Fuzzy Logic Based Approach for Object Feature Tracking

NUNO MANUEL LUCAS VIEIRA LOPES

A Thesis submitted to the University of Trás-os-Montes e Alto Douro in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering

SCIENTIFIC SUPERVISORS:
Professor Pedro Melo-Pinto, Ph.D.
Professor Pedro M. Couto, Ph.D.

University of Trás-os-Montes e Alto Douro
Department of Engineering
Vila Real
November 2011

N. V. Lopes: *Fuzzy Logic Based Approach for Object Feature Tracking.* A Thesis submitted to the University of Trás-os-Montes e Alto Douro in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.

SCIENTIFIC SUPERVISORS:
Professor Pedro Melo-Pinto, Ph.D.
Full Professor in the Engineering Department at University of Trás-os-Montes e Alto Douro
Professor Pedro M. Couto, Ph.D.
Assistant Professor in the Engineering Department at University of Trás-os-Montes e Alto Douro

INSTITUTION:
University of Trás-os-Montes e Alto Douro
Department of Engineering

LOCATION:
Vila Real

DATE:
November 2011

To my parents, ***Gilberto*** *and* ***Maria****.*

RESUMO

Nesta tese é introduzida uma nova técnica de seguimento de pontos característicos de objectos em sequências de imagens em escala de cinzentos baseada em lógica difusa. É apresentada uma metodologia versátil e modular para o seguimento de objectos utilizando conjuntos difusos e motores de inferência. É também apresentada uma extensão desta metodologia para o correcto seguimento de múltiplos pontos característicos.

Para se realizar o seguimento são definidas inicialmente três funções de pertença. Uma função de pertença está relacionada com a propriedade distintiva do objecto que desejamos seguir, outra está relacionada com o facto de se considerar que o objecto tem uma movimentação suave entre cada imagem da sequência e outra função de pertença referente à sua previsível localização futura. Aplicando estas funções de pertença aos píxeis da imagem, obtêm-se os correspondentes conjuntos difusos, que serão manipulados matematicamente e servirão como entrada num motor de inferência. Situações como a oclusão ou falha na detecção dos pontos característicos são ultrapassadas utilizando posições estimadas calculadas a partir do modelo de movimento e a um vector de estados do objecto.

Esta metodologia foi inicialmente aplicada no seguimento de um objecto assinalado pelo utilizador. Foram realizados vários testes de desempenho em sequências de imagens sintéticas e também reais. Os resultados experimentais obtidos são apresentados, analisados e discutidos. Embora esta metodologia pudesse ser aplicada directamente ao seguimento de múltiplos pontos característicos, foi desenvolvida uma extensão desta metodologia para esse fim. Nesta nova metodologia a sequência de processamento de cada ponto característico é dinâmica e hierárquica. Dinâmica por ser variável ao longo do tempo e hierárquica por existir uma hierarquia de prioridades relativamente aos pontos característicos a serem seguidos e que determina a ordem pela qual esses pontos são processados. Desta forma, o processo dá preferência a pontos característicos cuja localização é mais fácil de prever comparativamente a pontos característicos cujo conhecimento do seu comportamento seja menos previsível. Quando esse valor de prioridade se torna demasiado baixo, esse ponto característico deixa de ser seguido pelo algoritmo. Para se observar o desempenho desta nova abordagem foram utilizadas sequências de imagens onde várias características indicadas pelo utilizador são seguidas.

Na parte final deste trabalho são apresentadas as conclusões resultantes a partir do desenvolvimento deste trabalho, bem como a definição de algumas linhas de investigação futura.

ABSTRACT

This thesis introduces a novel technique for feature tracking in sequences of greyscale images based on fuzzy logic. A versatile and modular methodology for feature tracking using fuzzy sets and inference engines is presented. Moreover, an extension of this methodology to perform the correct tracking of multiple features is also presented.

To perform feature tracking three membership functions are initially defined. A membership function related to the distinctive property of the feature to be tracked. A membership function is related to the fact of considering that the feature has smooth movement between each image sequence and a membership function concerns its expected future location. Applying these functions to the image pixels, the corresponding fuzzy sets are obtained and then mathematically manipulated to serve as input to an inference engine. Situations such as occlusion or detection failure of features are overcome using estimated positions calculated using a motion model and a state vector of the feature.

This methodology was previously applied to track a single feature identified by the user. Several performance tests were conducted on sequences of both synthetic and real images. Experimental results are presented, analysed and discussed. Although this methodology could be applied directly to multiple feature tracking, an extension of this methodology has been developed within that purpose. In this new method, the processing sequence of each feature is dynamic and hierarchical. Dynamic because this sequence can change over time and hierarchical because features with higher priority will be processed first. Thus, the process gives preference to features whose location are easier to predict compared with features whose knowledge of their behavior is less predictable. When this priority value becomes too low, the feature will no longer tracked by the algorithm. To access the performance of this new approach, sequences of images where several features specified by the user are to be tracked were used.

In the final part of this work, conclusions drawn from this work as well as the definition of some guidelines for future research are presented.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

N. V. Lopes, H. Bustince, V. Filipe, and P. Melo-Pinto. *Fuzziness Measure Approach to Automatic Histogram Threshold*. In João Tavares and Natal Jorge, editors, Computational Vision and Medical Image Processing: VipIMAGE 2007, pages 295-299. Taylor and Francis Group, 2007. ISBN: 978-0415457774.

N. V. Lopes, P. Couto, H. Bustince, and P. Melo-Pinto. *Fuzzy Dynamic Matching Approach for Multi-Feature Tracking*. In EUROFUSE 2009: Eurofuse Workshop on Preference Modelling and Decision Analysis, pages 245-250, 2009. ISBN: 9788497692427.

N. V. Lopes, P. A. Mogadouro do Couto, H. Bustince, and P. Melo-Pinto. *Automatic Histogram Threshold Using Fuzzy Measures*. IEEE Transactions on Image Processing, volume 19, number 1, pages 199-204, 2010. ISSN: 1057-7149. DOI: 10.1109/TIP.2009.2032349.

P. Couto, N. V. Lopes, H. Bustince, and P. Melo-Pinto. *Fuzzy dynamic model for feature tracking*. In Fuzzy Systems (FUZZ), IEEE International Conference on, pages 1-8, 2010. DOI: 10.1109/FUZZY.2010.5583979.

N. V. Lopes, P. Couto, and P. Melo-Pinto. *Multi-feature Tracking Approach Using Dynamic Fuzzy Sets*. In P. Melo-Pinto, P. Couto, C. Serôdio, J. Fodor, and B. De Baets, editors, Eurofuse 2011: Workshop on Fuzzy Methods for Knowledge-Based Systems - Advances in Intelligent and Soft Computing, volume 107, pages 389-400. Springer-Verlag Berlin and Heidelberg GmbH, 2011. ISBN: 9783642240003.

ACKNOWLEDGMENTS

Throughout this difficult but exciting period of my life I had the support, directly or indirectly, of several persons and entities to whom I wish to express my sincere thanks.

Foremost, I would like to express my deep and sincere gratitude to my supervisor, Professor Pedro Melo-Pinto, Ph.D., who introduced me to the field of fuzzy logic in image processing. His wide knowledge, understanding, encouraging and personal guidance have provided a good preparation for the present thesis.

I would also like to express my warm and sincere thanks to my second supervisor, Professor Pedro Couto, Ph.D., for his constructive comments, his logical way of thinking, and for his important support and guidance throughout this work.

I wish to thank the School of Technology and Management, Polytechnic Institute of Leiria and Centre for the Research and Technology of Agro-Environmental and Biological Sciences at the University of Trás-os-Montes e Alto Douro, for providing me the conditions to carry out this research work.

I want to give my heartfelt gratitude to all my family, particulary to my parents Gilberto and Maria, and to my brothers Marina, João and André, for the support they provided me through my entire life.

I owe my loving thanks to Susana. Her support, encouragement and understanding have been very important for me to finish this work.

A special acknowledgment to all my Department colleagues and my friends for their friendship and encouragement during the difficult times.

My deepest gratitude to them all!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AFCM	Automated Fuzzy C-means
A-IFSs	Atanassov's Intuitionistic Fuzzy Sets
BFO	Bacterial Foraging Optimization
CAVIAR	Context Aware Vision using Image-based Active Recognition
CIPR	Center for Image Processing Research
CITAB	Centro de Investigação e de Tecnologias Agro-Ambientais e Biológicas
FCM	Fuzzy C-means
FKF	Fuzzy Kalman Filtering
FoV	Field of View
fps	frames per second
GLCM	Gray Level Co-occurrence Matrices
GOA	Greedy Optimal Assignment
GPS	Global Positioning System
HCI	Human Computer Interaction
HMI	Human Machine Interaction
HT	Hough Transform
IF	Index of Fuzziness
IKF	Interval Kalman Filtering
IVFS	Interval Valued Fuzzy Set
IVFSs	Interval Valued Fuzzy Sets
JPDAF	Joint Probability Data Association Filtering
LED's	Light Emitting Diodes
LoG	Laplacian of Gaussian
MHT	Multiple Hypothesis Tracking
MMS	Multimedia Messaging Service
NCC	Normalized Cross-Correlation
OTCBVS	Object Tracking and Classification in and Beyond the Visible Spectrum

PETS	Performance Evaluation of Tracking and Surveillance
RGB	Red, Green and Blue
SMS	Short Message Service
SNR	Signal-to-Noise Ratio
SPSS	Specific Persons Surveillance using Satellite
SRG	Seeded Region Growing
SVM	Support Vector Machine

Part I

MOTIVATION AND BACKGROUND

Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less.

— Marie Curie

1 INTRODUCTION

1.1 MOTIVATION

The understanding of the dynamics in the surrounding environment is a basic necessity for all animals. Environment changes are manifold and detected through physiological systems placed in the body. The conventional five sensitive systems are sight, hearing, touch, smell and taste. With these sensorial stimulus animals can interpret the scene and take the right actions to reach a goal. In non rational animals the goals are quite primitive and simple, including hunting, defence and safety. Moreover, with the cognitive property, humans can achieve higher level of understanding. Humans are capable to interpret the changes in the environment and, adding knowledge, think forward to anticipate future behaviours.

Computer vision systems are destitute of all sensitive senses rather than sight. Equipped only with "eyes" such systems attempt to understand the surrounding environment looking for scene modifications. In general, illumination changes, noise in the acquisition process and objects movement are the major causes for scene changes. Objects movement and behaviour are the most important features to be traced, since illumination changes is a global property and didn't give enough information about the dynamics of the scene, and noise is an extra and undesirable signal added, by the acquisition device, over the real representation of the scene. Moreover, illumination changes and noise are two causes, among others, that difficult the task of extraction of objects motion in an image. Other causes that difficult the extraction and analysis of objects motion are:

- information losses due the representation of a 3D world in a 2D plane;
- complex object motion;
- nonrigid or articulated nature of objects;
- complex object shapes;
- partial and full object occlusions.

In this work, the tracking concept refers to the process of locating a moving object (or several ones) and estimating its trajectory along time in a sequence of images. In its simplest approach, tracking involves two steps: detection of interesting objects and trace their trajectories. Furthermore, it could also involve behaviour analysis, object and activity classification, specific person identification, counting, flux statistics and alarming.

Normally, tracking is seen as a main task involving several subtasks such as image segmentation for object detection, object matching and position estimation. A myriad of algorithms has been developed to implement these subtasks but each one have their strengths and weaknesses and, over the last years, extensive research has been conduct in this field to find an optimal tracking system for a certain kind of applications. Consequently many approaches of tracking techniques have been proposed in the literature, however, they are not completely accurate for all kind of scenarios and just provide good results when a certain number of assumptions are verified. This reason is the motivation to study and implement new tracking approaches introducing new concepts, such as fuzzy logic, to improve the tracking process.

1.2 PROBLEM STATEMENT

As stated earlier, the process of object tracking is still a difficult task and remains a challenge.

The presented work aims to study the problem of object tracking in a sequence of images using fuzzy logic concepts. The object must denote a distinctive property that can conveniently characterise such object. In this work, only the value of grey level is used to distinguish the object. With the introduction of fuzzy logic concepts, robustness against data and accurate processing of noise and uncertainty are expected.

1.3 MAJOR CONTRIBUTIONS

In this work a novel object tracking approach is presented.

A singular methodology using fuzzy sets is presented here. Fuzzy sets are used, commonly, for control and, only recently, in image processing. In this thesis a distinct way of using fuzzy sets is introduced for object tracking.

Object detection and correspondence tasks are not made independently but they are incorporated in the same framework using fuzzy logic concepts such as mathematical relationships between fuzzy sets and rule based inference.

An improvement of an existing method for image threshold is also presented. A new automatic scheme with no requirements on image information is developed to overcome the initialisation of the original method. This segmentation algorithm was firstly developed as a feature detector in the object tracking process. The tracking method presented in this work incorporates a detection scheme that overcame the use of an external feature detection algorithm. However, it was an initial and innovative work that deserves to be mentioned.

1.4 THESIS OUTLINE

During the last years, extensive research has been conducted in this field and many types and applications of object tracking systems have been proposed in the literature. As a consequence, in Chapter 2, a review of object tracking is presented. After a brief introduction, a general definition of tracking is presented in Section 2.2 followed by the wide range of applications found in the literature. Methods and algorithms for object detection and matching are also presented in this chapter. At the end, a discussion about these topics is carried out.

In Chapter 3, the fuzzy logic concepts are introduced, as well the fuzzy logic concepts as a constituting part of the tracking algorithms. After a short introduction and a brief explanation of the historical genesis of fuzzy logic, the essential concepts in fuzzy logic theory are introduced in Section 3.3. These concepts are related with the definition, properties, operations and fuzziness measures in fuzzy sets and also with the most commonly used membership functions to construct such fuzzy sets. In Section 3.4 the introduction of fuzzy logic theory in object tracking is described. Fuzzy logic theory can appear in all steps of a tracking process such as object detection, position estimation or in the correspondence task.

An automatic histogram thresholding method is presented in Chapter 4. This automatic approach is an improvement of an existing method using fuzzy measures to classify each grey level as object or background. The introductory issues, theory of support and the explanation of the previous method are introduced, respectively, in Sections 4.2, 4.3 and 4.4. The new image segmentation approach is presented in Section 4.5. Experimental results and related discussion appear at the end of this chapter.

In Chapter 5, an object feature tracking approach using fuzzy concepts is introduced. The aim of this methodology is to solve the problem of feature tracking using fuzzy sets theory. Several fuzzy sets are defined using both cinematic and non cinematic properties. Properties such as grey level intensity, smooth motion and motion model of the feature are used to construct the fuzzy sets. Fuzzy operations are performed on these fuzzy sets and the resulting fuzzy sets are used as inputs in an inference engine. This way, problems of object detection and matching are performed exclusively using inference rules on fuzzy sets. If a feature is occluded during a period of time and no valid correspondence is performed, an estimated position of this feature is used. This estimated position is calculated based on the movement model and the previous movement behaviour of the feature. Experimental results using synthetic and non synthetic sequences are presented and discussed, respectively, in Sections 5.6 and 5.7.

A hierarchical matching approach for multiple feature tracking is presented in Chapter 6. In this chapter, the approach developed previously is extended to deal with multiple features. When multiple moving features appear in the scene, a matching scheme not using some kind of priorities could result in erroneous correspondence. Also, the viewing angle of the camera and the trajectory of the features in the real world can cause features to merge, divide or disappear in the image plane and these issues were taken into account. This hierarchical tracking approach is applied to several sequences to measure its performance. Synthetic and real sequences were used and experimental results are presented and discussed, respectively in Sections 6.5 and 6.6. In Section 6.7, a brief discussion concerning this hierarchical matching approach for multiple feature tracking is carried out.

Conclusion and future work are discussed in Chapter 7.

Do not talk a little on many subjects,
but much on a few.

— Pythagoras

2 OBJECT TRACKING - STATE OF THE ART

2.1 INTRODUCTION

Object tracking plays an important role within the field of computer vision. Tracking is essential to many applications, however, robust tracking algorithms are still a challenge. Difficulties can arise due to noise presence in images, quick changes in lighting conditions, abrupt or complex object motion, changing appearance patterns of the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, camera motion and real time processing requirements. Typically, assumptions are made to constrain the tracking problem in the context of a particular application. For example, almost all tracking algorithms assume that object motion is smooth or impose constrains on the object motion to be constant in velocity or acceleration. Multiple view image tracking or prior knowledge about objects, such as size, number or shape, can also be used to simplify this process.

In this chapter, the state of the art concerning object tracking is introduced. It begins with the definition of tracking and, in Section 2.3, the wide range of tracking applications is presented. Issues related with object detection and matching are discussed, respectively, in Sections 2.4 and 2.5.

2.2 DEFINITION

Object tracking can be described as the problem of estimating the trajectory of an object as it moves around a scene. Although this general concept is almost consensual, the specific definition of tracking could change in the literature. In Moeslund et al. [83] the tracking process is defined as consisting of two tasks: *figure-ground segmentation* and *temporal correspondences*. Figure-ground segmentation is the process of extracting the objects or features of interest from the video sequence. Segmentation methods are applied as the first step in many tracking systems and therefore is a crucial task. Object segmentation could be based on motion, appearance, shape, depth, etc. Temporal correspondence is concerning with the association of the detected objects in the current frame with those in the previous frames defining temporal trajectories. In Trucco and Plakas [125], tracking is described as the *motion problem* and the *matching problem*. In this work, the motion problem is related with the prediction of the object location in the next frame. The second step is similar to the explained above. However, Hu et al., Yilmaz et al. [47, 140] present a wider description of tracking with three steps: *detection of interesting objects*, *tracking such objects* and *analysis of object tracks to recognise their behaviour*. In Wang and Singh [133] this behaviour analysis is seen as a further interpretation of tracking results.

As we can see, tracking process can include two or more subtasks depending on the complexity of the final application. In this work, behavioural analysis is not discussed and it will be considered as future work.

2.3 APPLICATIONS

Most initial tracking algorithms were used in surveillance tasks but nowadays tracking algorithms are applied in a wide range of applications such as:

- Human Machine Interaction (HMI);
- traffic monitoring;
- medical applications;
- sports video analysis.

In new **automated surveillance systems**, the movement and location of people are monitored to uncover suspicious behaviours of people in a place such as airports or subways. People counting, crow flux and congestion analysis can also be performed.

A system that can provide extremely accurate location information for some specific person is proposed in Mazinan [80]. The proposed approach called Specific Persons Surveillance using Satellite (SPSS) is a person finder based on positioning information from Global Positioning System (GPS) Satellite. A system to detect and track multiple people using a stereo camera combining depth information, colour and position prediction is proposed in Muñoz-Salinas et al. [85]. A person detection and counting system in real world scenarios is introduced in Liu et al. [69]. The focus of this work is the segmentation of groups of people into individuals to achieve a correct counting of people. An outdoor surveillance video system is presented in Black et al. [15]. Using a set of calibrated cameras, a multi view image tracking system is implemented to track humans and vehicles. This approach tends to solve occlusions and can track objects between overlapping and non-overlapping camera views. An automated video surveillance and alarm system that can send alerts via email, SMS or MMS, with no human interaction, is found in Li et al. [65]. Smart surveillance systems (applications, architectures, technologies, challenges and implications), the state-of-the-art in the development of automated visual surveillance systems and a complete survey on applications involving the surveillance of people or vehicles are presented in Hampapur et al. [38], Valera and Velastin [128] and Hu et al. [47], respectively.

HMI or Human Computer Interaction (HCI) is concerned with gesture recognition or eye gaze tracking for data input in a computer or control command. A control system for an exoskeleton orthosis used by people suffering from tetraplegia is proposed in Baklouti et al. [7]. The aim of this work is to identify head motion and mouth expressions using a single camera and translate that information into control commands. In Hiley et al. [45], a low cost HCI based on eye tracking is presented. Using an infrared light source provided by Light Emitting Diodes (LED's) and a modified web cam to be sensitive to infrared radiation, pupil detection is achieved and the gaze point is then calculated. A more complex technique that does not rely on the use of infrared devices for eye tracking is presented in Amarnag et al. [3]. In Shi et al. [115, 116], is presented the design and implementation of a smart camera, called GestureCam, which can recognise simple hand and head gestures. The capture and recognition of user's head and hand gesture commands are used to control an internet browser. For further information, a extended survey on visual interpretation of hand gestures in the context of its role in HCI is discussed in Pavlovic et al. [93]. Also, Jaimes and Sebe [51] present a review of the major approaches in HCI, focusing body, gesture, gaze and facial expression recognition.

Traffic monitoring is related with real time traffic statistics, traffic flow analysis, velocity control, driver's violations or collision detections. In Atev et al. [4], a system capable to monitoring traffic intersections in real time and predicting possible collisions is proposed. In Kamijo et al. [55], the major goal is to track individual vehicles robustly against the occlusion and clutter effects which usually occur at intersections. In Zhang et al. [143], is obtained information of the traffic flow at intersections to be used in intelligent traffic management. Stopped vehicles in a highway is a possible sign of danger and the detection of stopped vehicles and stopped delay can be useful even to assess the level of service at urban intersections. Another approach to measure the maximum stopped delay is presented in Bevilacqua and Vaccari [12]. A vehicle tracking system which can detect and monitor vehicles as they break traffic lane rules is suggested in Choi et al. [22]. Lim et al. [66] introduces a system that detects all kinds of violations at a street intersection such as red light running, speed violation, stop line violation and lane violation by tracking individual vehicles. A method for accurately counting the number of vehicles that are involved in multiple-vehicle occlusions is proposed in Pang et al. [91]. In Tseng and Song [126], traffic parameters such as vehicle speed and number of vehicles can be calculated with satisfactory accuracy. A review on traffic visual tracking technology is presented in Liu et al. [68].

In **medical applications**, tracking is used to identify human structures behaviour, rehabilitation diagnosis and optimisation of athlete's performances. A technique to extract and track the tongue surface movements from ultrasound image sequences is presented in Akgul et al. [2]. These information is useful for a number of research areas, including disordered speech, linguistics, speech processing and swallowing. In Mikic et al. and Malassiotis and Strintzis [82, 76], an automatic identification and tracking of the boundary of heart structures from echocardiographic images is proposed. A method to gait analysis to use as a means to deduce the physical conditions of people is referred in Wang [134]. A review on recent progress in human movement detection and tracking systems and existing or potential application for stroke rehabilitation, can be found in Zhou and Hu [144].

Sports video analysis is related with training and gaming assistance and game statistics. It could include tactics analysis, balls or players tracking and computer-assisted referee. A real time tracking system applied in tennis matches is described in Pingali et al. [95]. This tennis tracking system provides spatio-temporal trajectories of the players and the ball in real time. This information is the basis for obtaining statistic data such as the distance travelled by the player, the instantaneous, average and peak speed and acceleration of the player and the covered areas of the court. The Hawk-Eye Tennis System is presented in Owens et al. [89]. This system provides a low-cost solution for tracking the tennis ball during a match. Real time requirements are not the crucial element but it provides the 3D reconstruction of the complete track of the ball no more than 5 seconds after the end of a point. In Xiong et al. [138] a system to track body contours of the athletes in diving videos is presented. This system can provide important information for coaches, such as the altitude of a jump, the vertical angle of the body and even the whole trajectory of the body. A soccer ball, players and referee detection, classification and tracking, team identification, and a field extraction approach is proposed in Naidoo and Tapamo [86].

Wang and Parameswaran [132] review current research in sports video analysis and discuss the research issues within this field and potential applications.

Numerous approaches for object tracking have been proposed in the literature. The classification of tracking algorithms is not consensual. This different types of classification could be explained by the fact there exists a myriad of tracking methods and authors could group them with respect to objects representation, relevant descriptive features, object motion or shape model and initial constrains. In Hu et al. [47], tracking methods are divided into four categories: region-based tracking, active-contour-based tracking, feature-based tracking and model-based tracking. In Trucco and Plakas [125], a concise survey towards to subsea video tracking is presented. In this work the review is divided into six categories increasing in its complexity: window tracking, feature tracking, planar rigid shapes, solid rigid objects, deformable contours and visual learning. More recently, in the survey presented in Yilmaz et al. [140], tracking methods are distributed into three main categories: point tracking, kernel tracking and silhouette tracking. Point tracking is subdivided in deterministic and statistical methods, kernel tracking could be classified in template and density based appearance models or multi-view appearance models and silhouette tracking could be seen as contour evolution or matching shapes.

Despite some different types of classification, all tracking systems are based in two main steps: feature detection and matching.

2.4 OBJECT DETECTION

Object detection is the first task in a tracking system and represents an important role in computer vision and image processing applications. Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the eventual success or failure of computerised analysis procedures. Segmentation of an image entails the division or separation of the image into regions of similar attribute. This situation can be mathematically described as follows:

Definition If I is the set of all pixels and $P()$ is a homogeneity predicate defined on groups of connected pixels, then segmentation is a partition of the set I into connected subsets or image regions $(R_1, R_2, ..., R_n)$, such that

$$\bigcup_{i=1}^{n} R_i = I, \text{ where } R_i \cap R_j = \emptyset, \forall i \neq j. \tag{2.1}$$

The uniformity predicate $P(R_i)$ is true for all regions, and $P(R_i \cup R_j)$ is false, when $i \neq j$ and R_i, R_j are neighbours.

It's easy to understand that the complexity of such process depends on the amount of information present in the input images. Segmenting objects with high contrast against background (characters detection or assembly operations) could be made with low processing level. However, in complex images such as aerial or urban scenes, segmentation task is a more complicated process.

No single standard method of image segmentation has emerged. Rather, there are a collection of heuristic methods that have achieved some degree of popularity. Because the methods are heuristic, it would be useful to have some means to assess their performance.

Haralick and Shapiro [41] have established the following qualitative guideline for a good image segmentation:

"Regions of an image segmentation should be uniform and homogeneous with respect to some characteristic such as grey tone or texture. Region interiors should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged, and must be spatially accurate."

Unfortunately, no standard quantitative image segmentation performance metric has been developed.

Several generic methods of image segmentation will be detailed described in the following sections.

2.4.1 *Motion based*

In applications such as moving object detection, surveillance or traffic monitoring, motion is a powerful property of that image sequence, revealing the dynamics of scene. The task of motion based segmentation remains a challenging and is a fundamental issue in computer vision systems. The difficulties in motion estimation are manifold and are concerned with the differences between the motion observed in the image plane and the real motion presented in the scene. As only the apparent motion in the sequence can be estimated, further assumptions on brightness changes, object properties, and the relation between relative 3D scene motion and the projection onto the 2D image plane are necessary for quantitative scene analysis.

Some issues frequently encountered in real-world sequences that difficult the motion based segmentation include overlay of multiple motions, occlusions, illumination changes, nonrigid motion, low Signal-to-Noise Ratio (SNR) levels, aperture and correspondence problems. These situations, however, are not always present and are usually not spread over the entire image area. Thus, there exist many applications where image segmentation by motion estimation could be implemented.

Several algorithms to estimate motion in the image plane are found in the literature. For further information, the interested reader is referred to reviews of motion understanding approaches (Shah [114]), hand gestures recognition (Pavlovic et al. [93]) and video analysis of human motion (Gavrila [35]) and dynamics (Wang and Singh [133]).

Frame differencing

One of the simplest approaches to detect changes and, consequently the movement, between two consecutive intensity image frames $I(x, y, t)$ and $I(x, y, t-1)$ taken at times t and $t-1$, respectively, is to make a difference between the two images pixel by pixel. A difference image between two images taken at instants t and $t-1$ may be defined as

$$I_{diff}(x, y, t) = \begin{cases} 1 & \text{if } |I(x, y, t) - I(x, y, t-1)| > T \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

where T is a specified threshold. The resulting image $I_{diff}(x, y, t)$ denotes a value 1 in spatial coordinates (x, y) if the difference in the grey level between the two images is reasonable different at those coordinates, as determined by the threshold value T, Radke et al. [100]. Frame differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, e. g., there may be holes left inside slowly moving objects. These small gaps in the extracted motion regions can be filled by performing a morphological closing operation, Pingali et al. [95]. This morphological closing operation can be mathematically represented as

$$B_{diff}(x, y, t) = (I_{diff}(x, y, t) \oplus g) \ominus g, \tag{2.3}$$

where $B_{diff}(x, y, t)$ is the binary result image at instant t, $I_{diff}(x, y, t)$ is the binary difference image at time t, g is a structuring element (e.g. a small circular element), and $\oplus$, $\ominus$ indicate morphological dilatation and erosion operations, respectively. In Lipton et al. [67], the detection of moving targets in real video streams using frame differencing is performed. If the illumination is not relatively constant, background pixels might be mistaken as the object. Suitable threshold calculation is critical in this method. Rosin [107] describes four different methods for selecting a correct threshold.

Optical flow

This group of methods performs segmentation based on displacement or optical flow of image pixels. The displacement or optical flow of a pixel is a motion vector represented by the translation between a pixel in one frame and its corresponding pixel in the following frame.

Commonly, it is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames, Horn and Schunck [46]. This constraint can be formally described as

Definition Let $I(x, y, t)$ represents the image intensity then

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t), \tag{2.4}$$

where δx and δy are the displacement of the local region at (x, y, t) after time δt.

Assuming that image motion is modelled by a continuous variation of image intensity as a function of position and time, then, expanding the left-hand side of the Equation 2.4 in a Taylor series yields

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + O^2, \tag{2.5}$$

where $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the first-order partial derivatives of $I(x, y, t)$, and O^2, the second and higher order terms.

Subtracting $I(x, y, t)$ on both sides, ignoring O^2 and dividing by δt yields

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} = 0. \tag{2.6}$$

In the limit as $\delta t \to 0$, Equation 2.6 becomes

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0. \tag{2.7}$$

Now $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are all measurable quantities and $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are estimates of what we are looking for: the velocity in x and y directions. Writing $\frac{dx}{dt} = u$, $\frac{dy}{dt} = v$, $\frac{\partial I}{\partial x} = I_x$, $\frac{\partial I}{\partial y} = I_y$ and $\frac{\partial I}{\partial t} = I_t$ gives

$$I_x u + I_y v + I_t = 0, \tag{2.8}$$

or equivalently,

$$\nabla I \cdot \mathbf{v} + I_t = 0, \tag{2.9}$$

where $\nabla I = (I_x, I_y)$ is the spatial gradient of the image, $\mathbf{v} = (u, v)$ is the velocity.

The Equation 2.9 is known as the *optical flow constraint equation*.

The constraint on the local flow velocity expressed by this equation is illustrated in Figure 1. The optical flow constraint equation defines a line in velocity space. This constraint is not sufficient to compute both components of $\mathbf{v}$ because the optical flow constraint equation is one linear equation with two unknowns $\mathbf{v} = (u, v)$. Without further assumptions only the component of the velocity in the direction of (I_x, I_y), represented as $\mathbf{v}_\perp$, perpendicular to the constraint line, can be estimated. This phenomenon is known as the *aperture problem* and only at image locations where there exists sufficient intensity structure can the motion be fully estimated with the use of the optical flow constraint equation (see Figure 2).

Through apertures (a) and (c), due to a lack of local structure, only normal motions of the edges forming the square can be estimated. Inside aperture (b), at the corner point, the motion can be fully measured as there is sufficient local structure, i. e., both normal motions are visible.

The normal velocity $\mathbf{v}_\perp$ and the normal direction n are given by

$$\begin{aligned} \mathbf{v}_\perp &= \frac{-I_t}{\|\nabla I\|}, \\ n &= \frac{\nabla I}{\|\nabla I\|}. \end{aligned} \tag{2.10}$$

This vector is referred to as *normal flow* as it points normally to lines of constant image brightness, parallel to the spatial gradient.

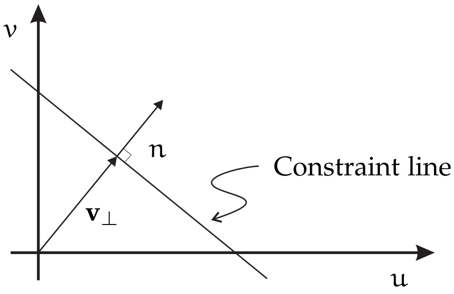


Figure 1: The optical flow constraint equation.

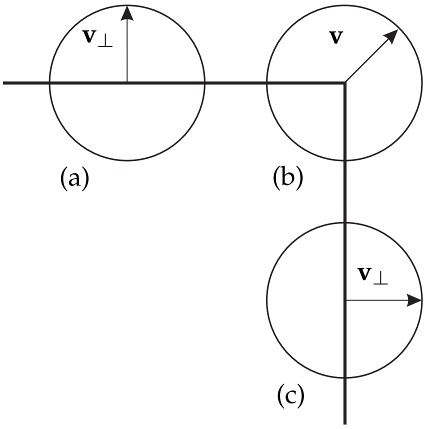


Figure 2: Illustration of the aperture problem.

To find the optical flow, another set of equations is needed, given by some additional constraint. All optical flow methods introduce additional conditions for estimating the actual flow. Popular techniques for computing optical flow include methods proposed by Horn and Schunck [46], Lucas and Kanade [75], Black and Anandan [16], and Szeliski and Coughlan [120].

Widely known methods for estimating optical flow are completely discussed in Beauchemin and Barron [9].

For the performance evaluation of the optical flow methods, the interested reader are referred to the surveys presented by Barron et al. [8] and Galvin et al. [33].

In Vega-Riveros and Jabbour [130], alternative methods to motion analysis rather than optical flow are also reviewed. These alternative methods are based in matching approaches such as cross-correlation based methods.

Background subtraction

Object detection can also be achieved by constructing a reference representation of the environment called background model and then finding deviations between this model and each incoming frame. A significant change between the background model and an image region denotes a moving object. This process is referred as background subtraction and represents a popular method for motion detection, especially under those situations with a relatively static background.

A simple way to represent the image background can be achieved by accumulating and averaging a sequence of frames for a certain interval of time, so called a temporal median filter [55, 66]. An alternative is to model the intensity of each pixel with a Kalman filter [103]. Using different Kalman filter gain, foreground pixels are adapted more slowly than the background pixels.

In Wren et al. [137], the background model initialisation is performed by acquiring a sequence of video frames without moving objects. The colour of each pixel is associated with a Gaussian distribution. Mean and variance are learnt from the colour observations in several consecutive frames. Once the background model is derived, for every pixel in the new frame, the likelihood of its colour is computed, and the pixels that deviate from the background model are labelled as foreground pixels. In each frame, background classified pixels have their statistics updated using an adaptive filter. This approach is also used in Wang [134]. A similar method is proposed by McKenna et al. [81]. Working in RGB colour space, a pixel in the background model is described by $[m_r, m_g, m_b, \sigma_r^2, \sigma_b^2, \sigma_b^2]$. Background model is adapted using a recursive estimation of mean and variance. Also it is assumed that the camera's R, G and B channels have Gaussian noise and therefore three variance parameters σ_{rcam}^2, σ_{gcam}^2, σ_{bcam}^2 are estimated for these channels. For a given pixel $\mathbf{x} = (r, g, b)$ in the current frame, if $|r - m_r| > 3\max(\sigma_r, \sigma_{rcam})$, or if similar test for components g or b is true, then the pixel is considered foreground.

In Haritaoglu et al. [43, 44], each background pixel is modelled by its minimum and maximum intensity values together with the maximum intensity difference between consecutive frames observed during the training period. A pixel in the new image is considered foreground if the absolute difference between the minimum or the maximum values of the background model is greater than the maximum intensity difference observed in the training period.

In Friedman and Russell [32] a mixture of three Gaussian distributions was used to model the pixel value for traffic surveillance applications. The pixel intensity was modelled as a weighted mixture of three Gaussian distributions: road, shadow and vehicle distribution. An incremental algorithm was used to learn and update the parameters of the model. A generalisation of the previous approach is presented in Stauffer and Grimson [117, 118]. The pixel intensity is modelled by a mixture of K Gaussian distributions (K is a small number from 3 to 5) to model variations in the background like tree branch motion and similar small motion in outdoor scenes. This mixture of Gaussian distribution is also used in Atev et al. [4] and Zhang et al. [143] to perform car detection.

In general, most tracking methods for fixed cameras use background subtraction methods to detect regions of movement. This fact occurs because recent subtraction methods have the capabilities of modelling illumination changes, noise, and the periodic motion of the background regions and, therefore, can accurately detect objects in a variety of circumstances. Moreover, these methods are computationally efficient.

In practice, background subtraction provides incomplete object regions in many instances, that is, the objects may be split into several regions, or there may be holes inside the object since there are no guarantees that the entire object will be different from the background model. However, the most important limitation of background subtraction is the requirement of stationary cameras. Camera motion usually distorts the background models.

2.4.2 *Feature based*

A feature is a distinctive characteristic or attribute of an object. The most desirable property of an object feature is its uniqueness so that the object can be easily distinguished in the image. Selecting the right feature plays a critical role in tracking. Features could be grey level, colour, edges or texture. In general, many tracking algorithms use a combination of these features.

Contours

It is possible to detect and represent an object by its boundaries. Object boundaries usually generate strong changes in image intensities. Edge detection is used to identify these changes. An important property of edges is that they are less sensitive to illumination changes compared to intensity or colour features. Usually, algorithms that track the boundary of objects, use edges as the representative feature.

First order derivatives of an image, ∇I, are a common approach to edge detection. They are computed from variations of the 2D image gradient defined as

$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}. \tag{2.11}$$

Roberts cross operator was one of the earliest edge detection operators [105]. It implements a basic first-order differentiation edge detection and uses two templates obtained by forming differences of diagonal pairs of pixels, Figures 3a, and 3b. The edge point is the maximum of the two values derived by convolving the two templates at an image point.

In order to reduce the effects of noise, local averaging is incorporated and the Prewitt edge detection operator arises [98]. This operator consists of two templates giving the rate of change of brightness along the two orthogonal axis, Figures 3c, 3d.

Sobel edge detection operator is obtained when the weight at the central pixels, for both Prewitt templates, is doubled, Figures 3e, 3f. It provides, in general, a better performance than Roberts and Prewitt operators.

However, the Canny edge detection operator is perhaps the most popular edge detection technique [18]. It was formulated with three main objectives:

1. Good detection - There should be a low probability of failing to mark real edge points, and low probability of falsely marking nonedge points.
2. Good localisation - The points marked as edge points by the operator should be as close as possible to the centre of the true edge.
3. Only one response to a single edge - This is implicitly captured in the first criterion since when there are two responses to the same edge, one of them must be considered false.

