2008

Probabilistic three-dimensional object tracking based on adaptive depth segmentation

Ehsan Parvizi
University of Windsor

NOTE TO USERS

Probabilistic 3D Object Tracking Based on Adaptive Depth Segmentation

by

Ehsan Parvizi

A Thesis
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada
2008

Your file Votre référence
ISBN: 978-0-494-47082-4
Our file Notre référence
ISBN: 978-0-494-47082-4

Declaration of Co-Authorship and Previous Publication

Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research with Professor Q.M. Jonathan Wu under his supervision. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from the co-author to include the above materials in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

Declaration of Previous Publication

This thesis includes three original papers that have been previously published in peer reviewed journal and conferences, as follows:

Thesis Chapter	Publication Title and Full Citation	Publication Status
Chapter 4	*E. Parvizi, Q.M. J. Wu, "Real-time Approach for Adaptive Object Segmentation in Time-of-Flight Sensors", 20th IEEE Int'l Conference on Tools with Artificial Intelligence, November 2008.*	*In Press*
Chapter 4	*E. Parvizi, Q.M. J. Wu, "Multiple Object Tracking Based on Adaptive Depth Segmentation", 2008 Canadian Conference on Computer and Robot Vision, CRV, pp. 273-277, May 2008.*	*Published*
Chapter 4	*L. Sabeti, E. Parvizi, Q.M. J. Wu, "Visual Tracking Using Color Cameras and Time-of-Flight Range Imaging Sensors", Journal of Multimedia, Vol. 3, No. 2, pp. 28-36, 2008.*	*Published*

I certify that I own the copyright of the above published material and that it describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted

referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owners to include such materials in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Object tracking is one of the fundamental topics of computer vision with diverse applications. The arising challenges in tracking, *i.e.*, cluttered scenes, occlusion, complex motion, and illumination variations have motivated utilization of depth information from 3D sensors. However, current 3D trackers are not applicable to unconstrained environments without a priori knowledge.

As an important object detection module in tracking, segmentation subdivides an image into its constituent regions. Nevertheless, the existing range segmentation methods in literature are difficult to implement in real-time due to their slow performance.

In this thesis, a 3D object tracking method based on adaptive depth segmentation and particle filtering is presented. In this approach, the segmentation method as the bottom-up process is combined with the particle filter as the top-down process to achieve efficient tracking results under challenging circumstances. The experimental results demonstrate the efficiency, as well as robustness of the tracking algorithm utilizing real-world range information.

To Samin
for her love, inspiration, and constant support.

Acknowledgements

I would like to express my deep appreciation to my supervisor, Dr. Jonathan Wu, for his invaluable direction and support throughout this work. My Master thesis could not have been accomplished without his academic guidance and insight. I would also like to thank my committee members, Dr. Majid Ahmadi and Dr. Boubakeur Boufama for their continuous attention and helpful suggestions.

I would like to thank the secretaries of the Electrical and Computer Engineering Department, Ms. Andria Turner and Ms. Shelby Marchand, for their friendly assistance and emotional support through my whole study at the University of Windsor. In addition, I would like to thank Mr. Frank Cicchello and Mr. Don Tersigni for their assistance. I am also thankful to my friends at the University of Windsor as well as my peers at the Computer Vision and Sensing Systems Laboratory for the useful discussions we had together and for their friendship throughout my graduate studies.

I would like to express my sincere appreciation to Samin, who has always supported and encouraged me to fulfill my degree and provided a warm environment at home. Further gratitude goes toward my parents for their love and confidence in me, without which I would have not been successful in achieving my goals.

Contents

List of Figures

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
CCTV	Closed-Circuit Television
CMOS	Complementary Metal-Oxide Semiconductor
CONDENSATION	Conditional Density Propagation
EKF	Extended Kalman filter
FOV	Field of View
HSV	Hue-Saturation-Value color space
i.i.d.	independent and identically distributed
MC	Monte Carlo
pdf	probability density function
PF	Particle Filter
RGB	Red-Green-Blue color space
SIS	Sequential Importance Sampling
SMC	Sequential Monte Carlo
TOF	Time of Flight
UKF	Unscented Kalman Filter

Chapter 1

Introduction

1.1 Motivation

Visual tracking is the process of detecting objects of interest from background and tracking them through consecutive frames in a video sequence. It has been one of the important topics of computer vision as it finds application in areas such as video surveillance, human-computer interaction, intelligent transportation, driver monitoring, pedestrian protection, medical diagnostics, and video compression.

There are several challenges that exist in tracking, including occlusion, noise in images, complex object motion, cluttered background, illumination variations, and real-time requirements. To address these challenges, extensive research activity has been dedicated to object tracking during the past years. Among different approaches, color tracking has been one of the most popular methods because of rich information content provided by using color as a feature for tracking. However, there is a limitation in choosing color for tracking due to its sensitivity to illumination variations, hence

encouraging incorporation of other features to increase the tracker efficiency.

In recent years, usage of depth information for object tracking is becoming popular due to the availability of information about the third dimension, which provides the distances of objects from the sensor. Stereo vision systems have been prevalently exploited to determine the depth-map of the scene by means of calculating disparities from images captured from two cameras separated by a baseline. Nevertheless, the process of stereo matching to obtain depth-map information tends to be computationally intense, and the results are not adequately accurate. In addition, passive stereo sensors require the presence of sufficient ambient illumination so that they can produce good quality shots. These limitations have motivated development of active depth sensors such as laser range scanners and time-of-flight (TOF) sensors [41]. TOF sensors have significant advantages over laser range scanners, which include higher accuracy, existence of vertical as well as horizontal scanning capability, pixel-level measurement quality, and considerably smaller weight and size [51].

Visual tracking can be further classified into low-level and high-level approaches. In a low-level approach, an image is segmented or classified in order to localize the blob or object without any initial hypothesis. The high-level approach, on the other hand, performs object association from one frame to the next, by generating an object hypothesis and then evaluating the likelihood of a set of given hypotheses for each frame, based on the most recent measurement. The particle filter [25, 27, 9] is one of the most successful object tracking methods that solves nonlinear cases in which noise may be non-additive and non-Gaussian, by representing simultaneous alternative hypotheses. The particle filter has been adopted as a recursive Bayesian filter in many research works such as [1, 11, 29, 39]. Besides, it has been shown to produce superior results as compared to mean shift, Kalman filter and the extended Kalman filter [1, 39].

This thesis aspires to develop a 3D object tracking algorithm based on a TOF

sensor, combining the high-level approach of particle filtering with a proposed bottom-up technique for object segmentation in depth images. One of the main applications of this research is in intelligent transportation systems, where the 3D profile of the driver and passengers are tracked to accomplish certain tasks. The corresponding environment consists of cluttered backgrounds, involving object occlusions and with possibly inadequate illumination settings or drastic lighting changes.

1.2 Thesis Organization

Following this chapter, chapter 2 presents a brief review of the literature in 2D and 3D visual tracking. Furthermore, the state of the art time-of-flight sensing technology for capturing 3D scene structure is described. This is followed by a review of the tracking approaches that exploit 3D sensors for acquiring input video sequences.

Chapter 3 describes the fundamentals of nonlinear Bayesian tracking and current approaches including the particle filtering method. It also presents a review of the significant research contributions in the area of particle filter tracking.

Chapter 4 is devoted to the elaboration of the developed probabilistic 3D tracking which is mainly based on adaptive depth segmentation of TOF range images. It is shown that depth histograms can be leveraged to derive a range segmentation approach, in order to be applied in object detection. In addition, the developed method is exploited to define parameters of the particle filter, which is used to associate and track objects throughout the video sequence.

Chapter 5 covers experimental results of both the adaptive depth segmentation and the probabilistic tracking presented in chapter 4.

Chapter 6 concludes this thesis by summarizing the contributions of this thesis and outlining suggestions regarding further development of the proposed research.

Chapter 2

Visual Tracking Literature

2.1 Overview

In visual tracking, there are three main factors that need to be considered in order to design an object tracking system. First, a suitable representation of the object should be defined. Another important step is to choose the appropriate input image features, and finally a strategy for detection of objects needs to be selected [53].

Objects can be represented by a point or by a set of points, geometric shapes, contours, silhouette representations, or using cylinders and ellipses. Point representation is generally used for tracking objects that occupy small regions in an image [48, 45]. Geometric shapes such as a rectangle, ellipse, *etc.* are more appropriate for representing simple rigid objects as well as nonrigid objects [13], and for tracking complex nonrigid shapes, contour and silhouette representations are exploited, which define the boundaries of an object and the region inside the object boundaries, respectively [54, 7]. Finally, cylinders or ellipses are used to model articulated objects such

as human body parts [53]. Also, object appearances can be represented using probability densities, which can be either parametric (Gaussian, mixture of Gaussians) or nonparametric (Parzen windows, histograms) [53]. Template representation is another approach that is most suitable for tracking objects whose poses do not change noticeably during tracking. One of the advantages of templates is that they convey both spatial and appearance information, but from a single view. There also exist other appearance representations including active models and multi-view models.

Common visual features that are used in tracking are color, edge, optical flow, and texture information [53]:

- Color: The color of an object is mainly affected by the environmental illumination as well as the reflectance properties of the object. Different color spaces are used for color representation in tracking, such as RGB, HSV, L*U*V, L*a*b, *etc.* Color is one of the most popular features used for tracking in the literature. However, color is sensitive to illumination variations, hence encouraging the incorporation of other features to increase the efficiency of the tracker.

- Edges: Edge information generally convey drastic intensity variations in an image, extracted using edge detection techniques. One of its significant properties is that edges are less sensitive to illumination variations compared to color features.

- Optical Flow: Optical flow is a dense field of displacement vectors which defines the translation of each pixel in a region. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications.

- Texture: Texture is a measure of intensity variation of a surface which quantifies properties such as smoothness and regularity. Texture requires a processing

step to generate descriptors compared to color. Also, texture features are less sensitive to illumination changes than color.

Each tracking algorithm is composed of an object detection module. Object detection is performed either once the object appears in the scene or in every frame, considering the temporal information of consecutive frames to increase the detection efficiency. Common object detection methods include point detection, segmentation, background modeling, and supervised classifiers [53].

2.2 Segmentation

Segmentation subdivides an image into its constituent regions or objects [24]. The level to which this subdivision is carried out depends on the problem being solved. In other words, the segmentation task should end when the objects of interest have been isolated. An optimal depth segmentation algorithm should partition an image into more meaningful and easier to analyze regions with no overlap, where the final depth scene is generated by arranging all these regions together. Furthermore, segmentation methods in tracking applications should consume the least amount of processing time, as well as incur the fewest possible computational operations, due to the real-time requirement in tracking approaches. On the other hand, increasing the time efficiency should not hinder achieving acceptable results. There are two main factors to be considered in order to evaluate the performance (speed) of segmentation algorithms, *i.e.* number of iterations, and computational complexity.

Image segmentation methods are generally based on one of the following: discontinuity and similarity of the image values. In the first category, an image is partitioned based on abrupt variations in pixel values, *i.e.*, image edges. The methods in the second category partition an image into regions that are similar according to a set of predefined criteria. These approaches include thresholding, region growing, and

region splitting and merging.

Edge detection has been one of the popular segmentation algorithms for years. Edge-based segmentation techniques apply edge detection to extract discontinuities in the scene and segment the image [20]. In order to segment the image correctly, the identified edges should form closed boundaries. However, the resulting edge maps are often disconnected. Hence, additional processing should be performed on edge boundaries to connect isolated edges if they are within a distance-threshold from each other. Another main drawback of edge-based range segmentation algorithms is that discontinuities are smooth and hard to locate for curved surfaces in depth images, resulting in under-segmentation of range images. Thus, it is essential to inspect each of the edge-separated regions iteratively to assure that no object of interest is missed. As a result, edge detection segmentation methods are computationally intense and therefore have limited applications in real-time vision systems.

Thresholding is another popular approach for segmentation especially in applications where speed is important, mainly because of its simplicity of implementation and intuitive properties. For instance, in a gray-level histogram of an image composed of light objects on a dark background, the objects can be separated from the background using a threshold level determined from the histogram. In this case, segmentation is carried out by scanning the image pixel by pixel and labeling each pixel as object or background, depending on whether the gray level of that pixel is greater or less than the threshold value. In general cases where there are three or more modes characterizing the image histogram, multilevel thresholding can be used to classify each object [24]. Note that the success of thresholding depends entirely on how well the histogram can be partitioned.

Region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria [24]. The basic approach is to start with a set of "seed" points and iteratively grow regions by appending to each seed those neighboring pixels

that have properties (*e.g.* color ranges) similar to the seed [2]. Generally, there can be one or more starting points based on the application. In cases where *a priori* information is not available, the same set of properties that will be used to assign pixels to regions during the growing process is calculated at each pixel. Based on the results of these calculations which form clusters, pixels close to the cluster centroids can be used as seeds. The selection of similarity criteria is crucial in the success of region growing and it depends on both the problem type and the type of image data available. Another issue in this technique is the determination of a stopping rule. As a rule, growing a region should stop when no more pixels satisfy the criteria for inclusion in that region.

Object tracking based on region growing in range images has been considered in [36], which mainly relies on road modeling. Here, a distance map is first calculated, and a region growing segmentation is performed within simulated and uncluttered traffic scenes. In [33], a depth-based tracking system in traffic scenes is described, where the employed segmentation method is based on a region growing scheme introduced in [36]. In order to achieve reliable results, it is necessary to apply some constraints regarding the identification of the ground surface. The background and foreground range images have to be predefined in this method, which is not suitable for real-time depth traffic environments where the camera is non-stationary. Furthermore, an additional preprocessing step is required to remove edges from objects using an edge detection technique. As discussed earlier, incorporating edge detection modules reduces the segmentation speed drastically, without even considering the usage of the computationally intense region growing algorithm. Depth and intensity information can be used together as in [52], where a visual surveillance system is presented based on depth sensing. Prospective locations of the objects of interest are determined through evaluation of intensity data, whereas the discontinuities among the detected objects are considered through processing of depth images. Evidently,

relying on intensity values requires the use of edge detection, not to mention the vulnerability to illumination variations. In addition, the distance map calculation is carried out in order to locate object boundaries in the scene. One necessary preprocessing step in this method is to eliminate out-of-range background clutter — which is done manually in each scenario, hence restricting its application.

