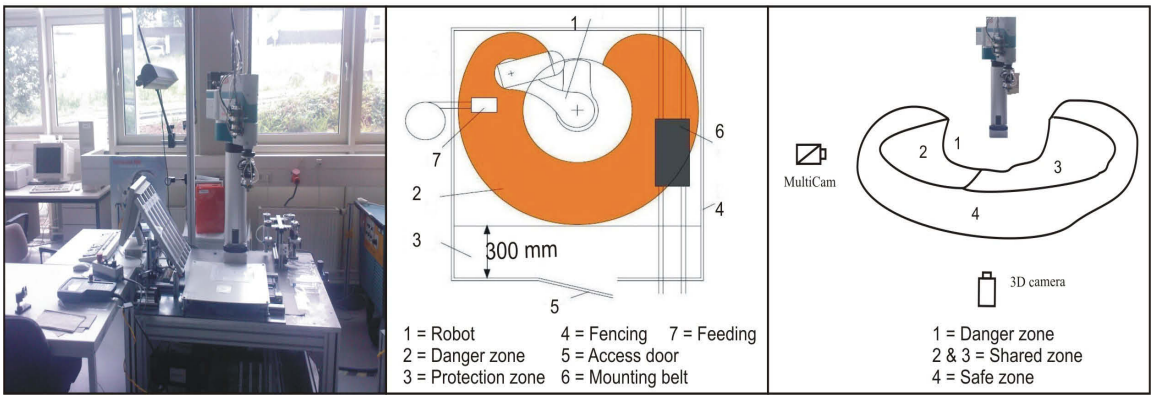

Figure 5.3: Left: Turboscara SR6, 4-axis robot which is used for our experiments. Middle: Space secured according to EN 775 given by the robot provider. Right: Dynamic safety system using MultiCam and 3D-PMD. The zones can be defined dynamically.

The moving objects in the predefined zones are detected and tracked. While the detection is done based on segmentation and classification techniques, the tracking is implemented using the CONDENSATION algorithm.

First the background image of the scene without the presence of any moving objects including the robot or personnel is captured and saved. By starting the object detection and tracking program, the background is subtracted from each frame. After the simple background subtraction, the intensity and range data are fused to be used as the input information for the segmentation algorithm. In this application, the segmentation is treated as a clustering problem which we have already discussed in section 3.2.3. Each segmented object in the image is then classified based on some heuristics. As the robot is a rigid object and it is made of metal one can take the size and modulation amplitude[39] as two explicit knowledge based features for the classification. Thus, we use the size of the robot and its corresponding modulation amplitude as important signatures to distinguish the robot from non-robot objects. After detection of the object(s), the tracking mechanism starts to estimate the position of the detected objects in each zone. As mentioned we have used the CONDENSATION algorithm, which was already discussed in the last chapter, for the tracking purpose. The observation model has been configured to track the pixels of the nearest cluster (the closest object) to each camera which are derived directly from range data. The number of particles in the CONDENSATION algorithm has been set to 50 samples and the particles are randomly initialized in a box form matching to the image size. The detection and tracking algorithms have been implemented using OpenCV library [3]. The robot

39 The objects made of metal can reflect the emitted infrared light from 3D camera much more than non-metal objects.

moves with the different velocities from the minimum 5 *cm/s* to the maximum 2 *m/s*.

In Fig. 5.4 some results of robot tracking in some instances are shown. The robot is detected and tracked after 5 iterations[40] and the presence of the human as a new moving object in the safety zone does not confuse the tracking system, i.e., the robot is detected and tracked continuously [41].

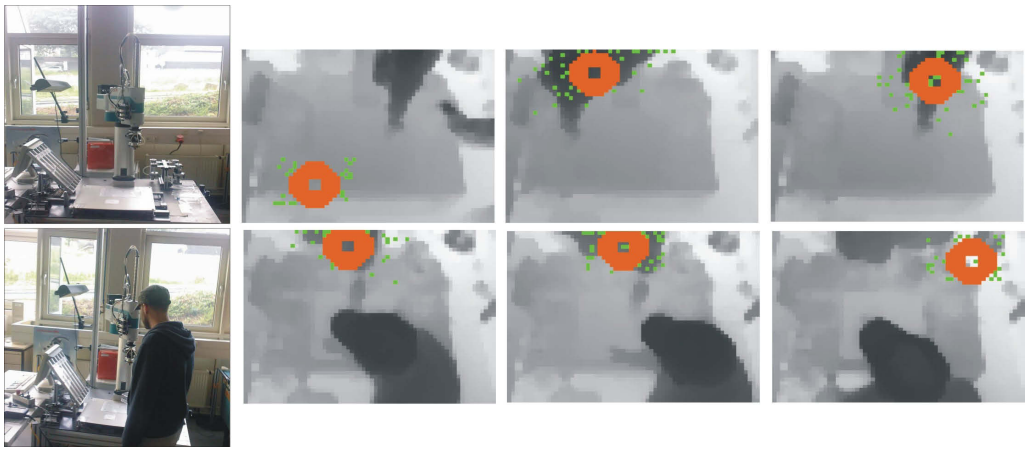

Figure 5.4: Tracking of robot in the workspace using range images. The presence of the human in the safe zone does not confuse tracking. First column: Images taken by a normal camera from the scene. Last three columns: Robot tracked in 3D range images, green points are the particles and orange circle represents the center of all particles.

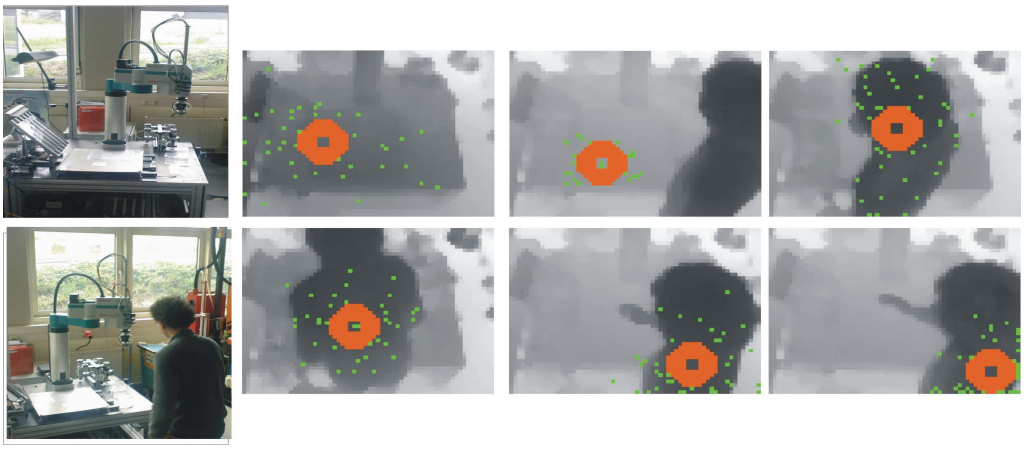

Figure 5.5: Tracking of human in the workspace using range images and without presence of the robot. First column: Images taken by a normal camera from the scene. Last three columns: Human tracked in 3D range images, green points are the particles and orange circle represents the center of all particles.

Fig. 5.5 shows some results of human tracking in the workplace where the robot is far from the human and is not detected in the image. In this case, same as the previous case, the person is detected and tracked after 5 iterations.

In Fig. 5.6, we have shown some of the tracking results using 2D/3D images of MultiCam. The results are presented in the 2D images. The first row shows some frames where the robot operates in the FOV of the camera in hazardous and shared zones and can be detected and tracked successfully. Second row shows some frames where the hand is moved close to the moving robot in the shared zone to grab

40 The time for an iteration depends on the frame rate of the camera, samples number and the processing capability of the computer.