A common approximation of this filter can be described in four steps:

1. Apply a Gaussian smoothing.
2. Use the Sobel operator.
3. Use non-maximal suppression: any pixel that is not at the maximum value is suppressed from the image.
4. Threshold with hysteresis to connect edge points: The gradient array is now further reduced by hysteresis by means of two thresholds: t1 and t2 where t1 > t2. If the magnitude is below t2, the pixel is set to zero (considered a nonedge). If the magnitude is above t1, it is considered an edge. If the magnitude is between t1 and t2, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above t2.

An alternative to first-order differentiation methods explained above, is to apply second-order differentiation and then find zero-crossings in the second-order information. The 2D Laplacian of an image I is a second order derivative defined as follows:

$$\nabla^2 I = \frac{\partial^2 I}{\partial^2 x} + \frac{\partial^2 I}{\partial^2 y}. \tag{2.12}$$

The Laplacian operator is a template which implements second-order differentiation which can be approximated using the difference between two adjacent first-order differences. An example of a 3 × 3 Laplacian operator template is portrayed in Figure 4. The Laplacian operator used in its basic form is more sensitive to noise than a first-order operator since it is differentiation of a higher order. To reduce noise effects and to increase stability in the differentiation step, Torre and Poggio [124] suggested a regularisation of the image intensities by a filtering operation preceding differentiation.

Also, to prevent erroneous responses due to noise, Marr and Hildreth [77] have proposed the Laplacian of Gaussian (LoG) edge detection operator in

1	0
0	-1

(a)

0	1
-1	0

(b)

1	0	-1
1	0	-1
1	0	-1

(c)

1	1	1
0	0	0
-1	-1	-1

(d)

1	0	-1
2	0	-2
1	0	-1

(e)

1	2	1
0	0	0
-1	-2	-1

(f)

Figure 3: First order derivative template operators.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 4: 3×3 Laplacian operator.

which Gaussian smoothing is performed prior to application of the Laplacian operator.

Since these operators are applied locally, it is difficult to extract edges in a global concept. In addition, if an image is noisy or if its region attributes differ by only a small amount between regions, the detected boundary may often be broken. In such case, edge linking techniques can be employed to assemble the edge fragments present in a boundary.

The two principal properties used in local edge linking approaches are the strength of the response of the gradient operator used to produce the edge pixel and the direction of the gradient vector [36]. A pixel in the neighbourhood is linked to the edge pixel if both magnitude and gradient direction criteria are satisfied.

The Hough Transform (HT), presented in Duda and Hart [29], can also be used as a means of global edge linking. It has been used to extract lines, circles and ellipses.

The HT involves a mapping from the image points into an accumulator space (Hough space). The mapping is achieved in a computationally efficient manner, accordingly the function that describes the target geometric shape. Higher the number of geometric shape parameters, higher the complexity. It is required significant storage and high computational requirements. However, the most important advantages of this approach is that HT based methods are insensitive to noise and to partial defined boundaries.

Different versions of the original HT have been proposed by several authors such as Princen et al. [99] and Guil et al. [37].

Active contours or snakes, introduced by Kass et al. [57], is a completely different approach to contour extraction. Object segmentation is achieved by molding a closed contour to the boundary of an object in an image. The snake model is a controlled continuity closed contour that changes their shape under the influence of internal forces, image forces and external constraint forces. These forces are combined to form an energy function that governs the evolution of the contour. The object boundary is obtained when this energy function reaches a local minimum. The use of snakes turns boundary extraction into an energy minimisation problem.

Several tracking approaches using active contours have been proposed in [60, 59, 2, 82, 126, 76, 50].

Intensity or colour

Many images can be described as containing some object of interest denoting a reasonably uniform intensity or colour placed against a background with different intensity or colour. If an object of interest is white against a black background, or vice versa, it is a trivial task to set a midgrey threshold to segment the object from the background. Practical problems occur when the observed image is influenced by noise and when both the object and background are nonuniform and assume some range of grey scales or colours. However, image thresholding is a simple segmentation method and plays a central role among the existing image segmentation techniques and is still widely used in several applications. Many different approaches are used in image thresholding.

Rosenfeld's convex hull method is based on analysing the concavity structure of the histogram defined by its convex hull [106]. When the convex hull of the histogram is calculated, the deepest concavity points become candidates for the threshold value.

Ridler and Calvard algorithm [104] uses an iterative technique for choosing the threshold value. At iteration n, a new threshold T_n is established using the average of the foreground and background class means. The process is repeated until the changes in T_n become sufficiently small.

Otsu's technique [88] is based on discrimination analysis, in which the optimal threshold value calculation is based on the minimisation of the weighted sum of the object and background pixels within-class variances. Let 1 and 2 be considered as two distinct classes. The weighted within-class variance can be expressed as

$$\sigma_w^2(t) = P_1(t)\sigma_1^2(t) + P_2(t)\sigma_2^2(t). \tag{2.13}$$

Where the class probabilities are estimated as

$$P_1(t) = \sum_{i=0}^{t} p(i),$$
$$P_2(t) = \sum_{i=t+1}^{255} p(i). \tag{2.14}$$

The probability density function $p(i)$ gives the probability that the intensity value i occurs in the image. It is estimated as the quotient between $h(i)$ and the total number of pixels in the image

$$p(i) = \frac{h(i)}{N}, \tag{2.15}$$

where N is the total pixels of the image and the histogram $H(i)$ gives the number of pixels in the image having the intensity value i.

The class means are given by

$$m_1(t) = \sum_{i=0}^{t} \frac{ip(i)}{P_1(t)},$$
$$m_2(t) = \sum_{i=t+1}^{255} \frac{ip(i)}{P_2(t)}, \tag{2.16}$$

and the individual class variances are computed as

$$\sigma_1^2(t) = \sum_{i=0}^{t} (i - m_1(t))^2 \frac{p(i)}{P_1(t)},$$
$$\sigma_2^2(t) = \sum_{i=t+1}^{255} (i - m_2(t))^2 \frac{p(i)}{P_2(t)}. \tag{2.17}$$

Finally, all we need to do is just run through the full range of t values, [0,255], and pick the value t that minimises $\sigma_w^2(t)$.

In Kittler and Illingworth's minimum error thresholding method it is assumed that the image can be characterised by a mixture distribution of object and background pixels [61].

An exhaustive survey of image thresholding methods can be found in Sezgin and Sankur [112] and Weszka [135]. In Weszka and Rosenfeld [136], two techniques for image threshold evaluation are proposed. These techniques use a cost function that should reach its minimum at the correct threshold value.

Theoretically, Seeded Region Growing (SRG) is one of the simplest approaches to provide image segmentation. SRG consists in grouping together neighbour pixels of similar homogeneity criterion to achieve a segmented region. This method starts with the definition of one or more initial points, so-called seed points, and neighbour pixels are assigned into a region if they satisfy the homogeneity criterion of the respective region. An usual homogeneity criterion uses the average grey level of the growing region [1, 53].

However, in practice, the selection of the number of seeds, their initial location and the definition of the homogeneity criterion to achieve acceptable results can be reasonably complex. Brice and Fennema [17] have developed a region-growing method based on a set of simple growth rules.

In Tuduki et al. [127], an automatic SRG approach is achieved using the resulting points of a previous image threshold and thinning algorithm as the initial seeds. Also, Fan et al. [30], propose a hybrid image segmentation technique by integrating the results of colour-edge extraction and SRG, in which the obtained edge regions are used as the initial seeds for the SRG procedure.

Another technique considered similar to SRG is region splitting and merging [92]. The region split and merge technique is based on a quad tree data representation. A square image segment is divided (split) into four quadrants if the homogeneity criterion of the original image is not satisfied. If four neighbouring squares are found to be uniform, they are merged forming a single square composed by the four adjacent squares. As in SRG techniques, the selection of the homogeneity criteria plays a central role in the performance of region split and merge algorithms.

Representing the image intensity grey levels using topographic concepts has been proved useful in the development of region segmentation methods [11, 131, 119, 40]. This concept based on visualising an image in three dimensions (two spacial coordinates versus grey levels) is called watershed. This algorithm consists on morphological operators and integrate many concepts of the approaches discussed above (thresholding, edge detection, region processing). Due to these factors, watershed segmentation often displays more effective and stable segmentation results, including continuous boundaries. In this context, an image is considered to be an altitude surface in which high intensity pixels correspond to ridge points and low intensity pixels correspond to valley points.

Two watershed approaches have been discussed in the literature: flooding and rainfall [97]. The flooding approach is the most used and is based on an immersion process analogy. The water will enter through the valleys at a uniform rate and flood the surface. The accumulation of water in the neighbourhood of a local minima is called a catchment basin. To avoid the water of distinct catchment basins from merging, a conceptual dam is build. The dam is made up of pixels with a height slightly higher than the peaks. The flooding process stops when only the top of the dams are visible. At this stage, the dam walls are called the watershed lines which form a closed contour and are the desired segmentation result.

In the rainfall approach, each local minima is tagged uniquely. Adjacent local minima are combined with a unique tag. Next, a conceptual water drop is placed at each untagged pixel. The drop moves to its lower-amplitude neighbour until it reaches a tagged pixel, at which time it assumes the tag value. The method stops performing when all the pixels are tagged.

Watershed algorithms as described above are computationally expensive and can easily produce over-segmented images due to noise or local irregularities of the gradient [73].

Texture Many images do not denote sharp edges over large areas, however, the scene can often be described as exhibiting several consistent textures and, therefore, texture can be used as a discriminant feature to segment an image. There is no formally definition of texture and consequently, there is no unique mathematical model to represent texture. Intuitively, texture can be defined in terms of fineness, smoothness, coarseness and regularity, and, therefore, texture descriptors (texture measurements that characterise a texture) have been developed. Three principal approaches used in image processing to deriving the measurements which can be used to describe textures are statistical, structural and spectral [36].

Statistical approaches describe texture as smooth, coarse, grainy and so on, defining qualities of texture based on spatial distribution of grey values. Haralick et al. [42] suggested the use of Gray Level Co-occurrence Matrices (GLCM) which have become one of the most well-known and widely used texture features. Spatial grey level co-occurrence estimates image properties related to second-order statistics, revealing certain properties about the spatial distribution of the grey levels in the textured image.

The repetitive nature of the placement of texture elements in the image is an important property of many textures. Therefore, the autocorrelation function can be used to assess the amount of regularity as well the fineness or coarseness of the texture present in the image.

Structural techniques attempt to describe a texture by rules which govern the placement of primitive elements (texture elements) which make up the texture. That placement is seemed to be approximately regular and spatially repetitive in the image. Structural texture analysis consists of two major steps: extraction of the texture elements and inference of the placement rule. Commonly, texture elements consist of regions in the image with uniform grey levels. Such approaches are appropriate for ideal, regular, artificial, or periodic textures with low noise.

A review on statistical and structural approaches applied to texture description is presented by Haralick [39].

An alternative to characterise texture is in the spectral domain. Primarily, the application of a Fourier transform to the image was used to detect periodicity in the image. The Fourier spectrum is suitable for describing the periodic or almost periodic patterns in an image. Gabor filters [94] and wavelet transform are also used to texture feature extraction. Randen and Husoy [101] review the most major filtering approaches to texture feature extraction and perform a comparative study with two non spectral techniques.

2.5 OBJECT MATCHING

The correspondence of detected objects in consecutive frames represents another important step in tracking systems. Object correspondence is a difficult problem in the presence of detection errors, occlusions (self, inter-object or background), exits, and entries of objects. Several approaches are presented in the literature, depending on the final application and feature properties, different matching techniques have been widely used.

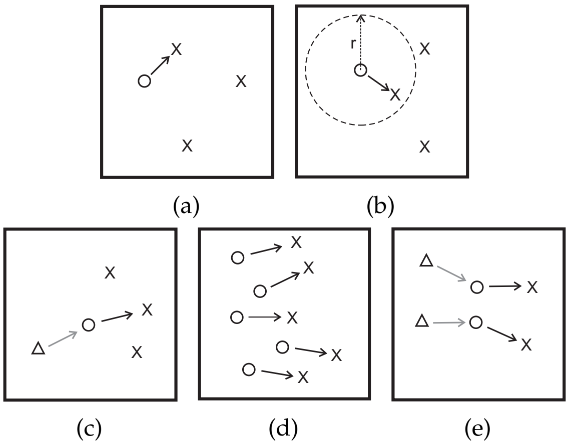


Figure 5: Different motion deterministic constraints. △ denotes object position at frame $t-2$, ∘ denotes object position at frame $t-1$, and finally × denotes object position at frame t.

2.5.1 *Point correspondence*

Concerning point correspondence, two categories of methods, called deterministic or statistical, appear in the literature. Using deterministic methods, qualitative motion heuristics are used to constrain possible tracks and to identify the optimal track set [129]. These qualitative motion heuristics could be proximity (Figure 5a), maximum velocity (Figure 5b), smooth motion (Figure 5c), common motion (Figure 5d), rigidity (Figure 5e) or any combination of them. Statistical methods use the measurement and the motion model uncertainties into account to establish the correspondence. Qualitative motion heuristics can also be used in the context of tracking using statistical methods. In this type of methods, statistical tools such as Kalman filter, particle filter, Joint Probability Data Association Filtering (JPDAF) or Multiple Hypothesis Tracking (MHT) are used.

Sethi and Jain [111] solve the correspondence problem as an optimisation problem using a greedy approach based on the notion of path coherence and assuming smoothness of motion. The algorithm is initialised by the nearest neighbour criterion. Using these nearest neighbours, m trajectories are initialised. Therefore, the problem is to find trajectories of m points in n frames. To solve this problem, general assumptions based on motion characteristics have been made. These assumptions are the following: an element in a frame can only belong to one trajectory; there should be m trajectories, each containing n points; for each trajectory, the deviation should be minimal; the sum of the deviations for trajectories should be minimal. The correspondences are exchanged iteratively to minimise the function cost. However, this method cannot handle occlusions, entries or exits.

Salari and Sethi [110] handle these problems by first establishing correspondence for the detected points using a modifying greedy algorithm capable to extend the tracking of missing objects by adding a number of hypothetical points. This is achieved by specifying local constraints that limits the acceptable location of a feature point in the next frame given its location in two previous frames.

Rangarajan and Shah [102] propose a greedy algorithm which is constrained by proximal uniformity. According to this constraint, most objects in the real world follow a smooth path and cover a small distance in a small time. Therefore, given the location of a point in a frame, its location in the next frame lies in the proximity of its previous location. Initial correspondences are obtained by computing optical flow in the first two frames. The method does not address entry and exit of objects. If the number of detected points decrease, occlusion or misdetection is assumed. Occlusion and misdetection are handled by predicting the position of the objects based on a constant velocity assumption.

Veenman et al. [129] extend the work of Sethi and Jain, and Rangarajan and Shah by introducing the Greedy Optimal Assignment (GOA) Tracker. The cost function is minimised by Hungarian assignment algorithm in two consecutive frames. The algorithm also assumes that the number of points remains the same over all frames. Shafique and Shah [113] propose a multiframe approach to preserve temporal coherency of speed and position. The correspondence problem is formulated as a graph theoretic problem. This work is closely related to Veenman et al. [129], however, multiple frame correspondence problem is introduced as opposition to the two frame correspondence problem presented in [129]. In addition, GOA assumes that the number of points in the scene remains constant, which is not a constrain for the algorithm proposed by Shafique and Shah. The correspondence is established by a greedy algorithm and uses a window of frames during point correspondence task to handle with occlusions whose durations are shorter than the temporal window used.

Statistical correspondence methods can also be used in correspondence problems by taking the measurement and the model uncertainties into account during object state estimation. The statistical correspondence methods use the state space approach to model the object properties such as position, velocity, and acceleration.

In situations where the state is assumed to be described by a Gaussian probability distribution, a Kalman filter can be used to estimate the state of a linear system. Kalman filtering is composed of two steps: prediction and correction. The prediction step uses the state model to predict the new state of the variables. Similarly, the correction step uses the current observations to update the state vector of the object. The Kalman filter has been extensively used in the vision community for tracking purposes [15, 66, 126, 143].

One limitation of the Kalman filter is the assumption that the state variables follow a normal probability distribution. This limitation can be overcome by using particle filtering.

When tracking multiple objects using Kalman or particle filters, it is necessary to associate the most likely measurement for a particular object state vector, that is, the correspondence problem needs to be solved before these filters can be updated. The simplest method to perform correspondence is to use the nearest neighbour approach. However, if two objects are close to each other, then there is a chance that the correspondence will be incorrect. An incorrectly associated measurement can cause the filter to fail to converge. There exist several statistical data association techniques to deal with this problem. Two widely used techniques for data association are the JPDAF and the MHT. The major limitation of the JPDAF algorithm is its inability to handle new objects since it performs data association of a fixed number of objects tracked over two frames. The MHT overcomes this limitation. It is an iterative algorithm that begins with a set of current track hypotheses. For each hypothesis, a prediction of the position of each object is made. These predictions are compared with actual measurements and a set of correspondences are established for each hypothesis based on a distance measure. Each measurement can belong to a new object, a previously tracked object a or a false measurement. Besides, a measurement may not be assigned to an object because the object may have exited or a measurement corresponding to an object may not be obtained due to occlusion or misdetection. In comparison with the JPDAF, the MHT algorithm is more computationally expensive both in memory and time. A detailed review of these two statistical data association techniques can be found in the survey presented by Cox [27].

An alternative to the point correspondence methods referred above, are the matching methods based in kernel, shape or contour matching.

2.5.2 *Kernel matching*

Tracking algorithms using kernel matching are typically performed by computing the motion of the object, which is represented by a primitive object region such as a rectangle, an ellipse, etc, from one frame to the next. Primitive geometric shapes are most suitable to represent simple rigid objects, however they could also be used for tracking nonrigid objects combining some primitive shapes. One approach is to use template matching. Templates are formed using simple geometric shapes carrying both spatial and appearance information. Template matching consists of searching in the current image for a region similar to the object template. The position and, consequently, the object matching between two consecutive frames is achieved by computing a similarity measure such as cross-correlation process. The cross-correlation concept and its normalised version are presented in Lewis [64].

The concept of cross-correlation for template matching is based in a distance measure (squared Euclidean distance). A squared Euclidean distance, $d^2_{I,T}(u,v)$ can be computed as

$$d^2_{I,T}(u,v) = \sum_{x,y} [I(x,y) - T(x-u,y-v)]^2, \tag{2.18}$$

where I is the image and the sum is over (x,y) under the window containing the feature T positioned at (u,v).

Expanding Equation 2.18 as

$$d^2_{I,T}(u,v) = \sum_{x,y} [I^2(x,y) - 2I(x,y)T(x-u,y-v) + T^2(x-u,y-v)], \tag{2.19}$$

the term $\sum T^2(x-u,y-v)$ is constant. If the term $\sum I^2(x,y)$ is approximately constant, then the remaining cross-correlation term

$$c(u,v) = \sum_{x,y} I(x,y)T(x-u,y-v), \tag{2.20}$$

is a measure of the similarity between the image and the template.

However, there are several disadvantages when using Equation 2.20 for template matching:

1. Equation 2.20 tends to match the brightest regions rather than the best topological fit. If the image energy $\sum I^2(x, y)$ changes with position, the correlation between the template and an exactly matching region in the image may be less than the correlation between the template and a bright spot in the image.

2. The range of $c(u, v)$ is dependent on the size of the template.

3. Equation 2.20 is not invariant to changes in image amplitude such as those caused by changing lighting conditions across the image sequence.

To overcome these difficulties the Normalized Cross-Correlation (NCC) is introduced

$$\gamma(u, v) = \frac{\sum_{x,y}[I(x, y) - \bar{I}_{u,v}][T(x - u, y - v) - \bar{T}]}{\left\{\sum_{x,y}[I(x, y) - \bar{I}_{u,v}]^2 \sum_{x,y}[T(x - u, y - v) - \bar{T}]^2\right\}^{\frac{1}{2}}}, \quad (2.21)$$

where $\bar{T}$ is the mean grey level of the template and $\bar{I}_{u,v}$ is the mean of $I(x, y)$ in the region under the template.

Despite of this improvements, NCC is not the ideal approach to feature tracking since it is not invariant with respect to imaging scale, rotation, and perspective distortions.

To overcome the shortcomings of the previous approach, Porikli et al. [96] proposed a covariance matrix representation to describe the object. A feature is defined by *d*-dimensional feature vectors $f_k k = 1..n$ constructed using two types of information: spatial attributes and appearance attributes, such as, colour, gradient, edge magnitude, edge orientation, filter responses, etc. These attributes may be associated directly to the pixel coordinates or they can be arranged in radial relationship. This latter association offers rotation invariant spatial formation of the features. The computed covariance matrix for a feature defined by a rectangular region is a symmetric matrix where its diagonal entries represent the variance of each attribute and the non-diagonal entries represent their respective correlations. For each frame, the whole image is searched to find the region which has the smallest distance between the covariance matrices of the target object window and the candidate regions. To incorporate a model update algorithm to adapt the object deformations and appearance changes, a mean covariance matrix of the T previous matrices is computed.

On the other hand, when the search region (current image) is too large, the process is more computational expensive during the matching process. To overcome this high computational cost, the object search area is reduced. Usually the search area is constrained at the neighbourhood of object previous position or at the neighbourhood of an estimated position obtained by a prediction and estimation algorithm such as Kalman filter.

Instead of templates, other object representations can be used for matching, for instance, colour statistics or histogram based information of the pixels inside the primitive shape boundaries. Fieguth and Terzopoulos [31] define a rectangular object region to be tracked and the mean colour of the pixels inside the region is computed to model the object. For each frame, different regions of search are defined. These regions are centred at different positions, encircling the predicted location. The similarity between the object model and each one of the hypothesised regions is calculated. The region that provides the highest ratio of similarity is selected as the current object location.

In Comaniciu et al. [24] a weighted histogram from a circular region is used. This weighted scheme assigns smaller weights to pixels farther from the centre of the region. To locate the object, an iterative algorithm consisting of a mean shift procedure followed by a histogram similarity measure is used. This technique of modelling an object by a weighted colour histogram is also adopted by Muñoz-Salinas et al. [85] where the information of the distance between the position of the object and its predicted position and the proximity between the colour model of the object are used as inputs in a cost function.

In the works presented by Avidan [5, 6] a Support Vector Machine (SVM) based classifier has been used for matching. SVM is a classification method that, given a set of positive and negative training examples, finds the best separating hyperplane between these two classes. Positive training examples are images of the object to be tracked and the negative examples consist, in general, of background regions that could be confused with the object. During the test period, the SVM indicates the degree of membership of the test data to the positive class. Avidan's tracking method is achieved by maximisation of the SVM membership degrees over image regions in order to estimate the position of the object.

In Nguyen and Bhanu [87] a tracking algorithm based on Bacterial Foraging Optimization (BFO) is introduced. Several bacterial particles, or agents, are generated and placed randomly in the image. These agents can move through the image looking for regions with higher fitness according a fitness function defined previously. Agents that cannot find areas of high fitness are relocated near to the agents with the best fitness. To ensure that the search space remains covered, a dispersal step is performed.

2.5.3 *Shape matching*

Shape matching can be performed in similar way than tracking based on kernel matching (Section 2.5.2). In this case an object silhouette and its associated model is searched in the current frame. The matching is performed using a similarity measure between the object shape in the current frame and the estimated shape model based on previous frame. The model of the object, usually an edge map, can be updated during time to accommodate viewpoint, changes in lighting condition and nonrigid object motion.

In the work presented by Huttenlocher et al. [49], a shape matching scheme is performed. The representation of the object is based on an edge map and the similarity measure is achieved using the Hausdorff distance. The main condition imposed by this method is that the two dimensional shape of an object not change rapidly between two consecutive frames. The model of the object is also updated during time to incorporate changes in the object shape.

One limitation of this approach could be the fact that only one object can be tracked in a sequence.

An alternative to model the object shape as an edge map representation is to use the silhouette of the object. Typically, the object silhouette detection is carried out using background subtraction (Section 2.4.1) and, consequently, an object model is constructed. Object models are, in general, in the form of colour or edges histograms, silhouette boundary or a combination of these models.

To distinguish people that were already tracked from new people that emerge in the scene, Haritaoglu et al. [44] combine together the grey scale appearance and shape information of a person in an appearance dynamic model.

Kang et al.[56] proposed an object model using a combination of colour and edges histograms. To provide rotation, translation and scale invariant models, they use a scheme of special histograms generated by concentric circles with different radius centred on a set of control points located in the smallest circle that involves the object silhouette. These way, the appearance model encodes both colour and shape information of the detected object. Using several distance measures such as cross-correlation, Bhattacharya distance and Kullback-Leibler distance, the similarity between the model is obtained. Comparing the performance of these distance measures, the authors conclude that both the Bhattacharya and Kullback-Leibler distance perform similarly and better than the cross-correlation based measure.

2.5.4 *Contour tracking*

Active contour based algorithms track objects by representing their outlines as bounding contours and updating these contours in successive frames. Two different approaches can be used to perform such evolution. One approach defines the contour by minimising the contour energy using minimisation techniques, the other uses state space models to model the contour shape and motion. The first approach is closely related with the contour based segmentation introduced earlier and, in this case, the shape of the contour is dictated only by the forces acting on it, and does not necessarily belong to any specific family of curves that could be mathematically described. The contour energy can be defined in terms of temporal information based on the optical flow.

In Kim et al.[60] the initialisation of the contour is assumed and, two modes of contour tracking are discussed. Rolling mode is used when the overlapped area exists between two sequential images for the object to be tracked, Figure 6. Contour nodes from the previous image are rolled into the boundary of the object of which location is moved to the other region in the next image. When an object area is not overlapped between two consecutive frames, a jump mode is used, Figure 7. In the jump mode the nodes of the contour are jumped to the next frame based on the information of the optical flow and the contour is reinitialised by performing a segmentation process. Optical flow is estimated using correlation between two regions.

A similar approach is presented by Bertalmío et al. [10] where the optical flow is computed for all pixels of the contour. In Akgul et al. [2] and

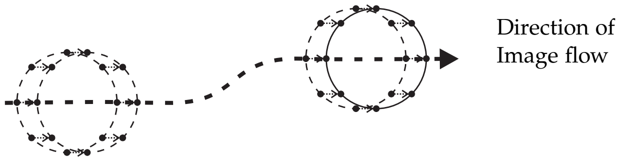


Figure 6: The concept of rolling (adapted from Kim et al.[60]).

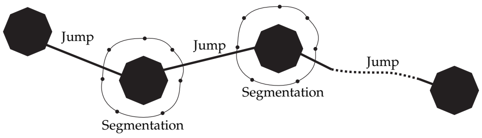


Figure 7: The concept of jumping (adapted from Kim et al.[60]).

Mikic et al. [82], an active contour tracking method guided by optical flow information is applied in ultrasound image sequences to track internal body structures for medical purposes.

An alternative of using minimisation techniques is proposed by Isard and Blake [50]. The contour state is defined in terms of spline shape parameters and motion parameters. The contour state is updated using a particle filter which measurements consist of image edges computed in the normal direction to the contour. With an appropriate parametrisation, a complex set of deformable contours could be described by a reasonable number of parameters. The key advantage over the previous method is that acceptable contours are guaranteed by the mathematical model itself, not by a particular energy function.

2.6 OCCLUSION

Occlusion is a frequent situation that appears in tracking systems. It can be classified into three categories: self occlusion, inter object occlusion, and occlusion by the background scene structure [140]. Self occlusion occurs when one part of the object occludes another part of the same object. Inter object occlusion occurs when one or more objects being tracked occlude other objects. Occlusion by the background occurs when a structure in the background occludes the tracked objects. In single camera tracking systems, misdetection errors are frequently considered as occlusion cases. Using different feature detection algorithms or multiple camera viewing, this erroneous consideration can be overcame.

A common approach to deal with the presence of misdetections or occlusions is to estimate objects position using the object motion model until the object reappears. In the traffic surveillance system for urban intersections presented by Zhang et al. [143], the occlusion problem is solved by establishing predictive positions for the vehicles, using a Kalman filter estimator.

An alternative approach to cope with occlusion is the implementation of a multiple view image tracking using a set of calibrated cameras. Black et al. [15] use collaboration between multiple views to solve dynamic (inter object) and static (by the background) occlusions. Kalman filter is also used to track each object in 3D world coordinates and 2D image coordinates. Information is shared between the 2D and 3D trackers of each camera in order to improve the performance of object tracking and trajectory prediction. When the Field of View (FoV) of two cameras is separated spatially by a small distance, the system tracks an object when it leaves one FoV and enters another using trajectory prediction in 3D world coordinates to determine when the object should reappear in the adjacent FoV. If the object does not appear when expected it is deleted.

2.7 DISCUSSION

The aim of tracking is to estimate a trajectory of an object when it moves around a scene. Object tracking is performed every day by humans, often in an unconscious way. Tracking is performed when we want to cross a road and we are attentive to the speed of the cars that are approaching. If their speed starts reducing and we estimate that it will stop in security, we can cross the road, otherwise we must wait. Tracking happens also when we walk into a busy street and we try not to lose sight of our friends and estimate their trajectories to stay close, or when we assist a football game and our eyes follow the ball or the players.

Situations where a tracking system can be applied are wide (surveillance, HMI, traffic monitoring, sports and medical applications) but, after years of improvements, a construction of a robust object tracking system is still a challenge. Despite its definition was not consensual, it is constituted by two general and important steps: object detection and matching.

Object detection is itself a challenge and some distinctive properties that represents the object must be defined. The performance of the detection task influences the subsequent process of matching and tracking. Good tracking systems need good object detection methods. Considering a moving object, object motion could be a suitable property that can be extracted from the sequence. Several techniques to estimate object motion in the image plane are frame differencing, optical flow or background subtraction. Frame differencing is a simple and not computational expensive technique but it requires a good selection of a threshold value, it is sensitive to noise and, objects can contain holes inside when they have regions with constant grey level intensity or colour. Optical flow is a more stable approach, however it requires high computation time and, since it is a neighbourhood operation, the aperture problem can occurs. Another method is background subtraction. This method is computationally efficient, but a representation of the stationary background is required. This representation can achieve high complexity when it incorporates illumination changes, noise or the periodic motion of background regions.

An alternative to locate the objects in an image is to search object properties such as contours, intensity, colour or texture.

Object boundaries generates strong changes in image intensity and edge detection techniques can be computed to extract the object in the image. Good results are obtained when using the popular Canny operator (first order differentiation) or the LoG (second order differentiation) but, in general, the resulting boundary is often broken. Some linking strategies are based in the magnitude and direction of the gradient. A global method to detect edges is the HT. It can detect lines, circles and ellipses even with partial occluded edges, however, it requires high computational requirements.

When the object denotes an intensity level different from the background, the well known technique proposed by Otsu can be applied. Nevertheless, this constraint is hard to achieve for all images in the sequence and the background suffers illumination changes and noise. SRG approaches sounds attractive but the definition of the initial seeds are a critical point. Watershed methods are a different approach that produce regions with well defined contours but it is more computationally expensive with comparison to threshold and SRG techniques.

Object matching strategy depends on the representation model of the object that is selected according the final application. Point, kernel or shape matching are common techniques to perform this task. Contour tracking are also used particularly in medical purposes. Point representation is the simplest and can be used when the dimensions of the object is considerably lower than the entire image or when we are tracking a specific and small feature present in the object. There exist deterministic and statistical methods to perform point matching, however, a combination of both is widely used. More complex and resource expensive matching schemes are based on primitive geometric shapes and a template or in shape information. In the first case a rectangle, circle or ellipse are used to represent the object region and an appearance model (colour statistics or region histogram) are used to represent the object and mathematical tools as correlation, mean shift or SVM are applied to perform the correspondence process. In shape matching the object region is defined by its edges or by its silhouette and dynamic models must be incorporated to handle with object shape changing. These methods, using further information about the object, are more robust than point matching, nevertheless, object models could be complex and higher computational resources are required. Contour tracking is based on energy minimisation process or spline shape parameters estimation. In both cases, information concerning image gradient or optical flow is required for contour convergence.

Interpreting and solve occlusion situations is a challenge in tracking algorithms. Several types of occlusion can appear and, with a monocular system is hard to distinguish them and often they are considered as misdetections situations. Using a more expensive vision system, with several cameras and information interchanging is possible to handle with inter object and background occlusions.

As far as the laws of mathematics refer to reality, they are not certain.
And as far as they are certain, they do not refer to reality.

— Albert Einstein

3 FUZZY LOGIC THEORY IN OBJECT TRACKING

3.1 INTRODUCTION

Fuzzy logic was first developed to be a representation scheme and calculus to handle with uncertain or vague notions. In their essence, it could be seen as a multi-valued logic incorporating intermediate categories or degrees between notations such as true/false, yes/no, 1/0, or similar, commonly used in Boolean logic. Therefore, fuzzy logic can be seen as an extension of conventional Boolean logic that was extended to deal with the concept of partial truth or partial false rather than the absolute values and categories present in Boolean logic. As its name suggests, it is the logic underlying modes of reasoning which are approximate rather than exact. The importance of fuzzy logic arises from the fact that most modes of human reasoning and especially common sense reasoning are approximate by nature.

In this chapter a brief presentation about the emerging of fuzzy concepts is introduced in Section 3.2. After this short historical introduction, the basic concepts of fuzzy logic concerning fuzzy sets, properties, operations and commonly used membership functions are presented, respectively, in Sections 3.3 and 3.3.5. Several fuzziness measures in fuzzy sets are presented in Section 3.3.6. The state of art concerning the application of fuzzy logic concepts in object tracking tasks, such as object detection and matching, is presented in Section 3.4. Finally, in Section 3.5, a discussion about this chapter is performed.

3.2 HISTORICAL FOUNDATIONS OF FUZZY LOGIC

The success of the precision in classical mathematics depends, in large part, to the efforts of the classical Greek philosopher Aristotle and the philosophers who preceded him. In their efforts to formulate a concise theory of logic, and later mathematics, the so-called "Laws of Thought" were posted. One of these thoughts, the Law of Excluded Middle, stated that every proposition must either be true or false. However, considering a middle filled glass, the glass can be half-full and half-not-full. Clearly, this disprove the Aristotle's law of bivalence. It was Plato (Aristotle's teacher) that laid the foundation for what would become fuzzy logic, by proposing a third region between true and false where the two notions collapse together. But it was Jan Lukasiewicz who first proposed a systematic alternative to the bi-valued logic of Aristotle. In his paper, published in 1920, titled *"On Three-Valued Logic"*[1], Lukasiewicz extended the conventional bi-valued logic of Aristotle and proposed a tri-valued logic where a new truth value, so-called possible, was added to the classical binary logic. Lukasiewicz made experiments with four and five valued logic and hypothesised the possibility of infinite-valued logic. This new logic approach, however, did not gain much prominence until Zadeh has introduced the notion of an infinite-valued logic, describing the mathematical support of fuzzy sets theory.

3.3 ESSENTIALS OF FUZZY LOGIC THEORY

In 1965, fuzzy sets were introduced by Zadeh [141] to represent or manipulate data and information containing non-statistical uncertainties. This theory was specifically created to mathematically represent uncertainty and vagueness and to provide tools for dealing with the imprecision intrinsic to many problems. Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic offers a mathematical framework to incorporate the uncertainties associated with human cognitive processes, such as thinking and reasoning. Some of the essential characteristics of fuzzy logic are related to the following (Zadeh [142]):

- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning;
- In fuzzy logic, everything is a matter of degree;
- In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables;
- Inference is viewed as a process of propagation of elastic constraints;
- Any logical system can be fuzzified.

3.3.1 *Fuzzy Sets*

A classical (crisp) set is defined as a collection of elements $x \in X$ where each single element can either belong to or not belong to a set A, $A \subseteq X$. Such classical set can be described in different ways: one can either enumerate the elements that belong to the set, describe the set analytically or define the member elements by using a characteristic function. Using this latter way, the subset A may be represented as a set of ordered pairs, with exactly one ordered pair presented for each element of X. The first element of the ordered pair is an element of the set X, and the second element is an element of the set $\{0, 1\}$. The value 1 is used to represent that the element belongs to the set and the value 0 means that the element doesn't belong to the set.

[1] Translation of the title of the original paper "J. Lukasiewicz, *O logice trójwartościowej*, Ruch filozoficzny, 5:170-171, 1920" written in Polish. English version available in "*Jan Lukasiewicz: Selected Works*, L. Borkowski (ed.), North-Holland, Amsterdam, 87-88, 1970. ISBN 0720422523".

Definition Let $X = \{x_1, ..., x_n\}$ be an ordinary finite non-empty set. A subset A in X is as set of ordered pairs $A = \{(x, \chi_A(x)) | x \in X\}$, where $\chi_A : X \to \{0, 1\}$ represents the characteristic function.

Knowing that, in a crisp set each element has a 0 or 1 membership to the set, one can look at a fuzzy set as an extension or a generalisation of a classical set since its elements can achieve various degrees of membership in the set. Therefore, fuzzy sets have more flexible membership requirements allowing the elements to have partial memberships between 0 and 1 rather than the unique memberships 0 and 1 like in classical sets.

Definition Let $X = \{x_1, ..., x_n\}$ be an ordinary finite non-empty set. A fuzzy set $\tilde{A}$ in X is as set of ordered pairs $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$, where $\mu_{\tilde{A}} : X \to [0, 1]$ represents the membership function.

Definition A fuzzy set $\tilde{A}$ is said to be empty, written $\tilde{A} = \emptyset$, if and only if

$$\mu_{\tilde{A}}(x) = 0 \forall x \in X. \tag{3.1}$$

Definition Two fuzzy sets $\tilde{A}$ and $\tilde{B}$ in X are equal, written $\tilde{A} = \tilde{B}$, if and only if

$$\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x) \forall x \in X. \tag{3.2}$$

Instead of writing $\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x) \forall x \in X$, it can be written, more simply, $\mu_{\tilde{A}} = \mu_{\tilde{B}} \forall x \in X$.

The membership function $\mu_{\tilde{A}}$ is also called grade of membership, degree of compatibility or degree of truth. The range of this function is a subset of the non-negative real numbers whose supremum is finite, normally 1.

Definition A fuzzy set $\tilde{A}$ defined in X, is said to be normal if and only if

$$\max \mu_{\tilde{A}}(x) = 1, \tag{3.3}$$

i. e., there exists at least an element $x \in X$, such that $\mu_{\tilde{A}}(x) = 1$. Otherwise, the fuzzy set is called subnormal.