2.3 Kernel Density Estimation

Kernel density estimation is a broadly applied technique in statistics and pattern recognition (also known as Parzen window method) [47, 19]. A kernel density estimate is a continuous function derived from discrete data [35]. To accurately determine the mode locations of a random variable x, which are the local maxima of its probability density function $p(x)$, a continuous estimate of the underlying density $\hat{p}(x)$ has to be defined. However, since only the discrete values of x are available, the data is convolved with a symmetric kernel function by placing a kernel in each point. Therefore, the density estimate in a given location is the average of the contributions from each kernel. However, due to the finite nature of the kernel support, only some of the points contribute to the density estimate. Let x_i, $i = 1, \ldots, n$, be scalar measurements drawn from an arbitrary probability distribution $p(x)$. The kernel density estimate $\hat{p}(x)$ of this distribution is achieved using a kernel function $K(u)$ and a bandwidth h

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{2.1}$$

The most significant properties of kernel functions are that they should be symmetric with bounded support and satisfy the following:

$$\begin{aligned} &K(u) = 0 && |u| > 1 \\ &\textstyle\int_{-1}^{1} K(u) = 1 \\ &K(u) = K(-u) \geq 0 \\ &K(u_1) \geq K(u_2) && |u_1| \leq |u_2| \end{aligned} \tag{2.2}$$

The above formulation can be further expanded by replacing the kernel bandwidth h with a symmetric, positive definite bandwidth matrix H to include multivariate measurements.

2.4 Time-of-Flight Sensors

Time-of-flight (TOF) depth sensors are non-contact optical measurement devices that are able to acquire the entire depth image of a scene in real-time. Depth information is delivered by the solid-state sensor without any need for external circuitry. They consist of a modulated light source such as infrared, a CMOS imaging sensor consisting of an array of pixels, as well as an optical focusing system [40, 41].

Overall, TOF sensors have significant advantages over laser range scanners, including higher accuracy, existence of vertical as well as horizontal scanning, pixel-level measurement quality, and considerably smaller weight and size. A comparison of TOF sensors and laser range scanners can be found in [51].

2.4.1 TOF Principle

TOF systems operate based on the TOF principle [32]. An intensity modulated wave is synchronously emitted through the light source, propagating from the TOF sensor to the scene and is reflected by the scene back to the sensor where the sensor captures its time of flight. The phase delay between the two signals is used to determine the object's distance from the sensor. The signal phase is detected by synchronously demodulating the incoming modulated light within the detector. Let $s(t)$ and $g(t)$ be the incoming optical signal (with amplitude A and phase φ) and the demodulation signal, respectively.

$$s(t) = 1 + A\cos(\omega t - \varphi) \tag{2.3}$$

$$g(t) = \cos(\omega t) \tag{2.4}$$

The cross correlation between the demodulation signal and the incoming signal is computed as:

$$c(\tau) = s(t) \otimes g(t) = lim_{T\to\infty}\frac{1}{T}\int_{-\frac{T}{2}}^{+\frac{T}{2}} s(t)\cdot g(t+\tau)dt \tag{2.5}$$

Evaluating (2.5) for phase delays of $\tau_0 = 0°$, $\tau_1 = 90°$, $\tau_2 = 180°$, and $\tau_3 = 270°$, the phase φ, the offset B, and the amplitude A of the incoming signal are determined as follows.

$$\varphi = \arctan\left(\frac{c(\tau_3) - c(\tau_1)}{c(\tau_0) - c(\tau_2)}\right) \tag{2.6}$$

$$B = \frac{c(\tau_0) + c(\tau_1) + c(\tau_2) + c(\tau_3)}{4} \tag{2.7}$$

$$A = \frac{\sqrt{[c(\tau_3) - c(\tau_1)]^2 + [c(\tau_0) - c(\tau_2)]^2}}{2} \tag{2.8}$$

The object's distance from the sensor d is thus determined from φ in

$$d = l\frac{\varphi}{2\pi} \tag{2.9}$$

where,

$$l = \frac{c}{2f_m} \tag{2.10}$$

represents the non-ambiguity distance range, f_m the modulation frequency, and c the speed of light [40].

2.5 TOF Literature

Since the introduction and implementation of TOF sensing systems, many researchers have contributed to the development of tracking approaches using TOF sensors. This section presents a review of the leading TOF-based tracking methods in recent years.

2.5.1 TOF Application in Head Tracking

In [23], a head-tracking algorithm using a time-of-flight depth sensor is described, where the depth sensor is exploited to segment the background and foreground. A depth signature is determined for each segmented foreground, followed by a comparison with depth signatures collected in training. K-means algorithm is used to cluster the training data to account for all possible cases. A correlation-based method allocates weights to the most possible head locations, and the final head location is determined by weighted-interpolating among these locations. Although it reported promising results, this work only addresses the tracking problem for one person sitting in front of the camera. Furthermore, only partial self occlusion between the object's head and hand is considered and the inter-object occlusion or occlusion by the background structure are not studied. The training session is required to obtain a good model for the head location, where the head location is chosen manually.

Another head tracking algorithm using model fitting of the head's 3D depth map as an ellipse and shape matching is presented in [37]. The ellipse properties, *i.e.*, position and size are constantly updated by a local search. Edge detection is applied on depth maps to provide depth discontinuities, followed by a Chamfer distance-based ellipse detection. The initialization of head position is not addressed in their paper and reader is referred to [44, 42]. The reason to use distance transform instead of the edge image is that the similarity measure becomes a smooth function of the shape model parameters and matching location, also allowing some degree of dissimilarity.

2.5.2 TOF Application in Visual Surveillance

Xu and Fujimura [52] present a visual surveillance method using depth and gray information from a single camera in a user-specified 3D environment. Depth data is used to extract the discontinuities between multiple objects in the scene, and gray data is exploited to extract contextual information about the location of the objects of

interest. This method can be used in non-stationary camera situations since it is not dependent upon background subtraction. The reported depth resolution for a 2.5 m range is 1 cm. The out-of-range background clutters are eliminated by manually setting a maximum acceptable distance, which is a constraint on its application. Region segmentation is performed based on a split-and-merge algorithm. Basically, foreground areas are split into smaller regions by separating their depth values into predefined depth layers (8 to 32) depending on each application. In this method, regions are grouped into sub-areas based on connected component analysis. Later on, sub-areas from different layers are merged with each other if the connections between them are regular and their layers are continuous. Next, a geometric representation is used to fit ellipses into the detected silhouette. Finally, tracking is performed by a simple method of identifying similar ellipses in the next frame. This method will achieve good results only if the relative movements are small and occlusions are rare. Also, the size of ellipsoidal model will vary for different object poses, causing inefficient results. Above all, this method depends much on the particular scene that is under investigation since it is dependent on manual selection of a depth threshold for foreground detection.

In [38], a method is presented for illumination-invariant tracking (head, hand, and body) in indoor cluttered environments using depth edges from a depth sensor. It mainly focuses on tracking the object as a whole, instead of using features for tracking. The operational domain is limited such that the target holds a distinct depth difference with respect to its surrounding environment. This technique uses potential fields, where the target is modeled as an attractor and each point outside the target is assigned a value based on its distance from the target. This task is carried out using depth image edges, based on distance transform and contour tracking.

An initial investigation of the use of TOF sensors for people tracking is studied in [6]. 3D tracking methods based on stereo vision and plan-view maps deal with

major issues such as occlusions and quick variations in body pose and appearance effectively. However, stereo systems degrade in performance in situations where there are untextured scenes because of homogeneous objects or poor lighting condition. The solution presented in this paper is to use TOF sensors that can operate under severe low-lighting conditions. Several geometrical constraints and invariants have been considered in order to simplify tracking. A simple background subtraction algorithm based on a pixel-wise parametric statistical model is applied to construct the background model. This model is not maintained over time once constructed, since the camera and background are assumed to be stationary. A plan-view map is also built using the intrinsic and extrinsic parameters of the camera considering the orthographic projection of the scene. This requires camera calibration which is done offline in the training stage. Tracking is performed over connected components in the blob level using a limited set of geometric features. This paper handles occlusion using Kalman filter which produces efficient results in linear situations. Based on the plan-view setup assumptions people cannot overlap each other and also should enter the scene separately.

2.5.3 TOF Application in Traffic Environment

A 3D multiple object tracking in traffic scenarios is investigated in [33]. The authors use a TOF range sensor mounted on a vehicle to acquire depth images. At first, several preprocessing filters are applied to eliminate noisy pixels from the image, as follows:

- Analysis of an amplitude minimum filter, which requires sorting all the amplitude values on the image and choosing an adaptive threshold to reject those pixels with low amplitude values as noise.

- Ground surface segmentation, which requires pre-defining a foreground range

image as well as a background range image to reject pixels with values outside the range of these two images.

- Edge pixel removal, which removes the edge pixels between objects as a necessary step before region growing.

The traditional region growing is exploited on the range image to segment regions of different objects. This is followed by a detailed region post-processing to deal with the problem of object over-segmentation caused by region growing. Next, the segmented objects in the current frame are associated with the objects in the previous frame. An object association strategy is proposed to deal with object tracking robustly in case of merging and splitting. A Kalman filter model is constructed in the last step to correct the object positions in the current frame and predict their positions in the next frame. These predictions will be used in the next iteration for the corresponding object association.

2.6 TOF Applications in Other Areas

Except for tracking, TOF sensors have been used in other research areas such as face detection, 3D pose estimation, human computer interaction, etc.

2.6.1 Face Detection

Hansen *et al.* [26] have developed a face detection method using images from a TOF sensor. They use cascade classifiers for face detection using both gray and depth information. Based on their argument, the distance to the object provides an important cue for face detection and size verifications. Depending on the distance from the camera, the apparent face size changes as well as the number of detailed face features.

2.6.2 3D Pose Estimation

In [55], the authors present a 3D head pose estimation technique using both gray and depth information from a TOF range sensor. Depth information is used for successful head segmentation even in a cluttered scene, where a sparse optical flow is exploited at head region to estimate the 3D head motion.

Also in the paper by Fujimura *et al.* [22], the authors present a 3D head pose estimation approach from a sequence of images taken by a single TOF camera. They partition the human body into a number of clusters and use machine learning techniques for pose extraction.

2.6.3 Human Computer Interaction

In [18], a virtual keyboard system consisting of a pattern projector and a TOF range sensor is presented. The depth information from the TOF sensor is used to detect the hand region with respect to a reference frame. Furthermore, the feature models of the depth curve is analyzed to determine the exact key that was pressed.

2.7 Stereo Vision Methods

In addition to TOF systems, passive sensors such as stereo vision cameras have been used for retrieving depth information for many years. They are less expensive than active sensors, but rely on 2D information in order to calculate the range values in the scene. Therefore, their performance is degraded in low illumination environments.

2.7.1 Integrated Stereo Visual Tracking

Darrell *et al.* [15] present a visual person tracking system combining stereo, color, and face detection modules. Depth information, through real-time stereo processing, is

used to localize users from other objects in the background. Also, skin classification detects body parts within the isolated user silhouette, and face detection localizes the faces within those identified body parts. Each method alone can track a user under optimal conditions, but each has substantial failure modes in unconstrained environments. They find that these failures are often independent, thus by combining them one can achieve relatively robust results. Head-size objects can cause false positives in the depth module, skin-colored objects can cause false positives in the color module, and face pattern detectors typically are slower and cause false positives in non-canonical poses or expressions. It is also mentioned that a key strength of this system is the use of depth estimation hardware. Tracking is performed on three different time-scales: short-term, *i.e.* frame to frame changes, medium-term for temporary occlusions or absences for a few minutes, and long term for absences of hours or more. In short-term tracking, region correspondences based on region position and size are considered. Here, a statistical model of multi-modal appearance is considered to resolve correspondences between tracked users over time. The incorporated features are body shape, face appearance location, color of hair, skin, and clothes determined at each time-step. Also, mean and covariance of the represented features are used to identify users on their return to the scene. In medium term tracking, lighting constancy and stable clothing color are assumed, as opposed to long-term tracking, where these criteria are neglected. In the dense domain, raw range signals are smoothed to reduce the effect of low confidence stereo disparities, using a morphological closing operator. A gradient operator is applied on the image, thresholded at a critical value based on the maximum expected depth discontinuity in the depth profile of one person (determined as 8 inches). Connected component analysis is performed on the regions of smoothly varying range, returning only those areas greater than a minimum threshold. Furthermore, estimates of head location, which are positioned below the vertical maxima, are calculated for each silhouette. Depth silhouettes are tracked

at each frame using position and size constancy through comparison of the centroid and size of each new depth region with those of the previous frame. In other words, for each new region, the closest old region within a minimum threshold is marked as the correspondence match. In the range module and for long-term tracking, height of the user is estimated and used as an attribute of identity. In the color module, the average color of skin and hair regions, as well as an optional color histogram of clothing are considered for identification. Although stereo is used in this paper, it mainly relies on intensity features rather than depth information.

2.7.2 Head Detection Using Stereo

In this paper, Krotosky *et al.* [31] propose a real-time head detection algorithm using stereo vision. Their work is based on posture analysis of occupants using a stereo camera mounted inside a car. Their method is intended to detect 3D head location information for in-car applications such as smart airbag deployment. Statistical background subtraction is used in their approach, constructing a background model from an empty car frame. Stereo-related post processing is carried out in order to obtain the disparity image, followed by its subtraction from the background model estimate. To eliminate small disparity blobs, median filter along with morphological opening and connected component analysis are applied to the result. The errors occur in the case of poor illumination, occlusion and competing objects. Since stereo sensors rely on texture of the objects in the scene, overexposed faces are impossible to be processed in order to produce elliptical objects using this method. In the case of low illumination, it is suggested to use near-infrared illumination, however with no further elaboration. As for occlusion handling, this method is able to resume detection after momentary partial occlusions. Competing objects cause errors in head detection since they have the same size, shape and disparity as the desired head.