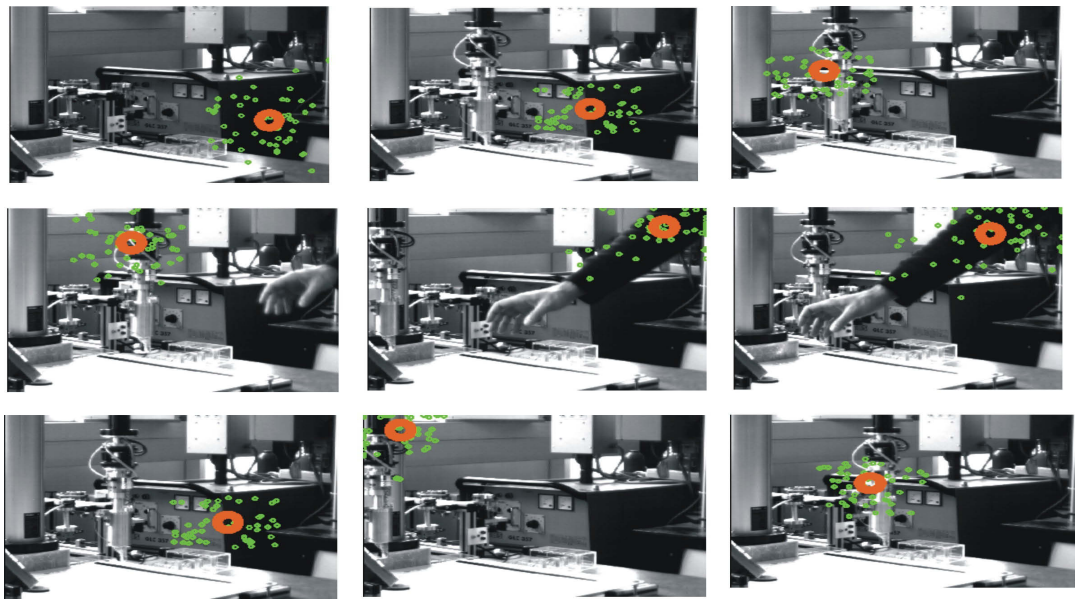

Figure 5.6: Tracking of the robot and the hand of an operator in the shared workspace using 2D/3D images, green points are particles and orange circle represents the center of the particle (final detected point).

something. In this case the hand is detected immediately and tracking is switched from the robot to the hand. This is because the hand is closer to the camera and assigned to the nearest cluster. By getting the exact position of the robot from the robot control system and having the position of the hand from tracking system, any occlusion can be avoided. After removing the hand from the occlusion area, the tracking system starts tracking the robot again and can find it after 5 to 6 iterations, dependent on the latest position of the robot, successfully.

5.2 Hand Based Robot Control

In the interaction between man and machine, an efficient, natural and intuitive commanding system plays a key role. Vision based techniques are usually used to provide such a system. In this section, we present our second application, so-called "Hactor", in which a real time hand tracking and classification system has been used as an interface for sending the commands to an industrial robot. The 2D/3D images, including low resolution range data and high resolution color information, are provided by a MultiCam at video frame rates. This real time application has shown promising results, even under challenging varying lighting conditions which was demonstrated at the Hannover Fair in 2008.

5.2.1 Background

Nowadays, robots are used in different domains ranging from search and rescue in dangerous environments to interactive entertainment. The more the robots are employed in our daily life, the more a natural communication with the robot is required. Current communication devices, like keyboard, mouse, joystick and electronic pen are not intuitive and natural enough. On the other hand, hand gestures, as a natural interface means, has been attracting so much attention for interactive communication with robots in recent years [60], [47], [74], [46]. In this context, vision based hand detection and tracking techniques are used to provide an efficient real time interface with the robot. However, the problem of visual hand recognition and tracking is quite challenging. Many early

approaches used position markers or colored gloves to make the problem of hand recognition easier, but due to their inconvenience, they cannot be considered as a natural interface for the robot control. Thanks to the latest advances in the computer vision field, the recent vision based approaches do not need any extra hardware, except for a camera. These techniques can be categorized as model based and appearance based methods [45]. While model based techniques can recognize the hand motion and its shape exactly, they are computationally expensive and therefore they are infeasible for a real time control application. The appearance based techniques, on the other hand, are faster but they still deal with some issues such as:

- complex nature of the hand with more than 20 Degree of Freedom (DOF)
- cluttered and variant background
- variation in the lighting conditions
- real time computational demand.

In this application, on the one hand, we address the solution to the aforementioned issues in the hand recognition problem, using 2D/3D images and on the other hand we propose an innovative natural commanding system for a Human Robot Interaction (HRI).

5.2.2 System Description

The system which is developed for hand based robot control consists of a set-up of the robot, 2D/3D imaging system and a control application.

- **Set-Up:** The set-up mainly consists of three parts:
  1. A six axis, harmonic driven robot from Kuka of type KR 3 with attached magnetic grabber. The robot itself has been mounted onto an aluminum rack along with the second system component.
  2. A dedicated robot control unit, responsible for robot operation and communication by running proprietary software from Kuka company.
  3. The main PC responsible for data acquisition from 2D/3D imaging system (MultiCam) and running the algorithms.

The communication between the robot control unit and the application PC is done by exchanging XML-wrapped messages via TCP/IP. The network architecture follows a strict client server model, with the control unit as the client connecting to the main PC, running a server thread, during startup.

- **2D/3D Imaging System:** A 2D/3D imaging system using Time-of-Flight (TOF) technique is used. The principle of the camera system was already discussed in chapter 2.
- **Control Application:** In order to make the interaction system with the robot more convenient for the user, all the necessary commands to control the robot such as moving the robot in 6 directions ($x^+, x^-, y^+, y^-, z^+, z^-$) or (de)activating the grabber are done by using a self developed GUI based application which is illustrated in Fig. 5.7.

As a first step, we track the user's hand movement in a predefined volume which is observed by the MultiCam. After finding the hand in the image space, its position in the world coordinate system is calculated and mapped into a virtual space inside the GUI (see Fig. 5.7). The virtual space is represented by a cuboid of defined size and correlates with the MultiCam's view frustum. Hand movement is visualized by placing a 3D hand model in the according location within the virtual space which can be observed by the user in the GUI. Thus, the user can see the movement of his hand virtually in the cuboid. Depending on the hand's distance from the center of the cuboid, a velocity vector is generated and wrapped into XML message. Some other state information are added to that message and it is sent to the robot's control unit which is in charge of unwrapping and sending the

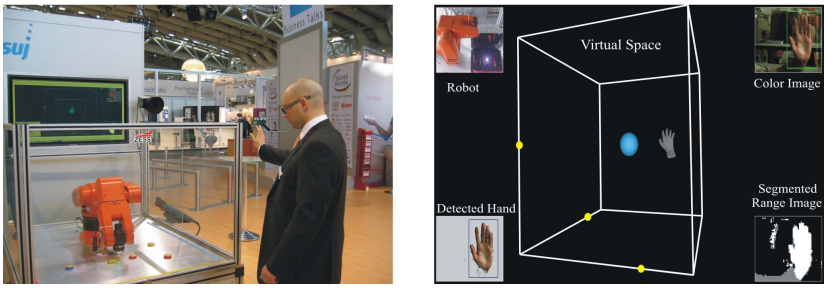

Figure 5.7: Left: Hand based robot control using Multicam, Hannover Fair 2008. Right: Graphical User Interface (GUI).

appropriate information to the robot itself.

By placing the hand model in the center of virtual space, the system can be put in a mode which is susceptible for special commands. For that matter, a rudimentary gesture classification algorithm has been implemented which is able to distinguish between a fist and a palm. We use self-defined fist to palm transition sequences (e.g., a palm-fist-palm transition) in order to perform a robot reset, put the system in predefined modes and to (de)activate the magnetic grabber which in turn enables the robot to handle ferric objects.

5.2.3 Algorithms Overview and Results

In order to remind the used techniques for this application, which have already been discussed in detail in the two previous chapters, an overview of the hand detection and posture classification has been shown in Fig. 5.8.