As stated previously, a fuzzy set is a generalisation of a classical set and the membership function is a generalisation of the characteristic function. Some elements of a fuzzy set may have the degree of membership zero. Often it is appropriate to consider those elements that have nonzero degree of membership in a fuzzy set.

Definition The support of a fuzzy set $\tilde{A}$ in X, written $\text{supp}(\tilde{A})$, is the crisp set

$$\text{supp}(\tilde{A}) = \{x \in X | \mu_{\tilde{A}}(x) > 0\}. \tag{3.4}$$

A more general and even more useful notion is that of an α-level set or α-cut.

Definition The α-cut of a fuzzy set $\tilde{A}$ in X, written $\tilde{A}_\alpha$, is the crisp set

$$\tilde{A}_\alpha = \{x \in X | \mu_{\tilde{A}}(x) \geqslant \alpha\}. \tag{3.5}$$

$\tilde{A}_\alpha^{'} = \{x \in X : \mu_{\tilde{A}}(x) > \alpha\}$ is called the strong α-level set or strong α-cut of a fuzzy set $\tilde{A}$ in X.

3.3.2 *Properties of Fuzzy Sets*

Since classical sets can be thought as a particular case of fuzzy sets, fuzzy sets follow the same properties as crisp sets. Frequently used properties of fuzzy sets are presented in Ross [108].

P1. Commutativity

$$\begin{aligned} &\tilde{A} \cup \tilde{B} = \tilde{B} \cup \tilde{A} \\ &\tilde{A} \cap \tilde{B} = \tilde{B} \cap \tilde{A} \end{aligned} \tag{3.6}$$

P2. Associativity

$$\begin{aligned} &\tilde{A} \cup (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cup \tilde{C} \\ &\tilde{A} \cap (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cap \tilde{C} \end{aligned} \tag{3.7}$$

P3. Distributivity

$$\begin{aligned} &\tilde{A} \cup (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cap (\tilde{A} \cup \tilde{C}) \\ &\tilde{A} \cap (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{C}) \end{aligned} \tag{3.8}$$

P4. Idempotency

$$\tilde{A} \cup \tilde{A} = \tilde{A} \text{ and } \tilde{A} \cap \tilde{A} = \tilde{A} \tag{3.9}$$

P5. Identity

$$\begin{aligned} &\tilde{A} \cup \emptyset = \tilde{A} \text{ and } \tilde{A} \cap X = \tilde{A} \\ &\tilde{A} \cap \emptyset = \emptyset \text{ and } \tilde{A} \cup X = X \end{aligned} \tag{3.10}$$

P6. Transitivity

$$\text{If } \tilde{A} \subseteq \tilde{B} \text{ and } \tilde{B} \subseteq \tilde{C} \text{ then } \tilde{A} \subseteq \tilde{C} \tag{3.11}$$

P7. Involution

$$\neg\neg\tilde{A} = \tilde{A} \tag{3.12}$$

3.3.3 *Basic Fuzzy Sets Operations*

The basic operations in fuzzy set theory are the complement, intersection and union. Since the membership function is the crucial component of a fuzzy set, it is therefore not surprising that operations with fuzzy sets are defined via their membership functions. These concepts, firstly suggested by Zadeh [141], constitute a consistent framework for the theory of fuzzy sets. However, they are not unique since Zadeh and other authors have suggested consistent alternative or additional definitions for fuzzy set operations.

Definition The complement of a fuzzy set $\tilde{A}$ in X, written $\neg\tilde{A}$, is the fuzzy set

$$\neg\tilde{A} = \{(x, \mu_{\neg\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x)) | x \in X\}. \tag{3.13}$$

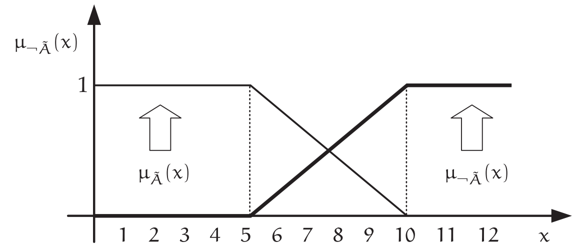


Figure 8: The complement of a fuzzy set (adapted from Kacprzyk [54]).

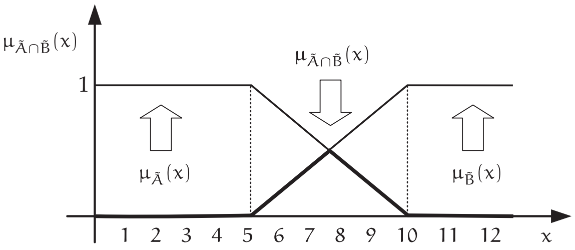


Figure 9: The intersection of two fuzzy sets (adapted from Kacprzyk [54]).

A graphical idea of the complement can be visualised in Figure 8.

Definition The intersection of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ in X, written $\tilde{A} \cap \tilde{B}$, is the fuzzy set

$$\tilde{A} \cap \tilde{B} = \{(x, \mu_{\tilde{A}\cap\tilde{B}}(x) = \wedge(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)))|x \in X\}, \tag{3.14}$$

where $\wedge$ is the minimum operator. The intersection of two fuzzy sets is illustrated in Figure 9.

Definition The union of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ in X, written $\tilde{A} \cup \tilde{B}$, is the fuzzy set

$$\tilde{A} \cup \tilde{B} = \{(x, \mu_{\tilde{A}\cup\tilde{B}}(x) = \vee(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)))|x \in X\}, \tag{3.15}$$

where $\vee$ is the maximum operator. The essence of the union can be portrayed as in Figure 10.

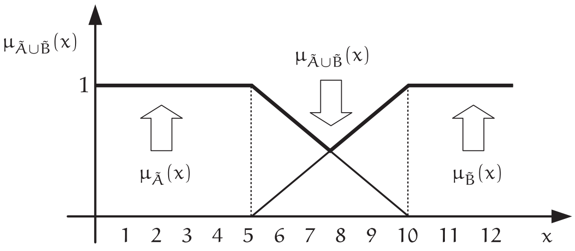


Figure 10: The union of two fuzzy sets (adapted from Kacprzyk [54]).

A closely related pair of properties which hold in ordinary set theory are the law of excluded middle, $A \cup \neg A = X$, and the law of non-contradiction principle, $A \cap \neg A = \emptyset$. However, the laws of excluded middle and non-contradiction are not satisfied in fuzzy logic. This laws are also known as the Laws of Aristotle.

Lemma 3.3.1 *The law of excluded middle is not valid.*
Let $\mu_{\tilde{A}}(x) = \frac{1}{2} \forall x \in X$, *then it is easy to see that*

$$\begin{aligned} \tilde{A} \cup \neg \tilde{A} &= \max\{\mu_{\tilde{A}}, \mu_{\neg \tilde{A}}\} \\ &= \max\{\frac{1}{2}, 1 - \frac{1}{2}\} \\ &= \frac{1}{2} \neq 1 \end{aligned}$$

Lemma 3.3.2 *The law of non-contradiction is not valid.*
Let $\mu_{\tilde{A}}(x) = \frac{1}{2} \forall x \in X$, *then it is easy to see that*

$$\begin{aligned} \tilde{A} \cap \neg \tilde{A} &= \min\{\mu_{\tilde{A}}, \mu_{\neg \tilde{A}}\} \\ &= \min\{\frac{1}{2}, 1 - \frac{1}{2}\} \\ &= \frac{1}{2} \neq 0 \end{aligned}$$

However, fuzzy logic satisfies De Morgan's laws.

$$\begin{aligned} \neg(\tilde{A} \cap \tilde{B}) &= \neg \tilde{A} \cup \neg \tilde{B} \\ \neg(\tilde{A} \cup \tilde{B}) &= \neg \tilde{A} \cap \neg \tilde{B} \end{aligned}$$

3.3.4 *General Fuzzy Sets Operations*

The operations on fuzzy sets listed in Equations 3.13, 3.14 and 3.15 are called the standard fuzzy operations [108]. However, these standard fuzzy operations are not the only operations that can be applied to fuzzy sets and, for each standard operation, there exists a class of functions that can be considered a general definition of these standard operations. General operators for the intersection and union of fuzzy sets are referred as triangular norms (*t-norms*) and triangular conorms (*t-conorms* or *s-norms*), respectively.

Definition A function

$$t : [0,1] \times [0,1] \to [0,1], \tag{3.16}$$

satisfying, for each $a, b, c, d \in [0,1]$, the following properties:

P1. it has 1 as the unit element: $t(a, 1) = a$;

P2. it is monotone: $t(a, b) \leqslant t(c, d)$ if $a \leqslant c$ and $b \leqslant d$;

P3. it is commutative: $t(a, b) = t(b, a)$;

P4. it is associative: $t[t(a, b), c] = t[a, t(b, c)]$.

is called a *t-norm*.

Some relevant examples of *t-norms* are referred in Kacprzyk [54]:

1. the minimum: $t(a, b) = a \wedge b = \min(a, b)$. Which was proposed by Zadeh [141].
2. the algebraic product: $t(a, b) = a \cdot b$
3. the Lukasiewicz *t-norm*: $t(a, b) = \max(0, a + b - 1)$

Definition A function

$$s : [0, 1] \times [0, 1] \to [0, 1], \tag{3.17}$$

satisfying, for each $a, b, c, d \in [0, 1]$, the following properties:

P1. it has 0 as the unit element: $s(a, 0) = a$;

P2. it is monotone: $s(a, b) \leqslant s(c, d)$ if $a \leqslant c$ and $b \leqslant d$;

P3. it is commutative: $s(a, b) = s(b, a)$;

P4. it is associative: $s[s(a, b), c] = s[a, s(b, c)]$.

is called a *t-conorm* or *s-norm*.

Some relevant examples of *t-conorms* are also referred in Kacprzyk [54]:

1. the maximum: $t(a, b) = a \vee b = \max(a, b)$. Which was proposed by Zadeh [141].
2. the probabilistic product: $s(a, b) = a + b - ab$
3. the Lukasiewicz *s-norm*: $s(a, b) = \min(a + b, 1)$

Note that a *t-norm* is dual to an *s-norm* in that:

$$s(a, b) = 1 - t(1 - a, 1 - b). \tag{3.18}$$

3.3.5 *Membership functions*

All information contained in a fuzzy set is provided by its membership function. The most common membership functions described in the literature are described in this section.

The simplest membership functions are piecewise linear mathematical functions, formed using straight lines. A membership function to represent the fuzzy set *"integer numbers which are greater than* a*"* can be mathematically defined as follows:

$$\mu_1(x) = \mu(x, a, b) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(b-a)} & a \leqslant x \leqslant b \\ 1 & x > b \end{cases} \tag{3.19}$$

With parameters $a = 5$ and $b = 10$, this membership function is illustrated in Figure 11.

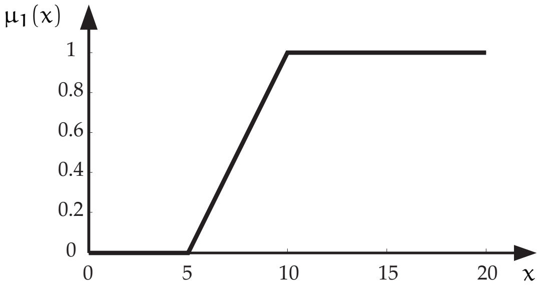


Figure 11: Piecewise linear membership function to represent the fuzzy set *"integer numbers which are much greater than* 5*"* with parameters $a = 5$ and $b = 10$.

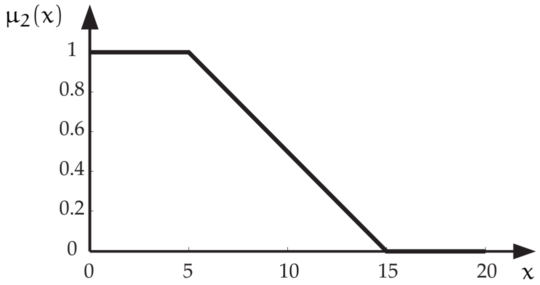


Figure 12: Piecewise linear membership function to represent the fuzzy set *"integer numbers which are much lower than* 15*"* with parameters $a = 5$ and $b = 15$.

In opposite, a membership function to represent the fuzzy set *"integer numbers which are lower than* b*"* is mathematically defined in Equation 3.20 and is depicted in Figure 12, with parameters $a = 5$ and $b = 15$.

$$\mu_2(x) = \mu(x, a, b) = \begin{cases} 1 & x < a \\ \frac{(b-x)}{(b-a)} & a \leqslant x \leqslant b \\ 0 & x > b \end{cases} \tag{3.20}$$

Membership functions to represent the fuzzy set *"integer numbers which are more or less* α*"* could be the triangular and trapezoidal functions. These membership functions are mathematically defined in Equations 3.21 and 3.22, respectively.

A triangular membership function is mathematically defined as

$$\mu_{tri}(x) = \mu(x, a, b, c) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(b-a)} & a \leqslant x \leqslant b \\ \frac{(c-x)}{(c-b)} & b \leqslant x \leqslant c \\ 0 & x > c \end{cases} \tag{3.21}$$

A triangular function with parameters $a = 7.5$, $b = \alpha = 10$ and $c = 15$ is depicted in Figure 13.

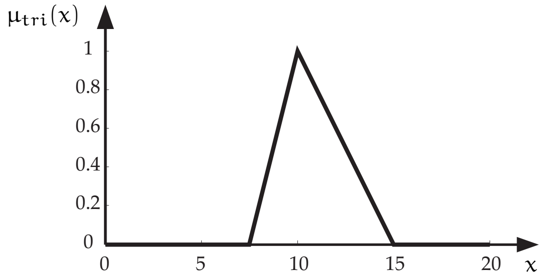


Figure 13: Triangular membership function to represent the fuzzy set *"integer numbers which are more or less 10"* with parameters $a = 7.5$, $b = \alpha = 10$ and $c = 15$.

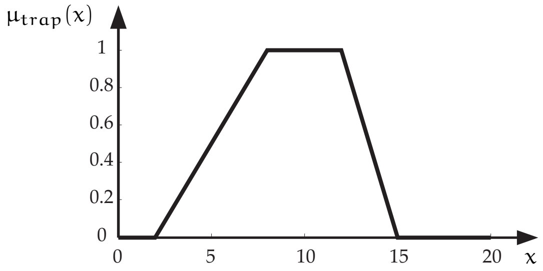


Figure 14: Trapezoidal membership function to represent the fuzzy set *"integer numbers which are more or less 10"* with parameters $a = 2$, $b = 8$, $c = 12$ and $d = 15$.

The trapezoidal membership function has a flat top and really is just a truncated triangle curve. It is defined as follows

$$\mu_{trap}(x) = \mu(x, a, b, c, d) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(b-a)} & a \leqslant x \leqslant b \\ 1 & b \leqslant x \leqslant c \\ \frac{(d-x)}{(d-c)} & c \leqslant x \leqslant d \\ 0 & x > d \end{cases} \tag{3.22}$$

A trapezoidal function with parameters $a = 2$, $b = 8$, $c = 12$ and $d = 15$ is depicted in Figure 14.

These piecewise linear membership functions have the advantage of simplicity, however, non linear membership functions can be employed to ensure variable slope behaviour.

The S-function can be used to represent fuzzy sets *"numbers which are greater than* a". This function is described as

$$\mu_S(x) = \mu(x, a, b, c) = \begin{cases} 0 & x \leqslant a \\ 2\left(\frac{x-a}{c-a}\right)^2 & a \leqslant x \leqslant b \\ 1 - 2\left(\frac{x-c}{c-a}\right)^2 & b \leqslant x \leqslant c \\ 1 & x \geqslant c, \end{cases} \tag{3.23}$$

where $b = \frac{1}{2}(a + c)$. The S-function can be controlled through parameters a and c. Parameter b is called the crossover point where $\mu_{AS}(b) = 0.5$. Considering $a = 10$, $b = 12.5$ and $c = 15$, its shape is illustrated in Figure 15.

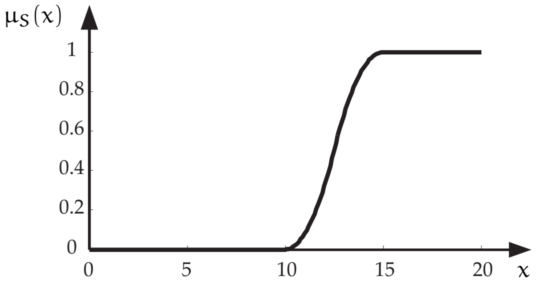


Figure 15: The S-function to represent the fuzzy set *"integer numbers which are much greater than 10"* with parameters $a = 10$, $b = 12.5$ and $c = 15$.

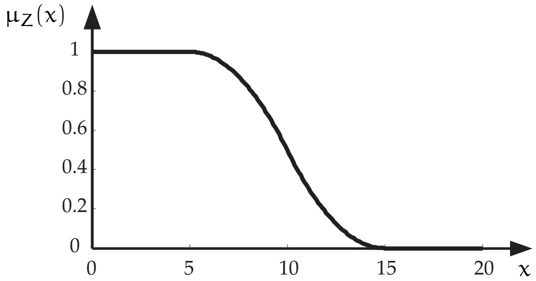


Figure 16: The Z-function to represent the fuzzy set *"integer numbers which are much lower than 15"* with parameters $a = 5$, $b = 10$ and $c = 15$.

The Z-function is defined by an expression obtained from S-function as follows

$$\mu_Z(x) = \mu(x, a, b, c) = 1 - \mu_S(x; a, b, c). \tag{3.24}$$

A Z-function with parameters $a = 5$, $b = 10$ and $c = 15$ is illustrated in Figure 16.

Non linear membership functions to represent the fuzzy set *"integer numbers which are more or less α"* could be the Gaussian and bell shaped functions.

The symmetric Gaussian membership function depends on two parameters σ and m as given by

$$\mu_{gauss}(x) = \mu(x, \sigma, m) = e^{\frac{-(x-m)^2}{2\sigma^2}}. \tag{3.25}$$

Considering $sigma = 2$ and $\alpha = m = 10$, a Gaussian function is portrayed in Figure 17.

The generalised bell shaped function depends on three parameters a, b, and c as given by

$$\mu_{bell}(x) = \mu(x, a, b) = \frac{1}{1 + \left(\frac{x-c}{a}\right)^{2b}}. \tag{3.26}$$

where the parameter b is usually positive. The parameter c locates the centre of the curve.

A bell-shaped function with parameters $a = 3$, $b = 4$ and $c = 10$, is illustrated in Figure 18.

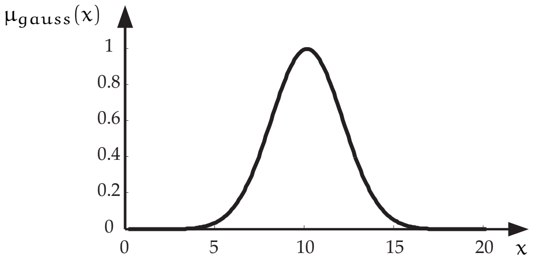


Figure 17: Gaussian membership function to represent the fuzzy set *"integer numbers which are more or less 10"* with parameters $\sigma = 2$ and $m = 10$.

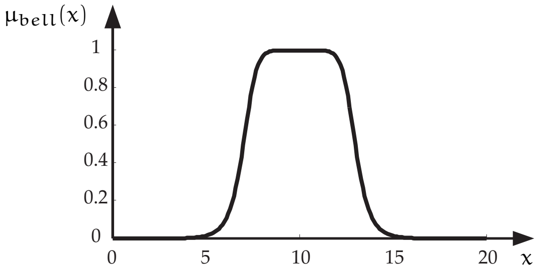


Figure 18: Bell-shaped function to represent the fuzzy set *"integer numbers which are more or less 10"* with parameters $a = 3$, $b = 4$ and $c = 10$.

3.3.6 *Measures of fuzziness*

A reasonable approach to estimate the average ambiguity within fuzzy sets is measuring their fuzziness [90]. The fuzziness of a crisp set should be zero, as there is no ambiguity about whether an element belongs to the set or not. A myriad of approaches to do this end are discussed in the literature.

In 1972, de Luca and Termini [28] introduced the concept of fuzziness measure in order to obtain a global measure of the vagueness connected with fuzzy sets.

Definition Let $\mu_{\tilde{A}}(x)$ be the membership function of the fuzzy set $\tilde{A}$ for $x \in X$, X finite. It seems desirable that the measure of fuzziness $d(\tilde{A})$ should have then the following properties:

P1. Crispness: $d(\tilde{A}) = 0$ if and only if $\mu_{\tilde{A}}(x) \in \{0, 1\}$ for all $x \in X$, i. e., $\tilde{A}$ is a crisp set in X.

P2. Maximality: $d(\tilde{A})$ assumes a unique maximum if and only if $\mu_{\tilde{A}}(x) = \frac{1}{2} \forall x \in X$.

P3. Resolution: $d(\tilde{A}) \geqslant d(\tilde{A}')$ if $\tilde{A}'$ is any sharpened version of $\tilde{A}$, that is: $\mu_{\tilde{A}'}(x) \leqslant \mu_{\tilde{A}}(x)$ if $\mu_{\tilde{A}}(x) \leqslant \frac{1}{2}$ and $\mu_{\tilde{A}'}(x) \geqslant \mu_{\tilde{A}}(x)$ if $\mu_{\tilde{A}}(x) \geqslant \frac{1}{2}$.

P4. Symmetry: $d(\tilde{A}) = d(\neg\tilde{A})$ where $\neg\tilde{A}$ is the complement of $\tilde{A}$.

Also, in de Luca and Termini [28], is suggested, as a measure of fuzziness, the entropy of a fuzzy set, defined as follows:

Definition The entropy, $d(\tilde{A})$, as a measure of fuzziness of a fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))\}$ is defined as

$$\begin{aligned} d(\tilde{A}) &= H(\tilde{A}) + H(\neg\tilde{A}), x \in X, \\ H(\tilde{A}) &= -K \sum_{i=1}^{n} \mu_{\tilde{A}}(x_i) \ln(\mu_{\tilde{A}}(x_i)), \end{aligned} \tag{3.27}$$

where n is the number of elements in the support (elements with a non-zero membership to the fuzzy set) of $\tilde{A}$ and K is a positive constant.

Using Shannons's function $S(x) = -x \ln x - (1-x)\ln(1-x)$, de Luca and Termini simplify the Equation 3.27 to arrive at the following definition:

Definition The entropy d as a measure of fuzziness of a fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))\}$ is defined as

$$d(\tilde{A}) = K \sum_{i=1}^{n} S(\mu_{\tilde{A}}(x_i)). \tag{3.28}$$

Yager [139] suggested that the basis for any measure of fuzziness should be a measure of the lack of distinction between $\tilde{A}$ and $\neg\tilde{A}$ or $\mu_{\tilde{A}}(x)$ and $\mu_{\neg\tilde{A}}(x)$. For that purpose, a possible metric to measure the distance between a fuzzy set and its complement is suggested by Yager:

Definition The distance between a fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))\}$ and its complement $\neg\tilde{A} = \{(x, 1 - \mu_{\tilde{A}}(x))\}$, written $D_p(\tilde{A}, \neg\tilde{A})$, is defined as

$$D_p(\tilde{A}, \neg\tilde{A}) = \left[\sum_{i=1}^{n} |\mu_{\tilde{A}}(x_i) - \mu_{\neg\tilde{A}}(x_i)| \right]^{1/p}, p = 1, 2, 3, \ldots \tag{3.29}$$

Definition The entropy d as a measure of fuzziness of a fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))\}$ can be defined as

$$d(\tilde{A}) = 1 - \frac{D_p(\tilde{A}, \neg\tilde{A})}{\|\mathrm{supp}(\tilde{A})\|}, \tag{3.30}$$

where supp denotes the support of $\tilde{A}$ and $\|.\|$ the cardinality of the set.

Kaufmann [58] introduced an Index of Fuzziness (IF) of any fuzzy set defined by a metric distance between its membership function and the membership function of its nearest crisp set.

Definition A fuzzy set $\tilde{A}^*$ is called the nearest crisp set of $\tilde{A}$ if the following conditions are satisfied

$$\mu_{\tilde{A}^*}(x) = \begin{cases} 0, & \text{if } \mu_{\tilde{A}}(x) < 0.5 \\ 1, & \text{if } \mu_{\tilde{A}}(x) \geqslant 0.5 \end{cases} \tag{3.31}$$

The index of fuzziness is then calculated by measuring the normalised distance between $\tilde{A}$ and $\tilde{A}^*$.

Definition The index of fuzziness of $\tilde{A}$ can be defined as

$$if_k(\tilde{A}) = \frac{2}{n^{1/k}} \left[\sum_{i=1}^{n} |\mu_{\tilde{A}}(x_i) - \mu_{\tilde{A}^*}(x_i)|^k \right]^{1/k}, \tag{3.32}$$

where n is the number of elements in $\tilde{A}, k \in [1, \infty[$. Depending if $k = 1$ or 2, the measure of fuzziness is called linear or quadratic.

Several well known measures of fuzziness for fuzzy sets are also presented and discussed in Pal and Bezdek [90].

3.3.7 *Extensions of Fuzzy Sets*

Since fuzzy sets were introduced, many approaches and theories dealing imprecision and uncertainty have been proposed. The use of extensions of fuzzy sets tends to overcome the problem of choosing the membership function since the success of the use of fuzzy sets theory depends on the choice of the membership function by an expert. However, there are some applications in which experts do not have precise knowledge about the membership function that should be taken.

Some of these new approaches like Interval Valued Fuzzy Sets (IVFSs), Type-2 Fuzzy Sets and Atanassov's Intuitionistic Fuzzy Sets (A-IFSs) are considered extensions of classical fuzzy sets theory.

Rather than assigning a membership degree to each element of the fuzzy set, in IVFSs theory the membership degree of each element of the set is given by a closed subinterval of the interval $[0, 1]$.

Definition Let $S([0, 1])$ denote the set of all closed sub-intervals of the interval $[0, 1]$ and let X be an ordinary finite non-empty set. An Interval Valued Fuzzy Set (IVFS) A in X is a mapping $A : X \to S([0, 1])$ such that

$$A = \{(x, A(x) = [\mu_L(x), \mu_U(x))] | x \in X\}, \tag{3.33}$$

where $\mu_L(x)$ and $\mu_U(x)$ are functions that satisfy the condition

$$\mu_L(x) \leqslant \mu_U(x) \forall x \in X. \tag{3.34}$$

$\mu_L(x)$ and $\mu_U(x)$ are called, respectively, the lower and upper membership functions. Figure 19 illustrates a Gaussian membership function of an IVFS.

The amplitude of the considered interval, $\delta_A(x)$ is given by $\delta_A(x) = \mu_U(x) - \mu_L(x)$.

IVFSs can be generalised by allowing their intervals to be fuzzy. Therefore, each interval becomes an ordinary fuzzy set and these new fuzzy sets are referred as Type-2 Fuzzy Sets.

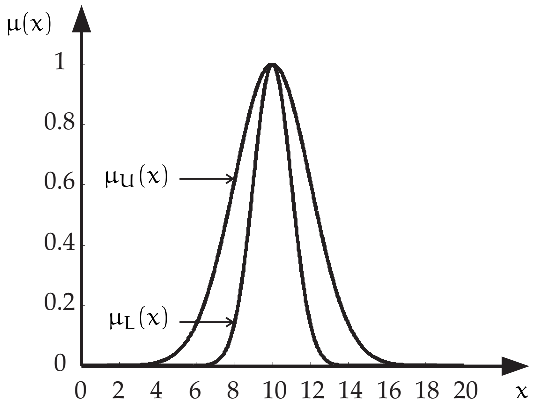


Figure 19: Gaussian membership function of an IVFS.

Atanassov suggested another generalisation of classical fuzzy sets, called an intuitionistic fuzzy sets.

Definition Let X be an ordinary finite non-empty set. An intuitionistic fuzzy set A in X is given by a set of ordered triples

$$A = \{(x, \mu_{\tilde{A}}(x), \nu_{\tilde{A}}(x)) | x \in X\}, \tag{3.35}$$

where $\mu_{\tilde{A}}, \nu_{\tilde{A}} : X \to [0, 1]$ are functions that satisfy the condition

$$0 \leqslant \mu_{\tilde{A}}(x) + \nu_{\tilde{A}}(x)) \leqslant 1 \forall x \in X. \tag{3.36}$$

For each x, the numbers $\mu_{\tilde{A}}(x)$ and $\nu_{\tilde{A}}(x)$ denote, respectively, the degree of membership and the degree of non-membership of the element x in set A.

For each element $x \in X$ we can compute, the so called, intuitionistic fuzzy index of x in A defined as follows

$$\pi_{\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x) - \nu_{\tilde{A}}(x). \tag{3.37}$$

Each fuzzy set $\tilde{A}$ can be seen as an intuitionistic fuzzy set where the non-membership function $\nu_{\tilde{A}}$ is given by $1 - \mu_{\tilde{A}}$. Since $\mu_{\tilde{A}}(x) + \nu_{\tilde{A}}(x) = \mu_{\tilde{A}}(x) + 1 - \mu_{\tilde{A}}(x) = 1$, in this sense fuzzy sets will be considered as a particular case of A-IFSs. Trivially, if the set $\tilde{A}$ is fuzzy, then $\pi_{\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x) - \nu_{\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x) - 1 + \mu_{\tilde{A}}(x) = 0$.

3.4 OBJECT TRACKING USING FUZZY LOGIC

Since the arise of fuzzy logic theory, it has been successfully applied in a large range of areas such as process and equipment control systems, automotive navigation systems, information retrieval systems, image processing, among others. Consequently, fuzzy logic is also present in numerous tracking systems, improving the performance by incorporate uncertainty in several crucial tasks of the tracking process. As presented beforehand, a tracking system can be seen as a multi-stage process that comprise figure-ground segmentation and temporal correspondences. Hence, fuzzy logic can appear in this two different stages working as a component of the global process. During the recent decades, several related works have been presented in the literature.

3.4.1 *Fuzzy logic in object detection*

In the work presented by Tao et al. [121], human knowledge is used to construct a set of linguistic if-then rules for edge detection. A set of fuzzy if-then rules is designed based on the relationship between each pixel and its eight closest neighbour pixels. Comparison studies with the gradient, Laplacian, and Laplacian of Gaussian edge detectors having fixed parameters (which means no prior information about the images is known) are provided. The empirical results show that the edge detector based on fuzzy if-then rules is generally more applicable to a wider class of images ranging from clear to very vague images.

An adaptation of the Newton's Law of Universal Gravitation using t-norms to edge detection is presented by Lopez-Molina et al. [74]. The intensity of each pixel is considered the body mass and the product of both masses is seen as a t-norm. Four definitions of t-norms are used and the results are also compared with the Prewitt, Sobel and Canny edge detectors.

Couto et al. [25] propose a segmentation approach using an extension of fuzzy set theory, so called A-IFSs. This approach uses the A-IFSs theory for representing the uncertainty of the expert in determining if a pixel belongs to the background or to the object. The optimal threshold value is associated with the intuitionistic fuzzy set of lowest entropy. Using the same concept, an extension to multilevel image segmentation is also presented.

Fuzzy C-means (FCM) clustering is an unsupervised technique that has been successfully applied in image segmentation [13, 14, 52]. The problem of fuzzy clustering is that of partitioning the set of N sample points, $x_1, x_2, ..., x_N$, into c classes. The algorithm is an iterative optimisation that minimises one cost function that is dependent on the distance of the pixels to the cluster centres defined as follows

$$J = \sum_{j=1}^{N} \sum_{i=1}^{c} u_{ij}^{m} \parallel x_j - v_i \parallel^2, \tag{3.38}$$

where u_{ij} represents the membership of pixel x_j in the *i*th cluster, v_i is the *i*th cluster centre, $\parallel . \parallel$ is a norm metric, and m is a constant.

The parameter $m \in [1, \infty[$ is a weighting exponent called the fuzzyfier that controls the fuzziness of the resulting partition. The Equation 3.38 reaches its minimum when pixels close to the centroid of their clusters are assigned with higher membership values, and lower membership values are assigned to pixels far from the centroid.

The membership values are randomly initialised according to

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1, ..., N. \tag{3.39}$$

The minimisation of the function J is accomplished by repeatedly adjusting the values of u_{ij} and v_i according to the following equations

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}}}, \tag{3.40}$$

and

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m}. \tag{3.41}$$

The algorithm stops when the changes in the cost function for two consecutive iteration steps are smaller than a predefined tolerance. In this algorithm the number of clusters c must be initialised. To implement a simple image threshold, the value of $c = 2$ is assigned. However, in some applications, several distinct regions must be identified and the value of c cannot be previously initialised.

In Sahaphong and Hiransakolwong [109], a so called FCM algorithm is presented, where the optimal cluster number is automatically obtained based in the size of the clusters. Two initial cluster centre are defined based on Otsu algorithm and the iterative algorithm is performed. The optimal clusters number will be reached when the area of at least two clusters is less than six percent of the whole image. In this situation the algorithm stops performing. Another extension of the standard FCM algorithm is proposed in Chuang et al. [23]. In this approach, spatial relationship between pixels is incorporated in the algorithm. Spatial information means that neighbouring pixels possess similar feature values and, therefore, the assumption that they belong to the same cluster is assumed. To exploit the spatial information, a spatial function is defined as

$$h_{ij} = \sum_{k \in NB(x_j)} u_{ik}, \tag{3.42}$$

where $NB(x_j)$ represents a square window centred on pixel x_j in the spatial domain. Then, spatial function is incorporated into membership function as follows

$$u'_{ij} = \frac{u_{ij}^p h_{ij}^q}{\sum_{k=1}^{c} u_{kj}^p h_{kj}^q}, \tag{3.43}$$

where p and q are parameters to control the relative importance of both functions. This new implementation reduces the number of spurious blobs and the segmented images are more homogeneous.

Huang and Wang [48] assign a membership degree to the pixel using the relationship between its grey value and mean grey value of the region to which it belongs. The image is regarded as a single fuzzy set where the membership distribution reflects the compatibility of the pixels to the region to which they belong. In this sense, the smaller the absolute difference between the pixel grey level and the mean of grey levels of its belonging region, the larger membership value the pixel has. For each grey level a fuzzy set is constructed and the optimal threshold value is the level of grey associated with the fuzzy set with lowest entropy. Using the same concept, an extension to multilevel thresholding is also presented.

A method for histogram thresholding which is not based on the minimisation of a criterion function is proposed in [122, 123]. Instead, the histogram threshold is determined according to the similarity between grey levels. Two initial fuzzy subsets, placed in the boundaries of the histogram, are defined and they are the seeds for the similarity process. A pixel grey level is assigned to the subset that denotes more similarity between them. For similarity purposes, the IF, explained already, was used. In [122] these initials subsets are defined manually. This initialisation is a critical issue since that subsets should contain enough information about the regions (object and background) to ensure the convergence of the method. Lopes et al. [72] proposes an extension of this method and in Lopes et al. [71] this new extension is supported with statistical analysis. In this new approach the initial subsets are defined automatically and they are large enough to accommodate a minimum number of pixels defined at the beginning of the process. To measure the performance of the proposed method the misclassification error parameter is calculated and results are compared with two well established methods: the Otsu's technique and the FCM clustering algorithm. After results analysis, is concluded that the proposed approach presents a higher performance for a large number of tested images.

Since texture is also a suitable property to perform object detection, Chamorro-Martínez et al. [19] introduce a model of the concept of coarseness, used in texture descriptors, by means of fuzzy sets. The assessments of human subjects have been aggregated by means of ordered weighted averaging aggregation operators and a fuzzy set which models the human perception of fineness is obtained.

3.4.2 *Fuzzy Kalman Filter*

Fuzzy logic concepts also incorporates the estimation and prediction tasks. Several extensions of the standard Kalman filter using fuzzy concepts are proposed in the literature.

Lalk [62] proposes the use of 28 fuzzy rules to adapt diagonal elements of the process noise covariance matrix or the measurement noise covariance matrix, or both, present in the traditional Kalman filter process. In Chen et al. [21] a new filtering algorithm, called Fuzzy Kalman Filtering (FKF) algorithm is presented. This algorithm is an evolution of the previous Interval Kalman Filtering (IKF) algorithm presented in Chen et al. [20]. In the IKF algorithm, all notations are the same as those of the classical Kalman filter. Nevertheless, all the matrices and vectors involved in the IKF algorithm are interval quantities, except the ordinary covariance process, measurement noise and the identity matrices. As a result, this algorithm produces interval estimation vectors, which are optimal in the same sense as the Kalman filter but for interval systems. The FKF algorithm follows the same of its predecessor but produces a scalar valued result using a three step procedure: fuzzification, fuzzy logic inference rules and defuzzification.

Most of the previous references on fuzzy Kalman filtering focus on using fuzzy rules or fuzzy relations. Matía et al. [79] propose to include fuzzy logic in the definition of the variables. The novelty of this approach derives from the fact that by using possibility distributions, instead of Gaussian distributions, a fuzzy description of the expected state and observation is sufficient to obtain a good estimation. Some characteristics of this approach are that uncertainty does not need to be symmetric and that a wide region of possible values for the expectations is allowed. This approach also contributes a method to propagate uncertainty through the process model and the observation model, based on trapezoidal possibility distributions.

3.4.3 *Fuzzy Tracking Systems*

In Lazoff [63] an active sonar system to track submarines using a Kalman filter and a posterior fuzzy rule logic association is presented. The Kalman filter provides a likelihood for associating each new detection to a track. Using the nearest neighbour approach, the detection with the highest likelihood is associated to the track and, in this case, only cinematic information was used. However, this data provided by the Kalman filter will be used as one input in a fuzzy logic association framework. Other non cinematic inputs include the neural net classifier score, the aspect angle, the SNR, and the distance to the nearest detected target. Using additional information, not used in the Kalman filter, a better method for associating a new detection with a track was achieved.

A fuzzy approach to assign one or several blobs (detected pixels grouped in a zone) to a track is described in Molina et al. and García et al., [84, 34]. This system is used for automatic surveillance in airport areas. The core of the system is the association function, in which the developed fuzzy system takes decision about what blobs belong to what tracks. This process is important when the segmentation process generates several non connected regions (blobs) from one moving object. Therefore, when multiple targets move closely spaced, their image regions overlap, appearing some targets occluded by other targets or obstacles, so that some blobs can be shared by different tracks. Three different fuzzy systems have been tested and compared with a hard decision system. The fuzzy rules for the first one were developed using an expert knowledge, the rules for the second one were learnt from recorded videos, and those for the third were developed as a refinement taking into account evaluation with ground truth information. For the tested scenarios, the fuzzy systems achieve better accuracy than the rigid scheme system.