2.8 Active Triangulation Methods

Active sensors have been exploited in [34] to present a head tracking algorithm using 3D data. A 3D sensor composed of a closed-circuit TV (CCTV) color camera and a standard slide projector is employed to acquire 3D data as well as color information. This method is based on the active triangulation principle, where color-encoded light pattern is projected onto the scene, and its deformation on the object surfaces is measured. The authors use an appearance-based 3D pose detection in a Bayesian tracking framework. Depth information is used to separate body from the background, while segmentation of the head from body relies on statistical modeling of the head–torso points in 3D space. However, their approach assumes only one person in the scene. As a result, the background separation technique will encounter difficulty when applied to a complicated setting with more than one person.

2.9 Summary

In this chapter, the key components of 2D and 3D visual tracking systems have been presented. Furthermore, TOF sensors and their applications have been discussed, followed by literature review of stereo vision and triangulation tracking methods. The TOF sensor has been chosen for this research due to its advantages over other 3D sensors, one of which is its ability to provide 3D depth profiles without further processing.

The following chapter addresses the probabilistic filtering approaches in nonlinear Bayesian domain, followed by a review of their applications in tracking.

Chapter 3

Nonlinear Bayesian Tracking

Tracking is one of the problems that require estimation of the state of a system using noisy measurements. [1] In this regard, the state-space approach is used to model discrete-time dynamic systems. In order to analyze a dynamic system, two models should be known: The system model, describing the evolution of the state with time, and the measurement model describing the relation between the noisy measurements and the state. In the Bayesian approach to dynamic state estimation, one can construct the posterior probability density function (pdf) of the state based on all available information, including the set of received measurements. In principle, this pdf is the complete solution to the estimation problem since it includes all available statistical information. Thus, an optimal estimate of the state may be obtained from the pdf. However, for many problems an estimate is required at each time-step when a measurement is received, which leads to a recursive filter solution. In a recursive filtering approach, received data can be processed sequentially rather than as a batch

[1] This chapter includes parts of material in [3], reproduced with the authors' permission.

so that it is not necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available [3]. A recursive filter consists of two main stages: prediction and update. The prediction stage uses the system model to predict the state pdf from one time-step to the next. Prediction generally translates, deforms, and spreads the state pdf, since the state is subject to unknown disturbances modeled as random noise. The update step uses the latest measurement to modify the prediction using the Bayes theorem. In the problem of tracking, the target is characterized by the state sequence $\{x_t, t \in \mathbb{N}\}$, assuming $\mathbb{N}$ as the set of natural numbers. The evolution of the state sequence is determined by the system model:

$$x_t = f_t(x_{t-1}, v_{t-1}) \tag{3.1}$$

where f_t is in general a nonlinear function of the state x_{t-1}, and $\{v_{t-1}, t \in \mathbb{N}\}$ is an i.i.d. process noise sequence. The objective of tracking is to recursively estimate x_t from measurements

$$z_t = h_t(x_t, n_t) \tag{3.2}$$

where h_t is in general a nonlinear function, $\{n_t, t \in \mathbb{N}\}$ is an i.i.d. process noise sequence. In other words, we are interested in filtered estimates of x_t based on the set of all available measurements $z_{1:t} = \{z_i, i = 1, \ldots, t\}$ up to time t. Therefore, it is necessary to have the pdf $p(x_t|z_{1:t})$. The initial pdf $p(x_0|z_0) \equiv p(x_0)$ of the state vector, *prior*, is assumed to be known. Then, in principle, the pdf $p(x_t|z_{1:t})$ may be obtained recursively in two stages: prediction and update. In the prediction stage, the system model (3.1) is utilized to obtain the prior pdf of the state at time t using the following equation, knowing that the pdf $p(x_{t-1}|z_{1:t-1})$ at time $t-1$ is available.

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{3.3}$$

In tracking, the system model is assumed to be a first order Markov process. Hence, $p(x_t|x_{t-1}, z_{1:t-1}) = p(x_t|x_{t-1})$, defined by the system model and the known statistics

of v_{t-1}. In the update stage and at time-step t, the measurement z_t becomes available, and this is used to update the prior using Bayes' rule

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \tag{3.4}$$

where the normalizing constant

$$p(z_t|z_{1:t-1}) = \int p(z_t|x_t)p(x_t|z_{1:t-1})dx_t \tag{3.5}$$

depends on the likelihood function $p(z_t|x_t)$ defined by the measurement model and the known statistics of n_t. Note that in the update stage (3.4), the measurement z_t is used to modify the prior density to obtain the required posterior density of the current state.

The optimal Bayesian solution is based on the recursive equations (3.3) and (3.4). However, this recursive propagation of the posterior density cannot be determined analytically. Instead, there are analytical recursive solutions, *i.e.*, the Kalman filter. Furthermore, in cases where an analytical solution is not present, extended Kalman filters and particle filters are the popular solutions that approximate the optimal Bayesian solution.

3.1 Kalman Filter

The posterior density in the Kalman filter is assumed to be Gaussian and parametrized by a mean and covariance. If $p(x_{t-1}|z_{1:t-1})$ is Gaussian it can be shown that $p(x_t|z_{1:t})$ is also Gaussian with the following assumptions:

- v_{t-1} and n_t are drawn from Gaussian distributions of known parameters.
- $f_t(x_{t-1}, v_{t-1})$ is a known linear function of x_{t-1} and v_{t-1}.
- $h_t(x_t, n_t)$ is a known linear function of x_t and n_t.

As a result, the system and measurement models, (3.1) and (3.2), become as follows:

$$x_t = F_t x_{t-1} + v_{t-1} \tag{3.6}$$

$$z_t = H_t x_t + n_t \tag{3.7}$$

F_t and H_t are known matrices defining the linear functions. The covariances of v_{t-1} and n_t, which are assumed to be of zero-mean and statistically independent, are Q_{t-1} and R_t, respectively. Note that the system and measurement matrices as well as noise parameters can be time variant.

The Kalman filter, derived from (3.3) and (3.4), can be expressed as the following recursive equations:

$$p(x_{t-1}|z_{1:t-1}) = \mathcal{N}\left(x_{t-1}; m_{t-1|t-1}; P_{t-1|t-1}\right) \tag{3.8}$$

$$p(x_t|z_{1:t-1}) = \mathcal{N}\left(x_t; m_{t|t-1}; P_{t|t-1}\right) \tag{3.9}$$

$$p(x_t|z_{1:t}) = \mathcal{N}\left(x_t; m_{t|t}; P_{t|t}\right) \tag{3.10}$$

where

$$m_{t|t-1} = F_t m_{t-1|t-1} \tag{3.11}$$

$$P_{t|t-1} = Q_{t-1} + F_t P_{t-1|t-1} F_t^T \tag{3.12}$$

$$m_{t|t} = m_{t|t-1} + K_t(z_t - H_t m_{t|t-1}) \tag{3.13}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1} \tag{3.14}$$

and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m, and covariance P. Moreover,

$$S_t = H_t P_{t|t-1} H_t^T + R_t \tag{3.15}$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1} \tag{3.16}$$

are the covariance of the innovation term $z_t - H_t m_{t|t-1}$, and the Kalman gain, respectively.

The above is the optimal solution to the tracking problem as long as the highly restrictive assumptions hold. According to literature, the Kalman filter provides the best result in a linear Gaussian environment.

3.2 Extended Kalman Filter

The above assumptions do not hold in most cases, and as a result, the Kalman filter cannot be exploited. In general, the system and measurement models, (3.1) and (3.2), are nonlinear and thus cannot be written as (3.6) and (3.7). The Extended Kalman Filter (EKF) approximates nonlinearity by local linearization of these functions. In this algorithm, $p(x_t|z_{1:t})$ is approximated by a Gaussian

$$p(x_{t-1}|z_{1:t-1}) \approx \mathcal{N}\left(x_{t-1}; m_{t-1|t-1}; P_{t-1|t-1}\right) \tag{3.17}$$

$$p(x_t|z_{1:t-1}) \approx \mathcal{N}\left(x_t; m_{t|t-1}; P_{t|t-1}\right) \tag{3.18}$$

$$p(x_t|z_{1:t}) \approx \mathcal{N}\left(x_t; m_{t|t}; P_{t|t}\right) \tag{3.19}$$

where,

$$m_{t|t-1} = f_t(m_{t-1|t-1}) \tag{3.20}$$

$$P_{t|t-1} = Q_{t-1} + \hat{F}_t P_{t-1|t-1} \hat{F}_t^T \tag{3.21}$$

$$m_{t|t} = m_{t|t-1} + K_t\left(z_t - h_t(m_{t|t-1})\right) \tag{3.22}$$

$$P_{t|t} = P_{t|t-1} - K_t \hat{H}_t P_{t|t-1} \tag{3.23}$$

and where $f_t(\cdot)$ and $h_t(\cdot)$ are nonlinear functions, and $\hat{F}_t$ and $\hat{H}_t$ are local linearization of these nonlinear functions:

$$\hat{F}_t = \left.\frac{df_t(x)}{dx}\right|_{x=m_{t-1|t-1}} \tag{3.24}$$

$$\hat{H}_t = \left.\frac{dh_t(x)}{dx}\right|_{x=m_{t|t-1}} \tag{3.25}$$

$$S_t = \hat{H}_t P_{t|t-1} \hat{H}_t^T + R_t \tag{3.26}$$

$$K_t = P_{t|t-1} \hat{H}_t^T S_t^{-1} \tag{3.27}$$

The EKF achieves linearization using the first term of the Taylor series expansion of the nonlinear function. A higher order EKF that considers further terms of the Taylor series has been achieved, but its intensive complexity has prevented it from being widely used.

3.3 Unscented Kalman Filter

Some researchers have proposed the use of the unscented transform in EKF, which yields the unscented Kalman filter (UKF) [30, 49, 50]. UKF considers a set of points that are deterministically selected from the Gaussian approximation to $p(x_t|z_{1:t})$. These points are all propagated through the nonlinearity, and the parameters of the Gaussian approximation are re-estimated. This filter has been shown to outperform EKF in some problems, mainly because of its better approximation of nonlinearity.

Nevertheless, the EKF and UKF both approximate $p(x_t|z_{1:t})$ to be Gaussian. If the true density is non-Gaussian (*i.e.*, bimodal), then a Gaussian will never be able to represent it satisfactorily, in which case, the particle filter will produce better results compared to EKF and UKF [4].

3.4 Particle Filter

The particle filter [9] or the sequential importance sampling (SIS) algorithm is a Monte Carlo (MC) method that forms the basis for most sequential Monte Carlo (SMC) methods developed so far [16, 17]. This SMC approach is also known as the CONDENSATION algorithm [27] and bootstrap filtering [25]. This technique implements the recursive Bayesian filter by MC simulations. In this approach, the posterior

density is represented by a set of random samples with associated weights, which are used to compute estimates. As the number of samples becomes very large, this MC approximation becomes an equivalent representation to the usual functional description of the posterior pdf, and the particle filter approaches the optimal Bayesian estimate.

Let $\{x_{0:t}^i, w_t^i\}_{i=1}^{N_s}$ denote a random measure that approximates the posterior pdf $p(x_{0:t}|z_{1:t})$, where $\{x_{0:t}^i, i = 0, \ldots, N_s\}$ is a set of support points with associated weights $\{w_t^i, i = 1, \ldots, N_s\}$ and $x_{0:t} = \{x_j, j = 0, \ldots, t\}$ is the set of all states up to time t. The weights are normalized such that $\sum_i w_t^i = 1$. Then, the posterior density at t can be approximated as

$$p(x_{0:t}|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_{0:t} - x_{0:t}^i) \tag{3.28}$$

which is a discrete weighted approximation to the true posterior, $p(x_{0:t}|z_{1:t})$. The weights are chosen using the principle of importance sampling [5, 17]:

Let $p(x) \propto \pi(x)$ denote a probability density function from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated (as well as $p(x)$ up to proportionality). Let $x^i \sim q(x), i = 1, \ldots, N_s$ be samples that are easily generated from a proposal $q(\cdot)$ called importance density. Then, a weighted approximation to the density $p(\cdot)$ is given by:

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \tag{3.29}$$

where

$$w^i \propto \frac{\pi(x^i)}{q(x^i)} \tag{3.30}$$

is the normalized weight of the i-th particle. As a result, considering the samples $x_{0:t}^i$ being drawn from the importance density $q(x_{0:t}|z_{1:t})$, the weights in (3.28) are defined by (3.30) to be

$$w_t^i \propto \frac{p(x_{0:t}^i|z_{1:t})}{q(x_{0:t}^i|z_{1:t})} \tag{3.31}$$

At each iteration of the sequential case, an approximation to $p(x_{0:t-1}|z_{1:t-1})$ is available, and it is desirable to approximate $p(x_{0:t}|z_{1:t})$ with a new set of samples. If the importance density is chosen such that

$$q(x_{0:t}|z_{1:t}) = q(x_{0:t-1}|z_{1:t})q(x_{0:t-1}|z_{1:t-1}) \tag{3.32}$$

then samples $x^i_{0:t} \sim q(x_{0:t}|z_{1:t})$ can be obtained by augmenting each of the existing samples $x^i_{0:t-1} \sim q(x_{0:t-1}|z_{1:t-1})$ with the new state $x^i_t \sim q(x_t|x_{0:t-1}, z_{1:t})$. To obtain the weight update equation, $p(x_{0:t}|z_{1:t})$ is expressed in terms of $p(x_{0:t-1}|z_{1:t-1})$, $p(z_t|x_t)$, and $p(x_t|x_{t-1})$.

$$\begin{aligned} p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|x_{0:t}|z_{1:t-1})p(x_{0:t}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x_{0:t}|z_{1:t-1})p(x_t|x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})}p(x_{0:t-1}|z_{1:t-1}) \\ &= \frac{p(z_t|x_t)p(x_t|x_{t-1})}{p(z_t|z_{1:t-1})}p(x_{0:t-1}|z_{1:t-1}) \end{aligned} \tag{3.33}$$

$$p(x_{0:t}|z_{1:t}) \propto p(z_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|z_{1:t-1}) \tag{3.34}$$

The weight update equation can be derived by substituting (3.32) and (3.34) into (3.31):

$$\begin{aligned} w^i_t &\propto \frac{p(z_t|x^i_t)p(x^i_t|x^i_{t-1})p(x^i_{0:t-1}|z_{1:t-1})}{q(x^i_t|x^i_{0:t-1}, z_{1:t})q(x^i_{0:t-1}|z_{1:t-1})} \\ &= w^i_{t-1}\frac{p(z_t|x^i_t)p(x^i_t|x^i_{t-1})}{q(x^i_t|x^i_{0:t-1}, z_{1:t})} \end{aligned} \tag{3.35}$$

In the common case when only a filtered estimate of $p(x_t|z_{1:t})$ is required at each time-step, the importance density becomes only dependent on x_{t-1} and z_t, *i.e.*, $q(x_t|x_{0:t-1}, z_{1:t}) = q(x_t|x_{t-1}, z_t)$. Therefore, only x^i_t need to be stored and the history of the states $(x^i_{0:t-1})$ and observations $(z_{1:t-1})$ is disregarded. The weight in this case becomes

$$w^i_t \propto w^i_{t-1}\frac{p(z_t|x^i_t)p(x^i_t|x^i_{t-1})}{q(x^i_t|x^i_{t-1}, z_t)} \tag{3.36}$$

Finally, the posterior filtered density $p(x_t|z_{1:t})$ can be approximated as

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w^i_t \delta(x_t - x^i_t) \tag{3.37}$$

It can be shown that as $N_s \rightarrow \infty$, this approximation approaches the true posterior density $p(x_t|z_{1:t})$. The particle filter consists of recursive propagation of the weights and support points as each measurement is received sequentially.