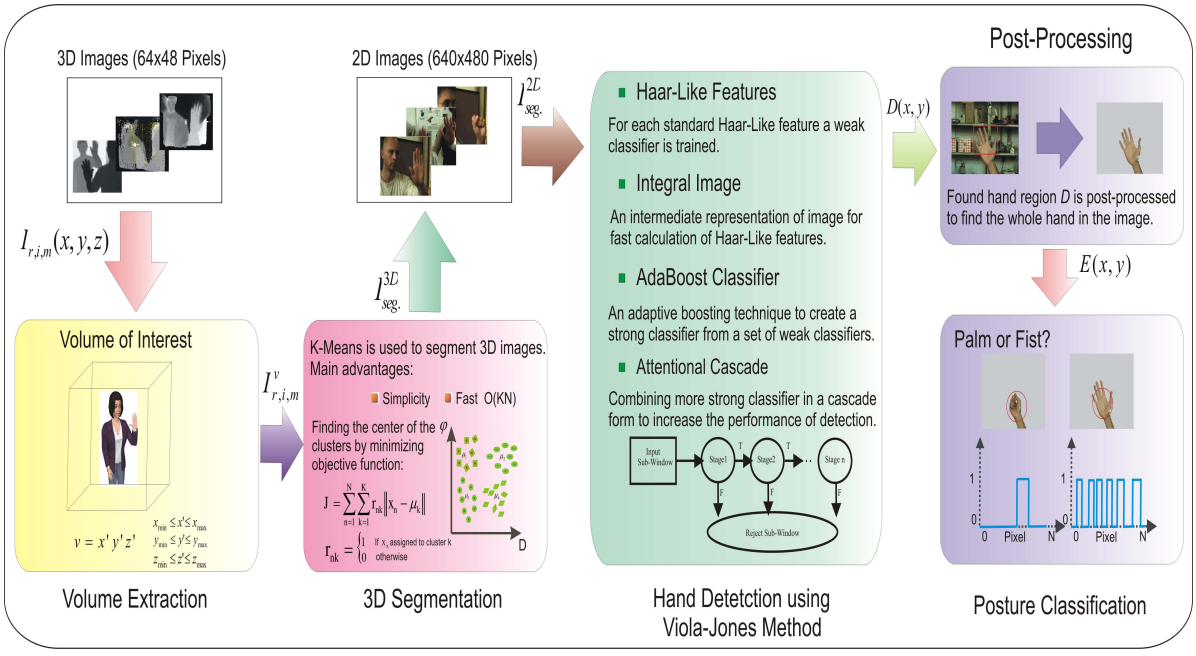

Figure 5.8: Hand detection and posture classification techniques for Hactor application.

As we can see, the K-Means, as the clustering approach, is used to segment the range image which was discussed in section 3.2.3. After segmenting the range image, the 3D segmented image is mapped to the 2D image in order to obtain a 2D segmented image. As discussed in the previous chapter, due to

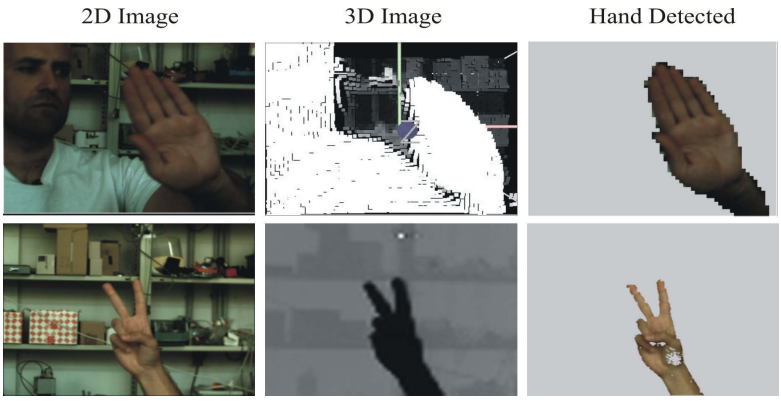

Figure 5.9: Some results of hand detection.

the monocular setup of the camera, mapping is trivial and fast. Having a 2D segmented image, the Viola-Jones method is applied to detect the hand in the image which we already explained in section 3.3.4. In fact, using the Viola-Jones technique we have implemented a binary classifier to distinguish between hand and non-hand objects. Since this classifier is very fast, we use it directly as a hand tracker which was stated in detail in section 4.2.3.

After the hand has been detected using the classifier, in the next step the pose of the hand should be classified. For this application, we consider a simple binary posture classifier to distinguish between palm and fist. As mentioned, any change from palm to fist and vice versa are interpreted as a special command for the robot. To solve the posture classification problem, a heuristic approach has been used which is very accurate and fast and it was already discussed in section 3.1.3.

Fig. 5.9 shows some qualitative results of hand detection and Fig. 5.10 depicts some pictures of the Hactor application in operation at the Hannover Fair in which the people control the robot using hand gestures. The quantitative results of this application was already stated in section 4.2.3.

Figure 5.10: Demonstration of Hactor application at the Hannover Fair 2008.

5.3 Summary

The results of object recognition and tracking using 2D/3D image data have been realized in two practical applications. These applications are a small part of the whole field where 2D/3D vision system can yield promising results.

In the first application, the safety of the personnel in a close cooperation with an industrial robot has been investigated. In fact, it was seen that using the MultiCam one can create a dynamic visual monitoring system to observe the predefined zones around the robot and control its operation in order to avoid any hazardous.

In the second application, as the complement to the first one, an intuitive and natural interaction between the human and a robot has been presented. This is done by implementing a real time hand detection, tracking and classification, using 2D/3D images, which is employed as an interface for sending the commands to the robot.

6

Discussion and Conclusion

A conclusion is the place where you got tired of thinking.
Arthur Bloch (1948-present)

This dissertation successfully investigates different aspects of employing a monocular 2D/3D vision system for a real time object recognition and tracking. The core contribution of this work is fourfold: In the first part, a novel monocular 2D/3D vision system which can provide 3D range and 2D color information, at video frame rate, is presented. In the second part, some aspects of object recognition using 2D/3D data are analyzed. This part consists of preprocessing and classification. While the preprocessing approaches are used to fuse 2D/3D data, extract the features and segment the objects in the scene, the classification techniques are employed to detect the objects of interest and put the same in a class. The third part of this work, as a complement to the previous part, deals with the object tracking to locate the position of the desired object at each frame and find its trajectory. Finally, the last part validates the results of object recognition and tracking in some practical applications.

6.1 Conclusions

In the following the main conclusion points of this work are summarized:

- Combination of TOF cameras with the high resolution standard cameras is a typical solution to the low lateral resolution issue in the current TOF cameras. In fact, such a combination can provide high resolution 2D images with distance information. However, combining these two cameras in a binocular setup deals with some issues such as sensor synchronization, image calibration and registration which can make the final solution practically complex or even infeasible for real world problems.

- The MultiCam which integrates a TOF sensor with a CMOS chip in a monocular setup has the main advantage that it does not require any complicated and time consuming calibration and registration techniques. Likewise, synchronization of 2D and 3D sensors in the MultiCam is simply attainable.
- Feature extraction is an important aspect in this work which deals with derivation of the informative attributes from acquired 2D/3D images. The features used in our work are either obtained by applying some mathematical approaches or based on some heuristics. While the former presents some similarities and differences in the data which cannot be observed directly by the human being, the latter indicates some prior knowledge about the desired object, known by the human. Therefore, the features in our work are categorized in two groups: Machine generated features and Human generated features.
- Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two main techniques to extract machine generated features. It was seen that the LDA usually outperforms the PCA because LDA is a supervised technique which takes the label of the observation into account. However, when the data are undersampled, LDA is unable to achieve good results. PCA+LDA which first project the data to an intermediate space using PCA and then apply LDA can be a good solution to this problem.
- Human generated features, which are derived using some simple heuristics, can be good features for some applications in which we have a prior knowledge about the object of interest. Also, extraction of such features is usually performed very fast which is a significant point in the real time object recognition tasks.
- Range images can yield good results for object segmentation in the real world problems with varying lighting conditions.
- Fusion of multimodal range, intensity and modulation amplitude, output from a TOF sensor, provides new information which can improve the results of object recognition.
- Integration of two different segmentation approaches, performed for each modality imaging, can improve the final result dramatically. For example, while edge detection yields good results on high resolution 2D color images, the unsupervised clustering performs very well on 3D range images. Integration of the 2D edge map with the clustered 3D image results in an improved 2D/3D segmented image.
- Classification of moving objects using range data which is based on Support Vector Machines (SVM) yields good results even with a small training data set. Therefore, SVM can be a good choice for such cases where the number of training data is limited or it is impossible to collect so much training samples.
- AdaBoost is a very fast classification technique. Therefore, it is employed in the Viola-Jones method for real time object detection. By employing 2D/3D images in the Viola-Jones method, on the one hand the solution to the noisy background issue is addressed by using 3D range data and on the other hand the Haar-like features are extracted from high resolution 2D color image.
- Background subtraction using range date is much more functional than using 2D images in cluttered scenes.
- Fusion of features derived from 2D and 3D images can improve the performance of object tracking dramatically.
- Occlusion is one of the most significant challenges in the object tracking which can be handled by using 2D/3D image data to some extent.
- Two merged objects during occlusion can be identified by applying range segmentation.

- If a classifier is fast enough it can be used directly as an object tracker. This is done by applying the Viola-Jones method and using 2D/3D images for hand gesture detection and tracking.
- Probabilistic object tracking methods like the Kalman filter and the CONDENSATION algorithm can address the solution to issues such as inaccuracies in sensor data, complex object motions, appearance changing, occlusion and sudden lost of observations.
- While the Kalman filter performs very well for tracking objects with unimodal Gaussian motion model, the CONDENSATION can perform much better in tracking objects with multimodal non-Gaussian motion models. This is shown in tracking a fast maneuvering person.
- The results of 2D/3D object recognition and tracking are validated in the personnel safety and robot control applications successfully.