3.5 DISCUSSION

Fuzzy logic can be seen as an extension of Boolean logic since it was intended to deal with the concepts of partial true or partial false rather than absolute values of true and false. Despite of some attempts to incorporate intermediate degrees between the notions of true and false, they did not gain much relevance until Zadeh has introduced the mathematical support of fuzzy logic theory. Zadeh introduced the mathematical definition of fuzzy sets where each element of the set is assigned with a membership degree between

0 and 1 allowing elements to have partial memberships rather than a unique membership 0 or 1 like in classical sets. Also, he introduced some properties and basic operations on fuzzy sets. After the firsts steps given by Zadeh other authors have suggested alternative definitions for fuzzy set operations and the interest in fuzzy logic starts emerging. Authors such as Lukasiewicz, Yager, Kacprzyk, Aldo de Luca, Kaufmann, Atanassov, among many others, have also played an important role by developing new definitions, concepts and measurements in the field of fuzzy logic.

Fuzzy logic was firstly applied in control systems but its concepts rapidly appeared in different areas such as image processing and tracking systems. Object detection started to be performed using fuzzy rules, the concepts of A-IFSs, using the FCM clustering method or by histogram thresholding using fuzzy measures on fuzzy sets. Also, several approaches to incorporate fuzzy concepts in the well known Kalman filter have been proposed in the literature. Finally, encouraging experimental results of fuzzy tracking systems that incorporate fuzzy concepts to improve the correspondence task have been presented by several authors and, therefore, they are referred in this chapter.

Part II

METHODOLOGY AND RESULTS

Those who become enamoured of practices without science
are like sailors who go aboard ship without a rudder and compass,
for they are never certain where they will land.

— Leonardo da Vinci

4 FUZZY SEGMENTATION ALGORITHM

4.1 INTRODUCTION

In order to implement a fuzzy feature tracking system both steps of feature detection and matching based in fuzzy concepts have to be developed. The first procedure is to study and develop a new and suitable fuzzy segmentation technique to perform feature detection to be incorporated in the tracking system. Among several fuzzy segmentation detection techniques, the one presented by Tobias and Seara [122] sounded very attractive since the histogram threshold is determined according to the similarity between grey levels instead the minimisation of a criterion function. Despite its attractiveness, the method didn't perform in an automatic mode and this automatic mode is needed to ensure that it could be incorporated in the global tracking system.

In this chapter an automatic histogram threshold algorithm based in fuzzy concepts will be presented. The threshold value is computed using a concept of similarity between pixels grey levels. This algorithm is an improvement of an existing method based on a fuzziness measure to find the threshold value in a grey image histogram previously presented in Tobias et al. and Tobias and Seara [123, 122]. The method incorporates fuzzy concepts that are more suitable to deal with imprecise object edges and ambiguity and avoids the issues involved in finding the minimum of a cost function. This method was developed to provide good results even in multimodal histograms where there is no clear separation between object and background. However it has some limitations concerning the initialisation of the method, and, to achieve an automatic process, these limitations must be overcome.

The remainder of this chapter is organised as follows. In Section 4.2 some preliminary considerations about this methodology are presented. An explanation about the used membership functions and the similarity measure is presented Section 4.3. After this introductory stage, the original method is discussed in detail in Section 4.4. To achieve an automatic version of the original method, a new algorithm is introduced in Section 4.5. In Section 4.6 the results of the proposed algorithm are compared with well established segmentation methods. Finally, this chapter ends with a discussion concerning this new approach, Section 4.7.

4.2 INITIAL CONSIDERATIONS

In order to implement the thresholding algorithm on a basis of the concept of similarity between grey levels, Tobias and Seara made the assumptions that there exists a significant contrast between the objects and background and that the grey level is the universe of discourse, a one-dimensional set, denoted by X. The first assumption means that is expected an input image with an object considerably brighter than the dark background, or vice versa. The second assumption tell us that membership functions used in the method assign membership values to the image grey levels.

At the beginning of the method, two initial fuzzy sub sets are manually defined called the seed sub sets. These fuzzy sub sets are constructed with pixels whose grey levels are present at histogram intervals located at the beginning and the end regions of the histogram. These dark and brighter sub sets will, with certainty, belong to the final sets object and background. These initial fuzzy sub sets must contain enough information to represent correctly the entities object and background.

4.3 MEMBERSHIP FUNCTIONS AND SIMILARITY MEASURE

Two distinct membership functions are used in this method for modelling the membership degrees. The S-function is suitable to represent the set of bright pixels and it was defined in Equation 3.23. Higher the grey level of a pixel (closer to white), higher membership value and vice versa. In opposition, the Z-function is used to represent the dark pixels and is defined by an expression obtained from S-function, defined previously in Equation 3.24.

The controlling parameters of both functions are variable according the properties of the fuzzy set. These parameters are calculated as follows:

$$b = \frac{\sum_{i=p}^{q} x_i h(x_i)}{\sum_{i=p}^{q} h(x_i)}, \tag{4.1}$$

$$c = b + \max\{b - (x_i)_{\max}, b + (x_i)_{\min}\}, p \leqslant i \leqslant q, \tag{4.2}$$

$$a = 2b - c, \tag{4.3}$$

where $h(x_i)$ denotes the image histogram and x_p and x_q are the limits of the subset being considered. The quantities $(x_i)_{\max}$ and $(x_i)_{\min}$, in Equation 4.2, represent the maximum and minimum grey levels in the current set for which $h((x_i)_{\max}) \neq 0$ and $h((x_i)_{\min}) \neq 0$. Note, that the crossover point b, Equation 4.1, is the mean grey level value of the interval $[x_p, x_q]$. Next, by using Equations 4.2 and 4.3, c and a are obtained.

Both membership functions could be seen, simultaneously, in Figure 20, Tobias and Seara [122]. The S-function in the right side of the histogram and the Z-function in the left side.

A reasonable approach to estimate the average ambiguity in fuzzy sets is measuring its fuzziness [90]. The fuzziness of a crisp set should be zero, as

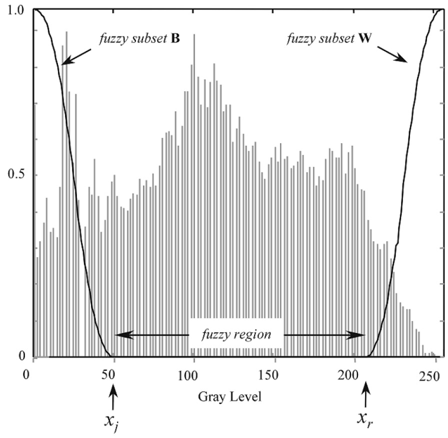


Figure 20: Histogram and the functions for the seed subsets.

there is no ambiguity about whether an element belongs to the set or not. If $\mu_A(x) = 0.5, \forall x$, the set is maximally ambiguous and its fuzziness should be maximum. Degrees of membership near 0 or 1 indicate lower fuzziness, as the ambiguity decreases. Kaufmann in [58] introduced an IF comparing a fuzzy set with its nearest crisp set. The mathematical definition of this index was previously described in Section 3.3.6.

4.4 ORIGINAL METHOD

The purpose is to split the image histogram into two crisp subsets, object subset O and background subset F, using the measure of fuzziness previously defined. A detailed description of this method can be found in Figure 21.

The initial fuzzy subsets, denoted by B and W, are associated with initial histogram intervals, $[x_{min}, x_j]$ and $[x_r, x_{max}]$, respectively, located at the beginning and the end regions of the histogram. The grey levels in each of these initial intervals have the intuitive property of belonging with certainty to the final subsets object or background. For dark objects $B \subset O$ and $W \subset F$, for light objects $B \subset F$ and $W \subset O$. These initial fuzzy subsets, W and B, are modelled by the S and Z membership functions, respectively. The parameters of the S and Z functions are variable to adjust their shape as a function of the set of elements as described above.

These subsets are a seed for starting the similarity measure process. A fuzzy region placed between these initial intervals is defined as depicted in Figure 20. Then, to obtain the segmented version of the grey level image, a classification for each grey level of the fuzzy region as being object or background is performed. The classification procedure is done by adding to each of the seed subsets a grey level x_i picked from the fuzzy region. Then, by measuring the index of fuzziness of the subsets $B \cup \{x_i\}$ and $W \cup \{x_i\}$, the grey level is assigned to the subset with lower IF (maximum similarity). Applying this procedure for all grey levels of the fuzzy region, we can classify them into object or background subsets. Since the method is based on measures of index of fuzziness, these measures need to be normalised by first computing

```
% ---------------------------------------------------------
% Histogram Threshold using Fuzzy Sets
% ---------------------------------------------------------
Initialization ();
Read Image (Image);
Compute Image Histogram (Image);
Normalize (Image Histogram);
Define Initial Intervals ($[x_{min}, x_j]$, $[x_r, x_{max}]$);
% White Region
Compute $\mu_s$ Parameters (a,b,c);
Compute Fuzzy Set ($W$);
Compute IF ($\psi(W)$);
% Black Region
Compute $\mu_z$ Parameters (a,b,c);
Compute Fuzzy Set ($B$);
Compute IF ($\psi(B)$);
Compute Normalization Factor ($\alpha = \frac{\psi(W)}{\psi(B)}$);
% Fuzzy Region
for Gray Level $x_i$ = $x_j$+1 : $x_r$-1,
    % White Region
    Define Interval ($W \cup \{x_i\}$);
    Compute $\mu_s$ Parameters (a,b,c);
    Compute Fuzzy Set ($W \cup \{x_i\}$);
    Compute IF ($\psi(W \cup \{x_i\})$);
    % Black Region
    Define Interval ($B \cup \{x_i\}$);
    Compute $\mu_s z$ Parameters (a,b,c);
    Compute Fuzzy Set ($B \cup \{x_i\}$);
    Compute IF ($\psi(B \cup \{x_i\})$);
    % Dark Objects
    if (Object == Dark && $\psi(W \cup \{x_i\})$ < $\alpha\psi(B \cup \{x_i\})$),
        $x_i$ is included in set $F$;
    else
        $x_i$ is included in set $O$;
    end
    % White Objects
    if (Object == White && $\psi(W \cup \{x_i\})$ < $\alpha\psi(B \cup \{x_i\})$),
        $x_i$ is included in set $O$;
    else
        $x_i$ is included in set $F$;
    end
end
```

Figure 21: Detailed steps of the original algorithm.

the index of fuzziness of the seed subsets and calculating a normalisation factor α according to

$$\alpha = \frac{\psi(W)}{\psi(B)}, \tag{4.4}$$

where $\psi(W)$ and $\psi(B)$ are the IFs of the subsets W and B, respectively. This normalisation operation ensures that both initial subsets have identical IF at the beginning of the process. It is a necessary condition since the method is based in the calculation of similarity between grey levels. Figure 22 illustrates how the normalisation works.

4.5 PROPOSED NEW METHOD

The concept presented above sounds attractive but has some limitations concerning the initialisation of the seed subsets. In the original method these subsets, that should contain enough information about the regions, are defined manually. Since the intervals are defined manually, there exists an

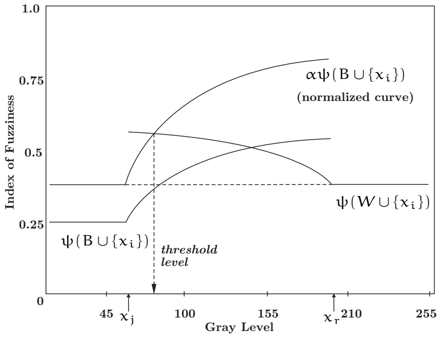


Figure 22: Normalisation step and determination of the threshold value. Adapted from Tobias and Seara [122].

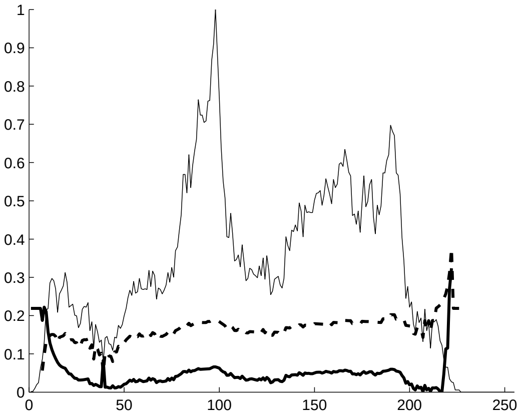


Figure 23: Divergence of the method. Solid thin line: image histogram; Bold lines: evolution of the IF's

initialisation process that is human dependent and, therefore, considering the same input image, different results can be achieved according the definition of these initial fuzzy subsets. If these initial fuzzy subsets are too large, the fuzzy region between them becomes smaller and, consequently, the grey levels considered as candidates for the threshold value are reduced. However, if the initial fuzzy subsets are too little, they do not contain enough information about the regions and the method can diverge, Figure 23. In this example, the image of the peppers is used (Appendix A, Figure 76t) and the initial fuzzy intervals $[1, 6]$ and $[223, 226]$ were considered.

When these initial fuzzy subsets are representative of both regions, the method performs as presented beforehand. This situation is depicted in Figure 24. In this case, the initial fuzzy intervals are $[1, 80]$ and $[177, 226]$, and the threshold value of 119 is obtained.

The proposed method aims to overcome some limitations of the existing method. In fact, the initial subsets are defined automatically and they are large enough to accommodate a minimum number of pixels defined at the beginning of the process. Considering the initial fuzzy subsets defined as $[x_{min}, x_j]$ and $[x_r, x_{max}]$, the values for x_{min} and x_{max} are, respectively,

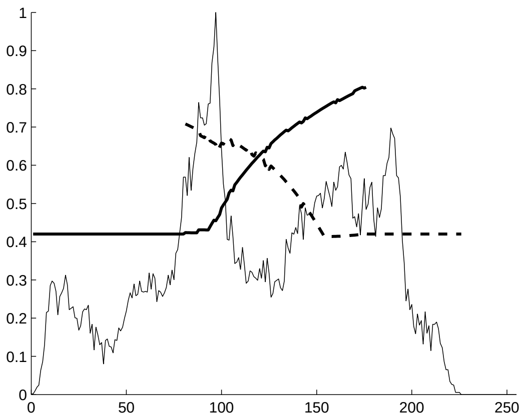


Figure 24: Method's convergence. Solid thin line: image histogram; Bold lines: evolution of the IF's

the lowest and highest grey levels of the image and, x_j and x_r are calculated to ensure that these intervals accommodate a required minimum number of pixels. This minimum number depends on the image histogram shape and it is a function of the number of pixels in the grey level intervals $[0, 127]$ and $[128, 255]$. It is calculated as follows

$$\text{MinPix}_{Bseed(Wseed)} = P_1 \sum_{i=0(128)}^{127(255)} h(x_i), \tag{4.5}$$

where $P_1 \in [0, 1]$ and $h(x_i)$ denote the number of occurrences at grey level x_i. Equation 4.5 can be seen as a special case of a cumulative histogram.

However, in images with low contrast the method performs poorly due to the fact that one of the initial regions contain a low number of pixels. So, previous histogram equalisation is carried out in images with low contrast aiming to provide an image with significant contrast in which the pixel's occurrences are distributed along the histogram. If the number of pixels belonging to the grey level intervals $[0, 127]$ or $[128, 255]$ is smaller than a value P_{MIN} defined by $P_{MIN} = P_2MN$, where $P_2 \in [0, 1]$ and M, N are the dimensions of the image, the image histogram is equalised.

Equalisation is carried out using the concept of cumulative distribution function [36]. The probability of occurrence of grey level x_i in an image is approximated by

$$p(x_i) = \frac{h(x_i)}{MN} \tag{4.6}$$

For discrete values the cumulative distribution function is given by

$$T(x_i) = \sum_{k=0}^{i} p(x_k) = \sum_{k=0}^{i} \frac{h(x_k)}{MN} \tag{4.7}$$

Thus, a processed image is obtained by mapping each pixel with level x_i in the input image into a corresponding pixel with level $s_i = T(x_i)$ in the output image using Equation 4.7.

Table 1: Minimum values of $P_1(\%)$.

IMAGE NAME	$P_1(\%)$
baboon	25
boats	40
cameraman	55
field	30
horses	50
lena	35
mouse2	25
peppers	40
potatoes	65
savanna	50
sea star	25
shadow	25
statues	35
stones	55
m	39.64
σ	13.37

4.5.1 *Calculation of parameters* P_1 *and* P_2

In Lopes et al. [72] the parameters P_1 and P_2 are determined using ad hoc heuristics after a test period with a set of images. However, in Lopes et al. [71], a statistical approach is used to obtain parameters P_1 and P_2. Parameters P_1 and P_2 are concerned with the number of pixels of the initial intervals and histogram equalisation, respectively. As the parameters are not mutually related, the statistical study is made independently.

In this study, 30 test images are used. These test images, illustrated in Appendix A, Figure 76, were obtained from Berkeley Segmentation Dataset[1] [78] and from Centro de Investigação e de Tecnologias Agro-Ambientais e Biológicas (CITAB).

To determine parameter P_1, images in the data base presenting a significant contrast are used. Such images exhibit a significant distribution of pixels with grey levels over the interval $[0, 255]$ and it is not necessary an histogram equalisation. For each image, the parameter P_1 is chosen to ensure that both the IF's of the subsets W and B provide an increasing monotonic behaviour. If P_1 is too high, the fuzzy region between the initial intervals is too small and the values of grey levels for threshold are limited. On the other hand, if P_1 is too low, the initial subsets are not representative and the method does not converge. With these minimum values of P_1 that ensure the convergence, Table 1 is constructed and the mean (m) and the standard deviation (σ) are calculated. After analysis of the results, the mean value of $P_1 = 39.64\%$ is adopted.

[1] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Table 2: Minimum values of P_2(%).

IMAGE NAME	P_2(%)
airplane	18.10
bath	36.29
bird	44.44
birds	10.92
blocks	17.02
blood	35.36
boat	20.42
gearwheel	45.99
moon	0.51
mouse	0.13
mush	20.62
newspaper	3.62
plane	29.32
rice	18.45
ski	11.43
zimba	18.51
m	20.70
σ	14.30

To determine the value of P_2, images with low contrast and parameter P_1, calculated earlier, are used. These images present a small contrast with most pixels concentrated in half side of the histogram. For these images, the minimum number of pixels in the grey level intervals $[0, 127]$ or $[128, 255]$ that ensures the convergence of the method is obtained by trial and error and the parameter P_2 is calculated. With these minimum values, Table 2 is constructed and the mean and standard deviation are also calculated. In this work, the value of $P_2 = 20.70\%$ is used.

A detailed description of this new method is presented in Figure 25.

4.6 EXPERIMENTAL RESULTS

In order to illustrate the performance of the proposed methodology, 14 images are randomly selected from our original 30 images database. A manually generated ground-truth image has been defined for each image and used as a gold standard.

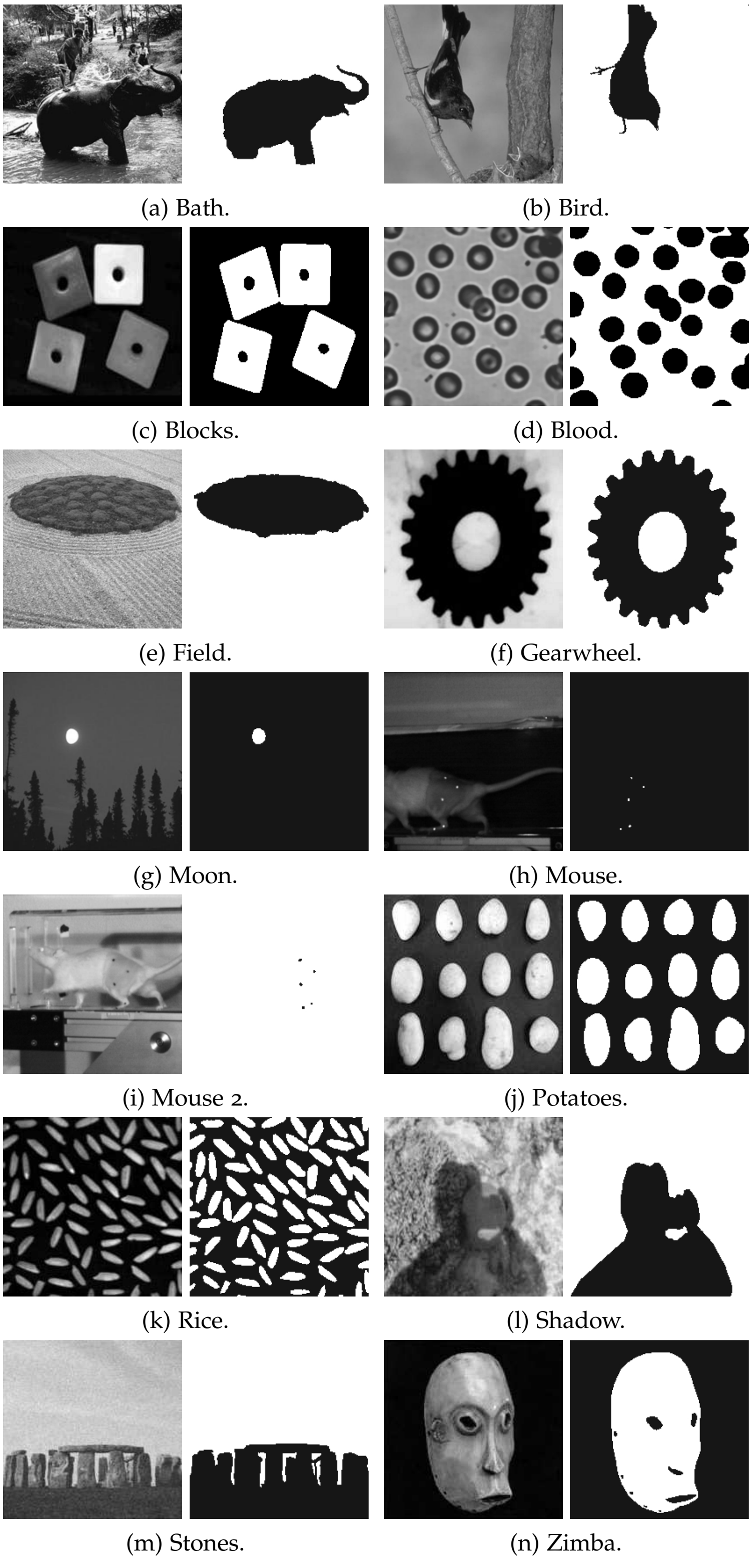
The ground-truth images were generated manually and represent the useful information of each grey level image that is relevant for further processing. In the image named bath (Figure 26a), the elephant was considered the relevant object, in the bird image (Figure 26b), the entire bird needs to be separated from the background, in the blocks image, the four blocks must be extracted, and similar procedure is applied to the remaining images. Note that the method works with both dark or bright objects since the method is a grey level histogram thresholding approach dividing a grey level image

```
% ----------------------------------------------------------
% Automatic Histogram Threshold using Fuzzy Sets
% ----------------------------------------------------------
Initialization ();
Read Image (Image);
Compute Image Histogram (Image);
Normalize (Image Histogram);
if (Image Histogram is not Equalized),
    Equalize (Image Histogram);
end
Find Initial Intervals ([$x_{min}$,$x_j$], [$x_r$,$x_{max}$]);
% White Region
Compute $\mu_s$ Parameters (a,b,c);
Compute Fuzzy Set ($W$);
Compute IF ($\psi(W)$);
% Black Region
Compute $\mu_z$ Parameters (a,b,c);
Compute Fuzzy Set ($B$);
Compute IF ($\psi(B)$);
Compute Normalization Factor ($\alpha = \frac{\psi(W)}{\psi(B)}$);
% Fuzzy Region
for Gray Level $x_i$ = $x_j$+1 : $x_r$-1,
    % White Region
    Define Interval ($W \cup \{x_i\}$);
    Compute $\mu_s$ Parameters (a,b,c);
    Compute Fuzzy Set ($W \cup \{x_i\}$);
    Compute IF ($\psi(W \cup \{x_i\})$);
    % Black Region
    Define Interval ($B \cup \{x_i\}$);
    Compute $\mu_s z$ Parameters (a,b,c);
    Compute Fuzzy Set ($B \cup \{x_i\}$);
    Compute IF ($\psi(B \cup \{x_i\})$);
    % Dark Objects
    if (Object == Dark && $\psi(W \cup \{x_i\})$ < $\alpha\psi(B \cup \{x_i\})$),
        $x_i$ is included in set $F$;
    else
        $x_i$ is included in set $O$;
    end
    % White Objects
    if (Object == White && $\psi(W \cup \{x_i\})$ < $\alpha\psi(B \cup \{x_i\})$),
        $x_i$ is included in set $O$;
    else
        $x_i$ is included in set $F$;
    end
end
```

Figure 25: Detailed steps of the proposed algorithm.

into two distinct regions, according the similarity between grey levels and, therefore, does not impose any limitation concerning on which region is the object or background. Another point is the fact that the regions that were selected to represent the ground-truth image are not homogeneous, i. e., in the selected regions could appear dark and bright pixels and, moreover, some pixels denoting similar grey levels are not considered as belonging to those regions. As a matter of fact there is no thresholding algorithm that performs correctly in such conditions, but it is reasonable to compare several thresholding algorithms with the desired and manually generated output image. Original images and their gold standard are illustrated in Figure 26.

Results are compared with two well established methods: the Otsu's technique (**OTSU**) [88] (explained in detail in Section 2.4.2) and Fuzzy C-means clustering algorithm (**FCM**) [52] (explained in detail in Section 3.4). In this way, a comparison between fuzzy and non-fuzzy threshold algorithms is carried out. Applying these different methods to the images, the correspond-



(a) Bath.

(b) Bird.

(c) Blocks.

(d) Blood.

(e) Field.

(f) Gearwheel.

(g) Moon.

(h) Mouse.

(i) Mouse 2.

(j) Potatoes.

(k) Rice.

(l) Shadow.

(m) Stones.

(n) Zimba.

Figure 26: Test images and the corresponding ground-truth images.

ing threshold values are calculated and Table 3 is constructed. The methods indicated by **IM1** and **IM2** represent the improved method without and with the possibility of histogram equalisation, respectively.

Table 3: Threshold values of individual methods.

IMAGE NAME	OTSU	FCM	IM1	IM2
bath	107	127	67	67
bird	96	113	93	93
blocks	82	127	30	44
blood	112	129	64	64
field	131	137	115	115
gearwheel	103	121	3	3
moon	55	136	254	82
mouse	47	91	20	64
mouse2	105	127	87	87
potatoes	129	126	129	129
rice	74	119	8	35
shadow	124	131	109	109
stones	137	132	142	142
zimba	71	120	10	59

With these values, grey level images are thresholded and binary images are obtained. The result binary images of the three techniques are presented in Figure 27.

Performance is obtained by comparing the gold standard image with the corresponding image provided by the three different methods. To measure such performance, a parameter η, based on the misclassification error, has been used [112]. Thus,

$$\eta_{(\%)} = \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|} \times 100 \tag{4.8}$$

where B_O and F_O are, respectively, the background and foreground of the original (ground-truth) image, B_T and F_T are the background and foreground pixels in the resulting image, respectively, and $|.|$ is the cardinality of the set. This parameter varies from 0% for a totally wrong output image to 100% for a perfectly binary image. The performance measure for every algorithm is listed in Table 4. Mean and standard deviation are also presented. After comparing results, the improved method with histogram equalisation provides, in general, satisfactory results with particular attention in images with imprecise edges.

4.7 DISCUSSION

Segmentation process is a critical task to achieve a robust tracking system since the results of the segmentation algorithm will determine the tracking performance. Therefore, to accomplish a robust segmentation method, an automatic histogram threshold approach based on index of fuzziness measure

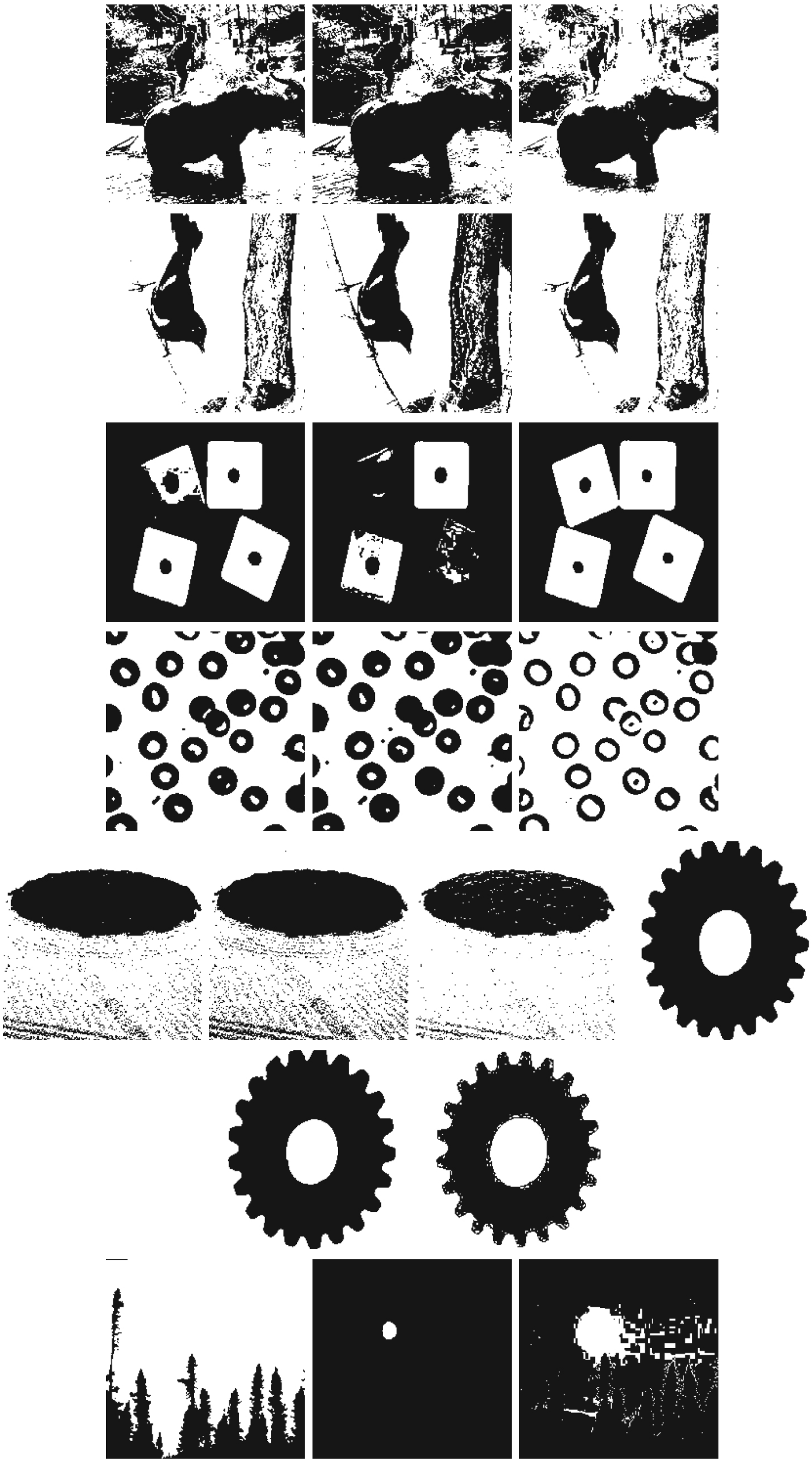


Figure 27: Results of three algorithms (first frames). For each image, from left to right: Otsu's technique, Fuzzy C-means algorithm and final improved method.

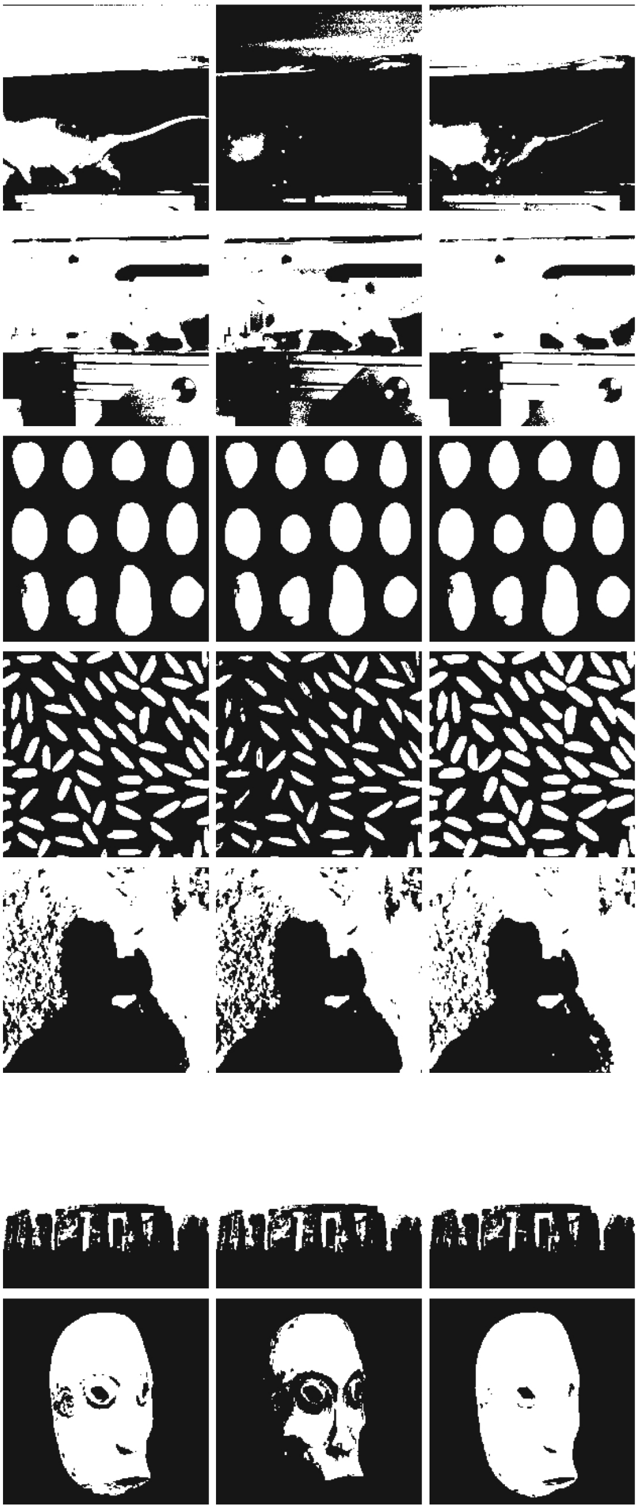


Figure 27: Results of three algorithms (last frames). For each image, from left to right: Otsu's technique, Fuzzy C-means algorithm and final improved method.

Table 4: Performance of individual methods (%).

IMAGE	OTSU	FCM	IM1	IM2
bath	62.65	55.92	76.32	76.32
bird	87.88	76.98	89.40	89.40
blocks	94.38	80.41	98.87	99.34
blood	95.61	95.73	85.09	85.09
field	93.36	90.71	96.28	96.28
gearwheel	97.85	97.07	95.59	95.59
moon	26.56	99.97	99.53	91.40
mouse	49.00	85.87	41.68	57.68
mouse2	73.56	59.09	79.63	79.63
potatoes	96.98	97.06	96.98	96.98
rice	93.51	85.84	82.06	95.91
shadow	90.46	88.30	93.26	93.26
stones	96.59	95.95	97.05	97.05
zimba	97.60	84.67	96.55	98.86
m	82.57	85.25	87.73	89.48
σ	21.91	13.58	15.28	11.58

is introduced. This work overcomes some limitations of an existing method concerning the definition of the initial seed intervals.

In the original method the definition of the initial intervals is made manually by the user, however, it is not a desirable procedure since the goal is to develop a method capable to correctly detect objects, in many images as possible, without the human supervision. Nevertheless, method convergence depends on the correct initialisation of these initial intervals and this point was the aim of this work. Using experimental data and statistical analysis, some global parameters were defined to ensure a correct definition of the initial seed intervals. After the definition of the initial seeds, a similarity process is started to find the threshold point. This property of similarity is obtained calculating an index of fuzziness of a fuzzy set.

To measure the performance of the proposed method the misclassification error parameter is calculated. For performance evaluation purposes, results are compared with two well established methods: the Otsu's technique and the Fuzzy C-means clustering algorithm.

After analyse these comparative results it can be concluded that the proposed approach presents a higher performance for a large number of tested images. However, no segmentation method performs correctly for all images tested, suggesting that the segmentation process must be carefully chosen according to the final application to ensure good results.

All truths are easy to understand once they are discovered;
the point is to discover them.

— Galileo Galilei

5 FUZZY FEATURE TRACKING METHODOLOGY

5.1 INTRODUCTION

In this chapter a new tracking approach based in fuzzy concepts will be introduced. The aim of this methodology is to solve the problem of feature tracking using the concepts of fuzzy sets theory. Several fuzzy sets are constructed according both kinematic (movement model) and non kinematic properties (image grey levels) that distinguish the feature. Meanwhile cinematic related fuzzy sets model the feature movement characteristics, the non cinematic fuzzy sets model the feature visible image related properties. The tracking task is performed through the fusion of these fuzzy models by means of a fuzzy inference engine. This way, object detection and matching steps are performed using inference rules on fuzzy sets. The user just needs to select the feature at the beginning of the sequence and the algorithm returns the estimated trajectory performed by the selected feature over the sequence.

The remainder of this chapter is organised as follows: in Section 5.2 some preliminary assumptions concerning this algorithm are presented. The definition and detailed explanation of the used membership functions are introduced in Section 5.3. In Section 5.4, the procedure to deal with a situation of occlusion is demonstrated. The proposed fuzzy tracking algorithm is presented in Section 5.5. Results with synthetic and non synthetic sequences are analysed in Sections 5.6 and 5.7, respectively. Further experimental results with low frame rate acquisition sequences and with alternative membership functions are also introduced in Section 5.7. An attempt to perform multiple feature tracking is demonstrated in Section 5.7.4. At the end of this chapter, in Section 5.8, a discussion related to this fuzzy tracking approach is presented.

5.2 INITIAL CONSIDERATIONS

The implementation of this methodology is based in some preliminary assumptions. These assumptions are commonly used in most tracking systems:

1. The feature has constancy of grey levels intensity;

2. The feature presents smooth motion;

3. For sake of simplicity, the motion between two consecutive frames can be described using a linear motion model;
4. The area occupied by the feature is small when compared with the total image area;
5. The size of the feature is preserved during the sequence.