3.5 Particle Filter in Tracking

In the high-level approach to tracking, objects are associated between consecutive frames by generating a set of object hypotheses, followed by evaluation of the corresponding likelihood for each frame based on the most recent measurement. The particle filter is able to represent multiple hypotheses simultaneously. In addition, it is one of the most efficient object tracking methods in nonlinear situations that involve non-additive and non-Gaussian noise.

3.5.1 CONDENSATION

CONDENSATION algorithm [27] uses "factored sampling", in which the probability distribution of possible interpretations is represented by a randomly generated set. It exploits dynamic models (transition or prior densities) along with visual observations (measurements), to propagate the random set over time. Given the prior, and an observation density that characterizes the statistical variability of image data z given a state x, a posterior distribution can, in principle, be estimated for x_t given z_t at successive times t.

Spatio-temporal tracking has been dealth thoroughly by Kalman filtering, in the relatively clutter-free case in which densities can be modeled as Gaussian. These solutions produce relatively poor results in clutter which causes the density for x_t to be multi-modal and therefore non-Gaussian.

The state of the modeled object at time t is denoted x_t and its history is $X_t = \{x_1, \ldots, x_t\}$. The set of observations at time t is z_t with history $Z_t = \{z_1, \ldots, z_t\}$.

No assumptions on linearity, Gaussian behavior or unimodal distribution are made. CONDENSATION is an iterative algorithm with each time-step a self-contained iteration of factored sampling. The output of an iteration at each time-step is a weighted sample set $\left\{s_t^{(n)},\ n = 1, \ldots, N\right\}$ with weights $\pi_t^{(n)}$, representing approximately the conditional state density $p(x_t|Z_t)$ at time t. The method begins with a prior density at time t, which is the posterior prediction at time $t-1$, *i.e.*, $p(x_t|Z_{t-1})$. This prior is derived from the output of the previous time-step, $\left\{\left(s_{t-1}^{(n)}, \pi_{t-1}^{(n)}\right),\ n = 1, \ldots, N\right\}$ of $p(x_{t-1}|Z_{t-1})$. The aim is to maintain, at successive time-steps, sample sets of fixed size N. The first step is to sample N times from the set $\left\{s_{t-1}^{(n)}\right\}$, choosing a given element with probability $\left\{\pi_{t-1}^{(n)}\right\}$. The elements with high weights may be chosen several times in the new set, while others with relatively low weights may not be chosen at all. Next, each element in the new set undergoes the predictive steps, *i.e.*, drift and diffusion. At this stage, the sample set $\left\{s_t^{(n)}\right\}$ for the new time-step has been generated with no associated weight. In the final step, the observation density $p(z_t|x_t)$ is used to generate weights, leading to the sample-set representation $\left\{\left(s_t^{(n)}, \pi_t^{(n)}\right)\right\}$ of state-density for time t.

3.5.2 ICONDENSATION

Followed by the introduction of CONDENSATION algorithm for visual tracking, a probabilistic framework, *i.e.* ICONDENSATION [28], was proposed to integrate the low-level and high-level tracking approaches using the statistical approach of importance sampling combined with the CONDENSATION algorithm.

Importance sampling provides a mathematically principled way of directing search, combining prediction based on the previous object state with any additional measurement available from auxiliary sensors. As a result, it allows the system to benefit from the diversity of information sources and avoid temporary tracking failures imposed by one of the measurement processes. An implementation of ICONDENSATION has

been performed using color segmentation to detect skin-colored blobs and incorporating this information with a hand contour tracker.

3.5.3 Color-based Probabilistic Tracking

The deterministic methods exploiting color histogram principle rely on the deterministic search of a window whose color content matches a reference histogram color model. Bradski [8] uses a histogram of skin color in HSV color space to determine the likelihood of skin occurring at each pixel, using histogram back-projection to replace each pixel with the probability associated with that HSV value in the skin color histogram. In [12] the target appearance model is a distribution of colors represented by a histogram probability $\hat{q}_u$, which is compared with a histogram of target candidate $\hat{p}_u$ observed within the current mean-shift window. The comparison is based on the histogram similarity using the Bhattacharyya coefficient. Basically, the current frame is deterministically searched for a region, a fixed-shape variable-size window, whose color content best matches a reference color model. Starting from the final location in the previous frame, it proceeds iteratively at each frame so as to minimize a distance measure to the reference color histogram. Excellent tracking results on complex scenes are demonstrated in [8, 10, 12]. This deterministic search might however run into problems when parts of the background nearby exhibit similar colors or when the tracked object is completely occluded for a while.

Perez *et al.* [43] have applied the SMC tracking technique on a tracker based on the color histogram distance. Incorporating the particle filter allows better handling color clutter in the background, as well as complete occlusion of the tracked objects over a few frames. Their goal is to track objects of *a priori* unknown nature but of a specific interest, *e.g.*, moving objects. In their approach, the input video frame is searched against a global color reference model describing the appearance of the object, and endogenous initialization, *i.e.*, extracted from the initial frame of the

studied sequence. This type of tracker is very useful for tracking objects of interest that are of any kind and show drastic spatial changes through the sequence, due to pose changes, partial occlusions, *etc.* It relies on the same principle of comparing color contents of candidate regions with a reference color histogram, while being embedded within a SMC framework. This requires construction of a color likelihood based on color histogram distances, coupling of this data model with a dynamical state space model, and sequential approximation of the resulting posterior distribution with a particle filter. The use of a sample-based filtering technique allows the simultaneous tracking of multiple posterior hypotheses, which is very crucial to avoid background distraction and recover after partial or complete occlusions. A second-order auto-regressive dynamics is chosen as the dynamic model of the system. The color model is obtained by histogram technique in the HSV color space [21]. Within the candidate region, a kernel density estimate of the color distribution at each time t, $q_t(x) = \{q_t(n;x)\}_{n=1...N}$, is used as the color model. This model associates a probability to each of the N color bins. At time t, the color model $q_t(x)$ associated with a hypothesized state x is compared to the reference color model $q^* = \{q^*(n)\}_{n=1...N}$, which is normalized and constructed at an initial time t_0 at a location $x^*_{t_0}$, either manually or automatically by a detection module. The likelihood function must give importance to the candidate histograms with minimum distance to the reference histogram. The Bhattacharyya distance based on the Bhattacharyya coefficient is used to identify the closest matches.

Another approach for applying the particle filter method in visual tracking was developed in [14]. Here, a color distribution in an upright circular region is used as target models, and an unconstrained Brownian model is considered for the dynamic model. In the tracking stage, the estimated state is updated at each time-step by incorporating new observations, using the Bhattacharyya coefficient as their similarity measure. In other words, the tracker uses the Bhattacharyya distance to update a

priori distribution provided by the particle filter. To weigh the sample set, the Bhattacharyya coefficient is computed between the target histogram and the histogram of the hypotheses.

3.6 Other Methods

3.6.1 Mean Shift Embedded Particle Filter

The particle filter performs a random search guided by a stochastic motion model to obtain an estimate of the posterior distribution describing the object's configuration. On the other hand, mean shift, a typical and popular variational method, localizes an object based on minimizing a cost function. The search method of the particle filter is stochastic and model-driven, while in mean shift, it is deterministic and data-driven. In addition, the particle filter applies a recursive Bayesian filter based on propagation of sample set over time, maintaining multiple hypotheses at the same time and using a stochastic motion model to predict the position of the object. Maintaining multiple hypotheses allows the tracker to handle clutters in background, also recover from failure or temporary distraction. Mean shift, on the other hand, uses only one hypothesis, which is computationally effective but is prone to converge to local maximum. A common problem in conventional particle filters is the degeneracy phenomenon, where all but one sample will have negligible weight after a few iterations [3]. In other words, these samples may have very low likelihood and their contribution to the posterior estimation becomes insignificant, which is computationally ineffective.

In [46], a combination of particle filtering and mean shift for object tracking is presented in the form of the mean shift embedded particle filter, integrating advantages of both methods. One outcome is to overcome the degeneracy problem of particle filters. They applied their algorithm on hand tracking, choosing skin color as the feature of the hand. The skin color model is adapted frame-by-frame during track-

ing to handle skin color variations over time due to illumination changes. In their approach, mean shift analysis is applied to each sample based on observation density, after being weighted by observation. After mean shift iterations, samples are "herded" to the local modes of the observation. Since the samples are moved to have large weights, the algorithm concentrates on samples with large weights. Therefore, the degeneracy problem is efficiently overcome. Also, if the iteration times are set properly, the resultant samples will not contain too many repeated points and the problem of impoverishment is reduced.

3.7 Summary

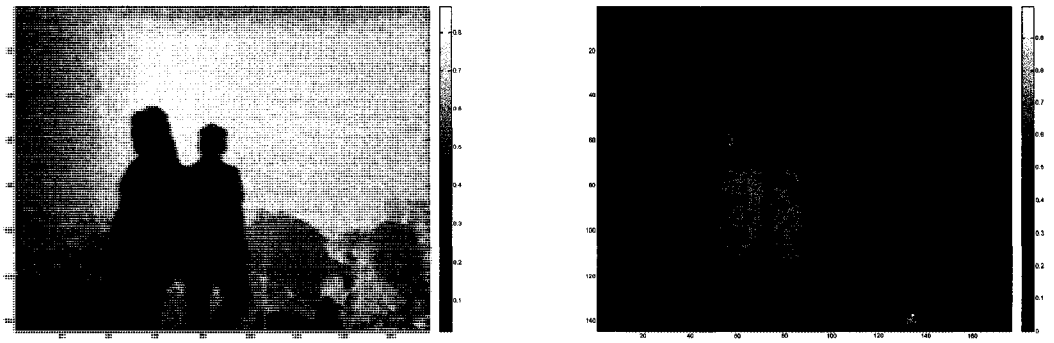
The probabilistic Bayesian filtering approaches including Kalman filter, EKF, UKF, and the particle filter have been discussed in this chapter. The particle filtering technique is chosen to be incorporated in this research, mainly because it is efficient in sampling the underlying state-space distribution of non-linear and non-Gaussian processes.

Chapter 4

Probabilistic 3D Tracking Based on Adaptive Depth Segmentation

This chapter presents the problem formulation and implementation steps of the proposed probabilistic object tracking based on the TOF sensor data. The goal is to detect objects of interest in the scene and consecutively track them through video sequences obtained by TOF sensor.

The 3D TOF sensor delivers for each pixel the coordinates x, y, z as well as the gray-scale intensity value i. By constructing an image of z values for all the pixels in the scene, the depth map image becomes available. As mentioned before, this work is focused on exploiting depth information for applications where intensity data are not promising — due to environmental illumination changes, absence of light source, *etc.* The depth and intensity outputs of the TOF sensor are shown in Fig. 4.1, and the corresponding 3D depth map is demonstrated in Fig. 4.2. The environment in which the tracking task is expected is supposed to be unconstrained. That is, the



(a) Depth image (b) Intensity image

Figure 4.1: Depth and gray-scale intensity outputs of a TOF sensor

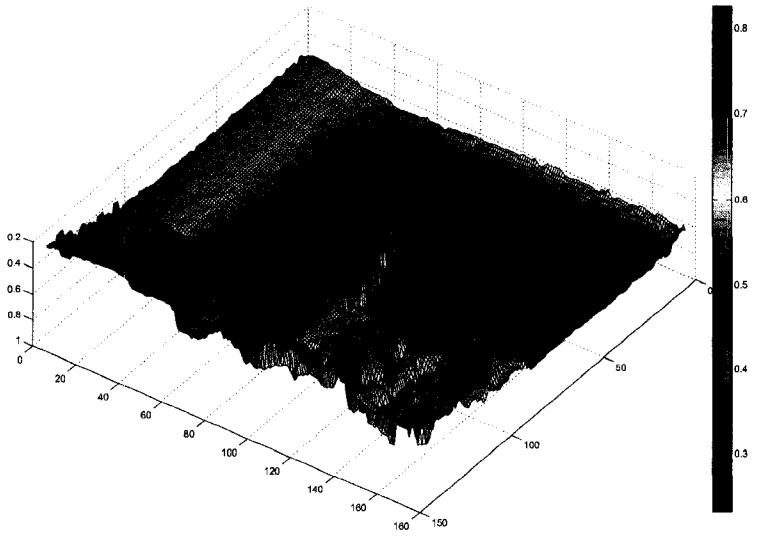


Figure 4.2: 3D Depth map representation of the depth output obtained from a TOF sensor

scene can have

- cluttered background
- various lighting settings
- and complex motion patterns.