6.2 Limitations

In this section, we will review some main limitations of this work which can be addressed in the future works.

- The unambiguous range measurement in the MultiCam is restricted. For example, in the frequency of $20\,MHz$, it is limited to $7.5\,m$. Therefore, while the objects over this distance can be observed in 2D image of the MultiCam, they do not have any reliable distance information in the 3D image. Although reducing the frequency can increase the unambiguity of range measurement, it reduces the resolution of range measurement as well.
- One of the main limitations of the MultiCam is its poor performance in outdoor applications. This is because the TOF range data are affected by the sun light to a great extent. This makes the use of current TOF range sensors impractical in outdoor environments like in automotive applications.
- The infrared lighting is one of the key components of TOF cameras. For a good depth perception of the scene, a powerful lighting system is required. For example, for some applications where a large scale scene should be observed, a very powerful illumination system is required which can increase the complexity of the camera system as well as its cost.
- The current TOF sensors are still more expensive than the conventional CCD and CMOS sensors which limits their large scale use for many applications.

6.3 Suggestions for Future Works

There are still some ongoing works at ZESS to improve the MultiCam as well as the techniques for object recognition and tracking. Some other points can be considered in future work. However, in the following we list some important points as suggestions:

- Improving the MultiCam by employing the new version of TOF sensors with higher resolution than what we have used in this work.
- Employing distributed lighting systems in those applications, where one lighting system is not enough to illuminate the whole scene.
- Upgrading the communication protocol from USB 2.0 to Gigabit Ethernet or USB 3.0 in order to increase the acquisition rate of the MultiCam.
- Implementation of some basic and general algorithms which might be used for many object

recognition problems inside the FPGA in the MultiCam.

- Using more than one MultiCam in the multi camera scenarios where the scene can be observed from different views.
- Investigation on the combination of the MultiCam with stereo vision systems or even setting up a stereo vision system using two MultiCams.
- Design and implementation of an automatic zoom for the MultiCam which can be interesting for some applications where an object in the scene can be zoomed to analyze it precisely.
- Integrating of the MultiCam with other sensors in a heterogeneous sensor network can address some solutions to the limitations of the MultiCam, especially in outdoor applications. For example, combining the MultiCam with other sensors such as thermal or acoustic sensors can be considered for a reliable object recognition and tracking.

Appendix A - Expectation Maximization

Probabilistic models are usually used in order to model the observed data. For example, a data set can be characterized by a mixture of Gaussian distributions as a probabilistic model. The aim is to find the parameters of the model which best fit to the observed data. Maximum Likelihood Estimation (MLE) is one of the popular approaches to find the parameters of the model. However, in many situations the data are incomplete, i.e., some parts of an observation are missing. For example, in the case where a whole data set is to be modeled with a mixture of Gaussian, we do not know which data belongs to which distribution, therefore the class labels are missing or hidden. This is called incomplete data and one cannot easily apply the MLE to estimate the parameters of a model for such a data. Expectation Maximization (EM) is a general approach to iterative computation of maximum-likelihood estimates when the observations can be viewed as incomplete data. This appendix reviews the main important points in derivation of EM technique [97], [107], [8].

A.1 Maximum Likelihood

We let $f_{X/\Theta}(\xi/P)$ be a probability density function which is governed by the set of unknown parameters $\Theta=\{\theta_1,...,\theta_n\}$. We also suppose that there is a sample data set $X=\{x_1,...,x_n\}$ with n data vectors which are independent and identically distributed (iid) with distribution f. For iid sample set the density function can be written as follows

$$f_{X/\Theta}(\xi/P)=f_{x_1,x_2,...,x_n/\theta_1,\theta_2,...,\theta_n}(\xi_1,\xi_2,...,\xi_n/\rho_1,\rho_2,...,\rho_n)$$

$$=f_{x_1/\theta_1}(\xi_1/\rho_1).f_{x_2/\theta_2}(\xi_2/\rho_2).\ ...\ .f_{x_n/\theta_n}(\xi_n/\rho_n)=\prod_{i=1}^{n}f_{x_i/\theta_i}(\xi_i/\rho_i) \qquad \text{A.1}$$

where ξ and $P=\{\rho_1,...,\rho_n\}$ are dummy vectors such that $\xi, P\in\mathbb{R}^n$.

This function can also be seen in such a way that the observed data X is fixed and the parameters Θ can be varied. Form this point of view, it is called the likelihood of parameters given the data or so-called likelihood function which is formulated as follows

$$L_{\Theta/X}(P/\xi)=\prod_{i=1}^{n}f_{x_i/\theta_i}(\xi_i/\rho_i)\,. \qquad \text{A.2}$$

The idea behind MLE is to maximize the likelihood of parameters with respect to the observation data. In other words, we would like to find $\hat{\Theta}$ where

$$\hat{\Theta}=\underset{P}{argmax}\,L_{\Theta/X}(P/\xi)\,. \qquad \text{A.3}$$

Since it is analytically easier to work with the logarithm of likelihood function we maximize the following function instead

$$\hat{L}_{\Theta/X}(P/\xi)=\sum_{i=1}^{n}\ln f_{x_i/\theta_i}(\xi_i/\rho_i)\,. \qquad \text{A.4}$$

The objective function $\hat{L}_{\Theta/X}(P/\xi)$ for some cases in which the data is complete has a single global optimum which can be found in closed form. In contrast, for the incomplete data cases the objective function has multiple local maxima and no closed form solution. More often, however, the observed data are incomplete and therefore the closed form solution to this maximization problem does not exist. In such cases we need to apply more elaborate techniques to the problem. The EM algorithm is one of such techniques which will be studied in next sections.

A.2 EM Algorithm

The Expectation Maximization (EM) algorithm is a generalization of maximum likelihood estimation for finding the parameters of a model from a data set which is incomplete or has missing values. In general, there are two types of the EM applications. In the first type, the EM algorithm is applied when the data has missing values due to the problems or limitation of observation process. The second type of the EM application is when optimizing the likelihood function is analytically intractable and the likelihood function can be simplified by considering the additional but hidden (missing) parameters. In both of these applications, the EM algorithm provides a simple and robust tool for parameter estimation in an iterative computation process. Since each iteration of the algorithm consists of an expectation step followed by a maximization step, it is called the EM algorithm [107].

We assume that the data X is an incomplete observed data set, and associated with a complete data set Z such that $Z = \{X,Y\}$ and there is a projection $Z \rightarrow X$ which is many to one. The joint density function for the complete data set can be written as follows

$$f_{Z/\Theta}(\lambda/P)=f_{X,Y/\Theta}(\xi,\zeta/P)=f_{Y/X,\Theta}(\zeta/\xi,P)\,f_{X/\Theta}(\xi/P) \qquad \text{(A.5)}$$

where λ, P, ξ and ζ are dummy variables in real coordinate space $\mathbb{R}^n$.

In fact, the EM algorithm is directed at finding the parameters Θ which maximizes $f_{X/\Theta}(\xi/P)$ given an observed data X by the use of the associated family $f_{Z/\Theta}(\lambda/P)$. In other words, for each observation in the incomplete data set X, we consider the corresponding value of the latent variable in Y such that $\{X,Y\}$ makes our complete data set. Now we define a new likelihood function called the complete-data likelihood which is as follows

$$L_{\Theta/Z}(P/\lambda)=L_{\Theta/X,Y}(P/\xi,\zeta)\,. \qquad \text{(A.6)}$$

This function is a random variable because the hidden variables Y is unknown. We will suppose, however, that maximization of the complete-data log likelihood function is straightforward.

Since we do not have the complete data set Z, we cannot use the complete-data log likelihood function. The solution, proposed in the EM algorithm, is to find the expected value of the complete-data log likelihood with respect to the unknown hidden data Y given the observed data X and the current parameters. This, in fact, corresponds to the E-step of the EM algorithm and is expressed in the following function

$$Q(P, P^{old}) = E[\ln f_{X,Y/\Theta^{old}}(\xi, \zeta/P)/\xi, P^{old}] \tag{A.7}$$

where Θ^{old} are the current parameter values with corresponding dummy variables P^{old} in the density function. In this equation, the data set X and parameters Θ^{old} are known and constant and Θ is a set of parameters that we want to adjust and Y is a random variable which is governed by the $f_{Y/X,\Theta^{old}}(\zeta/\xi, P^{old})$. Therefore, the equation A.7 can be written as follows

$$Q(P, P^{old}) = \sum_{\Omega} \ln f_{X,Y/\Theta}(\xi, \zeta/P) f_{Y/X,\Theta^{old}}(\zeta/\xi, P^{old}) \tag{A.8}$$

where Ω is the space of values the random variable Y can take on.