In this approach feature brightness constancy is assumed. This situation can mathematically be described as

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t), \tag{5.1}$$

where δx and δy are the displacement of the local region at (x, y, t) after time δt. Nevertheless, slightly changes in illumination, camera sensor noise, among other factors that cause variations in the intensity of the feature, are tolerated.

The smoothness of the movement concerns the continuity of the feature movement. The feature movement is assumed to be continuous and, therefore, using a typical acquisition frame rate and assuming there are no occlusions or misdetections, the next position of the feature lies inside a neighbourhood of its previous position.

It is also assumed that the feature movement between two consecutive frames can be represented by a linear motion model with constant acceleration. The feature can move along both the x and y axis and, therefore, the position $\mathbf{p}(t)$ can be obtained from the previous position $\mathbf{p}(t-1)$ applying the following equation:

$$\mathbf{p}(t) = \mathbf{p}(t-1) + \mathbf{v}\Delta t + \frac{1}{2}\mathbf{a}\Delta t^2, \tag{5.2}$$

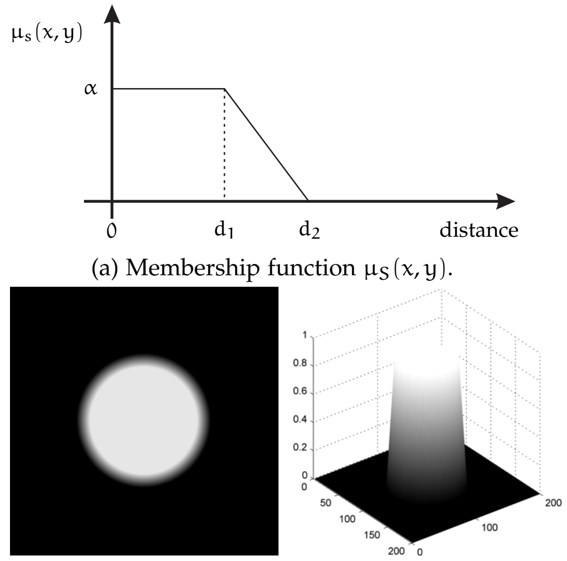
where $\mathbf{p}(t) = [x, y]'$ is the feature position at instant t, $\mathbf{p}(t-1) = [x_0, y_0]'$ is the feature position at instant $t-1$, Δt is the elapsed time from instant $t-1$ to instant t, $\mathbf{v} = [v_x, v_y]'$ and $\mathbf{a} = [a_x, a_y]'$ are, respectively, the observed velocity and acceleration in both axis during Δt.

The size of the feature is considerably small when compared with the total image area. Assuming this, the feature can be represented as a point or by a small region (represented by a $A \times B$ matrix) and, similar strategies to the ones used in point correspondence can be developed for feature matching.

Furthermore, the size of the feature is considered to be approximately constant over the entire sequence, however, the presented methodology is robust against slightly size changes.

5.3 MEMBERSHIP FUNCTIONS

The fuzzy sets constructed in this methodology are derived from the initial considerations presented beforehand. Three fuzzy sets, concerning the first three enumerated assumptions are constructed. Other fuzzy sets could be incorporated if necessary, in the methodology, addressing different properties of the features such as texture, colour or shape. In order to minimise computational resources and to increase speed and simplicity, the algorithm is



(a) Membership function $\mu_S(x, y)$.

(b) Fuzzy set S using $\mu_S(x, y)$.

(c) 3D view of fuzzy set S.

Figure 28: Membership function $\mu_S(x, y)$.

constructed based only on these three fundamental and generic assumptions. These generic assumptions are used in order to demonstrate the viability of the proposed methodology.

Assuming the smoothness of the feature movement, i. e., assuming that the feature position will not drastically change between two consecutive frames, it is plausible to suppose that the next location of the feature lies in a neighbourhood centred in its previous location.

Therefore, a fuzzy set S associated with each image pixel (x, y) by means of this proximity assumption related to the feature position in the previous frame is constructed. The membership function $\mu_S(x, y) \in [0, \alpha]$ can be graphically depicted as illustrated in Figure 28a, where the horizontal axis represents the Euclidean distance between the image pixels position and the previous location of the feature. In Figure 28b, is depicted a pictorial description of the fuzzy set S assuming a previous position $(x, y) = (100, 100)$, with $d_1 = 40$, $d_2 = 50$ and a maximum value $\alpha = 0.9$.

Three distinct zones of certainty are present in the definition of the membership function $\mu_S(x, y)$. For distances lower than d_1 the membership degree is maximum, defining a circular region centred in the feature previous position, where the new feature location is expected with equal certainty. For distances greater than d_1 the membership degree decreases in a linear way until it reaches the zero value at distance d_2. For distances lower than d_2 the membership degree is zero. This behaviour can be explained due to the fact that, for distances greater than d_1, the certainty of finding the feature is lower since the distance increases. The new position is not expected for distances greater than d_2 and the membership degree is zero for all these positions. These two controlling parameters d_1 and d_2 vary over time. Both values are directly proportional to the observed feature displacement $f_d(t)$. This displacement is based in an Euclidean distance defined as

$$f_d(t) = \sqrt{(\mathbf{p}(t) - \mathbf{p}(t-1))^2}, \tag{5.3}$$

where $\mathbf{p}(t) = (x(t), y(t))$ and $\mathbf{p}(t-1) = (x(t-1), y(t-1))$ are, respectively, the current and previous positions of the feature.

To avoid abrupt changes in this parameter, a weighted sum is performed, using the previous displacement information and an actualisation factor. This reasoning can be mathematically represented as

$$\Delta d = A_f f_d(t) + (1 - A_f) f_d(t-1), \tag{5.4}$$

where A_f is a constant within the interval $[0, 1]$, $f_d(t-1)$ is the previous displacement and $f_d(t)$ is the current observed displacement. Then, parameters d_1 and d_2 of $\mu_S(x, y)$ are defined as

$$d_1 = M_1 + \frac{\Delta d}{2}, \tag{5.5}$$

$$d_2 = d_1 + M_2 + \frac{\Delta d}{2}, \tag{5.6}$$

where M_1 and M_2 are two positive constants. These constants act as minimum values for these two parameters in order to deal with features presenting zero velocity. Parameter d_1 has a minimum value of M_1 and parameter d_2 has a minimum value of $d_1 + M_2$. Values M_1 and M_2 can be related with the dimensions of the feature.

The bright constancy assumed earlier ensures that the feature intensity level remains stable, or approximately stable, during the sequence. Hence, the initial grey level of the feature is considered unchangeable over time meaning that pixels denoting similar grey levels regarding the initial feature grey level are more likely to belong to the feature.

Under these conditions, a fuzzy set is constructed in order to access the certainty of a pixel belonging to the feature in such a way that higher similarity in grey levels intensity higher the membership degree to that fuzzy set. Gaussian or triangle membership functions completely satisfy this requirement and then, they can be employed to perform the operation of membership degree attribution. For simplicity reasons, the triangular shaped function is used.

Defining the membership function as $\mu_g(x, y)$, the initial feature grey level as q and a non negative value d, it can be graphically represented as depicted in Figure 29.

The membership function $\mu_g(x, y)$ assigns a membership degree to a pixel located in the image position (x, y) according the similarity between grey levels. Parameter d is used to adjust the shape of the function and the function can be more permissive or restrictive for, respectively, higher or lower values of d.

Applying this pixel-wise operation, adjacent pixels could be assigned with very different membership degrees. However, on the basis that, if a pixel

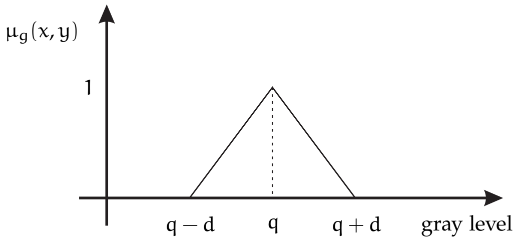


Figure 29: Illustration of the membership function $\mu_g(x, y)$.

belongs to a feature then its adjacent pixels have high probability of also belonging to that feature, spatial information is incorporated by representing the feature by means of a set of pixels rather than a single pixel. As a consequence, the effect of variations in the feature pixels due to noise is intended to be minimised.

A new membership function $\mu_G(x, y)$ is introduced and it can be mathematically described as follows:

Definition Let I be an image with dimensions $M \times N$, $I(x, y)$ is the grey level of the pixel (x, y) so that $0 \leqslant I(x, y) \leqslant L$ and $I_f(i, j)$ an intensity matrix of dimensions $A \times B$ representing the original feature grey levels, where $A = 2a + 1$, $B = 2b + 1$ and $\{a, b\} \in \mathbb{N}$.

For all (x, y) such that $a \leqslant x \leqslant M - a$ and $b \leqslant y \leqslant N - b$, the membership function $\mu_G(x, y)$ is defined as

$$\mu_G(x, y) = 1 - \frac{\sum_{i=-a}^{a} \sum_{j=-b}^{b} |I(x + i, y + j) - I_f(i + a, j + b)|}{A \times B \times L}. \qquad (5.7)$$

All pixels (x, y) of the image, such that $0 \leqslant x \leqslant a \vee M - a \leqslant x \leqslant M$ and $0 \leqslant y \leqslant b \vee N - b \leqslant y \leqslant N$ have zero membership values. This set of pixels are located at the boundaries of the image and, since a and b are small positive integers, this discontinuity does not change the global performance of the method.

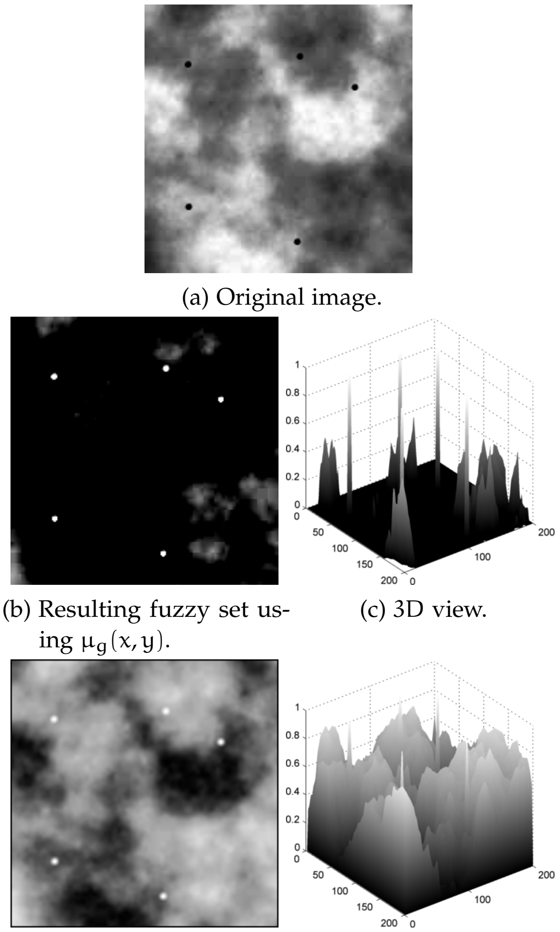
In Figure 30, comparative results using membership functions $\mu_g(x, y)$ and $\mu_G(x, y)$ applied to the same test image are illustrated.

In order to reduce processing time and increase computational speed, this membership function is applied locally in a neighbourhood centred in the previous position of the feature. The interested area could be seen as a square region whose sides have a length l defined as

$$l = 2K_G.d_2, \qquad (5.8)$$

where K_G is a positive constant and d_2 is the parameter defined previously in Equation 5.6.

Another membership function is constructed based on the assumption that the feature motion between two consecutive frames can be described using a linear motion model with constant acceleration. Stating this, a feature can increase its velocity between two consecutive frames. Several motion



(a) Original image.

(b) Resulting fuzzy set using $\mu_g(x, y)$.

(c) 3D view.

(d) Resulting fuzzy set using $\mu_G(x, y)$.

(e) 3D view.

Figure 30: Comparative results between membership functions $\mu_g(x, y)$ and $\mu_G(x, y)$.

models are discussed in the literature, however, the selected motion model is a compromise between the proximity with the real motion performed by the feature and computer processing requirements. The used motion model follows the equation described previously in Equation 5.2.

Kalman filter is a powerful tool to predict the feature position during the image sequence. Using information from previous frames, it is possible to predict the feature location in the current frame. The Kalman filter is used based on the assumption that the velocity and the direction of the feature doesn't suffer drastic changes from frame to frame, that is to say, the feature follow a linear motion model with constant acceleration.

In Kalman filter the motion model is introduced in state space representation. Using state space representation, a system can be defined by

$$\begin{aligned} \mathbf{x}(t) &= A\mathbf{x}(t-1) + B\mathbf{u}(t), \\ \mathbf{y}(t) &= C\mathbf{x}(t), \end{aligned} \tag{5.9}$$

where $\mathbf{x}(t)$ represents the state vector in the current time, A represents the motion model, B represents the state vector dependency matrix with respect the input $\mathbf{u}(t)$, $\mathbf{y}(t)$ represents the system output and C is called the output matrix.

For this particular motion model, previously described in Section 5.2, the state vector $\mathbf{x}(t)$ can be written as

$$\mathbf{x}(t) = \begin{bmatrix} x & v_x & a_x & y & v_y & a_y \end{bmatrix}', \tag{5.10}$$

where x and y are the location coordinates of the feature, v_x and v_y the velocities, a_x and a_y are the acceleration values.

Matrix A is defined as

$$A = \begin{bmatrix} 1 & t_{aq} & \frac{t_{aq}^2}{2} & 0 & 0 & 0 \\ 0 & 1 & t_{aq} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & t_{aq} & \frac{t_{aq}^2}{2} \\ 0 & 0 & 0 & 0 & 1 & t_{aq} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.11}$$

where t_{aq} is the elapsed time between two consecutive frames.

Matrix C is defined as follows

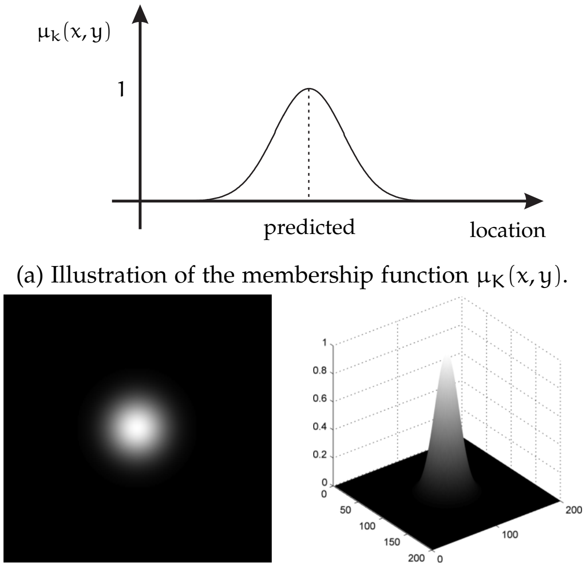
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.12}$$

Kalman filter is used to estimate the state vector of the feature, i. e., its position, velocity and acceleration. Using information provided by the state vector of the feature it is possible to predict the feature position (x, y) in the next frame. This thoughts led to the development of another membership function, called $\mu_K(x, y)$. This membership function assigns higher membership degree to pixels near the predicted location and its value decreases for locations far from this predicted point. To implement such behaviour, a Gaussian shape function is used, Figure 31a. A Gaussian function was used rather than a triangular shape function to ensure a higher decay in the membership degrees and, this way, is given more prominence for locations near the predicted one. The Gaussian function shape can be changed through the standard deviation parameter σ defined as follows

$$\sigma = \frac{3M_\sigma + \Delta d}{3}, \tag{5.13}$$

where M_σ is a minimum value to deal with stopped features and Δd is the previous observed displacement of the feature. Then, this parameter is changed according to the velocity of the feature with higher velocities giving rise to higher standard deviation values.

This membership function $\mu_K(x, y)$ is applied in a circular neighbourhood with radius equal to 3σ centred in the predicted position. For locations whose distance to the predicted position is greater than 3σ the membership degree is zero. In Figure 31b, is depicted the resulting fuzzy set K assuming a predicted position $(x, y) = (100, 100)$ and $\sigma = 15$. The corresponding 3D view is also illustrated in Figure 31c.

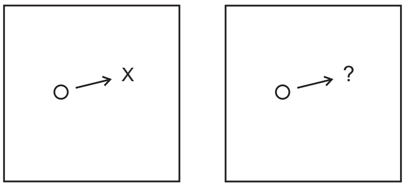


(a) Illustration of the membership function $\mu_K(x,y)$.

(b) Resulting fuzzy set K.

(c) 3D view.

Figure 31: Membership function $\mu_K(x,y)$.



(a)

(b)

Figure 32: Correspondence situations in single feature tracking.

5.4 OCCLUSION AND MISDETECTION

In single feature tracking systems, two distinct situations can occur. The tracked feature is assigned to a candidate point in the current frame or no correspondence is performed. These two situations are depicted, respectively, in Figures 32a and 32b, where $\circ$ denotes feature position at frame $t-1$, $\times$ denotes feature position at frame t and the question mark (?) indicates the absence of detected candidates. In the first situation, the current position of the feature is assumed to be the position of the candidate, however, when there are no candidates for matching, an occlusion or a misdetection situation is present.

When an occlusion or a misdetection occurs, the proposed method tends to solve this situation acting in several directions. In such situations, the proposed method uses a predicted position determined using the information given by the estimated state vector of the Kalman filter and the motion model equation described in Equation 5.2. During occlusion the position of the feature is predicted until the algorithm detects the feature once again.

On the other hand, dynamic membership functions $\mu_S(x, y)$ and $\mu_K(x, y)$ assume a different behaviour. When an occlusion or a misdetection are detected, parameters d_1 and d_2 of $\mu_S(x, y)$ and parameter σ of $\mu_K(x, y)$ increase. The purpose of this behaviour is to define a large region of search and to accommodate the uncertainty concerning the motion model.

During the occlusion period no matching is performed and the approach doesn't have reliable information about the position of the feature. Therefore, feature displacement Δd is not updated.

5.5 METHODOLOGY

The proposed methodology can be globally described through the pseudocode presented in Figure 33. This pseudocode will be explained in detail in this section.

In this methodology, two distinct stages are present: a setup stage and a closed loop stage.

```
% ---------------------------------------------------------
% Fuzzy Logic Tracking Approach for Single Feature Tracking
% ---------------------------------------------------------

Initialization Stage ();
Read Image (First Frame);
Show Image (First Frame);
Select Feature (By the User);
Save (Data of Selected Feature);

for Current Frame = Next Frame : Last Frame,
    Read Image (Current Frame);
    Compute ($\mu_G$, $\mu_K$, $\mu_S$);
    $\mu_M$ = Fuzzy Union ($\mu_K$, $\mu_S$);

    for All Pixels where $\mu_M$ > 0,
        Candidates = Find (Max($\mu_G$) > 0.8);
    end

    % Begin {Inference Engine}
    if (Number of Candidates == 1),
        Current Position = Position of the Candidate;
        Predicted = 0;
    elseif (Number of Candidates > 1),
        Nearest Candidate = Find (Candidate closest to Max($\mu_K$));
        Current Position = Position of Nearest Candidate;
        Predicted = 0;
    else
        Current Position = Position of Max($\mu_K$);
        Predicted = 1;
    end
    % End {Inference Engine}

    if (Predicted == 1),
        Increase $\mu_S$ Parameters (d1, d2);
        Increase $\mu_K$ Parameter ($\sigma$);
    end

    Update (Kalman Filter);
    Show Image (Current Frame);
    Draw (Current Position, Estimated Trajectory);

end
```

Figure 33: Detailed steps of the algorithm.

At the setup stage several steps are performed in order to define and initialise all the variables, matrixes and structures. It is assumed that the feature is described by a 3×3 matrix, α parameter of μ_S is 0.8, it has an initial displacement of 10 pixels and the displacement actualisation ratio $A_r = 0.75$. Since the feature is described by a 3×3 square matrix, parameters M_1 and M_2 of Equations 5.5 and 5.6, parameter M_G of Equation 5.8 and parameter M_σ of Equation 5.13 are considered equal to 3. Parameter K_G of Equation 5.8 is also considered 3 and controls the size of the interested region centred in the previous feature position in the input image. If this value is too high, local processing is not accomplished but if it is too low, this region couldn't accommodate the fuzzy set μ_M defined in Equation 5.14. A value of $K_G = 3$ also assumes that the feature displacement in the current frame is lower than three times the displacement observed in the previous frame. This assumption is valid since it is considered smooth motion.

Is also assumed that parameters d_1 and d_2 of $\mu_S(x, y)$ and parameter σ of $\mu_K(x, y)$ increase 10% in occlusion situations. The value of 10% is assumed to establish a smooth increasing of the region of search. In this stage is also defined the initial frame number and then, it is possible to observe the performance of the approach starting in different frames of the sequence. After these assignments, the initial frame of the sequence is displayed and the human operator must select the feature to track. Feature properties, such as intensity and initial position values are then recorded for further use.

In the closed loop stage, the algorithm performs as follows:

1. After opening the next frame of the sequence, a fuzzy set G is constructed by applying the membership function $\mu_G(x, y)$ to the image pixels;

2. Fuzzy set K is constructed through the definition of the membership function $\mu_K(x, y)$ using the Kalman filter information and motion model;

3. Fuzzy set S is constructed using the previous position of the feature and $\mu_S(x, y)$;

4. Fuzzy set μ_M is computed;

5. Using an inference engine with a set of rules the position of the feature is determined;

6. The membership function parameters and the state variables of the method are updated;

7. The estimated trajectory is illustrated and the algorithm jumps to the first step until it reaches the last frame of the sequence.

The first three steps are the direct implementation of the membership functions where three fuzzy sets are constructed and the last two steps are, respectively, an actualisation and display procedures. The fourth and fifth steps can be considered the most relevant and innovative steps in the method and, therefore, they will be explained in detail.

Before applying the rules to the fuzzy sets, a fuzzy set M is constructed from the fuzzy union between fuzzy sets K and S, using the maximum operator as follows

$$\mu_M(x,y) = \vee(\mu_K(x,y), \mu_S(x,y)) \tag{5.14}$$

After defining the fuzzy set M, an inference engine with the following set of rules is constructed. The output of the engine is a fuzzy set E that will ultimately lead us to the feature position which will be the pixel (x,y) that corresponds to the higher $\mu_E(x,y)$ value.

The proposed inference engine is constructed using the following three rules:

RULE 1 : **IF**, within the area defined by membership values of the fuzzy set M, such that $\mu_M(x,y) > 0, \forall (x,y)$, there is one and only one local maxima of $\mu_G(x,y) > \alpha$, **THEN**, fuzzy set E is the union between fuzzy sets S and G. This fuzzy set is constructed using the maximum operator in the following way:

$$\mu_E(x,y) = \vee(\mu_G(x,y), \mu_S(x,y)). \tag{5.15}$$

RULE 2 : **IF**, within the area defined by membership values of the fuzzy set M such that $\mu_M(x,y) > 0, \forall (x,y)$, there are $n \geqslant 2$ local maxima of $\mu_G(x,y)$, located at position $(x_i, y_i), \forall i = 1, \ldots, n$ that satisfy the condition $\mu_G(x_i, y_i) > \alpha$, **THEN**, fuzzy set E is the union between fuzzy sets S and $G'_i, \forall i = 1, \ldots, n$.

The fuzzy sets G'_i are constructed using fuzzy set G in the following way:

$$\mu_{G'_i}(x,y) = \psi_i \mu_G(x,y), \forall i = 1, \ldots, n, \tag{5.16}$$

with

$$\psi_i = 1 - \frac{d_i}{d_{MAX}}, \forall i = 1, \ldots, n, \tag{5.17}$$

and

$$d_i = \sqrt{(x_i - x_K)^2 + (y_i - y_K)^2}, \forall i = 1, \ldots, n, \tag{5.18}$$

$$d_{MAX} = max\{d_i\}, \forall i = 1, \ldots, n, \tag{5.19}$$

where (x_K, y_K) is the location of the maximum value of fuzzy set K.

Finally, fuzzy set E is constructed using the maximum operator

$$\mu_E(x,y) = \vee(\mu_{G'_1}(x,y), \ldots, \mu_{G'_n}(x,y), \mu_S(x,y)). \tag{5.20}$$

RULE 3 : **IF**, within the area defined by membership values of the fuzzy set M, such that $\mu_M(x,y) > 0, \forall (x,y)$, there is no local maxima of $\mu_G(x,y) > \alpha$, **THEN**, fuzzy set E is to be equal to fuzzy set K.

$$\mu_E(x,y) = \mu_K(x,y). \tag{5.21}$$

The design of this inference engine and its rules is based in human reasoning. People expect to find an object or a feature in its last known location or at locations in the neighbourhood, particularly when dealing with static objects or objects with small movement. This kind of human reasoning is modelled by fuzzy set S. When dealing with fast moving objects, people are capable to understand the feature motion pattern and consequently anticipate its next position. This thought is also valid when the object is occluded. Consequently, the fuzzy set K tends to incorporate this reasoning. According to these two behavioural attributes, the area defined by fuzzy set M, i. e., the image area where $\mu_M(x, y) > 0$, can be seen as the first area of search to locate an object or a feature. Looking for this region, if a person see an identical feature as expected, then it is plausible to consider this feature as the one that is been tracked. From RULE 1, the feature position will be the pixel, with coordinates (x, y), that denotes the maximum value of $\mu_G(x, y)$.

If multiple identical features are present in that region then, it is reasonable, based on the previous acquired motion pattern, to choose the feature near the predicted feature position (RULE 2).

In situations when the feature is not visible, its location can be only estimated by the understanding of the behaviour of the motion observed until that moment (RULE 3). In this case, the feature position will be the pixel, with coordinates (x, y) with the maximum value of $\mu_K(x, y)$.

When the feature is not detected, the search area increases due to the uncertainty of the movement described by the feature and the used motion model. Furthermore, if the output of the engine results from RULE 3 then, since all the $\mu_G(x, y)$ values in the considered image area are below than α, probably due to an occlusion, the membership functions of fuzzy sets S and K change in such a way that the region where $\mu_M(x, y) > 0$ becomes bigger, allowing the tracked feature to be searched in a wider area.

5.6 EXPERIMENTAL RESULTS USING SYNTHETIC SEQUENCES

In order to illustrate the performance of the proposed methodology, the proposed algorithm is foremost applied to four synthetic image sequences (S1, S2, S3 and S4) generated at 25 frames per second (fps) with a spatial resolution of 200 × 200 pixels. These sequential images illustrate a set of small moving points, to simulate object features moving across a scene. These small points could be completely black or have different grey levels and the scene could denote uniform or non-uniform background. With these assumptions four synthetic sequences were generated with the following properties:

- In the sequence S1, objects are similar having grey level equal to zero and the background is uniform with white colour.
- In the sequence S2 the objects remain the same but the background is non-uniform presenting grey level values ranging from 0 to 255.
- The sequence S3 is composed by moving objects denoting different grey levels in a uniform white scene.
- The sequence S4 is composed by objects denoting different grey levels that move across a non-uniform scene.

The human operator selects one point at the beginning of the process and the algorithm will track the selected point till the end of the sequence. The first image of the four sequences are illustrated in Figure 34 and the selected feature is located near the left lower corner of the image.

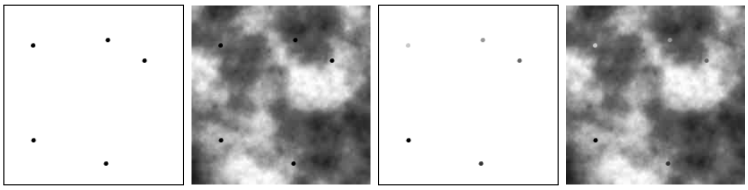
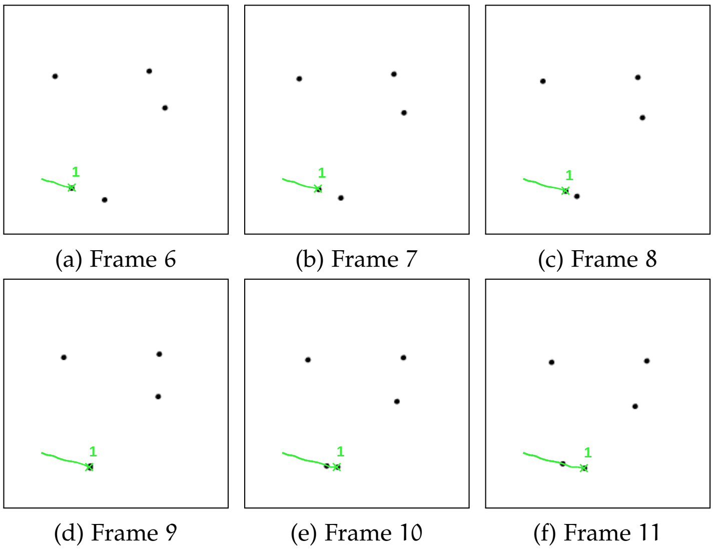


Figure 34: First frame of the four sequences. From left to right: S1, S2. S3 and S4

The first sequence used to test the performance of this approach is the sequence S1 whose characteristics were already presented previously.

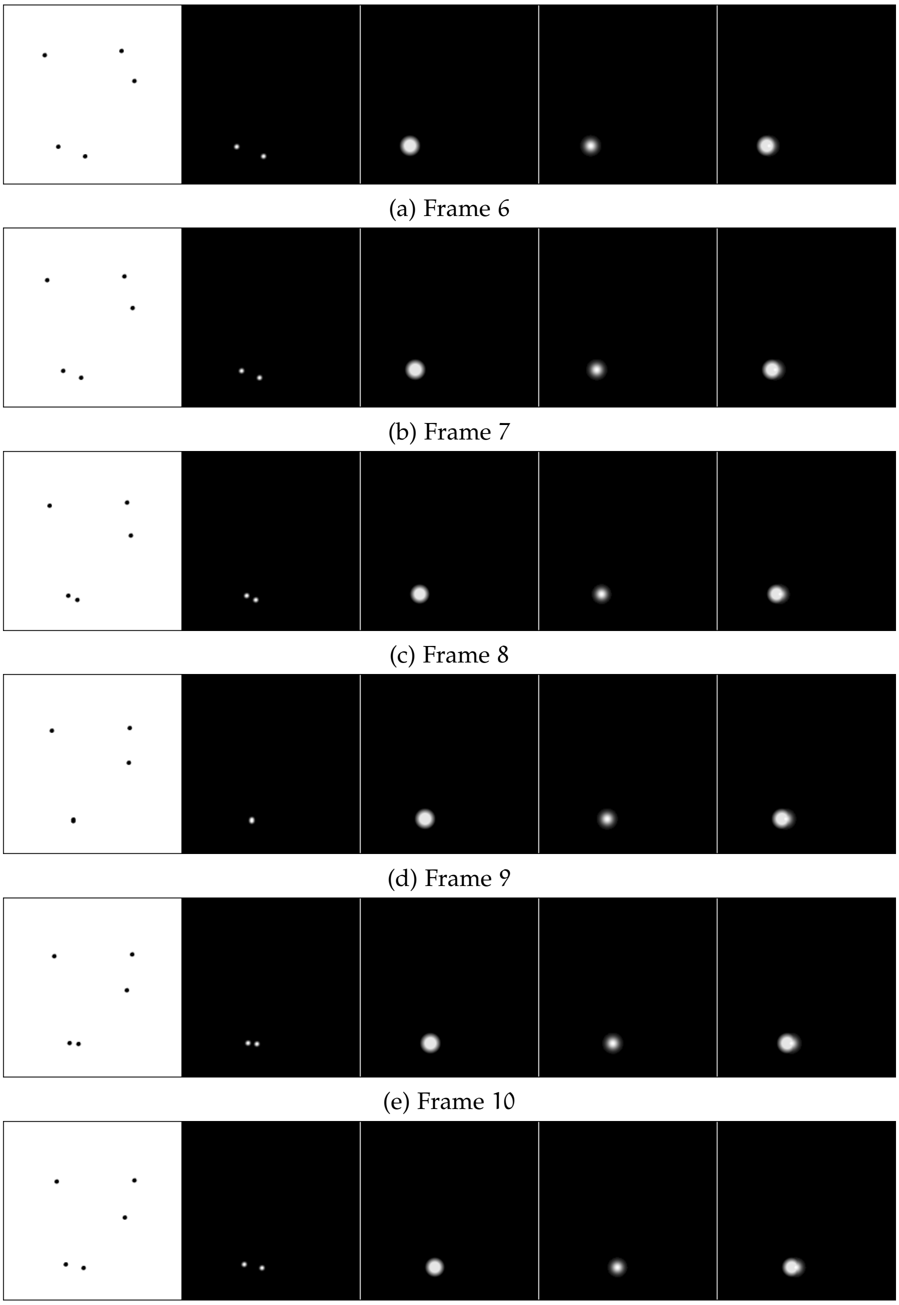
The first relevant situation occurs when two similar features cross each other and the estimated trajectory during this period is illustrated in Figure 35. This situation is observed from frame 6 to frame 11 as depicted in Figure 36. In Figure 36a, the leftmost image is the original image of the sequence and the four images of the right represents, respectively, fuzzy set μ_G, fuzzy set μ_S, fuzzy set μ_K and fuzzy set μ_M. Same image arrangement is considered for similar subsequent figures.



(a) Frame 6
(b) Frame 7
(c) Frame 8
(d) Frame 9
(e) Frame 10
(f) Frame 11

Figure 35: Estimated trajectory in crossing situation.

As explained earlier, fuzzy set G represents the proximity between grey levels of the feature and grey levels of the image pixels. In the image representation of this fuzzy set, higher proximity is represented by brighter pixels and, in opposition, darker pixels represent the absence of similarity between grey levels. As we can see, brighter regions in fuzzy set G correspond to darker regions in the original image because the chosen feature is dark. The fuzzy set S represents the smoothness of the feature motion is represented



(a) Frame 6

(b) Frame 7

(c) Frame 8

(d) Frame 9

(e) Frame 10

(f) Frame 11

Figure 36: Crossing situation in sequence S1.

by a circular region with higher values when locations become near the previous position of the feature. When the difference becomes lower than a predefined parameter d_1, the value of μ_S remains constant. The illustration of fuzzy set K is an image with a brightest pixel at the coordinates of the estimated next position of the feature and the bright is rapidly reduced for pixels far from this point.

In this crossing situation, features have different motion patterns, i. e., the tracked feature moves from left to right and other similar feature moves in the opposite way. Using the motion model of the feature and Kalman filter is possible to define a region in the next frame where the feature will be found. If the tracked feature denotes a movement from left to right, is expected to find the feature, in the consecutive frame, at the right of its previous position. When these two features are overlapped in the image, forming just one region, the tracked feature is expected to appears in the right side of the region, maintaining its motion behaviour.

In this particular situation, the fuzzy set K, which incorporates motion model information, and the inference engine play an important role to achieve a correct tracking of the feature.

The following pertinent situation is the feature occlusion that occurs from frames 35 to 39, as illustrated in Figures 37 and 38. To indicate the occlusion situation the colour of the estimated trajectory changes to red and the predicted location is indicated by a red $\circ$ rather than a green $\times$. In this case the feature is not present in the image but, using the information of the previous position and the model of the movement, a new feature position can be predicted. The motion model used to describe the dynamic behaviour of the feature assumes an important role, since it will define the estimated trajectory of the feature during occlusion situation.

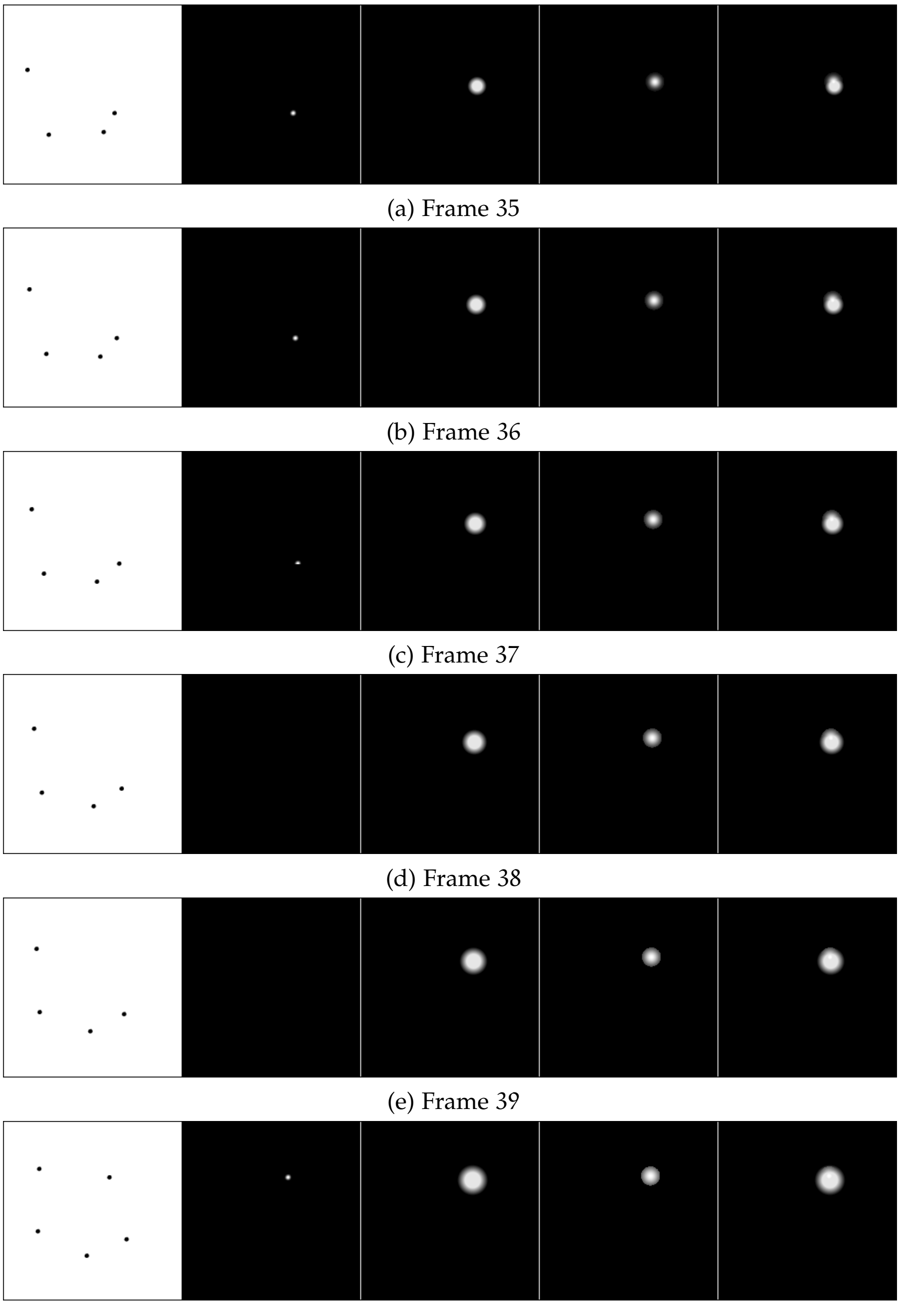
In the occlusion stage, the radius of the membership function μ_S increases resulting in a higher area of search. Furthermore, the parameter sigma in the membership function μ_K also increases. These two procedures tend to accommodate the differences between the considered motion model of the feature and the real motion described by the feature.