Also, multiple people can be present in the scene, navigate, enter, and exit the tracking environment.

4.1 Adaptive Depth Segmentation

This section describes a novel approach to segment objects in cluttered 3D environments using depth map distribution produced from a depth histogram of the scene. In the initial processing stage, depth images derived from the TOF sensor are passed through a noise-removing filter to remove noisy depth measurements around the object boundaries. The following processing steps are depth histogram evaluation, determination of the depth density function, segmentation using depth extrema, object detection and object association.

4.1.1 Depth Histogram Evaluation

Interpretation of depth histogram differs from color histogram in several aspects. The most important advantage of using depth histogram is that it provides the user with depth guidance to evaluate the scene. As shown in Fig. 4.3, the horizontal axis (bins) in this type of histogram represents the distance to the origin, *i.e.* the camera, and the vertical axis represents density of the pixels that fall into each distance bin. More specifically, higher values on the depth bin axis correspond to farther distances in the actual scene. Hence, using the depth histogram one can achieve a better depiction of the depth layout, which is not available in 2D images or color histograms. The depth

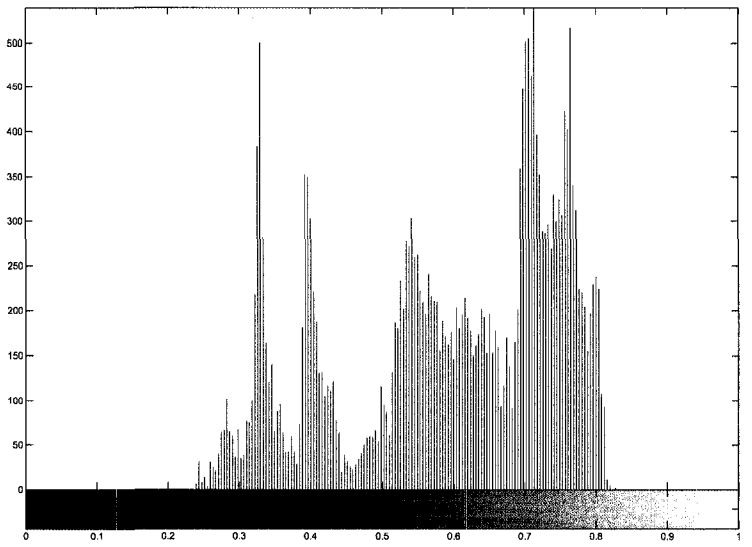


Figure 4.3: Depth histogram presentation of a 3D scene

image is denoted by $\mathbf{I} : \Re^2 \rightarrow (\alpha, \beta)$ where (α, β) is the dynamic range of the pixel values. The discrete depth histogram of $\mathbf{I}$ over N bins is defined as

$$h_z = \{h_z(i)\}_{i=1,2,..,N} \tag{4.1}$$

where, $h_z(i)$ corresponds to the number of pixels that are located at distance i from the camera.

4.1.2 Depth Density Function

In order to segment depth images, it is necessary to form the underlying continuous distribution that the discrete histogram measurements h_z are drawn from. For this purpose, the kernel density estimation technique is applied to approximate this distribution from depth histogram information to facilitate gradient estimation and local extrema detection. Denoting the scalar measurements z_i, $i = 1, \ldots, N$ from a depth distribution $H(z)$, the corresponding kernel density estimate is achieved using

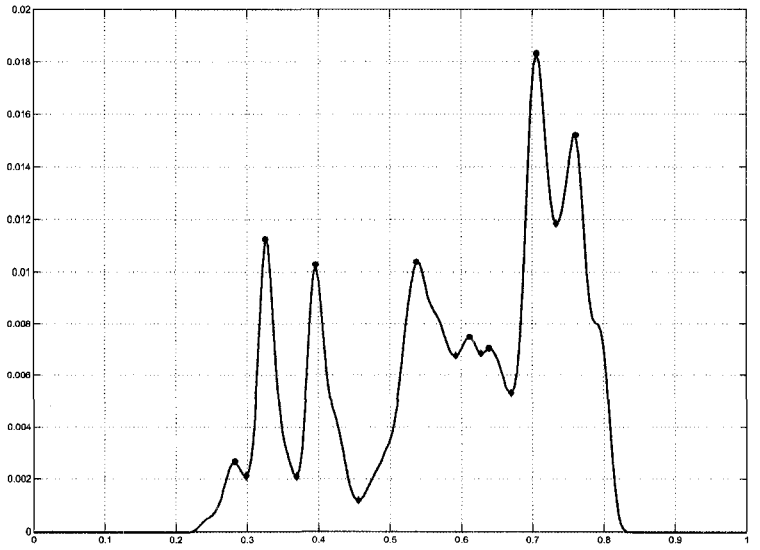


Figure 4.4: Depth density function of a depth histogram

a kernel function $K(u)$ with a bandwidth λ as

$$\hat{H}(z) = \frac{1}{N\lambda} \sum_{i=1}^{N} K\left(\frac{z - z_i}{\lambda}\right) . \tag{4.2}$$

$\hat{H}(z)$ has all the properties of a pdf, and thus is called the depth density function. The resulting depth density function for the depth histogram of Fig. 4.3 is given in Fig. 4.4.

4.1.3 Range Segmentation from Extremum Data

In the next step, the resulting depth density function $\hat{H}(z)$ is further analyzed to derive the local maxima and minima vectors, Γ and γ from equations (4.3) and (4.4), respectively. The *i-th* element of Γ over an interval $[a, b]$ where the distribution is unimodal is expressed as:

$$\Gamma_i = \arg\max_{z \in [a,b]} \hat{H}(z) . \tag{4.3}$$

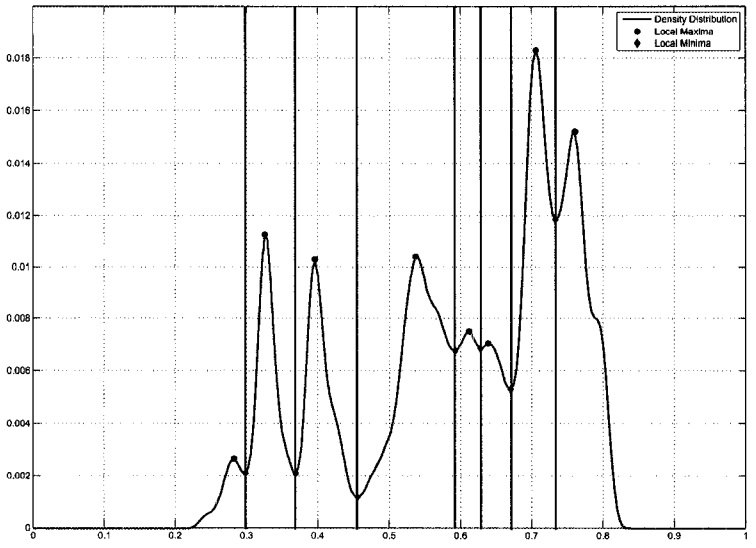


Figure 4.5: Range dividers for extremum segmentation

Then,

$$\gamma_j = \arg\min_{z \in \mathbb{R}_j} \hat{H}(z) \ , \tag{4.4}$$

$$\mathbb{R}_j = \{\Gamma_j \leq z < \Gamma_{j+1}\} \tag{4.5}$$

where, $j = 1, 2, \ldots, M-1$, and M is the total number of the local maxima.

Upon determination of the local extremum points of the depth density function, a set of range dividers S_k, $k = 1, \ldots, M+1$, can be evaluated from (4.6). To better illustrate this process, the corresponding local extremum points as well as range dividers for Fig. 4.4 are highlighted in Fig. 4.5.

$$S_k = \begin{cases} \alpha & k = 1 \\ \gamma_{k-1} & 2 \leq k \leq M \\ \beta & k = M+1 \end{cases} \tag{4.6}$$

These dividers are used to partition the scene into different depth divisions $\mathbf{D}_l$ in

Figure 4.6: Binary depth divisions resulted from the range segmentation approach

order to separate adjacent and overlapping objects, denoted by:

$$\mathbf{D}_l(x,y) = \begin{cases} 1 & S_l \leq \mathbf{I}(x,y) < S_{l+1} \\ 0 & \text{otherwise} \end{cases}, \tag{4.7}$$

where $l = 1, 2, \ldots, M$, and $\mathbf{I}$ is the depth image. (x, y) correspond to the horizontal and vertical pixel coordinates in the image, respectively.

In essence, each division forms a binary image containing pixels with depth values between two consecutive range dividers. A set of depth divisions achieved with this approach is demonstrated in Fig. 4.6. It is noteworthy to mention that range dividers are chosen adaptively for each image. Adaptive selection ensures that the algorithm can be applied on unconstrained environments without *a priori* information about the scene, including number of objects and background settings.

4.1.4 Object Detection

In the final stage of segmentation, connected component analysis is exploited on each of the derived depth divisions to detect object blobs in the scene. This is followed by a size filter with a minimum area threshold of Δ_a to eliminate insignificant and trivial

regions of isolated noises or inter-objects pixels. There exists at least one object for each division $\mathbf{D}_l$ that is localized in that depth range.

$$\mathbf{D}_l = \sum_{i=1}^{N_1} \Omega_i + \sum_{j=1}^{N_2} \varepsilon_j \ , \tag{4.8}$$

where N_1 and N_2 represent the total number of significant objects and insignificant regions in the corresponding depth division, respectively.

The total number of objects in the scene is determined by inspecting each division's objects, as stated above. These objects can be further classified based on their properties such as their associated mean depth in order to be exploited in the subsequent procedures. Also, human candidates are detected based on their geometric features, *e.g.*, aspect ratio and relative size to depth mean. Segmentation output of this method is further illustrated in Fig. 4.7, where each object is assigned a segmentation label. By further analyzing the properties of segmented objects, objects of interest can be detected, as demonstrated in Fig. 4.8.

4.1.5 Object Association

To compare and match two objects from consecutive frames, it is necessary to form a similarity measure using a distance metric between their signatures. An object signature is defined by a concatenation of its X, Y, and Z histograms as

$$s = [h_x \ h_y \ h_z] \ . \tag{4.9}$$

There exist several metrics including the Euclidean distance, histogram intersection, Bhattacharya distance, *etc.* Here, a similarity metric derived from the Bhattacharya coefficient is used since it has been established as an efficient metric for comparing arbitrary histogram-based distributions [13]. The distance between two discrete distributions is defined as

$$d(h_i, h_j) = \sqrt{1 - \rho[h_i, h_j]} \tag{4.10}$$

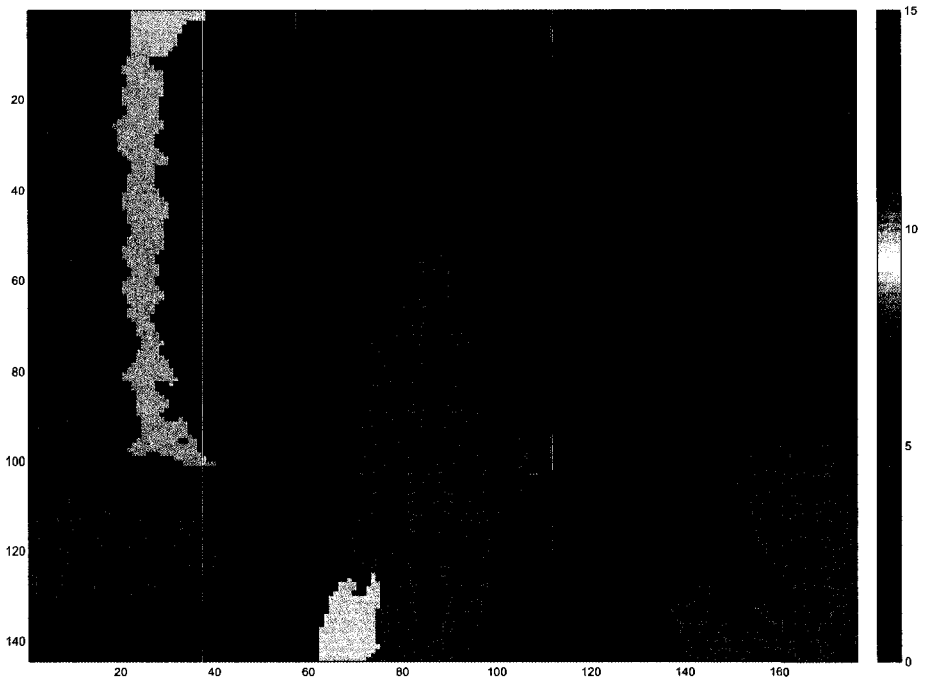


Figure 4.7: Object segmentation output from analysis of depth divisions

where

$$\rho[h_i, h_j] = \sum_{n=1}^{N} h_i(n) h_j(n) \tag{4.11}$$

4.2 Probabilistic Method: Particle Filter

In this work, the particle filter is employed to probabilistically associate objects between consecutive frames. The advantage of incorporating particle filtering in tracking is that it is highly efficient in cluttered environments as well as being robust to object occlusion. This section aspires to explain the characteristics of the particle filter, including the proposal and likelihood distributions, as well as the required formulation to update the particle weights during the update stage. By employing the particle fil-

Figure 4.8: Objects of interest, detected from segmentation image using geometric features

ter, it is feasible to achieve an efficient, depth-based human tracking algorithm using a TOF range sensor.