It should be mentioned that $f_{Y/X,\Theta^{old}}(\zeta/\xi, P^{old})$ is the marginal distribution of the hidden variables and is dependent on the observed data X and the current parameters Θ^{old} .

In the next step, the EM algorithm determines the revised parameter estimate Θ^{new} by maximization of the Q function. This is called M-step in the EM algorithm.

$$P^{new} = argmax\, Q(P, P^{old}) . \tag{A.9}$$

In fact, the EM algorithm starts by choosing some values for the parameters as the initial values and then these two steps are repeated till the algorithm converges to a local maxima of the likelihood function.

A.3 The EM for Gaussian Mixtures

The estimation of parameters for a mixture of Gaussian distributions is one of the widely used applications of the EM algorithm in the pattern recognition field. In fact, Gaussian mixture model, as a simple linear superposition of Gaussian components, can provide a richer class of density models than a single Gaussian. In this case, the sample data set $X=\{x_1,...,x_n\}$ can be modeled using a mixture density function as follows

$$f_{X/\Theta}(\xi/P) = \sum_{k=1}^{K} \alpha_k f_{X/\theta_k}(\xi/\rho_k) \tag{A.10}$$

where each Gaussian density function is parametrized by $\theta_i \in \Theta$ and the parameters $\alpha_k \in \Theta$ are called mixing coefficients such that $\sum_{k=1}^{K} \alpha_k = 1$.

Therefore the incomplete-data log likelihood function for A.10 is as follows

$$\ln(L_{\Theta/X}(P/\xi)) = \ln \prod_{i=1}^{n} f_{x_i/\Theta}(\xi_i/P) = \sum_{i=1}^{n} \ln\left(\sum_{k=1}^{K} \alpha_k f_{x_i/\theta_k}(\xi_i/\rho_k) \right) . \tag{A.11}$$

Optimizing the log likelihood function A.11 is difficult because it contains the log of the sum. In order to solve this problem, we consider X as incomplete data set and associate unobserved data $Y=\{y_1, y_2, ..., y_n\}$ in which $y_i \in \{1,..., K\}$ and $y_i = k$ if the i^{th} data sample is generated by the k^{th} mixture component.

Now by assuming the existence of the hidden variables Y and making an initial guess for the parameters, Θ^{old}, we can obtain the marginal density function $f_{Y/X,\Theta^{old}}(\zeta/\xi, P^{old})$ as follows

$$f_{Y/X,\Theta^{old}}(\zeta/\xi, P^{old}) = \prod_{i=1}^{n} f_{y_i/x_i,\Theta^{old}}(\zeta_i/\xi_i, P^{old}) . \tag{A.12}$$

Therefore the equation Q in A.8 for this case takes the form of

$$\begin{aligned} Q(P, P^{old}) &= \sum_{\Omega} \ln f_{X,Y/\Theta}(\xi, \zeta/P) \prod_{i=1}^{n} f_{y_i/x_i,\Theta^{old}}(\zeta_i/\xi_i, P^{old}) \\ &= \sum_{y_i \in \Omega} \sum_{i=1}^{n} \ln(\alpha_{y_i} f^{y_i}_{x_i/\theta_{y_i}}(\xi_i/\rho_{y_i})) \prod_{i=1}^{n} f_{y_i/x_i,\Theta^{old}}(\zeta_i/\xi_i, P^{old}) \end{aligned} \tag{A.13}$$

we follow the same mathematical simplification which is done in [107] to take the Q equation as follows

$$\begin{aligned} Q(P/P^{old}) &= \sum_{k=1}^{K} \sum_{i=1}^{n} \ln(\alpha_k)\, p\{\omega: k=\kappa/x_i=\xi, \Theta=P^{old}\} \\ &+ \sum_{k=1}^{K} \sum_{i=1}^{n} \ln f^{k}_{x_i/\theta_k}(\xi_i/\rho_k)\, p\{\omega: k=\kappa/x_i=\xi, \Theta=P^{old}\} \end{aligned} \tag{A.14}$$

where κ is a dummy variable corresponding to the k^{th} Gaussian distribution.

Now, in order to find the parameters θ_k and α_k for the k^{th} distribution, we can maximize the two terms in Q function independently because these parameters are not related.

By introducing the Lagrange multiplier τ with the constraint $\sum_k \alpha_k = 1$ we can find α_k by solving the following equation

$$\begin{aligned} &\frac{\partial}{\partial \alpha_k}\left[\sum_{k=1}^{K} \sum_{i=1}^{n} \ln(\alpha_k)\, p\{\omega: k=\kappa/x_i=\xi, \Theta=P^{old}\} + \tau\left(\sum_k \alpha_k - 1\right)\right] = 0 \\ &\rightarrow \quad \sum_{i=1}^{n} \frac{1}{\alpha_k}\, p\{\omega: k=\kappa/x_i=\xi, \Theta=P^{old}\} + \tau = 0 . \end{aligned} \tag{A.15}$$

Summing over k, we get $\tau = -n$ which consequently results in

$$\alpha_k = \frac{1}{n}\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\} \ . \tag{A.16}$$

For the k^{th} Gaussian distribution with the dimension of d, we are looking for the parameters consisting of mean and covariance $\theta_k = (\mu_k, \Sigma_k)$ in the following density function

$$f^{k}_{x/\mu_k, \Sigma_k}(\xi / \mu_o, \Sigma_o) = \frac{1}{(2\pi)^{d/2}|\Sigma_o|^{1/2}} \exp\left\{-\frac{1}{2}(\xi - \mu_o)^T \Sigma_o^{-1}(\xi - \mu_o)\right\} \ . \tag{A.17}$$

By substituting the Gaussian distribution in the second component of equation A.14, and taking the derivative with respect to μ_k and setting to zero, we get the following equation

$$\sum_{i=1}^{n} \Sigma_k^{-1}(x_i - \mu_k)\, p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\} = 0 \tag{A.18}$$

in which we can find the parameter μ_k as follows

$$\mu_k^{new} = \frac{\sum_{i=1}^{n} \xi_i \, p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}}{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}} \ . \tag{A.19}$$

By taking the derivative of the same with respect to Σ_k and taking some assumptions, we can obtain the Σ_k as follows

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}(\xi_i - \mu_k)(\xi_i - \mu_k)^T}{\sum_{i=1}^{n} p\{\omega : k = \kappa / x_i = \xi, \Theta = P^{old}\}} \ . \tag{A.20}$$

Appendix B- The CONDENSATION Algorithm

The CONDENSATION algorithm is based on factored sampling. It is a solution to the problems with multimodal non-Gaussian observations. In the following this technique will be reviewed [20], [91], [94].

Table B.1- Notation of vectors and probability distributions.

Symbol	meaning
$x(t)$	State vector at time *t* with corresponding dummy vector ξ
$z(t)$	Measurement vector at time *t* with corresponding dummy vector ζ
$X(t)$	History of the state $\{x_1,..., x_t\}$ up to time *t*
$Z(t)$	History of all observations $\{z_1,..., z_t\}$ up to time *t*
$f_{x(t)/Z(t)}(\xi/z_t)$	The *a posteriori* density function
$f_{x(t)/Z(t-1)}(\xi/z_{t-1})$	The *a priori* density function
$f_{x(t)/x(t-1)}(\xi/x_{t-1})$	The process density describing the stochastic dynamics
$f_{z(t)/x(t)}(\zeta/\xi)$	The observation density
$f_{x(t-1)/Z(t-1)}(\xi_{t-1}/Z_{t-1})$	The initialization density

- **Stochastic Dynamics**

  Based on general assumption from Markov chain, the new state is conditional directly only on the immediately preceding state, but not on any function prior to *t*-1.

$$f_{x(t)/X(t-1)}(\xi/X_{t-1}) = f_{x(t)/x(t-1)}(\xi/x_{t-1}) . \qquad \text{B-1}$$

- **Measurement**

  Observations are assumed to be independent and therefore defined by specifying the conditional density $f_{z(t)/x(t)}(\zeta/\xi)$.