When the feature stops moving, the algorithm still detects its presence, Figure 40. This fact can be considered an advantage comparing this approach with tracking algorithms whose feature detection is based uniquely in motion analysis such as frame differencing, background subtraction or optical flow.

Since the detection is performed using the information about grey levels of the features given by fuzzy set G, the feature continues to be detected using this tracking approach when it stops. In this stage, the searching region defined by μ_S and the sigma parameter in membership function μ_S start reducing as observed in Figure 39.

The estimated trajectory of the feature over the sequence is illustrated in Figure 40f. The presented trajectory represents the estimated path described by the feature in a smooth way. Since the motion model of the feature is described by a linear motion with constant acceleration, in occlusion situations, it is assumed that feature denotes a linear movement, and this linear behaviour can also be seen in this picture when the feature starts to move straight forward (red line). When it reappears in the scene it is immediately detected by the approach and the location is then updated.

To understand the evolution of the outer radius d_2 of membership function μ_S, its evolution is indicated in Figure 41a. At the beginning of the process this parameter is initialised. During the sequence, this value is directly dependent on the velocity of the feature and, in occlusion situations, this value starts to increase from its last value. With these statements, the shape



(a) Frame 35

(b) Frame 36

(c) Frame 37

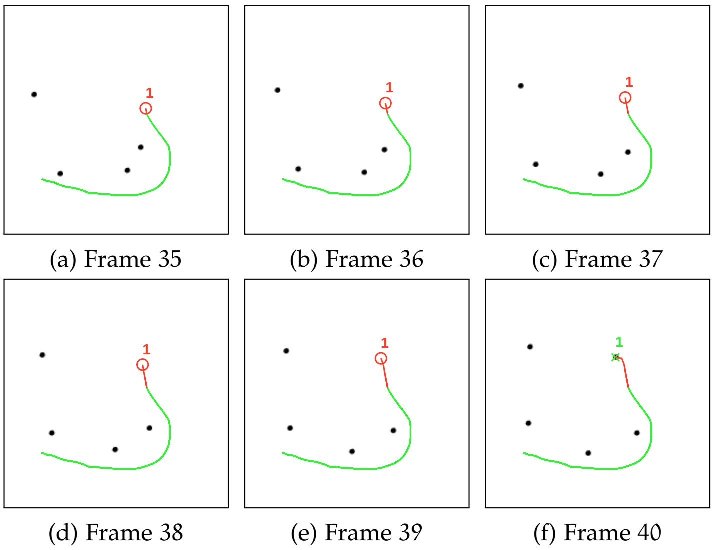
(d) Frame 38

(e) Frame 39

(f) Frame 40

Figure 37: Occlusion situation in sequence S1 (frame 35 to 40).

of Figure 41a is easily understood. This value decreases from a starting value and remains substantially constant until the occlusion period. In the occlusion situation this value starts to increase until the feature reappears in the scene. After this moment, this parameter returns to similar values before the occlusion. When the feature stops, at the final of the sequence, this parameter reaches its minimum value since the feature doesn't denote any



(a) Frame 35 (b) Frame 36 (c) Frame 37

(d) Frame 38 (e) Frame 39 (f) Frame 40

Figure 38: Estimated trajectory in occlusion situation.

movement. Similar behaviour describes the evolution of σ value concerning the membership function μ_K which is depicted in Figure 41b.

For the remaining sequences the results are very similar. The algorithm performs as expected, tracking the selected feature during the sequence. Because of this similarity, the illustration of fuzzy sets and result images are neglected and just the estimated trajectory and the evolution of parameters of membership functions μ_S and μ_K are shown. Making an analysis to the graphics depicted in Figures 42, 43 and 44, its easily noticed that non-uniform background and different features grey level don't decrease the performance of this methodology.

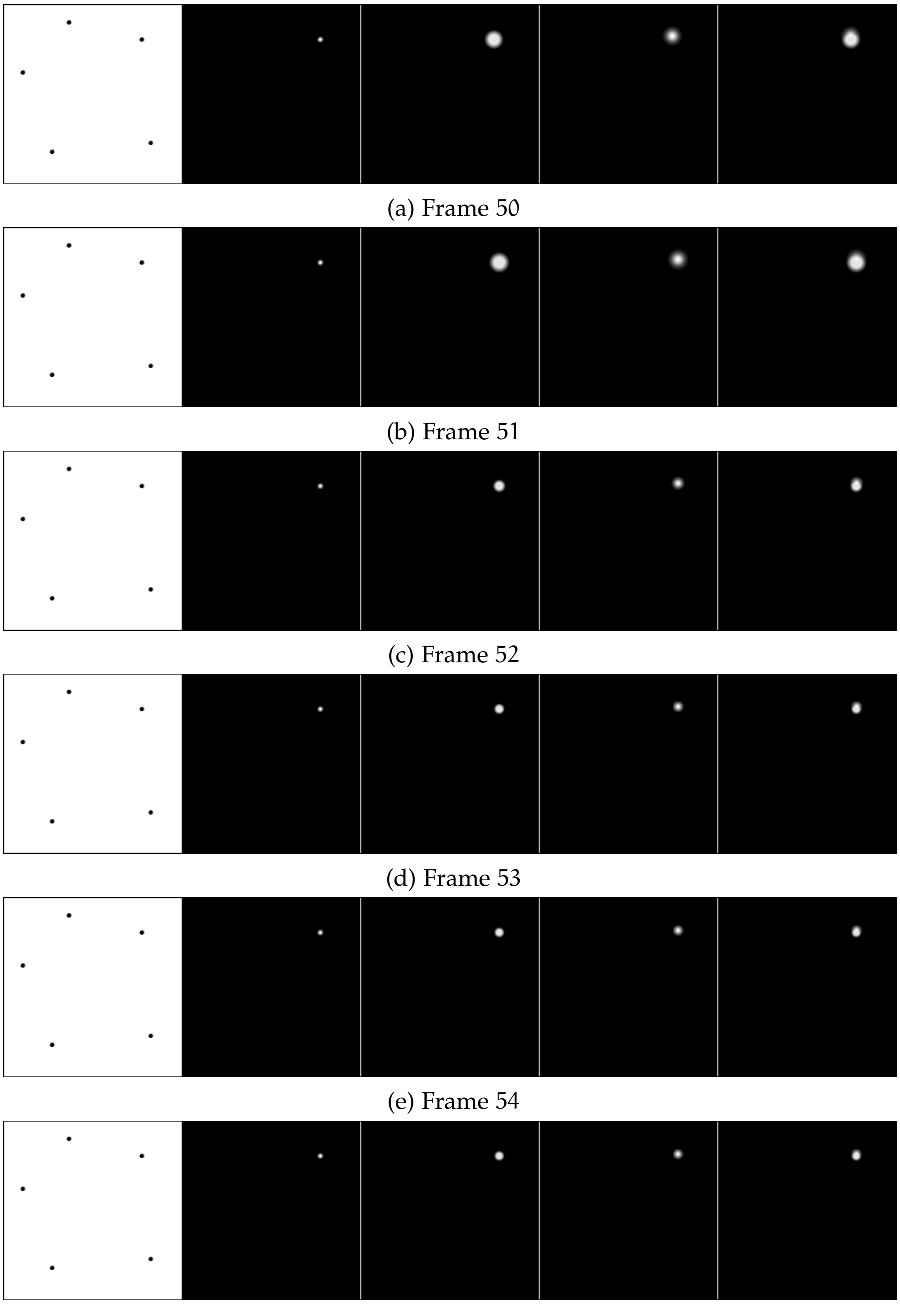
5.7 EXPERIMENTAL RESULTS USING NON SYNTHETIC SEQUENCES

After some preliminary tests with synthetic sequences, the algorithm was applied to several non synthetic image sequences. These microscopic sequences were obtained from Bacterial Motility and Behavior group located in the Rowland Institute at Harvard[1] with approval of Professor Howard C. Berg.

The first sequence shows a bacteria called *Serratia Marcescens* moving over the surface. This sequence is constituted by 300 frames recorded with a frame rate of 13 fps and each frame has a size of 320 × 240. In this sequence, the bacteria denotes a white colour against a dark background. To illustrate the effectiveness of the approach, one bacteria is picked at the initial frame and the trajectory of such bacteria is estimated until it reaches a predefined final frame. The initial and final frames are defined by the user. In Figure 45a the selected bacteria is indicated and, in Figure 45b, its estimated trajectory is depicted with a solid green line indicating the absence of occlusion or misdetection situations.

The other test sequence is also a bacteria, an organism called *Synechococcus*, swimming in a medium. The sequence was recorded with an acquisition

[1] http://www.rowland.harvard.edu/labs/bacteria/index.html



(a) Frame 50

(b) Frame 51

(c) Frame 52

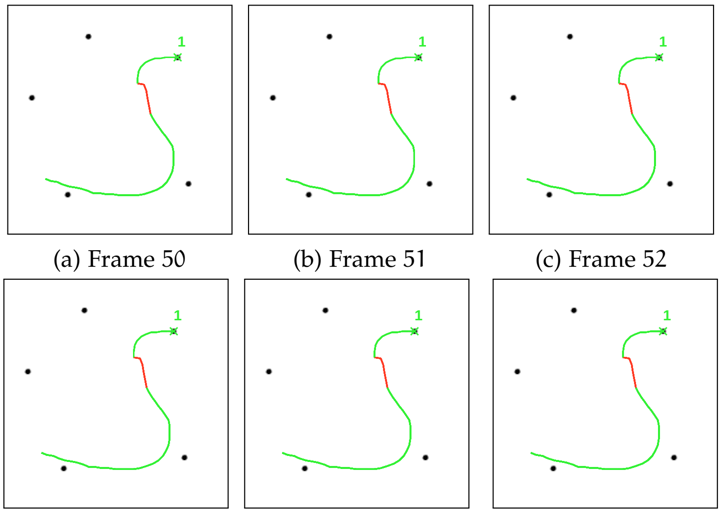
(d) Frame 53

(e) Frame 54

(f) Frame 55

Figure 39: Static situation in sequence S1 (frame 50 to 55).

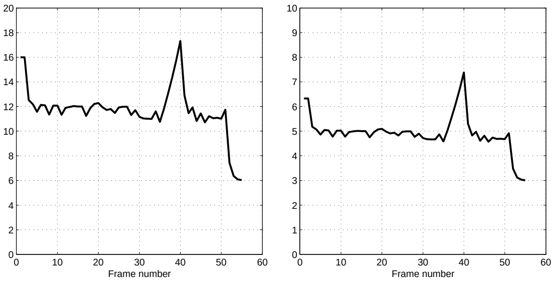
rate of 30 fps, with an image size of 320 × 240 and a total of 103 frames. This sequence is more complex than the previous one. There are more similar features in the scene, the background is not uniform and the difference of grey levels between feature and background is lower. Once again, one feature is selected in frame number 1, Figure 46a, and the estimated trajectory until frame number 100 is depicted in Figure 46b.



(a) Frame 50 (b) Frame 51 (c) Frame 52

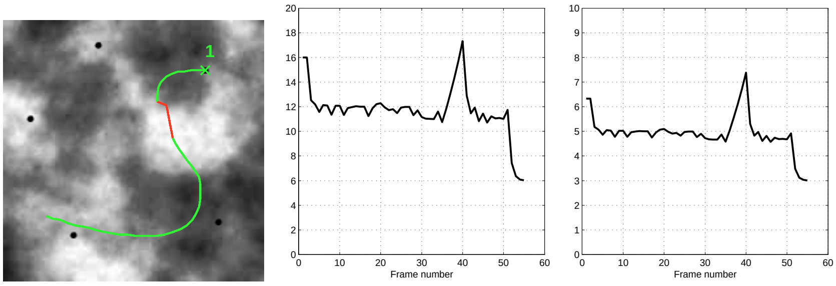
(d) Frame 53 (e) Frame 54 (f) Frame 55

Figure 40: Estimated trajectory in static situation.



(a) Outer radius of μ_S (b) Sigma value of μ_K

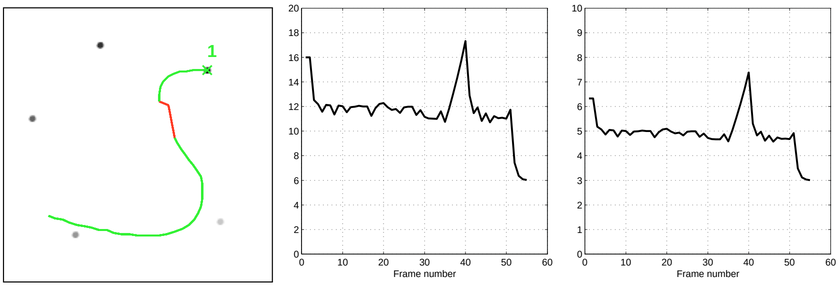
Figure 41: Parameters of membership functions μ_S and μ_K during sequence S1.



(a) Estimated position (b) Outer radius of μ_S (c) Sigma value of μ_K

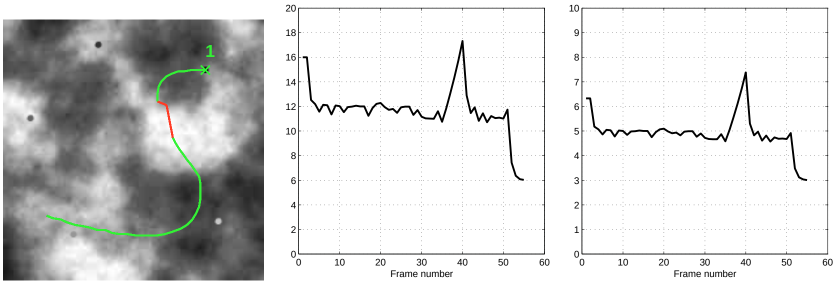
Figure 42: Estimated position and evolution of parameters in membership functions μ_S and μ_K during sequence S2.

The third sequence is a *Mycoplasma* bacteria gliding. 704 frames with a spatial resolution of 720 × 480 were recorded with a temporal resolution of 30 fps. An initial bacteria is selected in frame number 1 and its estimated trajectory is presented until frame number 540. These two situations are depicted in Figures 47a and 47b, respectively.



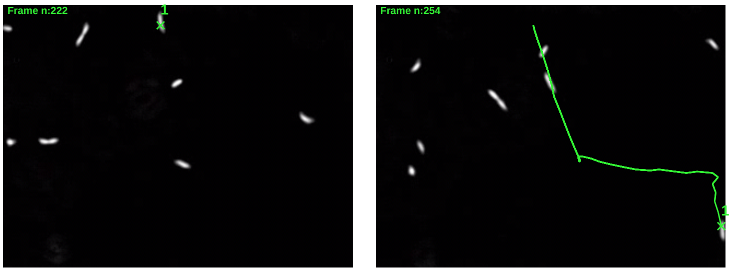
(a) Estimated position (b) Outer radius of μ_S (c) Sigma value of μ_K

Figure 43: Estimated position and evolution of parameters in membership functions μ_S and μ_K during sequence S3.



(a) Estimated position (b) Outer radius of μ_S (c) Sigma value of μ_K

Figure 44: Estimated position and evolution of parameters in membership functions μ_S and μ_K during sequence S4.

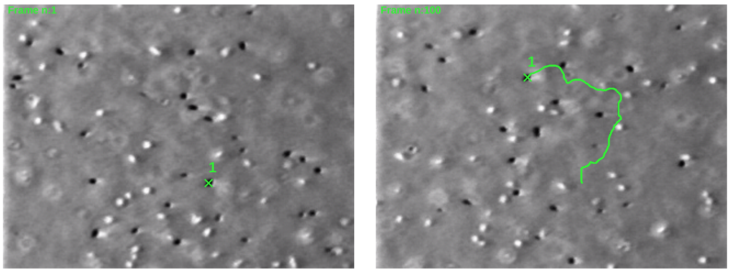


(a) Selected bacteria. (b) Estimated trajectory.

Figure 45: Estimated trajectory of a bacteria between frames 222 and 254.

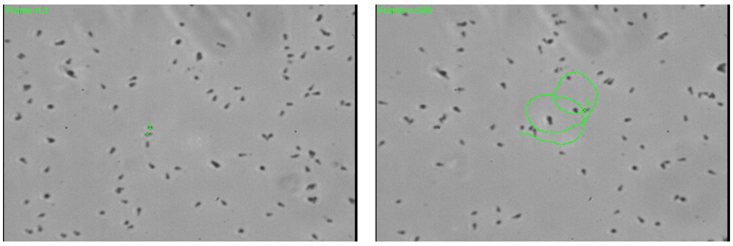
5.7.1 *Performance Measurement*

In order to access a performance measure of the proposed approach, experimental results are compared with manually generated feature positions. These ground-truth positions have been manually defined for each frame of the sequences presented previously. They represent the centre of mass of the feature and they will be used as a gold standard. Using these ground-truth positions, a tracking error measurement could be developed calculating the average and standard deviation of an Euclidean distance between experimental and ground-truth positions.



(a) Selected organism. (b) Estimated trajectory.

Figure 46: Estimated trajectory between frames 1 and 100.



(a) Selected organism. (b) Estimated trajectory.

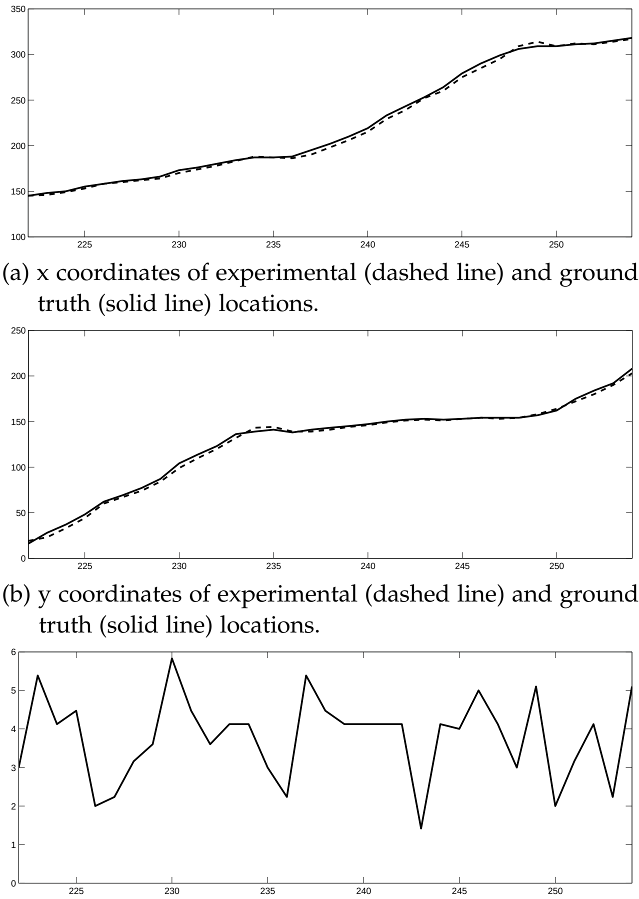
Figure 47: Estimated trajectory between frames 1 and 540.

In Figures 48, 49 and 50, a pictorial comparison between the experimental and ground truth (x,y) coordinates and the distance error for all frames are presented, respectively, for the three image sequences introduced previously. Since both experimental and ground-truth positions have single pixel resolution, the distance error values are discrete rather than continuous.

Table 5 gives the average and standard deviation of the distance errors. The first sequence denotes higher values since the moving bacteria denotes the tracked feature (high grey level intensity) in all their surface and the bacteria presents an elongated shape leading the method to provide experimental positions not close enough to the centre. However, since the tracking process is performed correctly, these errors could be solved using morphological approaches. In the *Mycoplasma* bacteria sequence the results are better but not the best since this bacteria presents grey level variations over the sequence leading the method to provide positions not close enough from its centre and fuzzy boundaries causing difficulties to achieve ground-truth positions with high accuracy. The second sequence provides best results since the feature is radial, with well defined boundaries and no significant variations in its grey level intensity are observed over the sequence.

5.7.2 *Feature Tracking with Low Frame Rate*

Performance tests concerning low frame rate acquisition were also subject of attention. To simulate a low frame rate acquisition, the images of the sequence used in the algorithm are not taken in a consecutive manner but a jump of several frames is made. The first sequence was recorded at 13 fps, if the method gets a frame, discard the consecutive frame and uses the next



(a) x coordinates of experimental (dashed line) and ground-truth (solid line) locations.

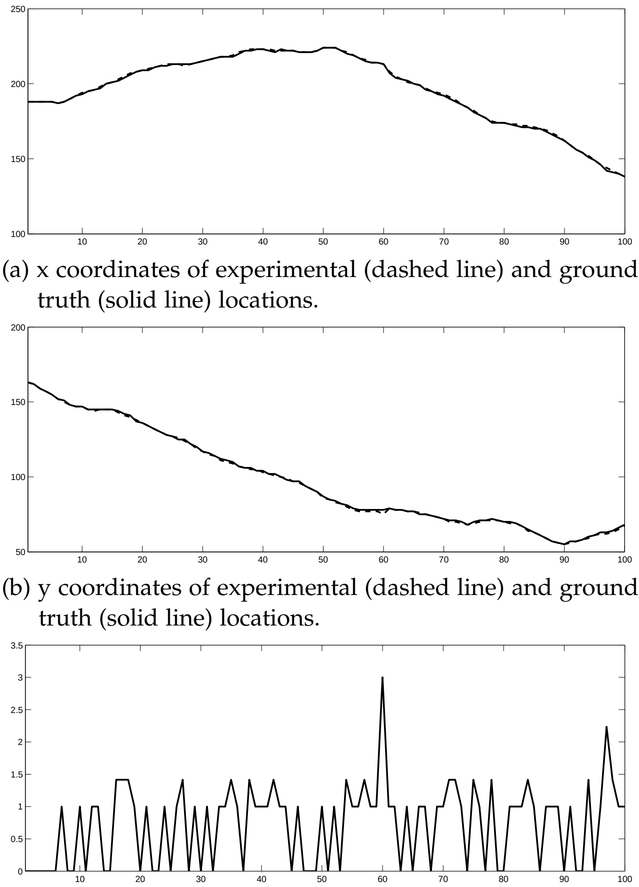
(b) y coordinates of experimental (dashed line) and ground-truth (solid line) locations.

(c) Distance between experimental and ground-truth locations.

Figure 48: Ground truth comparison between frames 222 and 254 of a *Serratia Marcescens*.

frame, i. e., the frames are used with an interval of 2 frames, the frame rate is halved. If the sequence was acquired at 30 fps, if the method makes a jump of 5 frames, its equivalent to have a frame rate of 6 fps. These procedures were used in the sequences to simulate low frame rate.

Figure 51a illustrates the estimated trajectory between frames 1 and 76 with a frame rate of 6.5 fps, Figure 51b represents the trajectory between frames 1 and 100 and Figure 51c depicts the estimated trajectory between frames 1 and 300. The frame rate used in these two latter sequences was 6 fps. Since the time interval between two frames used by the methodology increases, less information is used and therefore, the trajectory is not so detailed as shown previously. However, the selected features still continue to be tracked correctly. An important issue is related with the velocity of the feature between two consecutive frames, if it is too high, the current position of the feature could not lie inside the search region, specially at the beginning of the sequence when the state vector of the feature is not stable yet.



(a) x coordinates of experimental (dashed line) and ground-truth (solid line) locations.

(b) y coordinates of experimental (dashed line) and ground-truth (solid line) locations.

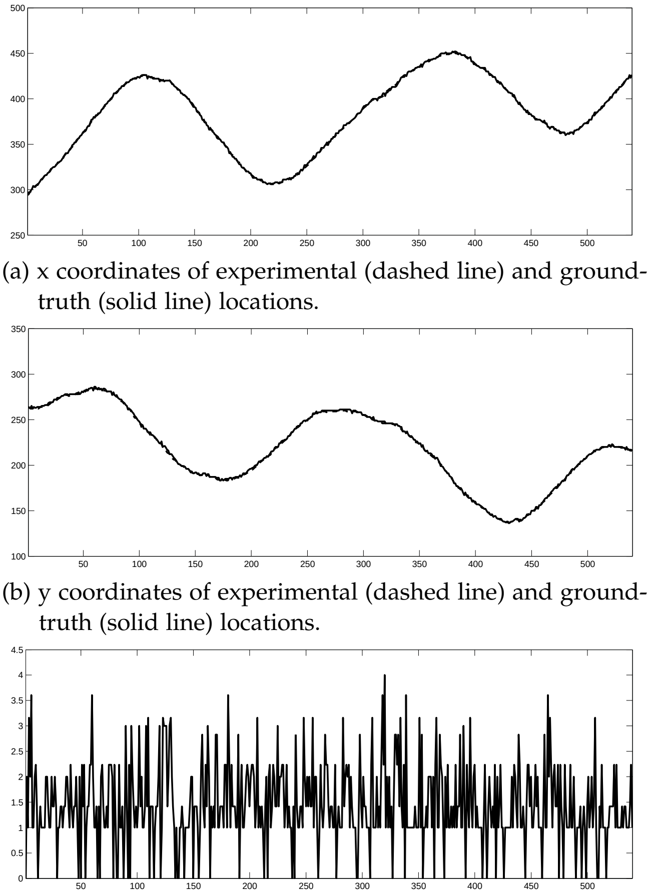
(c) Distance between experimental and ground-truth locations.

Figure 49: Ground truth comparison between frames 1 and 100 of a *Synechococcus*.

5.7.3 *Feature Tracking with Alternative Membership Functions*

Since this methodology is modular it can be easily adapted according the user end application. The number and nature of used membership functions can change to deal with the suitable distinguish properties denoted by the feature. For instance, instead of using only the grey level to distinguish the feature it can be used also its template. With a template of the feature, the concept of cross-correlation between the template and the image can be used. According to Equation 2.21, a new membership function μ_C can be developed. Image regions where the template matches perfectly have higher cross-correlation values, otherwise, regions not denoting a similarity between the template have lower μ_C values. The range of the membership function μ_C belongs to the interval $[0, 1]$. Since the range of Equation 2.21 is $[-1, 1]$, negative values are truncated to zero. This procedure doesn't affect the performance of the method because the interested values are the maximum values of μ_C.

To evaluate the performance of using cross-correlation in the methodology, a feature is selected to be tracked during several frames of the sequence. At



(a) x coordinates of experimental (dashed line) and ground-truth (solid line) locations.

(b) y coordinates of experimental (dashed line) and ground-truth (solid line) locations.

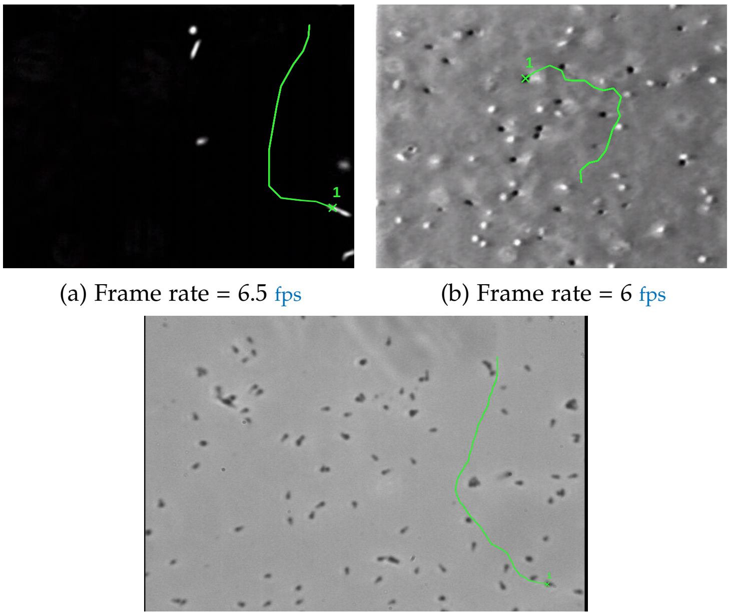
(c) Distance between experimental and ground-truth locations.

Figure 50: Ground truth comparison between frames 1 and 540 of a *Mycoplasma*.

frame number 8 the feature is selected by the user, Figure 52a, and a template of the selected feature is defined, Figure 52b. Between frame number 8 and frame number 40 an estimated trajectory is constructed as illustrated in Figure 52c. This trajectory is represented by a green line, but after several frames this line becomes red. Red colour indicates a period where the feature is occluded or not detected by the algorithm and the trajectory is constructed using predicted positions calculated using the motion model and the information of the feature state vector. During this period the use of this alternative membership function fails because the feature changes significantly its appearance and the membership function μ_C returns low values. If the values of membership function μ_C are lower than a predefined value, all the regions of the image are considered quite different from the template and, in this situation, the feature will not be detected. Fortunately, after some time the feature returns to an appearance very similar with the template and it will be detected once again. Figure 53 illustrates a situation where the tracked feature changes permanently the similarity between its template and, consequently, the tracking fails. This tracking failure is related

Table 5: Average distance error and standard deviation.

SEQUENCE	INTERVAL OF FRAMES	MEAN	STANDARD DEVIATION
Serratia Marcescens	[222, 254]	3.7911	1.1000
Synechococcus	[1, 100]	0.7286	0.6188
Mycoplasma	[1, 540]	1.4452	0.7991



(a) Frame rate = 6.5 fps

(b) Frame rate = 6 fps

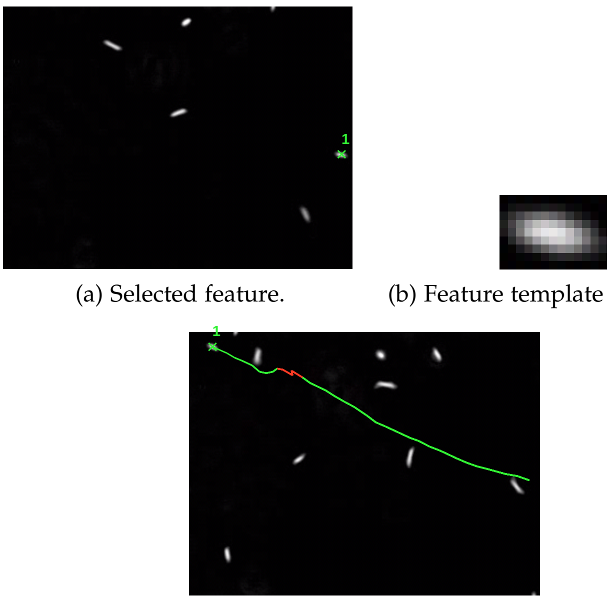
(c) Frame rate = 6 fps

Figure 51: Estimated trajectory with low frame rate.

with the inner limitations of the cross-correlation concept because cross-correlation is not invariant to template scale and rotation.

Also in Lopes et al. and Couto et al. [70, 26] slightly different membership functions were used to construct the fuzzy sets S and K. In Lopes et al. the membership function μ_K, illustrated in Figure 54a, is very similar with the one described in this work but the sigma value of μ_K is defined exclusively by $\frac{\Delta p}{3}$, where Δp is equal to half of the displacement of the feature from frame $t-2$ to frame $t-1$. When feature stops the sigma value becomes zero and the membership function fails since the Gaussian function needs a non-zero sigma. To overcome this situation a minimum sigma value should have been established in this work. Also, the membership function μ_S depicted in Figure 54b, denotes an exponential decay, however, this behaviour introduces more computational resources and, when replaced by a linear decay, no performance losses were observed. Therefore, this latter behaviour has been adopted in future work.

In Couto et al. the membership function μ_K is also a Gaussian shaped function but the membership function μ_S is asymmetrical and two distinct

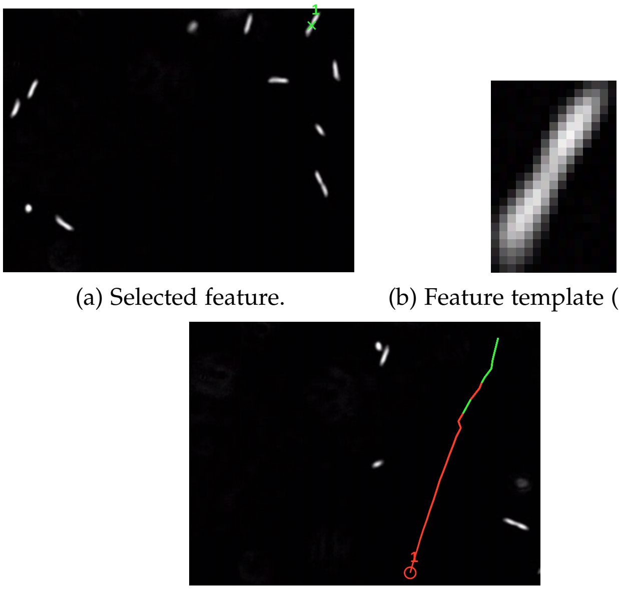


(a) Selected feature.

(b) Feature template (zoomed).

(c) Estimated trajectory.

Figure 52: Estimated trajectory between frames 1 and 40.



(a) Selected feature.

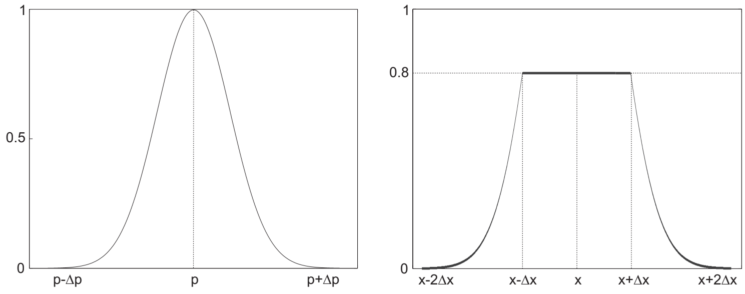
(b) Feature template (zoomed).

(c) Estimated trajectory.

Figure 53: Estimated trajectory between frames 50 and 75.

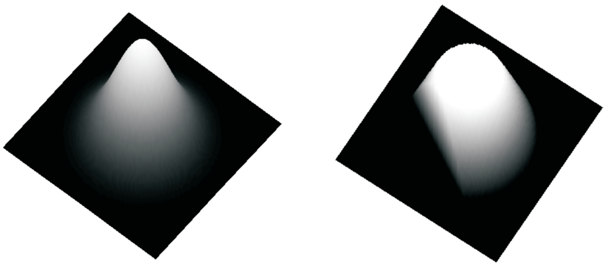
zones of certainty are present in the definition of this membership function. More certainty is given in the feature movement direction. A 3D representation of these two membership functions is illustrated in Figures 55a and 55b, respectively.

In order to illustrate the experimental results of both methods, two sequences are used. The used sequences were taken from the Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS) Benchmark Dataset Collection[2] and from the Center for Image Processing Research (CIPR) [3].



(a) μ_K (b) μ_S

Figure 54: Alternative used membership functions.



(a) μ_K (b) μ_S

Figure 55: 3D representation of alternative used membership functions.

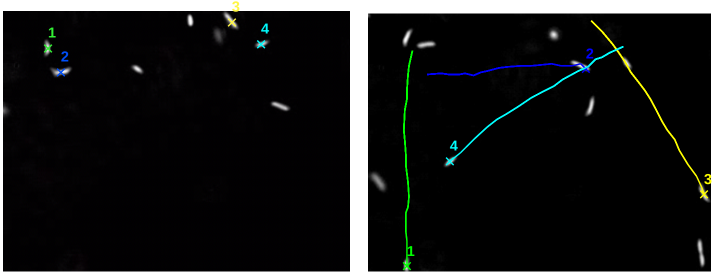
5.7.4 *Multiple Feature Tracking*

After testing the performance of the tracking algorithm for one feature, the methodology is extended to track simultaneously more than one feature in the sequence. This extension was achieved repeating the tracking method for each feature. At the beginning, the user selects several features to track and the method starts finding the best correspondence in the next frame, and consequently, defines each trajectory. Figures 56, 57 and 58 illustrate the estimated trajectories of several selected features. This correspondence process starts to be performed from the first selected feature to the last. However, this fixed matching sequence based uniquely in the order of the selected features couldn't be the optimal scheme during the entire sequence and another matching sequence must to be implemented. This new matching sequence must be dynamic to ensure that well behaved and mature features are processed and assigned first than erratic features. Mature features are features that had been tracked correctly for a long time and being well behaved they are expected to maintain the same motion pattern. Therefore,

[2] http://www.cse.ohio-state.edu/OTCBVS-BENCH/index.html

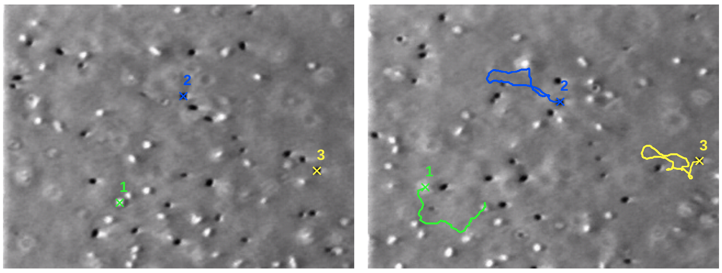
[3] http://www.cipr.rpi.edu/resource/sequences/sif.html

the next position of these features could be predicted, potentially, with more confidence and the matching process would perform better.



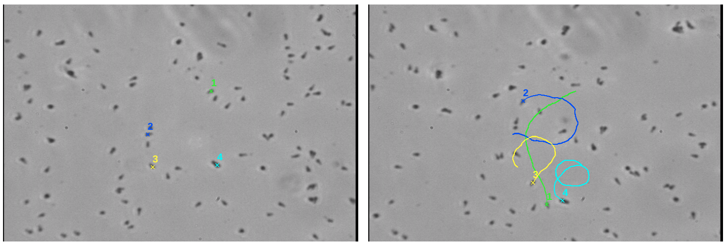
(a) Selected feature. (b) Estimated trajectory.