Based on the discussed information in chapter 3, particle filtering — as a nonlinear filtering method — characterizes the tracking target by the state sequence $\{x_t\}_{t=0,1,\ldots}$, whose transition density is specified by the dynamic model $x_t = f_t(x_{t-1}, v_{t-1})$. Only the measurements $\{z_t\}_{t=1,2,\ldots}$ are available, which are used along with the likelihood model to estimate the state x_t through the prediction and update stages. In the particle filter, the posterior density is given by (3.37) and the weight update equation is determined from (3.36). In order to determine an estimation of the state at time t, the posterior density should be known, necessitating the calculation of the transition,

likelihood, and proposal distributions of (3.36).

4.2.1 Proposed Transition Distribution

Human body at time t is represented by a scalable rectangle, $x_t = \{c_t^x, c_t^y, w_t, h_t\}$ where, (c_t^x, c_t^y) are the center coordinates and (w_t, h_t) are the corresponding width and height of the bounding box. Human motion is modeled with a first order dynamic model as (4.12):

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{K}_{t-1}\mathbf{N}_{t-1} \tag{4.12}$$

where, $\mathbf{A}$ and $\mathbf{N}_{t-1}$ represent the deterministic component of this model and a multivariate Gaussian random variable, respectively. Constant velocity model with $\mathbf{A} = \mathbf{I}$ is chosen for the nature of random human walk. $\mathbf{K}$ is related to the object's velocity in the sense that it increases when the person is moving with higher velocity — causing an increase in the variance of the process noise. As a result, samples are propagated over a larger area in the state space to increase the efficiency of object localization for faster human motions.

4.2.2 Proposal Distribution

The proposal distribution is a combination of high- and low-level approaches in the sense that particles are drawn from the transition prior distribution by using the CONDENSATION algorithm for tracking. In addition, samples are propagated through a Gaussian distribution obtained from low-level processing of the proposed adaptive depth segmentation in Section 4.1, centered at the object's center coordinates. The proposed segmentation process is exploited to detect human objects as they enter the scene, while detecting them in each frame, using the corresponding depth density function. A Gaussian distribution is considered as the system's proposal, having the object's center and covariance.

4.2.3 Likelihood Distribution

For each particle i, a likelihood distribution is considered by using a distance measure between the corresponding target and the object model.

$$p\left(z_t|x_t^i\right) = 1 - d^i\left(S_T, S_M\right) \tag{4.13}$$

d^i is the distance measure between the depth signatures of the object model and the target corresponding to particle i. As mentioned before in Section 4.1.5, the Bhattacharya distance is employed to derive a similarity measure that can be leveraged to form the likelihood model in (4.13).

4.2.4 Weight Measurement

At each iteration, samples are drawn from the discussed proposal distribution in the previous frame. The prediction step includes sampling from a Gaussian distribution resulted from the segmentation algorithm, as well as sampling from the predicted area determined by the transition density. At the update stage, each of the new samples is weighted based on the available measurements z_t obtained from segmentation of the current frame. Sample weights are then normalized such that:

$$\sum_{i=1}^{N_s} w_t^i = 1 \tag{4.14}$$

In the final step, the person's location is determined as follows:

- The achieved weights are categorized into L clusters.
- Centers of the clusters are assigned to their corresponding segmented objects, known from the most recent measurement.
- The Bhattacharya distance metric is calculated between the signatures of the object model in the previous frame and the object candidates defined by cluster means in the current frame.

- The cluster with the least Bhattacharya distance is investigated and if the corresponding distance satisfies a high confidence, the person's location is updated to be the expected value of that cluster.

4.3 Summary

In this chapter, design and implementation of the proposed TOF object tracking method has been demonstrated. An adaptive depth segmentation technique has been developed to detect objects of interest in each frame. The depth histogram of TOF input has been leveraged to derive a depth density distribution conveying key range segmentation information. For tracking the objects, the particle filter has been utilized to perform object association between frames. Also, the proposal distribution used in the particle filter has been designed so that it includes both bottom-up and top-down approaches to enhance the tracking efficiency in nonlinear and complex situations.

The next chapter will address the experimental results performed in order to verify the operational efficiency and performance of the developed TOF object segmentation and tracking approach discussed in this chapter.

Chapter 5

Experimental Results

Performances of the proposed segmentation and tracking algorithms have been evaluated by carrying out experiments using SR-3000 TOF sensor with a resolution of 176×144 pixels. The operating range of this sensor is from 30 cm to 7.5 m with a field of view (FOV) of 47.5° × 39.5°. Range images can be acquired up to a frame rate of 29 fps with and without environmental illumination. More than 10 video sequences, each including over 200 frames, were used for the experiments. Various object tracking scenarios have been considered, while no constraint has been assumed on the environment. There has been diversity in background selection including cluttered, semi-cluttered, and plain cases. Also, a diverse number of people with differences in size and sex have been present in tracking scenes.

5.1 Proposed Depth Segmentation Evaluation

5.1.1 Performance Comparison

As the representative of iterative segmentation techniques, an edge-based segmentation method is chosen for comparison with the proposed approach in chapter 4. Both algorithms have been applied to the same TOF dataset for all of the sequences. As discussed earlier, the performance of edge-segmentations highly depends on the number of iterations, as well as the choice of edge threshold. The edge-based algorithm has been applied with two and three iteration cycles, and with adaptive thresholds. The results were compared to those of the proposed segmentation (Fig. 5.1), where it was observed that the edge-method produces under-segmented results, even though the number of iterations has been increased. This is due to the fact that discontinuities in curved surfaces are not easily recognized by edge detection. Furthermore, edge removal at each iteration leads to distortion of the neighboring regions, as seen in Fig. 5.1.

5.1.2 Operational Efficiency

To verify the operational efficiency of the proposed segmentation method, performance analysis has been exploited on a diverse database of TOF depth images. Table 5.1.2 demonstrates the significant efficiency of the adaptive depth segmentation method compared to the employed edge segmentation technique. On average, the proposed algorithm operates 1.5 times faster than an edge segmentation with two iteration levels, and 2.5 times faster than a three iteration-level edge segmentation. In addition, the number of segmented objects using the proposed method is significantly higher compared to the other methods. These experiments show the non-promising results of iterative segmentation approaches, and verify the efficiency and high performance of incorporating depth-based segmentation in the tracking problem.

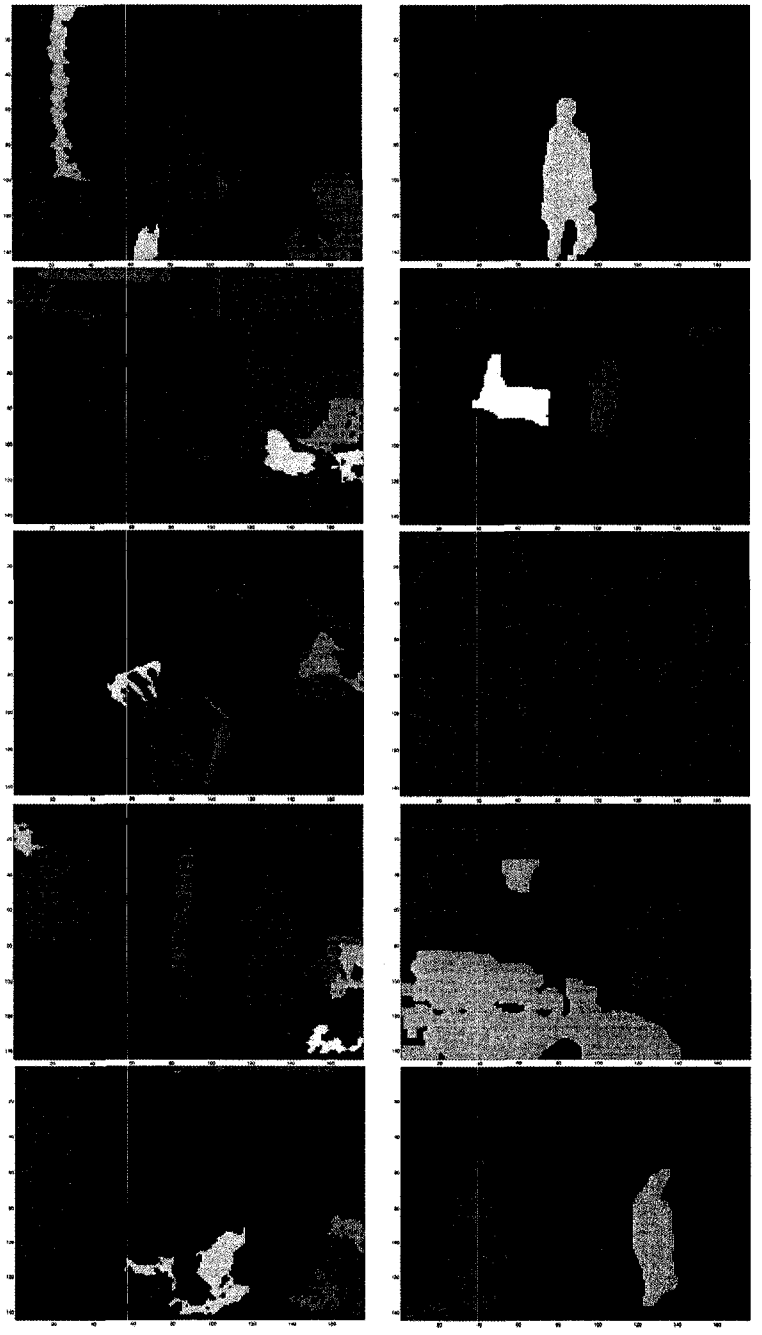


Figure 5.1: Comparison of the proposed depth-based segmentation (left) with edge segmentation algorithms (right)

Segmentation Method	Mean No. of objects	Run-time
Proposed adaptive depth approach	17.97	1×
Two iteration edge-based method	7.92	1.49×
Three iteration edge-based approach	6.70	2.55×

Table 5.1: Performance Analysis of the proposed depth segmentation method

5.2 Proposed Tracking Approach Evaluation

5.2.1 Handling Scale Variations in Cluttered Scenes

The proposed tracker was evaluated in a cluttered office environment with a person entering the scene and moving through the background. The corresponding tracking results of this person is demonstrated in Fig. 5.2. The tracker has been shown to be efficient throughout rapid scale changes.

5.2.2 Handling Occlusions in Inadequate Illumination

In low illumination settings, successful tracking is feasible, as the results show the robustness of the tracking system against weak lighting conditions in Fig. 5.3. Furthermore, the tracker is able to handle occlusions since it uses particle filtering with multiple hypothesis consideration. In Fig. 5.3 and Fig. 5.4, multiple people enter and walk through the scene, while coinciding with each other sporadically from different directions and with various poses. Tracking each of the objects is achieved in spite of undergoing several partial and complete occlusions for a few frames, in which case particles are diffused around the proximity of the previous state until one of the hypotheses satisfies the confidence measure and updates the measurement weights accordingly.

Figure 5.2: The proposed tracker's results for rapid scale variation in cluttered background setting