- **Propagation**

  Given a continuous-valued Markov chain with independent observations, the conditional state density at time *t* is defined by $f_{x(t)/Z(t)}(\xi/Z_t)$. This can be calculated as follows

$$f_{x(t)/Z(t)}(\xi/Z_t)=k_t f_{z(t)/x(t)}(\zeta/\xi) f_{x(t)/Z(t-1)}(\xi/Z_{t-1}) . \quad \text{B-2}$$

where k_t is a normalization constant and

$$f_{x(t)/Z(t)}(\xi/Z_t)=\int_{x_{t-1}} f_{x(t)/x(t-1)}(\xi/x_{t-1}) f_{x(t-1)/Z(t-1)}(x_{t-1}/Z_{t-1})\,dx_{t-1} . \quad \text{B-3}$$

In fact, the problem can be summarized as follows

$$f_{x(t-1)/Z(t-1)}(x_{t-1}/Z_{t-1}) \rightarrow^{dynamics} \rightarrow f_{x(t)/Z(t-1)}(\xi/Z_{t-1}) \rightarrow^{measurement} \rightarrow f_{x(t)/Z(t)}(\xi/Z_t) \quad \text{B-4}$$

.

To solve this problem, the CONDENSATION algorithm, as opposed to the analytical solutions, employs factored sampling. The factored sampling generates a random variate x' from a distribution $\tilde{f}_x(\xi)$ which approximates the posterior $f_{x(t)/Z(t)}(\xi/Z_t)$. To do that, first a sample set $\{s^{(1)},...,s^{(N)}\}$ is generated from the priori density $f_x(\xi)$ with probability π^j as follows

$$\pi^j=\frac{f_z(s^{(j)})}{\sum_{j=1}^{N}(f_z(s^{(j)}))} \quad \text{B-5}$$

where

$$f_z(\xi)=f_{z(t)/x(t)}(\zeta/\xi) \quad \text{B-6}$$

is the conditional observation density.

In fact, the CONDENSATION algorithm applies factored sampling iteratively to calculate the *a posteriori* density $f_{x(t)/Z(t)}(\xi/Z_t)$. An iteration step starts with a sample set *s* representing the *a posteriori* density $f_{x(t-1)/Z(t-1)}(x_{t-1}/Z_{t-1})$ from the previous time step. In the prediction step, *s* is propagated to obtain a new sample set s' according to the system model. s' represents the a priori density $f_{x(t)/Z(t-1)}(\xi/Z_{t-1})$. In updating step the observation density is used to derive a new sample set s'' by applying factored sampling. Thus s'' represents the new *a posteriori* density $f_{x(t)/Z(t)}(\xi/Z_t)$.

One time step in the CONDENSATION algorithm has been shown in Fig. B.1.

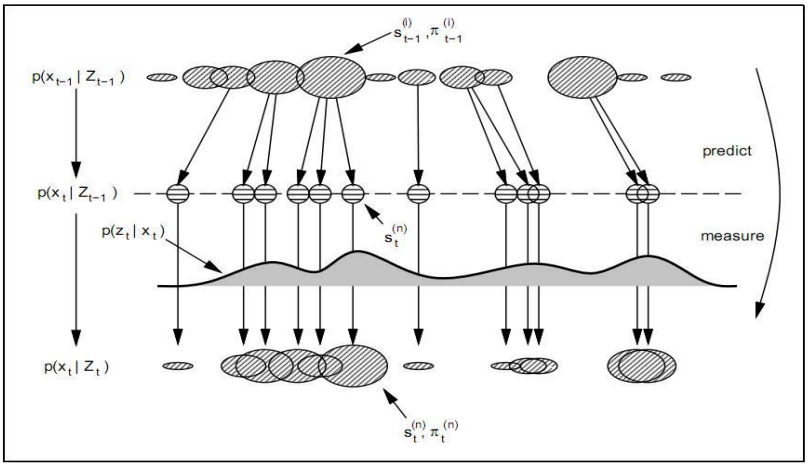

Figure B-1: One time-step in the CONDENSATION algorithm [20].

Appendix C- The MultiCam's Data Sheet

Preliminary 2D/3D Multi CAM Data Sheet

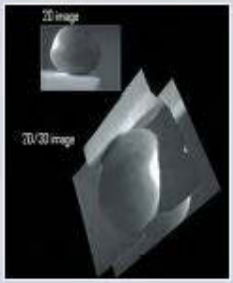

FIGURE 1

FEATURES

- fast acquisition of synchronized 2D and 3D scene data
- 2D sensor: 640x480 pixels, greyscale or Bayer pattern
- 3D sensor: 64x48 pixels with built-in suppression of background illumination (SBI)
- monocular set-up with beamsplitter for simple-range independent - image registration; bandpass filters to eliminate ambient light influence; standard F-Mount adapter for different fields of view
- USB 2.0 interface
- external trigger input/output
- support for second modulation output (to couple two illumination systems)
- adoption of the unambiguous measurement range by freely choosing the modulation frequency up to 20MHz

SPECIFICATIONS - CAMERA

Pixel Array Resolution	2D: 659(H) x 494(V) 3D: 64(H) x 48(V)
Lens Mount	F-Mount
Field of View	20,2° (H) x 15,2° (V)
Unambiguous Range	20 MHz: 7,5 Meter
Frame Rate	Up to 100 fps, combined 2D/3D Images (depends on exposure time)
Interface	USB 2.0

Output Data	x,y,z coordinates, i - intensity
Camera Body Size	80 x 85 x 85 mm³
Power Consumption (without illumination)	3 Watt
Operation temperature	-10°C bis 50°C
Weight	700 gr (1600 gr, with Illumination and Lens)
range resolution (90% reflectivity target)	1m: ±2mm 2m: ±3mm 5m: ±12mm 10m: ±21mm 20m: ±48mm (burst mode) 25m: ±59mm (burst mode) 30m: ±66mm (burst mode)

ILLUMINATION

Illumination power (optical)	6 Watt (continuous)
Power consumption	30 Watt (12V/2.5A)
Wavelength	870 nm (Peak)
LED Type	50 mW (optisch) bei 100 mA (30.000 Stunden)
Emitting Angle LED	± 10°

No LEDs	240
Power Supply	12V ± 5%
12V +- 5%	Between 1 MHz and 20 MHz, default 20 MHz
Size	160 mm (outer diameter)

FEATURES

- USB driver and SDK for Microsoft Windows ® XP / Vista
- documentation as PDF included in the installer
- sample program to work with the camera and save / load / view 2D/3D images (source code included)
- system parametrization (shutter time etc.)
- synchronisation of the 2D and 3D data
- several data acquisition modes
  - external trigger enabling synchronisation with external events (e.g. movements)
  - sequence recording to capture a sequence with maximal performance
  - single-shot
- different levels of data processing (raw data, processed data)
- Matlab and Labview sample programs

Universität Siegen
ZESS - Zentrum für Sensorsysteme
Paul-Bonatz-Str. 9-11
57076 Siegen

Tel.: +49 271-7403323
Fax: +49 271-7402336
www.zess.uni-siegen.de

ZESS ZENTRUM FÜR SENSORSYSTEME

Bibliography

[1] Cyganek, B.; Siebert, J.P.: *An Introduction to 3D Computer Vision Techniques and Algorithms,* Wiley-Blackwell, 2009.

[2] Bishop, Christopher M.: *Pattern Recognition and Machine Learning*, Springer, 2008.

[3] Bradski, Gray; Kaebler, Adrian: *Learning OpenCV: Computer Vision with the OpenCV Library*, Friends of OpenDocument Inc., 2008.

[4] Ohta J.: *Smart CMOS Image Sensors and Applications*, Crc Press Inc., 2007.

[5] Abe, Shigeo: *Support Vector Machines for Pattern Recognition*, Springer, 2005.

[6] Kecman, Vojislav: *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, Mit Pr, 2001.

[7] Cristianini, Nello; Shawe-Taylor, John: A*n Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[8] McLachlan, Geoffrey J.; Krishnan, T.: *The EM Algorithm and Extensions*, Wiley-Interscience, 1996.

[9] Michie, D.; Spiegelhalter, D.J.; Taylor, C.C.: *Machine Learning, Neural and Statistical Classification*, Prentice Hall, 1994.

[10] Loffeld, Otmar: *Estimationstheorie, Bd.2, Anwendungen, Kalman-Filter*, Oldenbourg, 1990.

[11] Maybeck, P.S.: *Stochastic models, estimation, and control*, Academic Press, 1979.

[12] Lottner, O., *Monocular 2D/3D Camera System Platform in Dynamic Environments and Multi-Lighting Configurations*, ZESS, University of Siegen, 2010- Unpublished PhD Thesis.