Figure 56: Estimated trajectories between frames 26 and 43.



(a) Selected features. (b) Estimated trajectory.

Figure 57: Estimated trajectories between frames 1 and 80.



(a) Selected feature. (b) Estimated trajectory.

Figure 58: Estimated trajectories between frames 1 and 200.

5.7.5 *Feature tracking using macroscopic sequences*

In the microscopic sequences introduced previously the tracking feature is present in all the area occupied by the moving object. However, in most macroscopic object tracking systems, the tracked feature is present only in a particular region of the moving target. To access the behaviour of this methodology in macroscopic scenes, two sequences obtained from Context Aware Vision using Image-based Active Recognition (CAVIAR) [4] and Performance

[4] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

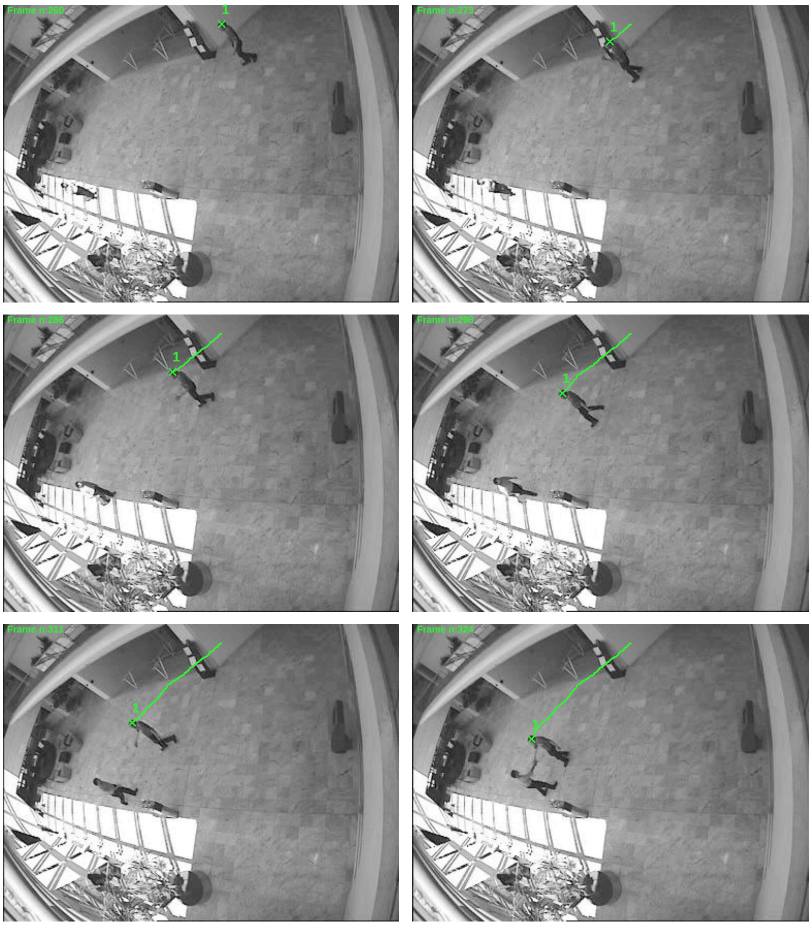


Figure 59: Estimated trajectory between frames 260 and 400 (first frames).

Evaluation of Tracking and Surveillance (PETS) [5] test case scenarios are used. These two sequences are used because the selected features to track denote different motion patterns and occlusion situations are also present.

The first sequence, used in CAVIAR test case scenarios, represents two people meeting and walk together in an entrance lobby. The selected tracking feature is the black hair of one person chosen by the user at the beginning of the sequence. This feature starts to move in a single direction, stops for a few moments and then takes another direction to continue its motion. Figure 59 illustrates the effectiveness of the proposed methodology between frames 260 and 400.

To achieve a performance evaluation of the approach, experimental positions computed by the proposed methodology for all frames of the tested sequence are compared with ground-truth positions generated manually. The evolution of the Euclidean distance between these two positions can be observed in Figure 60.

The maximum value of the error distance is lower than 3 pixels as observed in Figure 60. This maximum value is considerable small and can be explained

[5] http://www.cvg.rdg.ac.uk/PETS2010/a.html

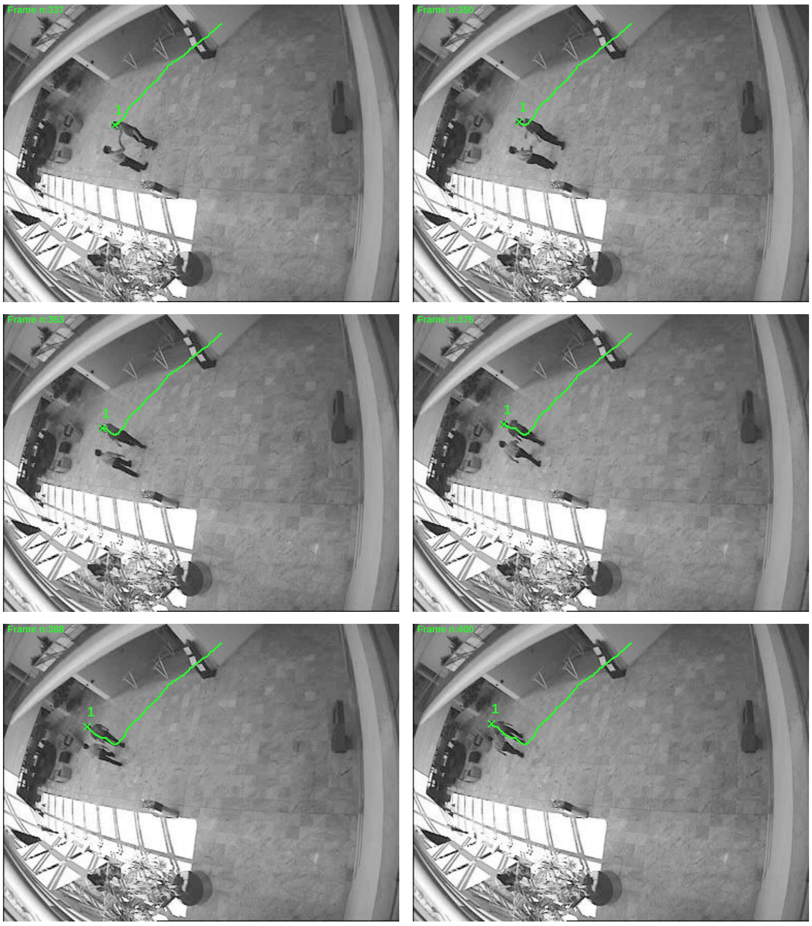


Figure 59: Estimated trajectory between frames 260 and 400 (last frames).

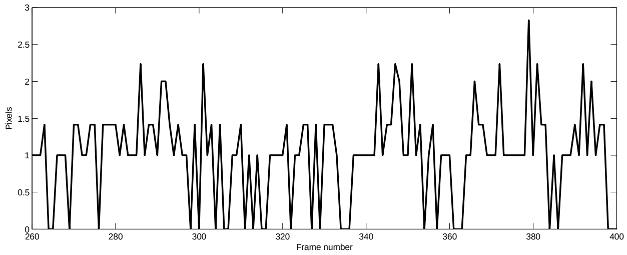


Figure 60: Distance between experimental and ground-truth positions.

due to errors introduced in the ground-truth generation and also because the proposed method doesn't ensure that the output estimated position lies at the centre of mass of the feature. However the methodology performs a correct feature tracking.

The other sequence, presented in PETS test case scenarios, represents two people walking together in the street and one person is carrying several white paper sheets in its hand. The high grey level intensity of these white paper sheets is the selected tracking feature.

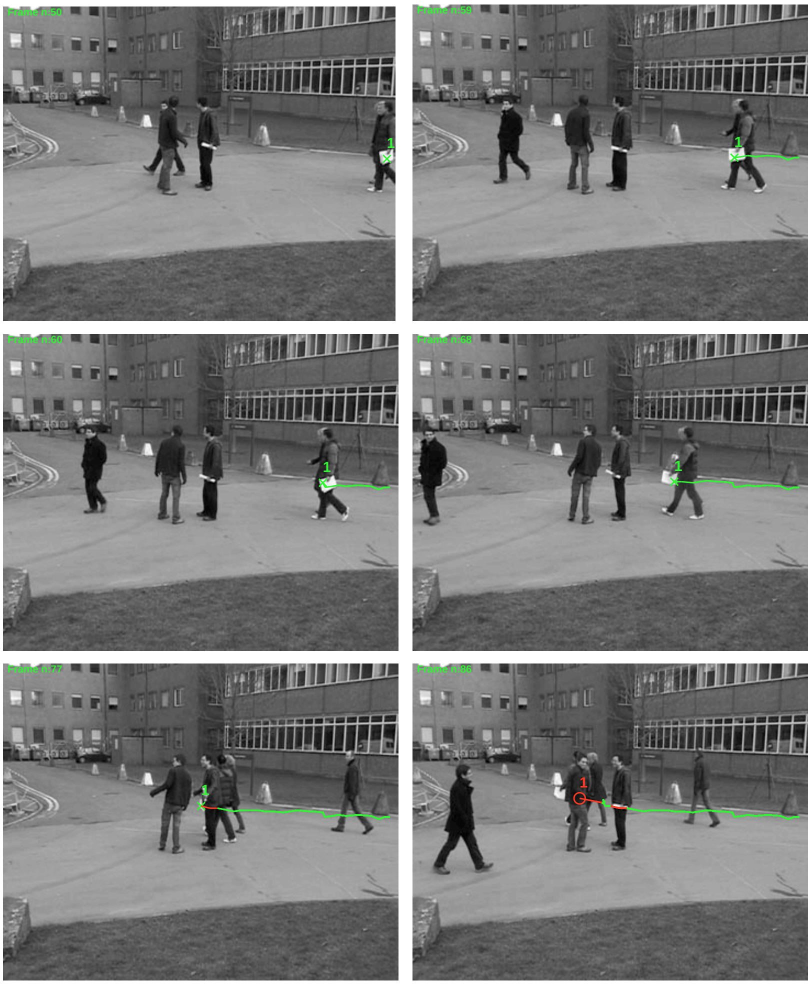


Figure 61: Estimated trajectory between frames 50 and 150 (first frames).

This sequence is more challenging than the previous because the feature presents more complex motion behaviour since it denotes a swinging motion pattern due to the human hand movement during walking and it suffers several occlusions along the sequence.

The result of the proposed methodology between frames 50 and 150 is illustrated in Figure 61.

The evolution of the Euclidean distance between experimental and ground-truth positions is depicted in Figure 62.

Higher error values at the beginning of the sequence are easily explained by the large size and also by the swinging pattern of the feature observed in the image plane leading the estimated positions given by the proposed methodology to be not close enough from its centre of mass. Another issue is related with the occlusion situations that could increase the errors between the ground-truth positions since the method doesn't have any information about the feature. Despite these source errors, the maximum error value observed in the first half of the sequence doesn't degrade the tracking process of the proposed methodology. In the second half there are no presence of

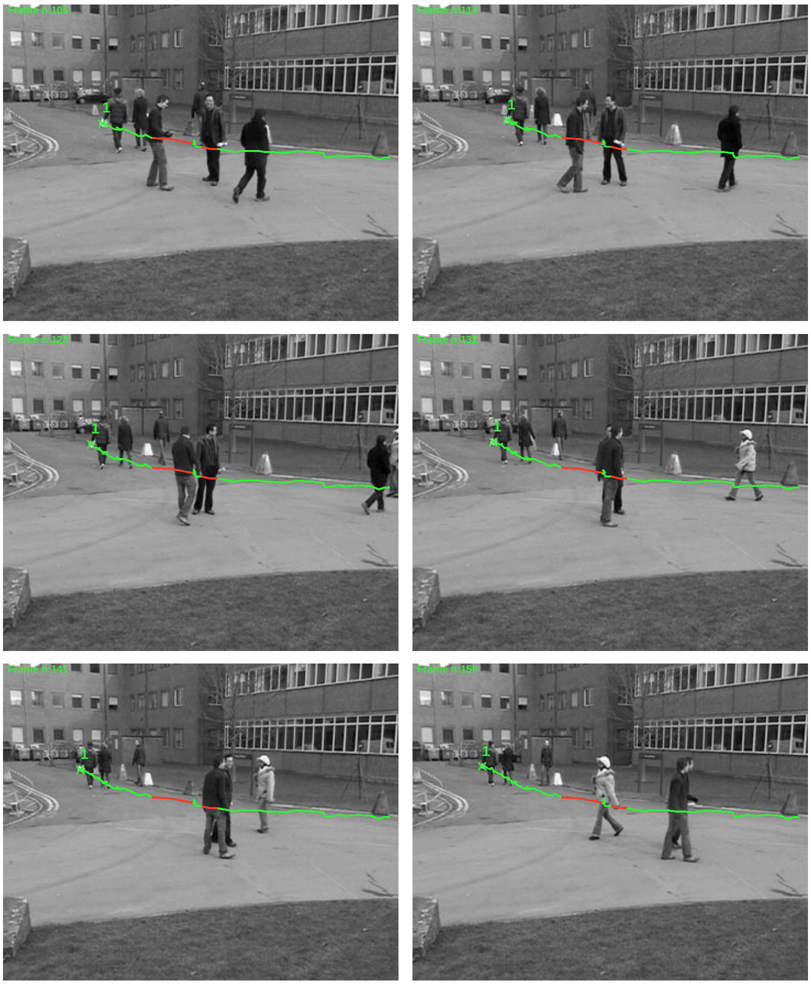


Figure 61: Estimated trajectory between frames 50 and 150 (last frames).

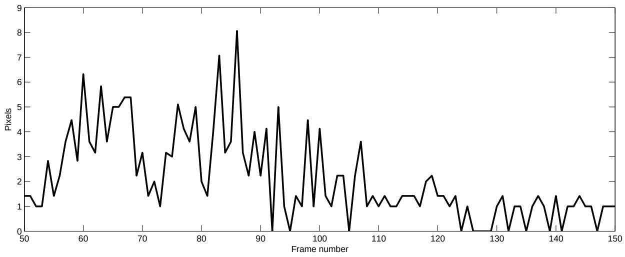


Figure 62: Distance between experimental and ground-truth positions.

occlusions and the feature starts to move along a direction that reduces the swinging motion of the feature observed in the image plane, therefore, the distance errors are significantly decreased.

The average and standard deviation of the Euclidean distance between experimental and ground-truth positions for the tracked feature in these two sequences are presented in Table 6.

Table 6: Average distance error and standard deviation.

SEQUENCE	INTERVAL OF FRAMES	MEAN	STANDARD DEVIATION
CAVIAR	[260, 400]	1.0186	0.6307
PETS	[50, 150]	2.1320	1.7490

In Table 6, it is possible to observe that the average distance error and standard deviation values are low for both sequences, indicating the correct performance of the method.

5.8 discussion

In this chapter a new feature tracking approach based in fuzzy sets was introduced. This approach adopts the concepts introduced in fuzzy logic theory but implements a new viewpoint related to the use of fuzzy logic concepts to perform feature tracking.

With the definition of three basic membership functions and the construction of a rule based inference engine, an algorithm providing quite satisfactory results was achieved. Using fuzzy logic theory, human like reasoning approaches have been used to develop this methodology. Humans are capable to interpret changes in the environment and think forward to predict future behaviours. One distinctive property sensitive to human eyes is the grey level intensity, therefore, a membership function is constructed related with the feature grey level. This membership function assigns a value, to each pixel of a frame, according its proximity to the grey level of the selected feature. Humans always start to find an object in its last known position, so, a membership function is used to define a search region around the last position of the feature. Finally, a third membership function is constructed related with the predicted position of the feature in the next frame. When a feature denotes a constant motion pattern, humans are capable to think forward and focus their attention in regions where it is expected to find the near feature. Even in cases of occlusion, its plausible to estimate the localisation of the feature using its previous position and motion behaviour. These three membership functions combined with a rule inference engine are an attempt to implement a human inspired reasoning based system.

This approach was formerly applied to several synthetic sequences and then to real image sequences. The results are quite satisfactory and when the velocity of the feature is low, the algorithm can perform with low acquisition rates without losing its performance. If the feature changes rapidly its velocity, the next position of the feature could lie outside the expected search region and the method fails.

Since this methodology is modular, alternative membership functions could be used and an example of feature tracking using cross-correlation was presented. A preliminary test to achieve a multiple feature tracking system is performed, however, this simplest attempt is not acceptable because well behaved features should be always processed first and this issue is not ensured.

If I find 10000 ways something won't work, I haven't failed.
I am not discouraged, because every wrong attempt discarded
is another step forward.

— Thomas A. Edison

6 HIERARCHICAL MATCHING APPROACH FOR MULTIPLE FEATURE TRACKING

6.1 INTRODUCTION

In this chapter an extension of the fuzzy based tracking algorithm presented previously will be introduced to deal with multiple feature tracking. The purpose of this extension is to incorporate a hierarchical scheme in the processing sequence of the features. When the user picks up several features to track, $1, \ldots, n$, the natural order is to start processing the first selected feature and continuing till the last, for all frames in the sequence. This initial schedule could be suitable during the first frames of the sequence, however, after some time it could bring worst results. Performance reduction could be explained due to the fact that along time, some features could become occluded, due to background or other existing features, and less information about these occluded features is available to the algorithm. This lack of information causes degradation in the estimation of the state vector of such features leading to predicted positions with higher uncertainty. If these features were the ones selected first by the user, the algorithm will start to match these low certainty features rather than well behaved features.

Hence, to avoid these features to have higher priority, and being assigned first than features that provide more information to the algorithm, its plausible to implement a hierarchical matching scheme. In this hierarchical matching scheme, features that provide more information to the algorithm have priority in the matching step. Furthermore, when features became occluded, their location will be predicted over several frames, if the algorithm continues to not get reliable information about the location of the features, such features will be deleted from the list of tracked features.

The remainder of this chapter is organised as follows: In Section 6.2 some preliminary assumptions concerning this algorithm are presented. In Section 6.3, the procedure to deal with situations of inter feature and background occlusion is explained. The proposed hierarchical fuzzy tracking approach is presented in Section 6.4. Results with synthetic and non synthetic sequences are analysed in Sections 6.5 and 6.6, respectively. At the end of this chapter, in Section 6.7, a brief discussion concerning this hierarchical matching approach for multiple feature tracking is presented.

6.2 INITIAL CONSIDERATIONS

This new approach assumes the same initial considerations introduced in the method presented previously: the feature has constancy of intensity, has smooth motion, the feature motion between two consecutive frames can be described using a linear motion model, the area occupied by the feature is small compared to the total image area and the shape of the feature doesn't suffer relevant changes over the sequence.

The used membership functions and, consequently, the used fuzzy sets remain the same. One fuzzy set S related to the previous location of the feature, one fuzzy set G concerning the feature grey level and another fuzzy set K concerning the predicted position of each feature. The rule inference engine is also maintained in this extension, however, this extension will implement a hierarchical matching scheme. To perform this hierarchical scheme each feature is associated with a confidence degree modelling the amount of knowledge about the feature. Higher the knowledge about the feature higher the confidence degree and, therefore, higher the priority.

At the beginning of the process, when the user selects the features to track, all features are associated with a predefined maximum value of confidence. If a feature becomes occluded its confidence degree decreases, otherwise increases until reaches the predefined maximum value. The fuzzy matching approach explained in Chapter 5 is then first applied for features denoting higher confidence degree. When its confidence degree becomes too low, the approach doesn't have reliable information about the location of the feature for a long time and its location can't be predicted with reasonable certainty. In this case this feature is no longer tracked.

6.3 OCCLUSION AND MISDETECTION

In multiple feature tracking several correspondence situations can occur. Figure 63 depicts these situations, where ○ denotes the feature position at frame $t-1$ and × denotes the feature position at frame t. The question mark (?) represents the absence of correspondence at frame t. The first situation depicted in Figure 63a indicates that each feature is matched with a different candidate point in the next frame and the current position, at frame t, of the feature will be the position of the corresponding candidate.

Sometimes different features in frame $t-1$ will be assigned to the same point in frame t. When two moving features pass close each other or when one feature occludes an other, or even also due to the representation of a 3D world in a 2D plane, they can appear as being just one region in the image. However one region in the image plane represents several features. This situation could be seen as a merging of features or a inter feature occlusion, Figure 63b.

The opposite situation is also considered, i. e., several features could have different motion directions and one single region, representing multiple features, could result in multiple matching points. It could be seen as a split of features, Figure 63c.

Finally, at some instant, there are no candidate points for a feature and, without correspondence, a predicted position is assumed, Figure 63d. Its

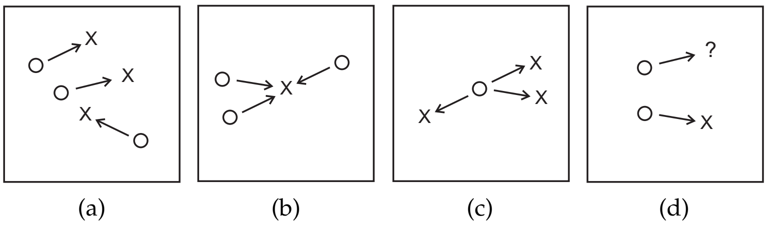


Figure 63: Correspondence situations in multi feature tracking.

a typical situation of background occlusion. All these situations must be considered in the tracking algorithm and appropriate procedures must be applied in each case.

In this multiple feature tracking approach it is possible to distinguish between inter feature occlusion and background occlusion and different actions could be performed for each case.

6.4 METHODOLOGY

According the assumptions and situations described previously, a new tracking approach was developed. At the beginning, the user selects multiple features to track and each feature is associated with a confidence level. This confidence level is related with the trust or knowledge about a feature. At the beginning of the sequence, each feature has its confidence degree at the maximum value meaning the position of each feature is known with higher certainty or confidence. When the algorithm doesn't have information about the feature, its confidence degree decreases. It is considered that situations of inter feature or background occlusion and misdetections introduce absence of information about the feature. Inter feature occlusion is considered when the distance between the location of several features is lower than the size of each feature. Using the similar assumption indicated previously in the fuzzy tracking algorithm, each feature is represented by a 3×3 matrix, then, when the distance between the position of two features is lower than 3 pixels, it indicates the presence of an inter feature occlusion.

For all situations previously described, the algorithm performs as follows

1. When only one feature is matched with a candidate, its confidence degree increases until it reaches a maximum value, Figure 63a;

2. When several features are matched with the same candidate, the feature assigned first with that candidate see its confidence degree increasing and the other features see their confidence degree decreasing slowly, Figure 63b;

3. When there is no candidate to be matched with a feature, a predicted position is used and its confidence degree decreases faster, Figure 63d;

4. Features with low confidence degree are removed from the list of tracked features.

All the procedures indicated previously are first applied to features with higher confidence level and this way the hierarchical matching approach is performed.

Situation depicted in Figure 63c is covered by the first step since one single feature is matched with a candidate and, in this situation, the confidence degree for such features increases. To perform the fourth step, a threshold of minimal confidence degree is assumed and features denoting a confidence degree below this minimum value will not be tracked.

In inter feature occlusion situation there exists one region visible in the frame that represents at least one visible feature. Some features could have disappeared at this moment but at least one visible feature is present and continues to be tracked. The feature with higher confidence degree continues to be tracked without decreasing its confidence degree but the occluded features suffer a slow decrease in their confidence degree. When background occlusion occurs, there is no candidate present in the frame and the feature suffers a higher decrease on its confidence degree. The slow decay in confidence degree when inter feature occlusion situation occurs ensures that inter occluded features continue to be tracked over a large number of frames, giving the opportunity to some inter occluded features come out from the common region before being deleted.

The proposed approach can be globally described through the pseudocode presented in Figure 64.

At the initialisation stage several steps are performed in order to define and initialise all the variables, matrixes and structures. It is also defined a maximum and minimum confidence degrees equal to 16 and 6, respectively. After this point, the first image of the sequence is displayed and the user must select the features to track. Features properties, such as grey level intensity and initial position values are recorded for further use. The next frame of the sequence is opened and the features are sorted according their confidence degree. Features with higher confidence degree are processed first by the fuzzy logic algorithm introduced previously. The fuzzy logic algorithm performs the matching for that feature and returns the current position. If that current position was not been assigned to previous processed features and if such position was not predicted using motion model and vector state information then the confidence degree increases. However, if the current position is predicted due to an occlusion situation, the confidence level decreases. Moreover, if the current position of this feature lies in a circular neighbourhood of 3 pixels from a previously processed feature (with more confidence degree), this feature is considered occluded by that feature. It's the case when these two features appear in the image close together forming one single region. When this situation occurs, the feature with lower confidence degree suffers a reduction on its confidence degree. These steps are repeated for the remaining features. After all features had been processed by the fuzzy algorithm, an update stage is needed to update the Kalman filter and to remove features with lower confidence degree. If there exists more frames to analyse, the next frame is open and the cycle is repeated until it reaches the end of the sequence.

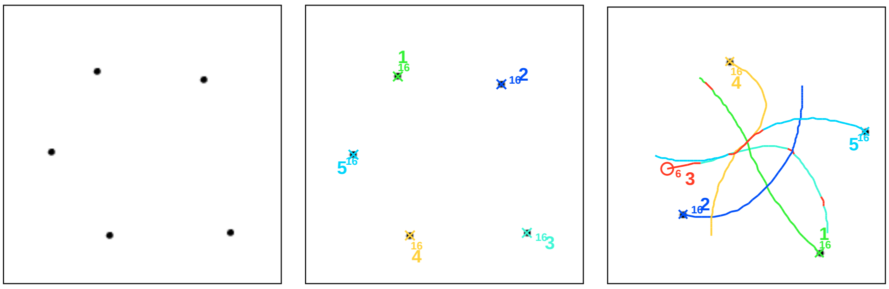
```
% --------------------------------------------------------
% Hierarchical Fuzzy Logic Tracking Approach
% for Multiple Feature Tracking
% --------------------------------------------------------
Initialization Stage ();
Read Image (First Frame);
Show Image (First Frame);
Select Features (By the User);
Save (Data of Selected Features);

for Current Feature = First Feature : Last Feature,
    Current Feature.Confidence = Maximum Confidence;
end

for Current Frame = Next Frame : Last Frame,
    Sort Descending (Feature.Confidence);
    Read Image (Current Frame);

    for Current Feature = Higher Confidence : Lower Confidence,
        [Current Position, Predicted] = function Fuzzy Tracking(Current Feature);

        if (Predicted == 0),
            Merged = 0;
            for All Previous Features
                if (Current Position == Previous Feature.Position),
                    Merged = 1;
                end
            end

            if (Merged == 1)
                % Inter Feature Occlusion or Merging
                Decrease Current Feature.Confidence;
            else
                % Correspondence One to One
                Increase Current Feature.Confidence;
            end

        else  % Background Occlusion or Misdetection
            Decrease Current Feature.Confidence;
            Increase $\mu_S$ Parameters (d1, d2);
            Increase $\mu_K$ Parameter ($\sigma$);
        end
    end

        Remove (Lower Confidence Features);
        Update (Kalman Filter);
        Show Image (Current Frame);
        Draw (Current Positions, Estimated Trajectories);
end
```

Figure 64: Detailed steps of the hierarchical tracking approach.

6.5 EXPERIMENTAL RESULTS USING SYNTHETIC SEQUENCES

In order to test the performance of this new approach, the methodology is first applied to a synthetic sequence. In this synthetic sequence all the situations described above are represented in different moments to evaluate the effectiveness of this methodology.

At the beginning of the sequence the user selects all the features to be tracked and the methodology must estimate each trajectory until the end of the sequence. The first frame of the sequence, the indication of the selected features with their confidence degrees and the final frame of the sequence showing the estimated trajectories are illustrated, respectively, in Figures 65a, 65b and 65c.



(a) Initial frame (b) Selected features (c) Estimated trajectories

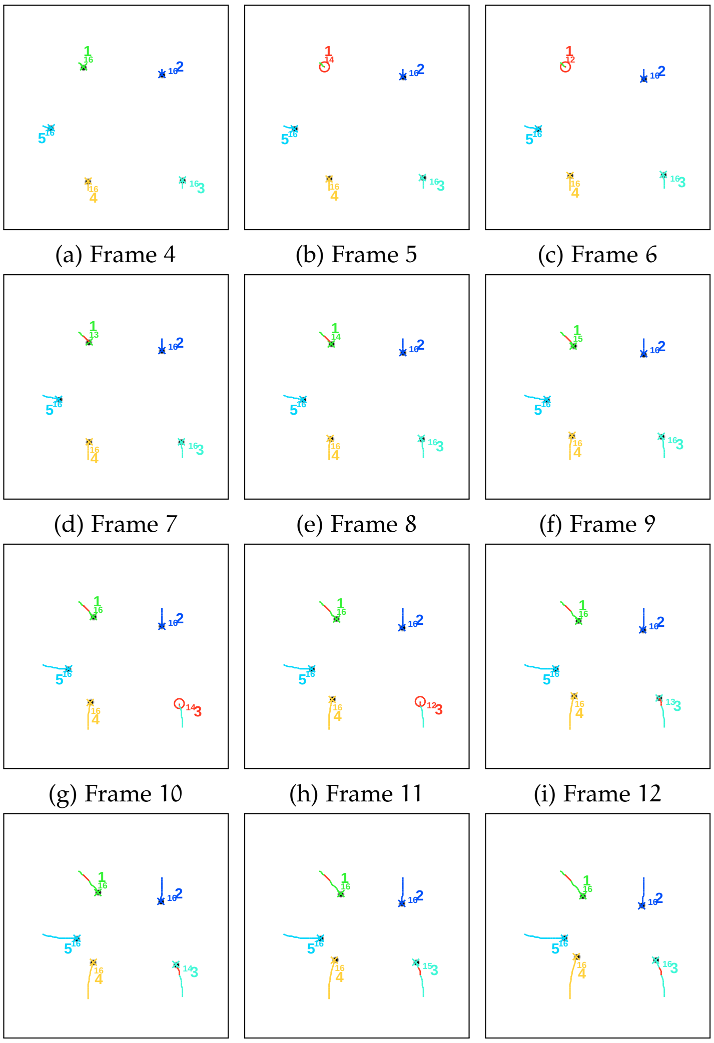
Figure 65: Multi feature tracking.

When the user selects the features to track, a numeric label and a confidence degree are assigned to each feature. The label will be exclusive for each feature and is preserved until the end of the sequence and the confidence degree starts with its maximum value and could suffer changes according to the assumptions described earlier. In this particular test, the maximum and minimum confidence values are, respectively, 16 and 6. The relevant issue in the selection of these confidence values is just the difference between them. The difference between both values determines the amount of time that a feature continues to be tracked when it suffers an occlusion or a misdetection.

The first relevant moment occurs between frames 4 and 15 as depicted in Figure 66. During this period, when features 1 and 3 are not detected, their confidence level decreases and only start to increase when the correspondence is established once again. When a feature is occluded or not detected, its estimated trajectory line turns red and the marker becomes a circle rather than a cross.

Another interesting moment to observe the behaviour of this method takes place between frames 22 and 36. At frame number 23 an inter feature occlusion is observed, i. e., features 4 and 5 become merged. Since they had the same confidence level just before this merging, no priority according their confidence degree could be performed. In draw situations, the feature selected first by the user gains priority. As illustrated in Figure 67, feature 4 is matched correctly but feature number 5 is assumed occluded and then its confidence level decreases. This behaviour remains until frame number 27. At frame 28 it is considered that feature 1 overlapped features 4 and 5. In the previous frame, features 1 and 4 had equal confidence degree but as previous explained, feature 1 prevails. Feature 2 also overlaps feature 3 and feature 2 prevails. At frame 29 all features increase their confidence degree except feature number 5. Despite the consecutive reduction on its confidence level, it remains above the minimum confidence degree defined previously and the feature continues to be tracked. At frame 33 the same region where it is supposed to accommodate features 4 and 5 suffers a division. At this point, feature number 5 is the least to be processed by the corresponding task but a correspondence was performed and then its confidence level starts increasing.

At frame 45, feature number 3 completely disappears, however, its trajectory will be estimated until its confidence degree reaches the established



(a) Frame 4 (b) Frame 5 (c) Frame 6

(d) Frame 7 (e) Frame 8 (f) Frame 9

(g) Frame 10 (h) Frame 11 (i) Frame 12

(j) Frame 13 (k) Frame 14 (l) Frame 15

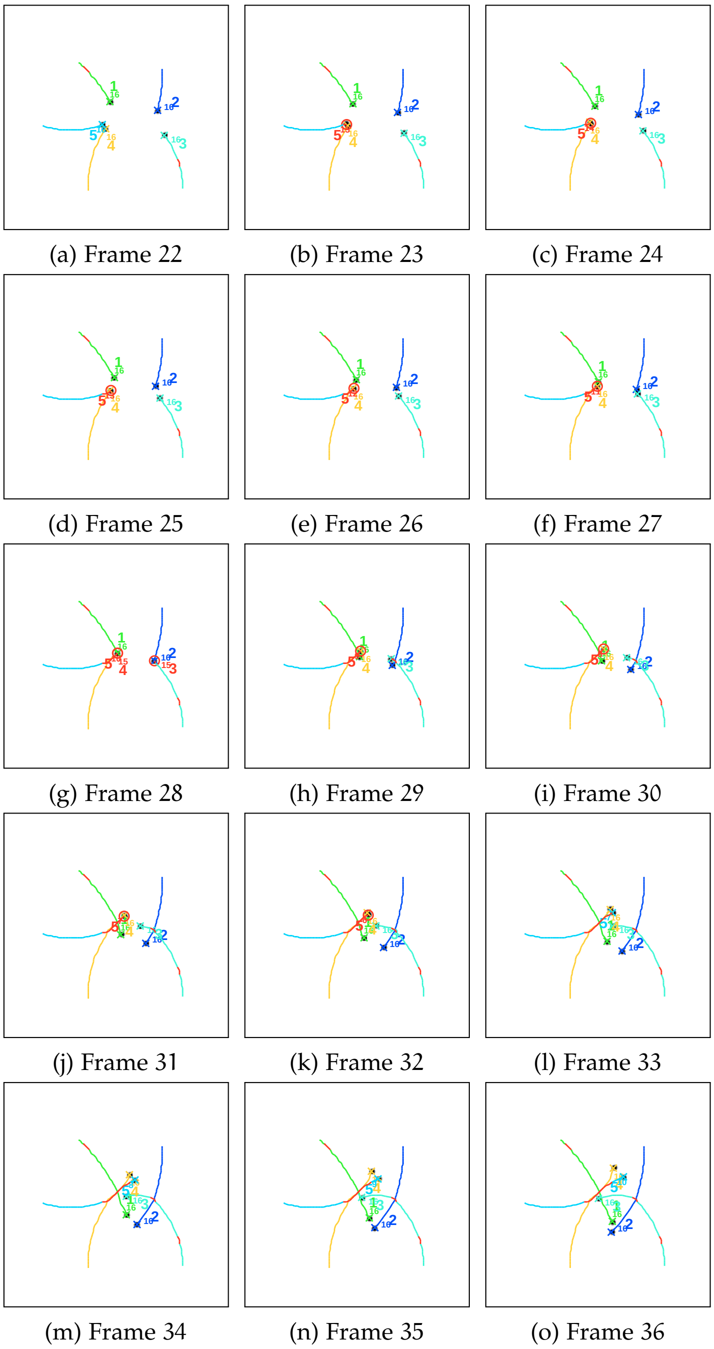
Figure 66: Background occlusion or misdetection.

minimum value. At frame number 50 this feature is removed from the list of tracked features and its trajectory is no longer updated. This situation is depicted in Figure 68.

Since the hierarchical matching approach performed as expected using synthetic images, the methodology will be tested in non synthetic sequences where more challenging tracking conditions were observed.

6.6 EXPERIMENTAL RESULTS USING NON SYNTHETIC SEQUENCES

After observing the results obtained from the preliminary tests with synthetic sequences, the algorithm is tested in several non synthetic image sequences. These sequences are the same used in Section 5.7 but now several features are selected at the initialisation stage.



(a) Frame 22 (b) Frame 23 (c) Frame 24

(d) Frame 25 (e) Frame 26 (f) Frame 27

(g) Frame 28 (h) Frame 29 (i) Frame 30

(j) Frame 31 (k) Frame 32 (l) Frame 33

(m) Frame 34 (n) Frame 35 (o) Frame 36

Figure 67: Features merging and splitting.

6.6.1 *Multi feature tracking using microscopic sequences*

The first microscopic sequence shows a bacteria called *Serratia Marcescens* moving over the surface. In this sequence, the bacteria denotes a bright intensity against a dark background. To illustrate the effectiveness of the approach, several bacterias are picked up and the trajectories are estimated,

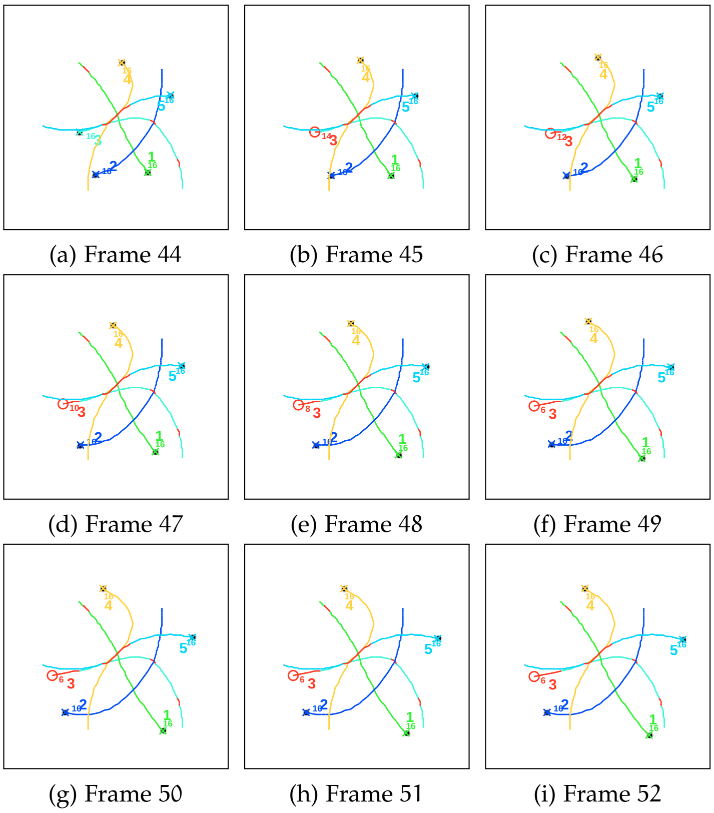

(a) Frame 44
(b) Frame 45
(c) Frame 46
(d) Frame 47
(e) Frame 48
(f) Frame 49
(g) Frame 50
(h) Frame 51
(i) Frame 52

Figure 68: Permanent occlusion.