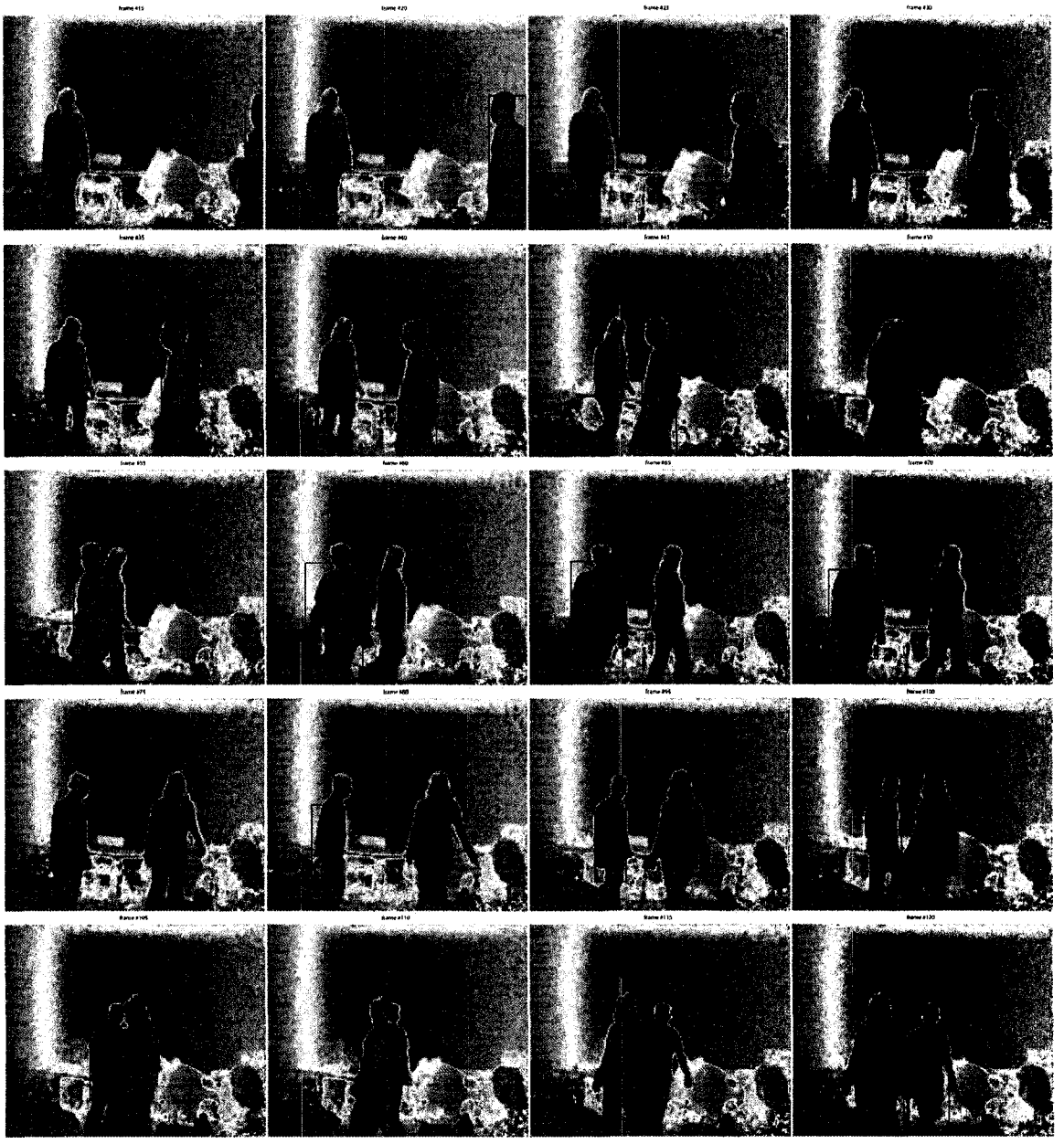


Figure 5.3: Tracking results under low illumination and occlusion

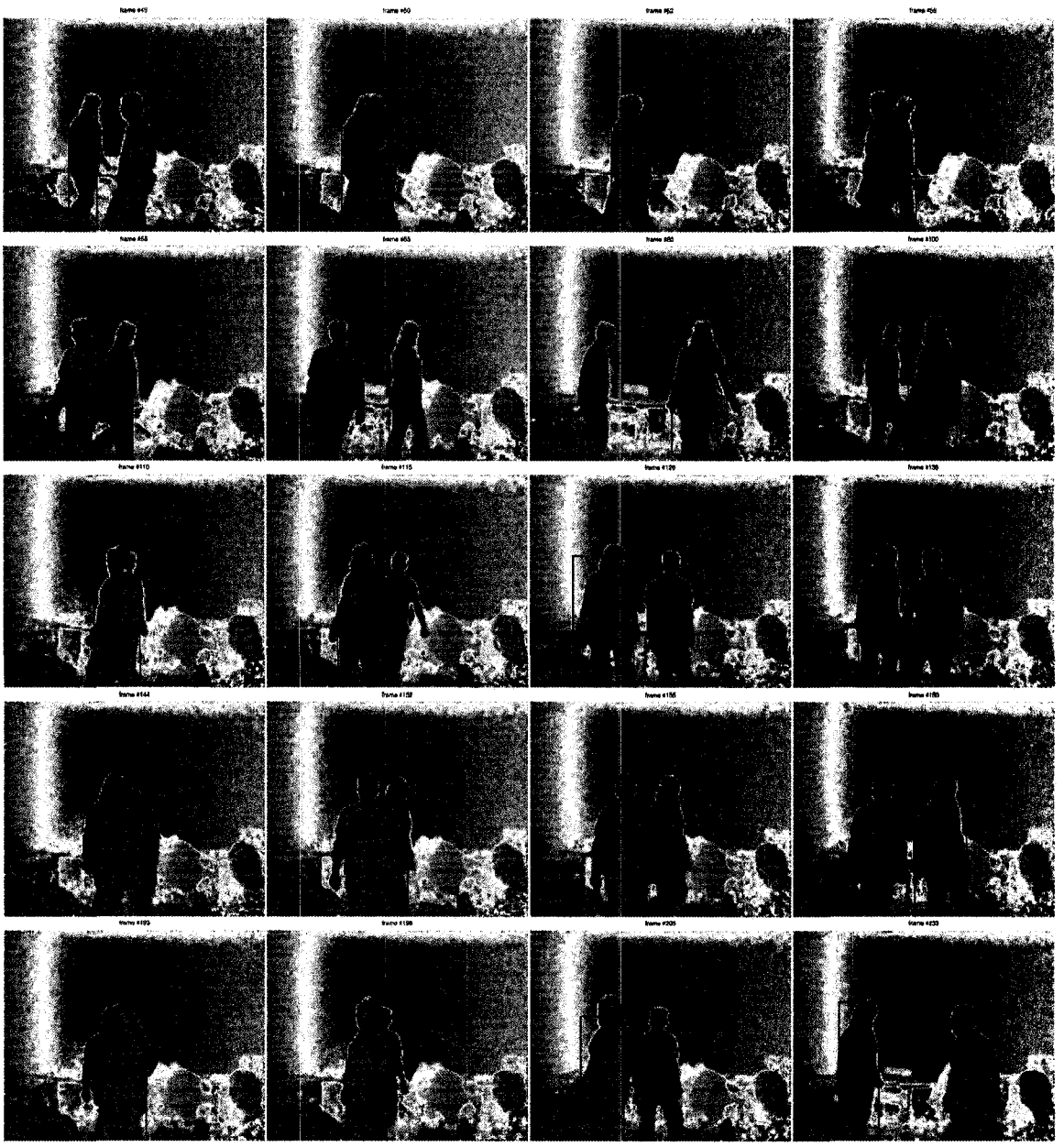


Figure 5.4: Tracking results of multiple people under low illumination and occlusion

5.2.3 Handling Rotation, Complex Motion and Self-Occlusion

One of the circumstances in which tracking methods fail is the occurrence of out-of-plain rotation. Since the exploited detection method in this thesis mainly relies on the depth information of the object, out-of-plain rotation will not issue a problem since depth values of the person's body do not change radically. In addition, complex object motion and rapid scale changes could result in tracking failure. In these cases, the transition distribution in section 4.2.1 is challenged, as the object motion does not comply with the considered constant velocity model. Fig. 5.5 and Fig. 5.6 summarize the output of the proposed algorithm, while evaluated under rapid scale changes and complex motion, as well as out-of-plain rotation and self-occlusion.

5.2.4 Performance in Noisy Environments Due to the TOF Nature

To evaluate the noise response of the system, the system has been set up in a noisy environment composed of a long, narrow hallway. The reflectance of the surrounding walls as well as the floor and ceiling on the TOF sensor leads to incorrect depth measurement of the objects. Nevertheless, the tracker is able to track the object and retrieve from noisy readings although the detection results are highly distorted. However, as the object moves away from the camera, it is impossible to distinguish its depth from noisy surroundings, which eventually results in tracking failure. Later on, as the object moves toward the camera, distortion in the detection stage is reduced and the tracker picks up the object again. The results of this test are demonstrated in Fig. 5.7.

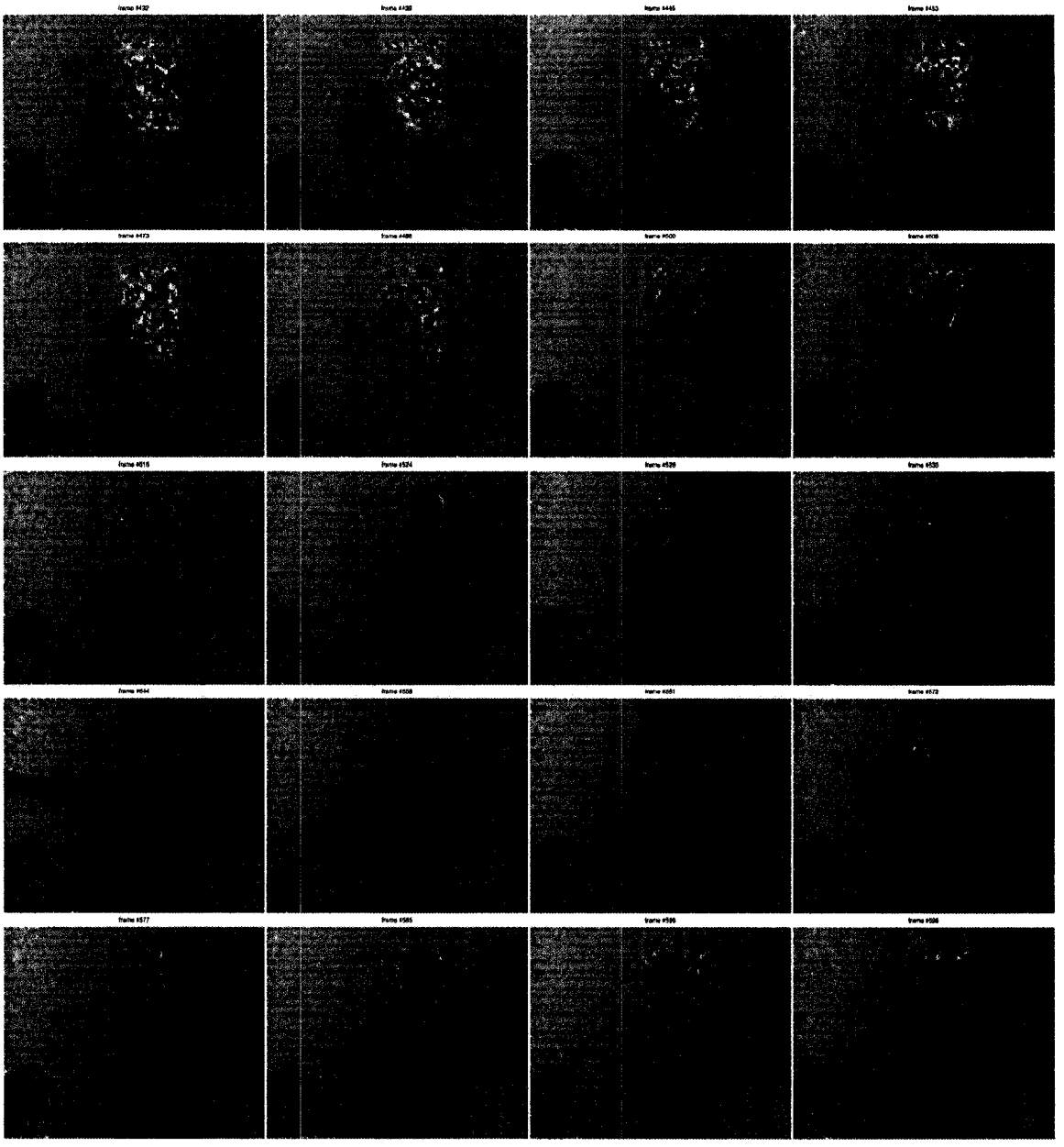


Figure 5.5: Successful tracking results under out-of-plane rotation with rapid pose change and complex motion

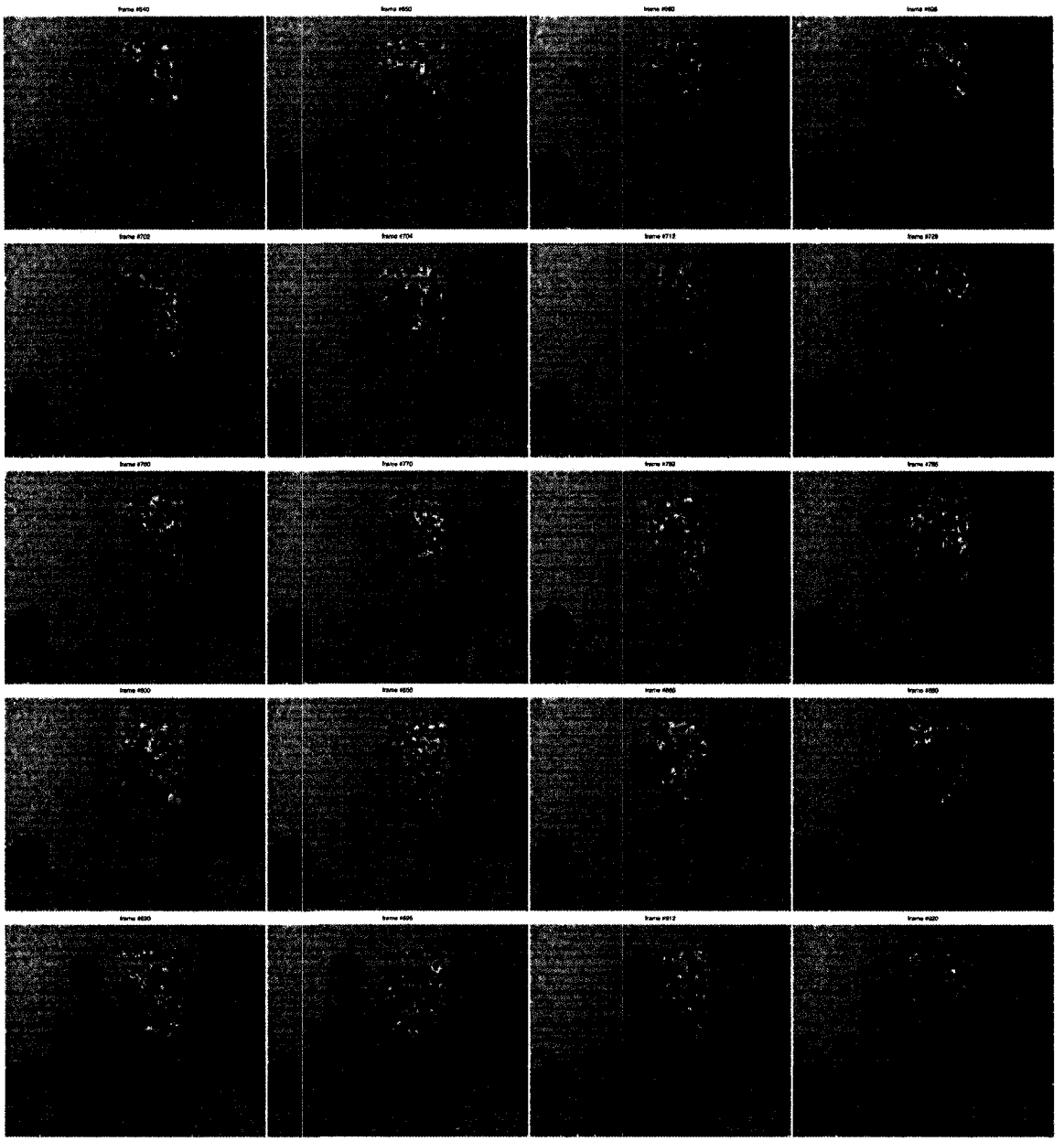


Figure 5.6: Successful tracking results under self-occlusion, rapid pose change and complex motion