[13] Kulic', D., *Safety for Human-Robot Interaction*, The University of British Columbia, 2005.

[14] Estrada, F.J., *Advances in Computational Image Segmentation and Perceptual Grouping,* University of Toronto, 2005.

[15] Cantzler, H., *Improving architectural 3D reconstruction by constrained modelling,* University of Edinburgh, 2003.

[16] Justen, D., *Untersuchung eines neuartigen 2D- gestützten 3D-PMD*, University of Siegen, 2001.

[17] Luan, Xuming, *Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology*, 2001.

[18] Sobottka, K., *Analysis of Low-Resolution Range Image Sequences*, University of Bern, 2000.

[19] Pollak, I., *Nonlinear Scale Space Analysis in Image Processing,* Massachusetts Institute of Technology, 1999.

[20] Isard, M.A., *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*, 1998.

[21] Weber, J., *Ein visuell unterstütztes laseroptisches Multisensorsystem zur automatisierten Erfassung dreidimensionaler Objekte*, University of Siegen, Shaker, 1998.

[22] Skarbek, W.; Koschan, *A.: Colour Image Segmentation- A Survey,* Institute of Computer Science Polish Academy of Sciences, 1994.

[23] Principle of SVM, http://imtech.res.in/

[24] Graphical Representation of PCA, http://web.media.mit.edu/~tristan/phd/dissertation/index.html

[25] SafetyEye, 2009, http://www.pilz.de/

[26] Safety, 2009, http://www.bircheramerica.com/

[27] The Berkeley Segmentation Dataset and Benchmark, 2009, http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

[28] Canesta, Inc., 2009, http://www.canesta.com/

[29] Swissranger; C.C.S. d'Electronique SA, 2009, http://www.mesa-imaging.ch/

[30] 3DV Systems, ZCam, 2009, http://www.3dvsystems.com/

[31] PMD Tech, 3D Video Sensor Array with Active SBI, 2009, http://www.pmdtec.com/

[32] DFG Dyn3D Project- LO 455/10-2, ZESS, University of Siegen, 2006-2010.

[33] Valls Miro, J.; Dissanayake, G., Robotic 3D visual mapping for augmented situational awareness in unstructured environments, *International Workshop on Robotics for Risky Interventions and Surveillance of the Environment (RISE'08).*

[34] Abmayr, T.; Härtl, F.; Wagner, A.; Burschka, D.; Hirzinger, G.; Fröhlich, C., Calibration and registration framework for 3D reconstruction of the Kirche Seefeld, P*roceedings of the 3rd ISPRS International Workshop.*

[35] Ghobadi, S.E.; Loepprich, O.E.; Lottner, O.; Hartmann, K.; Weihs, W.; Loffeld, O., 2D/3D Image Data Analysis for Object Tracking and Classification, *Advances in Machine Learning and Data Analysis, 2009.*

[36] Lipnickas, A.; Knys, A., A Stereovision System for 3-D Perception, E*lectronics And Electrical Engineering, 2009.*

[37] Ghobadi, S.E.; Loepprich, O.E.; Ahmadov, F.; Bernshausen, J.; Hartmann, K.; Loffeld, O., Real Time Hand Based Robot Control Using 2D/3D Images, Advances in Visual Computing, 5*th International Symposium on Visual Computing, 2008.*

[38] Benezeth, Y.; Jodoin, P.M.; Laurent, H.; Rosenberger, C., Review and Evaluation of Commonly Implemented Background Subtraction Algorithms, *Proceedings of the 19th*

International Conference on Pattern Recognition, 2008.

[39] Dalley, G.; Migdal, J.; Grimson, W.E.L., Background Subtraction for Temporally Irregular Dynamic Textures, *IEEE Workshop on Applications of Computer Vision, 2008.*

[40] Ghobadi, S.E.; Loepprich, O.E.; Lottner, O.; Hartmann, K.; Loffeld, O.; Weihs, W., Improved Object Segmentation Based on 2D/3D Images, *The Fifth IASTED International Conference on Signal Processing, 2008.*

[41] Ghobadi, S.E.; Loepprich, O.E.; Lottner, O.; Hartmann, K.; Loffeld, O.; Weihs, W., Analysis of the personnel safety in a man-machine-cooperation using 2D/3D images, *IARP/ EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance of the Environment, 2008.*

[42] Zhu, J.; Wang, L.; Yang, R.; Davis, J., Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps, *Computer Vision and Pattern Recognition (CVPR), 2008.*

[43] Muehlbauere, Q.; Kuehnlenz, K.; Buss, M., Fusing Laser and Vision Data with a Genetic ICP Algorithm, *10th International Conference on Control, Automation, Robotics and Vision, 2008.*

[44] Fechteler, P.; Eisert, P., Adaptive Color Classification for Structured Light Systems, I*EEE Conference on Computer Vision and Pattern Recognition Workshops, 2008.*

[45] Fang, Y., Wang, K., Cheng, J., Lu, H., A real-time hand gesture recognition method, *IEEE International Conference on Multimedia and Expo, 2007.*

[46] Cerlinca, T., Pentiuc, S., Cerlinca, M., Hand posture recognition for human-robot interaction, *Proceedings of the 2007 workshop on Multimodal Interfaces Insemantic Interaction, 2007.*

[47] Wang, C., Wang, K., Hand posture recognition using adaboost with sift for human robot interaction, *International Conference on Advanced Robotics, 2007.*

[48] Kahlmann, T.; Remondino, F.; Guiliaume, S., Range imaging technology: new developments and applications for people identification and tracking, *Conference on Videometrics IX, part of the IS&T/SPIE Symposium on Electronic Imaging, 2007.*

[49] Chen, Q.; Petriu, E.M.; Georganas, N.D, 3D Hand Tracking and Motion Analysis with a Combination Approach of Statistical and Syntatic Analysis, *International Workshop on Haptic Audio Visual Environments and their Applications, 2007*

[50] Ghobadi, S.E.; Loepprich, O.E.; Hartmann, K.; Loffeld, O., Hand Segmentation Using 2D/3D Images, *Proceedings of Image and Vision Computing, New Zealand, 2007.*

[51] Robards, M.; Gao, J.; Chartlon, P., A Discriminant Analysis for Undersampled Data, *2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM 07), 2007.*

[52] Lottner, O., Hartmann, K., Weihs, W., Loffeld, O., Image Registration and Calibration Aspectsfor a New 2D /3D Camera, *EOS Conference on Frontiers in Electronic Imaging, 2007.*

[53] A'rni, S.; Aanas, H.; Larsen, R., Fusion of Stereo Vision and Time-of-Flight Imaging for Improved 3D Estimation, *International workshop in Conjunction with DAGM'07: Dynamic 3D Imaging, 2007.*

[54] Hahne, U.; Alexa, M., Combining Time-of-Flight Depth and Stereo Images without Accurate Extrinsic Calibration, *International workshop in Conjunction with DAGM'07: Dynamic 3D Imaging, 2007.*

[55] Huhle, B.; Fleck, S.; Schilling, A., Integrating 3D Time-of-Flight Camera Data and High Resolution Images for 3DTV Applications, *3DTV CON - The True Vision, 2007.*

[56] Haindl, M.; Zid, P., Multimodal Range Image Segmentation, *Vision Systems: Segmentation and Pattern Recognition, 2007.*

[57] Alami, R.; Albu-Schaeffer, A.; Bicchi, A.; Bischoff, R.; Chatila, R.; De Luca, A.; De Santis, A.; Giralt, G.; Guiochet,J.; Hirzinger,G.; Ingrand, F.; Lippiello, V.; Mattone, R.; Powell, D.; Sen, S.; Siciliano, B.; Tonietti, G.and Villani L., Safe and Dependable Physical Human-Robot Interaction in Anthropic Domains: State of the Art and Challenges, *Physical Human-Robot Interaction in Anthropic Domains, 2006.*

[58] Olesya, O. , Human-Robot Interaction. Safety problem, *International Workshop on Robotics in Alpe-Adria-Danube Region, Hungary, 2006.*

[59] Nasser, S.; Alkhaldi, R.; Vert, G., A Modified Fuzzy K-means Clustering using Expectation Maximization, *IEEE World Congress on Computational Intelligence,, 2006.*

[60] Malima, A., Ozgur, E., Cetin, M., A fast algorithm for vision-based hand gesture recognition for robot control, *Conference on Signal Processing and Communications Applications, 2006.*

[61] Ghobadi, S.E.; Hartmann, K; Weihs, W.; Netramai, C.; Loffeld, O.; Roth, H., Detection and classification of moving objects-stereo or time-of-flight images, *Computational Intelligence and Security, 2006.*

[62] Kahlmann, T.; Remondino, F.; Ingensand, H., Calibration for increased accuracy of the range imaging camera swissranger, *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology', 2006.*

[63] Kuhnert, K.D.; Stommel, M., Fusion of stereo-camera and pmd-camera data for real-time suited precise 3D environment reconstruction, *International Conference on Intelligent Robots and Systems, 2006.*

[64] Reulke, R., Combination of Distance Data with High Resolution Images, *ISPRS, Commission V Symposium, Image Engineering and Vision Metrology, 2006.*

[65] Perrollaz, M.; Labayrade, R.; Royère, C.; Hautière, N.; Aubert, D., Long Range Obstacle Detection Using Laser Scanner and Stereovision, *Intelligent Vehicles Symposium, 2006.*

[66] Ghobadi, S.E., Hartmann, K., Weihs, W., Prasad, A. and Sluiter A., Classificationof 3D Moving Objects using Support Vector Machines and a 3D-Time of Flight Camera, *3rd International Conference on Artificial Intelligence in Engineering and Technology, 2006.*

[67] Peters, V.; Hasouneh, F.; Knedlik, S.; Loffeld, O., Simulation of PMD based self-

localization of mobile sensor nodes or robots, *19th Symposium on Simulation Technique, 2006.*

[68] Amiri Parian, J.; Gruen, A., Integrated laser scanner and intensity image calibration and accuracy assessment , *ISPRS workshop Laser scanning, 2005.*

[69] Goktuk, S.B., Rafii, A., An Occupant Classification System-Eigen Shapes or Knowledge-Based Features, *Computer Vision and Pattern Recognition (CVPR'05), 2005.*

[70] Möller, T.; Kraft, H.; Frey, J.; Albrecht, M.; Lange, R., Robust 3D Measurement with PMD Sensors, *Range Imaging Day, Zürich, 2005.*

[71] Sato, K.; Aggarwal, J.K., Temporal spatio-velocity transform and its application to tracking and interaction, *Computer Vision and Image Understanding, 2004.*

[72] Romero, L.; Nunez, A.; Bravo, S.; Gamboa, L.E., Fusing a Laser Range Finder and a Stereo Vision System to Detect Obstacles in 3D, *Advances in Artificial Intelligence–IBERAMIA, 2004.*

[73] Gokturk, S.B., Yalcin, H. and Bamji, C., A Time of Flight Depth Sensor, System Description, Issues and Solutions, *IEEE workshop on Real-Time 3D Sensors and Their Use in conjunction with IEEE Conference on Computer Vision and Pattern Recognition, 2004.*

[74] Rogalla, O., Ehrenmann, M., Zoellner, R., Becher, R., Dillmann, R., Using Gesture and Speech Control for Commanding a Robot Assistant, *IEEE International Workshop on Robot and Human Interactive Communication, 2002.*

[75] Cremers, D. and Kohlberger, T., Nonlinear Shape Statistics in Mumford-Shah Based Segmentation, *European Conference on Computer Vision (ECCV), 2002.*

[76] Baltzakis, H.; Argyros, A.; Trahania, P., Fusion of range and visual data for the extraction of scene structure information, *Intl. Conf. on Pattern Recognition, (ICPR), 2002.*

[77] Forster, F.; Lang, M.; Radig, B., Real-Time Range Imaging for Dynamic Scene Using Colour-Edge Based Structured Light, *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), 2002.*

[78] Senior,A.; Hampapur, A.; Tian, Y.; Brown, L.; Pankanti, S. and Bolle, R., Appearance Models for Occlusion Handling, *2nd IEEE Int. Workshop on PETS, 2001.*

[79] KaewTraKulPong, P.; Bowden, R., An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection, *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, 2001.*

[80] Viola, P., Jones, M., Rapid object detection using a boosted cascade of simple features, *Conference on Computer vision and Pattern Recognition, 2001.*

[81] Surmann, H.; Lingemann, K.; Nüchter, A.; Hertzberg, J., A 3D laser range finder for autonomous mobile robots, *Proceedings of the 32nd ISR (International Symposium on Robotics), 2001.*

[82] Forster, F.; Rummel, P.; Lang, M.; Radig, B., The Hiscore Camera- A Real Time Three

Dimensional and Color Camera, *International Conference on Image Processing, 2001.*

[83] Beymer, D.; Konolige, K., Real-time tracking of multiple people using continuous detection, *IEEE International Conference on Computer Vision (ICCV), 1999.*

[84] Stauffer, C.; Grimson, W.E.L., Adaptive Background Mixture Models for Real-Time Tracking, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), 1999.*

[85] Zhuanga, X.; Dai, D.: Improved discriminate analysis for high-dimensional data and its application to face recognition, *Pattern Recognition 40(5), 1570–1578, 2007.*

[86] Yilmaz, A.; Javed, O.; Shah, M.: Object tracking: A survey, *ACM computing surveys, Vol. 38 No.4, 2006.*

[87] Bab-Hadiasha, A.; Gheissari, N.: Range Image Segmentation Using Surface Selection Criterion, *IEEE Transaction on Image Processing, Vol. 15 No. 7, p. 2006-2018, 2006.*

[88] Ye, J.; Li, Q.: A Two-Stage Linear Discriminant Analysis via QR-Decomposition, I*EEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, pp. 929-941, June 2005.*

[89] Yilmaz, A.; Li; X. and Shah, M.: Contour-Based Object Tracking with Occlusion Handling inVideo Acquired Using Mobile Cameras, *IEEE Transaction On Pattern Analysis and Machine Intelligence, 26 (1), 2004.*

[90] Comaniciu, D.; Ramesh, V.; Meer, P.: Kernel-Based Object Tracking, I*EEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, n.5, P.P. 564–575, 2003.*

[91] Koller-Meier, E.B. and Ade, F.: Tracking Multiple Objects Using the Condensation Algorithm, J*ournal of Robotics and Automation Systems, vol. 34, pp. 93-105, 2001.*

[92] Martinez, A.; Kak, C.: PCA versus LDA, *Transaction on Pattern Analysis and Machine Intelligence, vol. 23, No.2, pp. 228-233, 2001.*

[93] MacCormick, J.; Blake, A.: Probabilistic exclusion and partitioned sampling for multiple object tracking, *International Journal of Computer Vision, 39(1), P.P 57-71, 2000.*

[94] Isard, M. and Blake, A.: CONDENSATION—Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision, vol. 29, P.P. 5-28, 1998.*

[95] Hoover, A.; Jean-baptiste, G.; Jiang, X.; Flynn, P.; Bunke, H.; Goldgof, D.; Bowyer, K.; Eggert, D.; Fitzgibbon, A.; Fisher, R.: An Experimental Comparison of Range Image Segmentation Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18 , Issue 7, PP. 673-689, 1996.*

[96] Turk, M.; Pentland, A.: Eigenfaces for recognition, *Journal of Cognitive Neuroscience,vol. 3, No. 1 PP. 71-86, 1991.*

[97] Dempster, A. P.; Laird, N. M.; Rubin, D. B.: Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society. Series B (Methodological), vol. 39, No. 1, pp.1-38, 1977.*

[98] DIN EN 775: Manipulating industrial robots; safety (ISO 10218:1992, modified); German

version EN 775:1992 + AC:1993, DIN-adopted European Standard.

[99] ANSI/RIA R15.06-1999: American National Standard for Industrial Robots and Robot Systems- Safety Requirements, American National Standard Institute, New York, NY.

[100] Wirjadi, O.: Survey of 3D Image Segmentation Methods, Fraunhofer, 2007.

[101] Loepprich, O.E.: Translation Unit API, ZESS, University of Siegen, 2007.

[102] Santrac, N.; Friedland, G., Rojas, R.: High Resolution Segmentation with a Time-of-Flight 3D-Camera using the Example of a Lecture Scene, Department of Computer Science, Freie Universitaet Berlin, 2006.

[103] Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers, 2004.

[104] Jain, S.: A survey of Laser Range Finding, 2003.

[105] Schutter, J.D; Geeter, J.D; Lefebvre, T. and Bruyninckx, H.: Kalman Filters: A Tutorial, 1999.

[106] Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer Academic Publishers, 1998.

[107] Jeff Bilmes: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, 1998.

[108] Welch, G. and Bishop, G.: An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, 1995.

[109] Kohoutek, T., Monitoring of an Industrial Robot by Processing of 3D Range Imaging Data Measured by the SwissRange SR-3000, 2007.