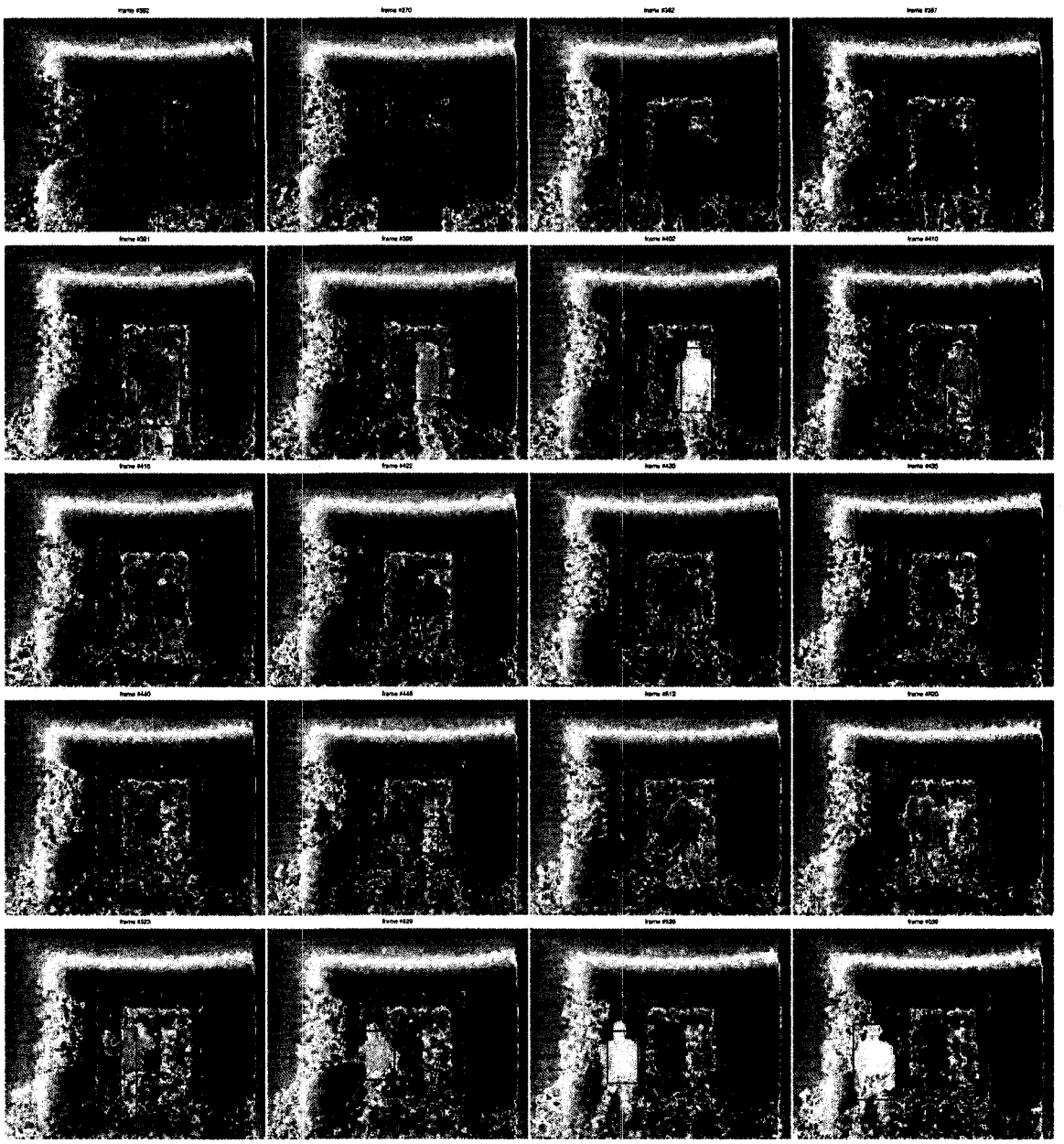


Figure 5.7: Tracking evaluation under noisy depth measurements

Chapter 6

Conclusions and Future Work

The main contributions of this thesis are summarized in this chapter, followed by suggestions regarding future development of the proposed research.

Visual tracking has been an active area of research for many years. With the advent of active 3D sensors such as TOF systems, which provide system designers with accurate pixel-level depth measurements with no post-processing, there has been an increasing interest in deploying these sensors in tracking systems. However, the massive dimensions as well as high costs of early active sensors have been an issue in developing tracking approaches using depth information. In recent years, CMOS-based implementation has led to the development of small TOF cameras with much lower cost. Benefits of exploiting TOF sensors are their low sensitivity to lighting variations and the availability of the third dimension that was not feasible without extensive computation in previous years. Using this added information, tracking system designers can develop more efficient algorithms to handle challenging cases including occlusion, rotation, and insufficient illumination.

6.1 Contributions

This thesis proposed a 3D object tracking methodology using TOF sensor information, which combines bottom-up and top-down approaches to achieve more efficient results. The main idea was to develop an adaptive depth segmentation method for object detection as the bottom-up process, while integrating it with the particle filter as the top-down association approach.

The major contributions of this thesis are as follows:

- An original range segmentation approach was presented, which segments images based on the depth density of the corresponding range histograms. The motivation to use depth histograms for segmentation originates from the fact that other segmentation techniques in literature do not translate to operate on depth images efficiently, as discussed in earlier chapters. The performed experiments verify the efficiency of the proposed segmentation algorithm as compared to other techniques.

- The segmentation process is adaptive to any scene structure since there is no manual threshold selection for object and background segmentation. In essence, depth density of each environment is evaluated independently to derive the required range dividers.

- An improved particle filter was introduced to be exploited in tracking. The proposal distribution of the particle filter was improved so that particles were drawn both from bottom-up and top-down approaches in order to achieve more efficient tracking results. As previously mentioned, the proposed depth segmentation method was chosen as the bottom-up process, while the transition distribution was selected to propagate particles in the top-down process.

- The proposed method is unaffected by illumination variation, as it relies on TOF information, which are not sensitive to environmental lighting changes. It

also produces promising results under no illumination due to the near-infrared nature of the sensor, as illustrated in chapter 5.

- The tracker produces efficient outcomes in cluttered backgrounds. This is due to the fact that particle filtering assumes multiple hypotheses simultaneously and evaluation of these hypotheses are maintained through a confidence measure. Therefore, clutter will not distort the results of tracking.

- One of the main challenges in tracking is object occlusions. Object — either in part or as a whole — can become occluded by itself (self occlusion) or by another object (inter-object occlusion). The proposed method has been verified to successfully track and retrieve objects after all of the discussed occlusion types.

- The proposed particle filter tracker has been designed and verified to succeed in complex object motion and in the case of multiple objects present in the sequence.

6.2 Future Work

The following are some of the possible avenues of future research that may be taken to further enhance the performance and functionality of the probabilistic 3D tracking approach:

- Current detection accuracy is at the blob level, since object details or micro-features cannot be determined based on range images alone, if not in close proximity. Thus, one approach would be to integrate range information with color data, leading more detailed detection results and enhanced tracking performance.

- The proposed research can be further expanded to include human behavior analysis and pose estimation based on the tracking outcomes.

- Current TOF sensors are applicable in indoor environments only, although the proposed method is not restricted to indoor scenes. With future advancements in design and implementation of TOF sensors, probabilistic 3D tracking can be achieved in outdoor applications by further investigating the proposed particle filter tracker using outdoor depth information.

References

[1] F. Ababsa, M. Mallem, and D. Roussel, "Comparison between particle filter approach and Kalman filter-based technique for head tracking in augmented reality systems," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1, pp. 1021–1026, 2004.

[2] R. Adams and L. Bischof, "Seeded region growing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 6, pp. 641–647, 1994.

[3] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. Adelaide, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[4] S. Arulampalam and B. Ristic, "Comparison of the particle filter with range-parameterized and modified polar EKFs for angle-only tracking," *Proceedings of SPIE*, vol. 4048, p. 288, 2000.

[5] N. Bergman, "Recursive Bayesian Estimation," *Linkping Studies in Science and Technology, Dissertation*, no. 579, 1999.

[6] A. Bevilacqua, L. Di Stefano, and P. Azzari, "People Tracking Using a Time-of-Flight Depth Sensor," *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, 2006.

[7] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion.* Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1998.

[8] G. Bradski, "Computer vision face tracking as a component of a perceptual user interface," *Workshop on Applications of Computer Vision*, vol. 1, pp. 214–219, 1998.

[9] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *Radar, Sonar and Navigation, IEE Proceedings-*, vol. 146, no. 1, pp. 2–7, 1999.

[10] H.-T. Chen and T.-L. Liu, "Trust-region methods for real-time tracking," *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 717–722, 2001.

[11] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 485–494, 2004.

[12] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 2000.

[13] ——, "Kernel-Based Object Tracking," *IEEE Computer Society*, 2003.

[14] E. Cuevas, D. Zaldivar, and R. Rojas, "Particle Filter in Vision Tracking," *Technical Report B-05-13, Freie Univ., Fachbereich Mathematik und Informatik*, 2005.

[15] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, 2000.

[16] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," *Sequential Monte Carlo Methods in Practice*, pp. 3–14, 2001.

[17] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[18] H. Du, T. Oggier, F. Lustenberger, and E. Charbon, "A Virtual Keyboard Based on True-3D Optical Ranging," *Proceedings of the British Machine Vision Conference, Oxford, UK*, vol. 1, pp. 220–229, 2005.

[19] R. Duda, P. Hart, and D. Stork, "Pattern Classification. 2001," *NY John Wiley*, 2001.

[20] T. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-D objects using surface descriptions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 11, pp. 1140–1157, 1989.

[21] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer graphics: principles and practice.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990.

[22] K. Fujimura, Y. Zhu, and V. Ng-Thow-Hing, "Estimating pose from depth image streams," *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pp. 154–160, 5, 2005.

[23] S. Gokturk and C. Tomasi, "3d head tracking based on recognition and interpolation using a time-of-flight depth sensor," *Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington DC, USA*, vol. 02, pp. 211–217, 2004.

[24] R. Gonzalez and R. Woods, *Digital Image Processing.* Prentice Hall, 2002.

[25] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[26] D. Hansen, R. Larsen, and F. Lauze, "Improving Face Detection with TOF Cameras," *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*, vol. 1, 2007.

[27] M. Isard and A. Blake, "CONDENSATION — Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[28] ——, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.

[29] Y. Jin and F. Mokhtarian, "Towards Robust Head Tracking By Particles," *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3, pp. 864–867, 2005.

[30] S. Julier, "Skewed approach to filtering," *Proceedings of SPIE*, vol. 3373, p. 271, 2003.

[31] S. Krotosky, S. Cheng, and M. Trivedi, "Real-time stereo-based head detection using size, shape and disparity constraints," *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pp. 550–556, 2005.

[32] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *Quantum Electronics, IEEE Journal of*, vol. 37, no. 3, pp. 390–397, 2001.

[33] B. Liu, O. Jesorsky, and R. Kompe, "Robust Real-Time Multiple Object Tracking in Traffic Scenes Using an Optical Matrix Range Sensor," *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 742–747, 2007.

[34] S. Malassiotis and M. Strintzis, "Real-time head tracking and 3D pose estimation from range data," *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, 2003.

[35] G. Medioni and S. Kang, *Emerging Topics in Computer Vision.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.

[36] E. Meier and F. Ade, "Object detection and tracking in range image sequences by separation of image features," *IEEE International Conference on Intelligent Vehicles*, pp. 176–181, 1998.

[37] H. Nanda and K. Fujimura, "A robust elliptical head tracker," *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 469–474, 2004.

[38] ——, "Visual Tracking Using Depth Data," *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, pp. 37–37, 2004.

[39] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.

[40] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc, "An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)," *Proceedings of SPIE Optical Design and Engineering*, vol. 5249, pp. 534–545, 2004.

[41] T. Oggier, F. Lustenberger, and N. Blanc, "Miniature 3D TOF Camera for Real-Time Imaging," *Springer-Verlag, Lecture Notes in Computer Science*, vol. 4021, pp. 212–216, 2006.

[42] E. Osuna, R. Freund, F. Girosi *et al.*, "Training support vector machines: an application to face detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 24, 1997.

[43] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *European Conference on Computer Vision*, vol. 1, pp. 661–675, 2002.

[44] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 1, pp. 23–38, 1998.

[45] D. Serby, E. Meier, and L. Van Gool, "Probabilistic object tracking using multiple features," *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, 2004.

[46] C. Shan, Y. Wei, T. Tan, and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 669–674, May 2004.

[47] B. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, 1986.

[48] C. Veenman, M. Reinders, and E. Backer, "Resolving Motion Correspondence for Densely Moving Points," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 1, pp. 54–72, 2001.

[49] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pp. 153–158, 2000.

[50] E. Wan and R. van der Merwe, "The Unscented Kalman Filter," *Kalman Filtering and Neural Networks*, pp. 221–280, 2001.

[51] J. Weingarten, G. Gruener, and R. Siegwart, "A state-of-the-art 3D sensor for robot navigation," *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2155–2160, 2004.

[52] F. Xu and K. Fujimura, "Human detection using depth and gray images," *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, pp. 115–121, 2003.

[53] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, 2006.

[54] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1531–1536, 2004.

[55] Y. Zhu and K. Fujimura, "3D head pose estimation with optical flow and depth constraints," *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pp. 211–216, 2003.

VITA AUCTORIS

Ehsan Parvizi received his Bachelor of Applied Science degree in electrical engineering from the University of Tehran, Tehran, Iran in 2003. He is currently a candidate for the Master of Applied Science degree in electrical engineering at the University of Windsor, Windsor, Canada with specialization in Computer Vision. During his master's work, he was involved with the design and implementation of 3D object tracking systems. His recent publications include "Real-time Approach for Adaptive Object Segmentation in Time-of-Flight Sensors", in *Tools with Artificial Intelligence, 2008, ICTAI 2008*, "Visual Tracking Using Color Cameras and Time-of-Flight Range Imaging Sensors", in *Journal of Multimedia, Vol. 3, No. 2, 2008*, "Multiple Object Tracking Based on Adaptive Depth Segmentation", in *Computer and Robot Vision, 2008, CRV 2008*, and "Real-Time 3D Head Tracking Based on Time-of-Flight Depth Sensor", in *Tools with Artificial Intelligence, 2007, ICTAI 2007*. His research interests include real-time object detection and tracking, 3D image analysis, machine learning, pattern recognition, image processing, multimedia, and biometrics.