Figure 69. The second test sequence is an organism called *Synechococcus*, swimming in a medium. This sequence is more complex than the previous one: there are more similar features in the scene, the background is not uniform and the difference of grey levels between feature and background is lower (Figure 70). The third sequence is a *Mycoplasma* bacteria gliding. Several initial bacterias are selected in frame number 1 and their estimated trajectory is presented until frame number 500. The evolution of their trajectories is depicted in Figure 71.

These sequences where chosen because they present additional difficulties to the tracking process. There are features crossing each other, there are permanent occlusions, the trajectories described by the features are complex, the boundaries of the features are not defined clearly and their grey level changes over time. In these sequences all features are labelled with a unique identification number and their confidence level is also shown.

To implement a performance measure of the proposed approach, experimental results are compared with manually generated feature positions that represent the centre of mass of the features. Using these ground-truth positions, a tracking error measurement could be developed calculating the average and standard deviation of an Euclidean distance between experimental and ground-truth positions as presented in Table 7. Due to the large number of analysed frames of the sequence *Mycoplasma*, the ground-truth

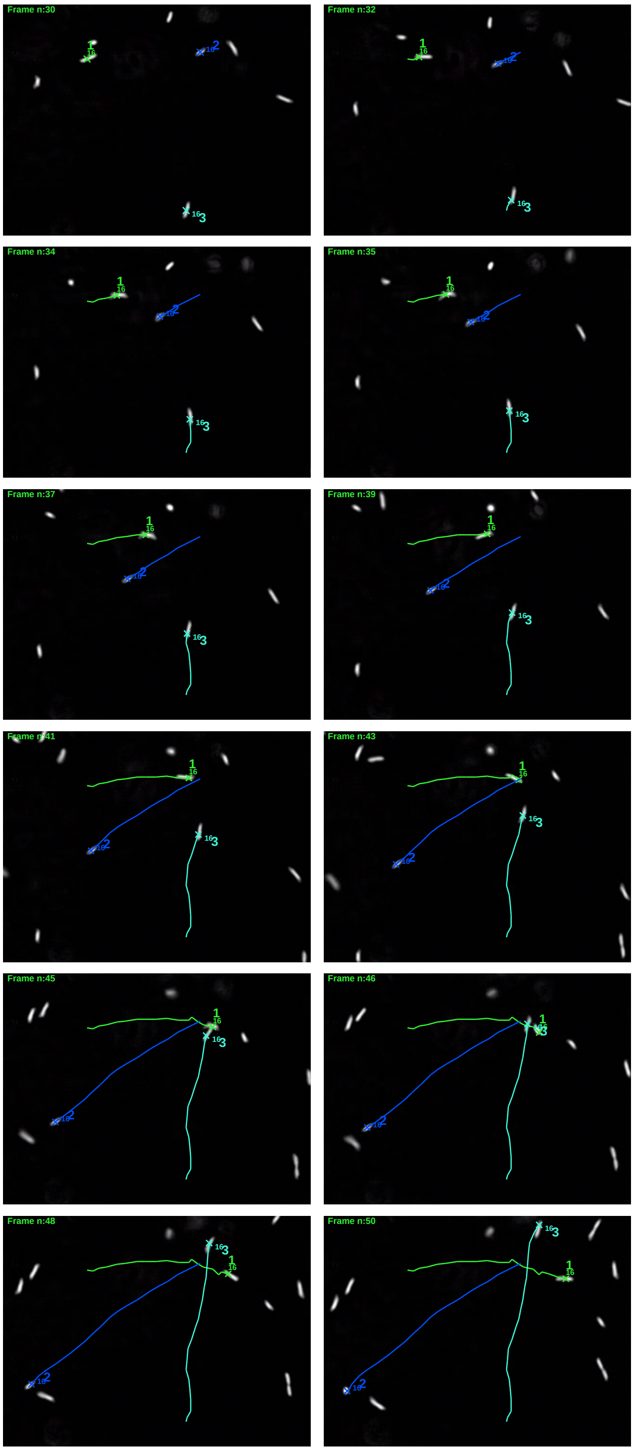


Figure 69: Estimated trajectories between frames 30 and 50.

positions were extracted just for features number 1 and 2. However, the obtained results for these two features could be extended to the remaining features (3 and 4) since they are similar in shape, size and behaviour.

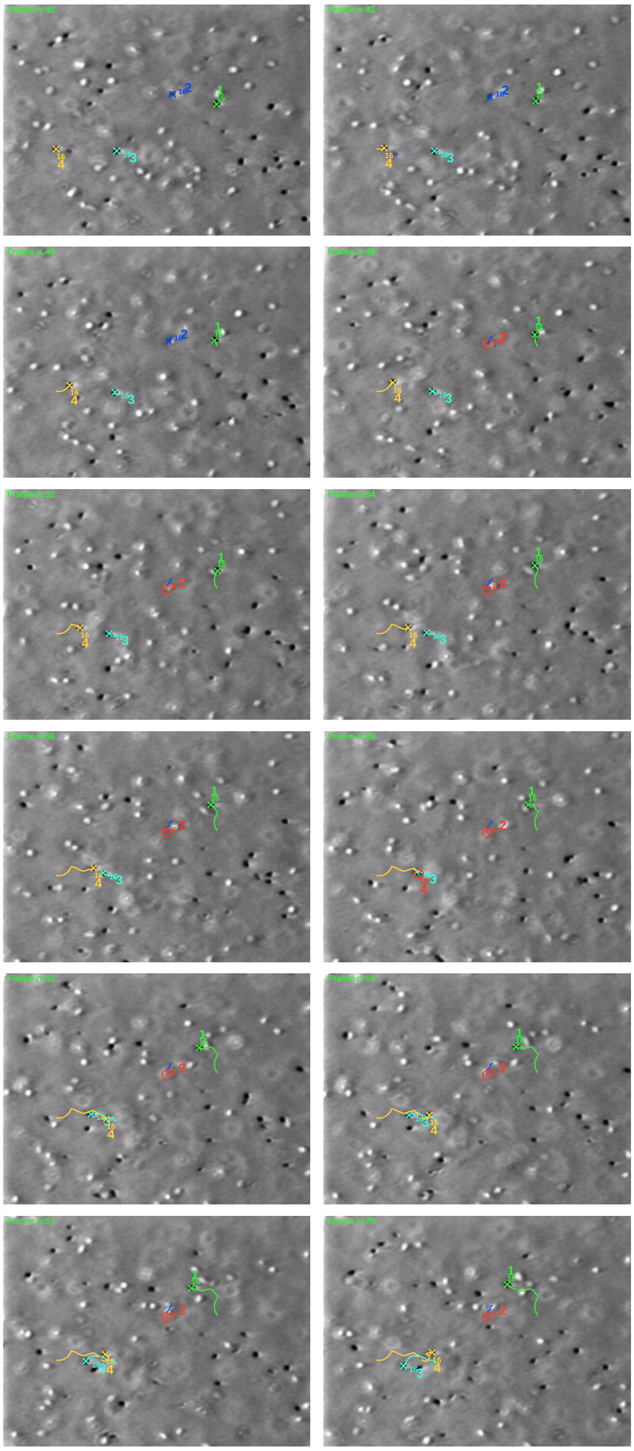


Figure 70: Estimated trajectories between frames 40 and 70.

Results presented in Table 7 denote coherency with the results presented previously in Table 5 since the fuzzy matching strategy remains the same. As explained earlier, the first sequence generates higher error values since

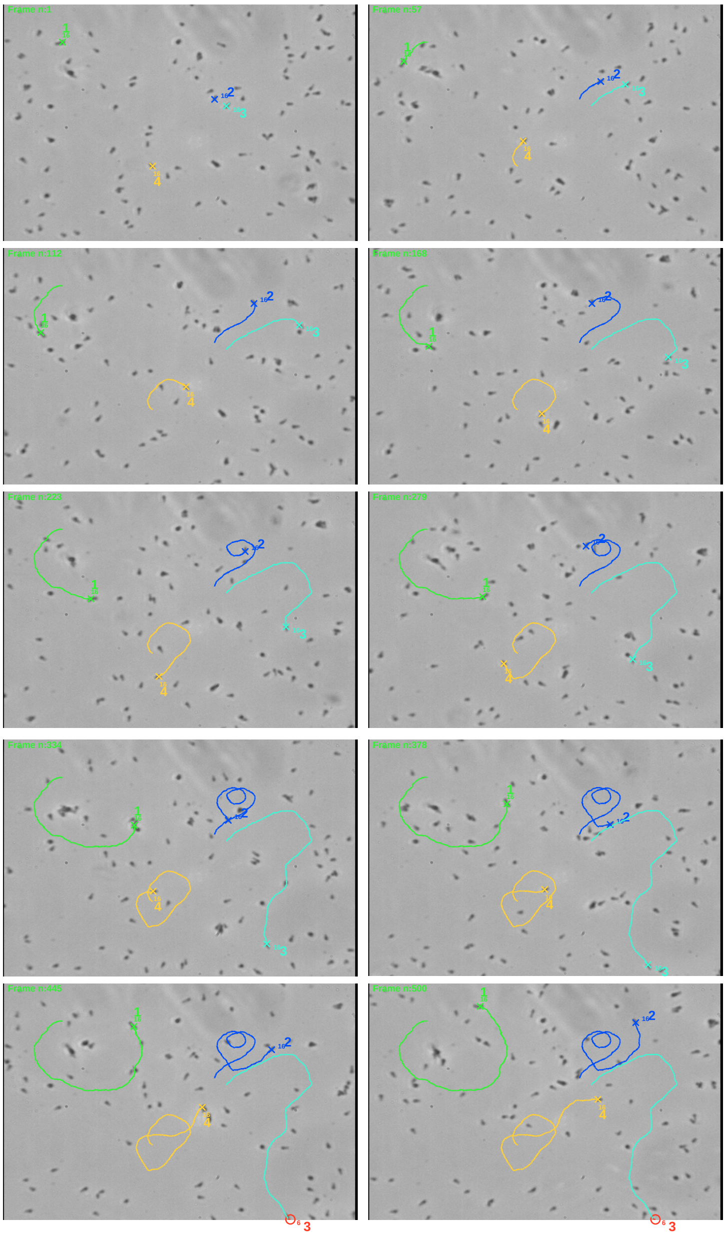


Figure 71: Estimated trajectories between frames 1 and 500.

the moving bacterias denote the tracked feature (high grey level intensity) in all their surface and these bacterias present an elongated shape, leading the method to provide experimental positions not close enough to the centre of mass.

The tracked features in the second sequence denote well defined boundaries and intensity over time and, therefore, the approach provides best results.

The selected features to be tracked in the third sequence denote fuzzy boundaries causing difficulties to the extraction of precise ground-truth positions and, moreover, their grey level intensity is not constant over the

Table 7: Average distance error and standard deviation.

SEQUENCE	FEATURE	MEAN	STANDARD DEVIATION
Serratia Marcescens	1	2.7752	1.5534
	2	1.0908	0.8948
	3	2.7443	1.5628
Synechococcus	1	0.8051	0.6809
	2	1.0598	0.7572
	3	0.8462	0.6782
	4	0.8032	0.6583
Mycoplasma	1	1.9122	1.0972
	2	1.7377	0.9613

entire sequence, leading the method to provide results not close enough from the ground-truth positions.

Despite of these errors between experimental and ground-truth positions, the tracking task is performed correctly for all selected features presented in each sequence, as illustrated in Figures 69, 70 and 71.

6.6.2 *Multi feature tracking using macroscopic sequences*

After applying the proposed approach in microscopic sequences, two macroscopic sequences are used. The first sequence is a test case scenario used in CAVIAR project. The second was obtained from PETS datasets.

The first tested sequence comprises 150 frames starting at frame number 420. At this starting frame, three persons are present in the scene: one person will walk straight on the corridor, one will go inside a store and the other will wait outside. The selected feature to track is the dark intensity of their hairs. The result of the proposed methodology is illustrated in Figure 72. In this sequence there are no occlusions or misdetections and therefore the confidence level of each feature remains at its maximum.

In the second macroscopic sequence, the white pixels denoted by a mark of concrete placed in the floor and the white paper sheets carried by a person are selected to be tracked. In this sequence both features suffer occlusions and their confidence level changes accordingly. The result of the proposed methodology between frames 50 and 150 is illustrated in Figure 73. Despite of the static behaviour of the concrete mark it suffers occlusions by the pedestrians in such a way that its confidence level decreases below the predefined minimum value of 6. When it happens this feature is no longer tracked by the approach. To ensure that features that don't provide information to the method are still tracked by the approach for a long period of time, the difference between the maximum and minimum values of the confidence level must be higher.

Once again, experimental positions of each feature in the sequences computed by the proposed methodology are compared with a ground-truth position generated manually. The evolution of the Euclidean distance be-

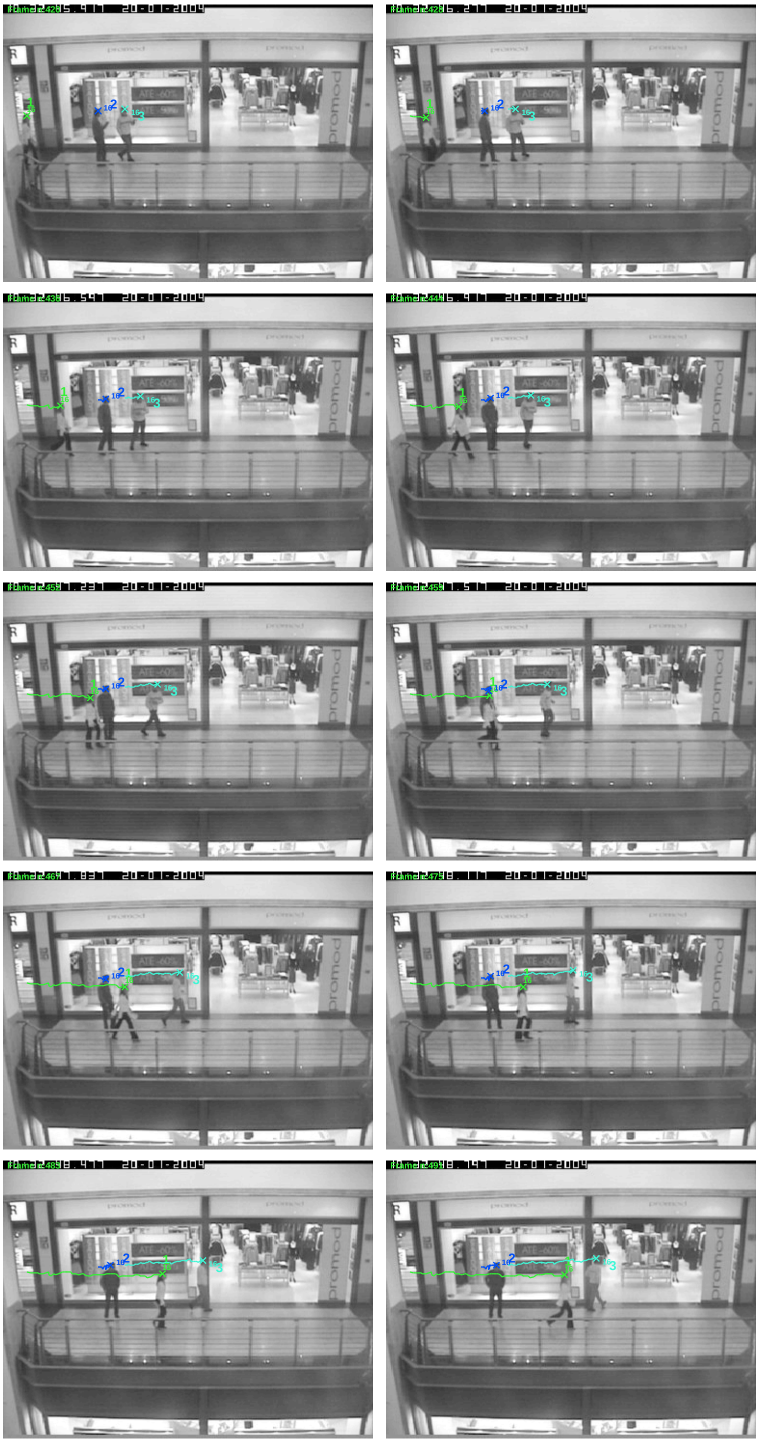


Figure 72: Estimated trajectories between frames 420 and 570 (first frames).

tween them, for all frames of the sequence, can be observed in Figures 74 and 75, respectively, for CAVIAR and PETS sequences.

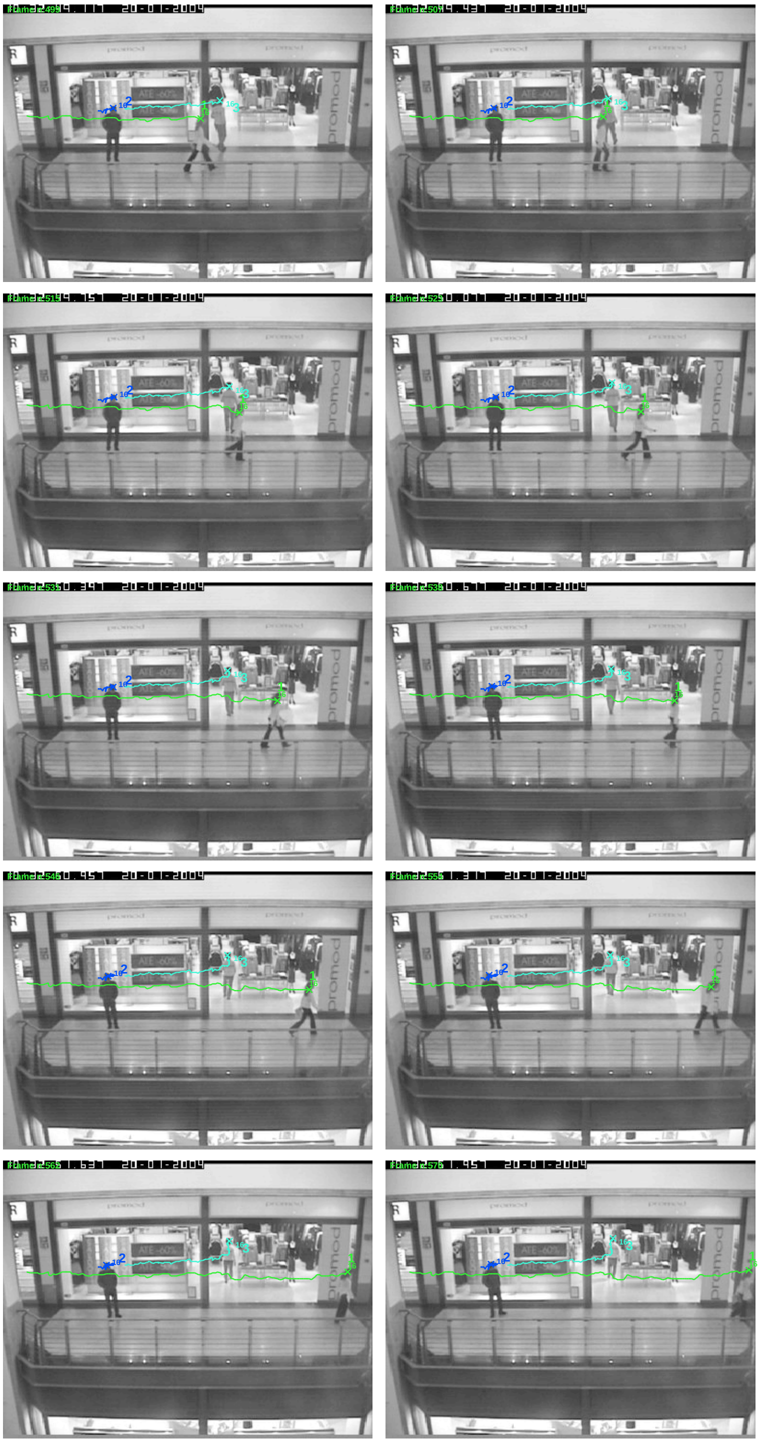


Figure 72: Estimated trajectories between frames 420 and 570 (last frames).

The average and standard deviation of the Euclidean distance between experimental and ground-truth positions for all tracked features in these two sequences are presented in Table 8.

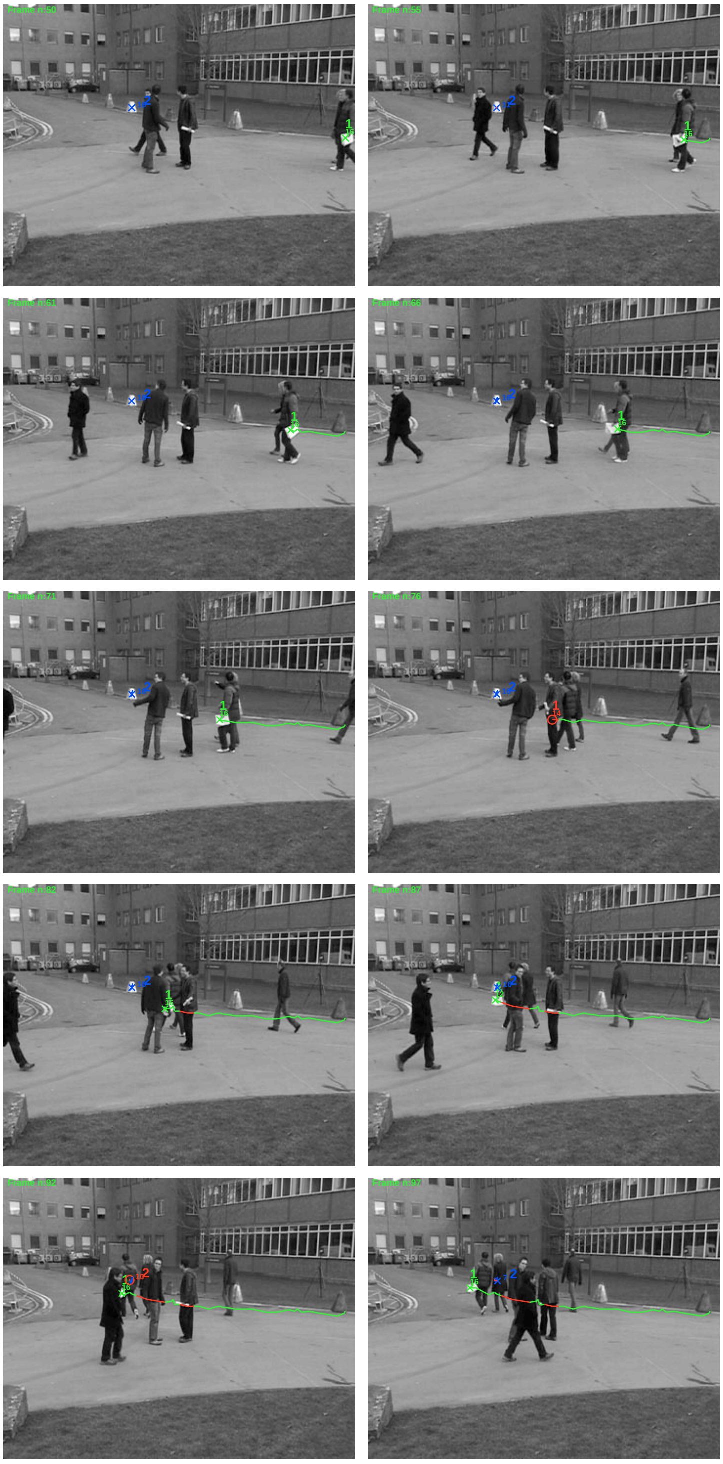


Figure 73: Estimated trajectories between frames 50 and 150 (first frames).

After result analysis, some remarks could be performed. In the sequence obtained from CAVIAR, the distance error for feature number 1, depicted in Figure 74a, assumes higher values at the end of the sequence. This issue

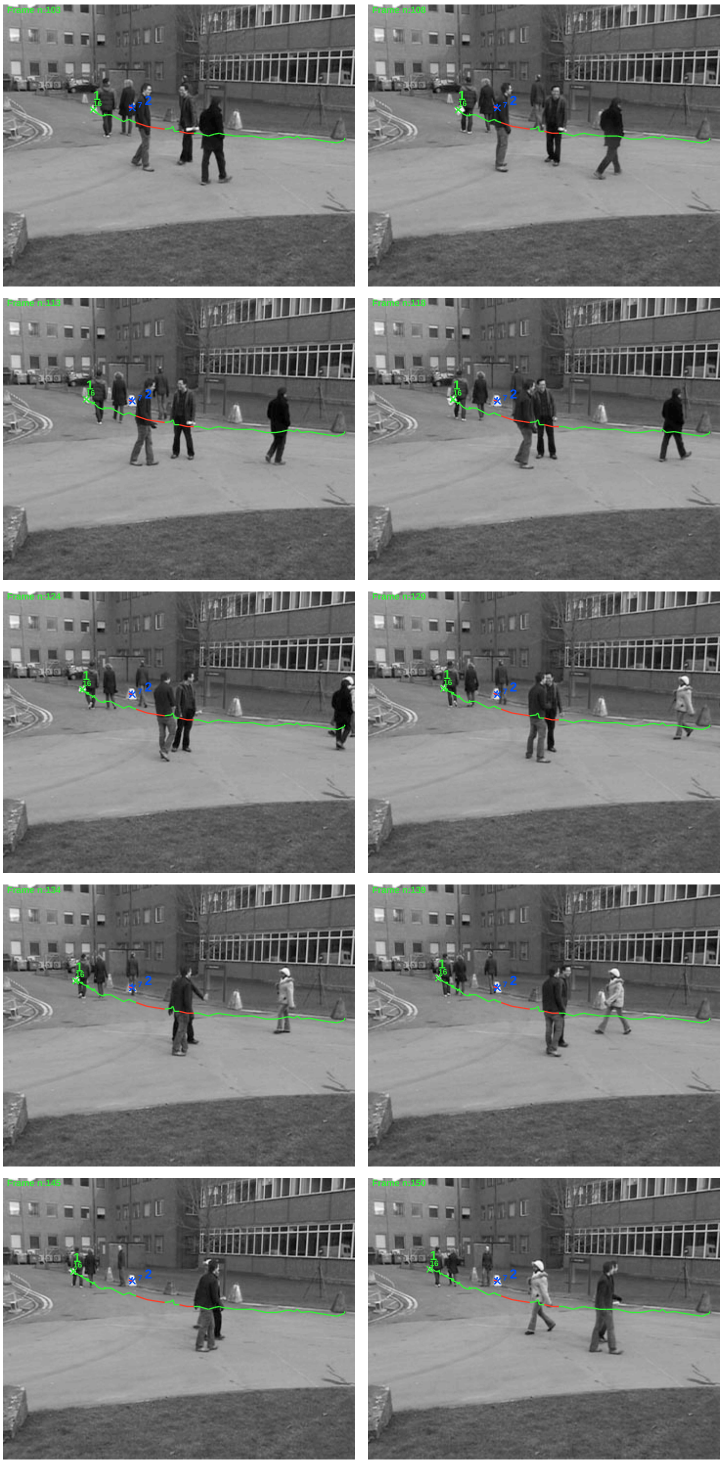
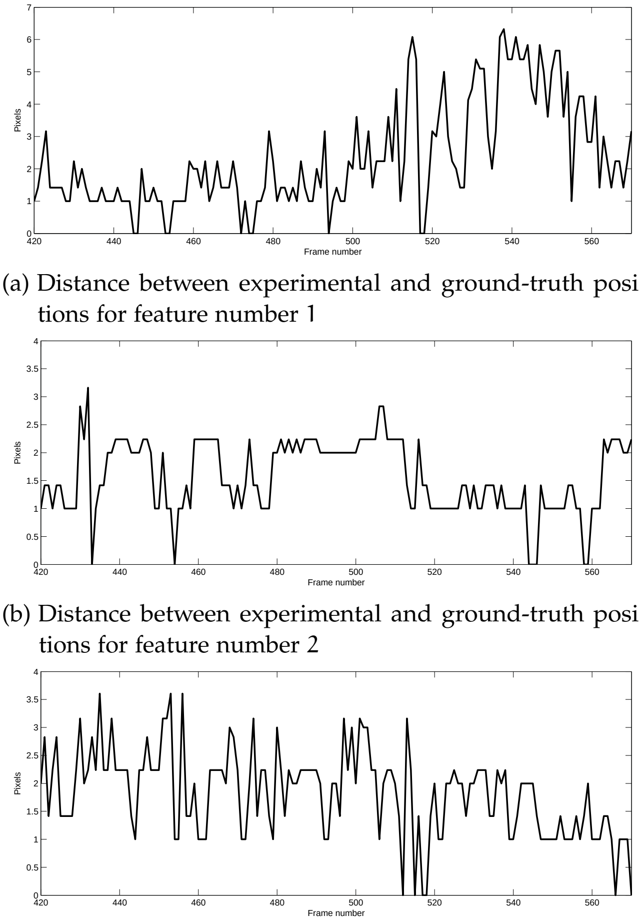


Figure 73: Estimated trajectories between frames 50 and 150 (last frames).

is explained since the long hair of the woman becomes more exposed in the image plane and also it changes in intensity. These two factors lead the method to provide positions not equal with the ground-truth. The error



(a) Distance between experimental and ground-truth positions for feature number 1

(b) Distance between experimental and ground-truth positions for feature number 2

(c) Distance between experimental and ground-truth positions for feature number 3

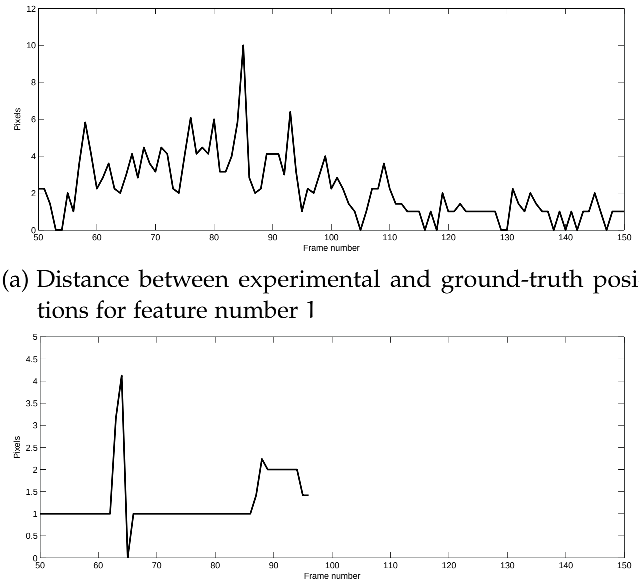
Figure 74: Distance between experimental and ground-truth positions for all features.

Table 8: Average distance error and standard deviation.

SEQUENCE	FEATURE	MEAN	STANDARD DEVIATION
CAVIAR	1	2.3208	1.6405
	2	1.5577	0.6460
	3	1.8297	0.7723
PETS	1	2.2577	1.7355
	2	1.2716	0.6595

values observed in the other two features are considerable lower and don't denote a clear variation.

In feature number 1, related with the sequence obtained from PETS, higher error values at the beginning of the sequence could be explained by the large size, intensity changes and the swinging pattern of the feature observed in the image plane. Another issue is related with occlusion situations that increase the errors between the ground-truth positions since the method



(a) Distance between experimental and ground-truth positions for feature number 1

(b) Distance between experimental and ground-truth positions for feature number 2

Figure 75: Distance between experimental and ground-truth positions for all features.

doesn't have any new information about the location of the feature. An interesting analysis concerning feature number 2 could be presented since it is a landmark and the camera is static, its position should remain constant along time, but observing Figure 75a the error between experimental and ground-truth positions changes over the sequence. Moreover, after frame number 96 the approach stops to track this feature. Error variations can be explained by partial occlusions made by pedestrians passing across the concrete mark that introduce an apparent displacement. After frame 96 the confidence degree of this feature becomes lower than the established minimum value and it is removed from the tracking process because this feature suffers several consecutive total occlusions leading a fast decreasing on its confidence degree.

Despite these errors, feature tracking is performed successfully for all features in both sequences. The maximum error values are not significant if compared with the dimensions of the frames of the sequence and with the size of the selected features.

6.7 DISCUSSION

In this chapter a hierarchical matching approach for multiple feature tracking is introduced. This approach is an extension of the proposed methodology presented in Chapter 5. It adopts similar initial assumptions but the methodology is changed to achieve a hierarchical scheme to perform multiple feature tracking.

In order to implement this hierarchical scheme, each selected feature is associated with a confidence degree that represents the amount of information provided by such feature to the methodology. This information is related with the visibility of each feature, i. e., when a feature is visible to the method its confidence degree increases, otherwise, when the feature is occluded, its confidence degree decreases. To avoid features with low information to be assigned first than features that had been present in the scene for a large number of frames, a hierarchical scheme must be implemented.

In this hierarchical scheme, features denoting higher confidence degree are assigned first than features with lower confidence degree. Moreover, it is possible to discern two occlusion situations: inter feature and background occlusion and different procedures are performed. When two or more tracked features became merged in the image plane the approach is capable to detect that one or more feature are occluded by other features. Since the approach continues to see one region with similar attributes to those features, the confidence degree of occluded features suffers a slow decay. In opposition, when a background occlusion occurs, no region is present in the image and the confidence degree of such features have a higher decay. To avoid tracking features whose information about them is too low, features with a confidence degree lower than a minimum are deleted from the list of tracked features.

The hierarchical multiple feature tracking approach was firstly applied in synthetic sequences and after that, microscopic and macroscopic sequences were also used to assess its performance. A performance measurement using the difference between experimental and ground-truth positions is used and the observed errors are not significant and do not affect the tracking process.

Since the approach must to track several features, computational time and resources increase proportionally with the number of selected features.

Part III

CONCLUSION

If I have seen further,
it is by standing on the shoulders of giants.
— Isaac Newton

7 CONCLUSION AND FUTURE WORK

Despite of all the work developed over the last decades, the problem of feature tracking is still challenging and plays an important role within the research field of image processing. The difficulty of feature tracking derives from its inner steps of segmentation and matching. These two steps had been also subject of intensive study and finding optimal segmentation and matching algorithms is not trivial, and is indeed a very difficult task.

With the introduction of fuzzy logic concepts by Zadeh, human based reasoning could be incorporated in digital systems trough the definition of membership functions, construction of fuzzy sets and rules inference engines that translate linguistic variables and common sense to a computational domain.

The concepts of fuzzy logic were rapidly adopted in control purposes specially in cases where the model of the control system couldn't be mathematically defined with accuracy and a large amount of uncertainty was present. The observed results in control applications encouraged researchers to develop and apply fuzzy logic concepts in a large number of areas. Recently, several well known image processing tasks find their replica using fuzzy concepts.

Developing a new framework using fuzzy logic concepts to implement a tracking system sounds very attractive due to its singularity. Furthermore, using fuzzy concepts, human reasoning and uncertainty could be incorporated in the method. The first step to achieve a fuzzy tracking system would be the development of a fuzzy segmentation algorithm to perform feature detection. Assuming that features denote a significant contrast against background, an automatic fuzzy segmentation algorithm was introduced. This automatic fuzzy approach was an improvement of an existing method based on a fuzziness measure to find the optimal threshold value in a grey image histogram. After comparing the experimental results with other well known methods, the results were very satisfactory. This way we have been able to think forward to the fuzzy correspondence process.

At this stage, new developments have emerged and a novel fuzzy framework was constructed. This framework uses fuzzy concepts as fuzzy sets and inference rules on fuzzy sets to perform both detection and matching tasks. Then, a novel fuzzy tracking system was introduced.

In this work a compromise between functionality and simplicity was taken in consideration. With the definition of three membership functions related with kinematic and non kinematic properties of the feature and with a construction of an inference engine with three rules the fuzzy tracking approach is implemented. Experimental tests with synthetic and non synthetic sequences were performed to evaluate the performance of the approach. Also, a extension of this approach to deal with multiple feature tracking is presented. It was developed a hierarchical approach since features that provided more information to the method are assigned first than features with low confidence degree. Therefore, candidate points in the next frame are first matched with features denoting high confidence degree.

A comparison between other classic tracking approaches can also be carried out. Since the concept of this methodology is considerably different from the classic ones it could be difficult to perform an accurate comparison. Foremost, there is no clear separation between the tasks of feature detection and matching as in classical methods and, the decision block is performed by a fuzzy inference engine that incorporates human reasoning. In classic tracking approaches two distinctive levels of processing were performed to implement the feature detection and the correspondence tasks. In the proposed approach these two steps are performed in a single level through the definition of several fuzzy sets and the uncertainty inherent to each task is considered in the process. Concerning classical approaches based on template matching such as cross correlation, the presented method allows more significant changes in feature shape and intensity without decreasing its performance. Moreover, methods of tracking moving features based in frame differencing, optical flow or even background subtraction are not suitable to deal with static features. When a feature stops, these strategies are not capable to perform the feature detection. The proposed approach can deal with this particular situation without losing its performance. Since the proposed approach is based on human reasoning described by a set of rules, the minimisation of a cost function, the formulation as a graph theoretic problem and the use of statistical data association techniques to solve the correspondence problem are avoided.

The presented hierarchical matching approach for multiple feature tracking has provided encouraging results, however this results lead us to further work with intend to improve robustness, introduce new capabilities and achieve computational efficiency over different image sequences.

Further work is intended on the validation of the methodology through an exhaustive comparative study with other state of the art techniques.

It will be relevant the introduction and performance evaluation of different distinctive feature properties such as colour or other descriptors, in order to construct suitable fuzzy sets and introduce new rules in the inference engine. This way, it's expectable an increasing of robustness in the estimated trajectories.

The introduction of an automatic capability to deal with entries and exits of features over the sequence is also an important issue to be studied and implemented.

It's also a purpose the construction of an expert tracking system capable to satisfy real time requirements, meaning the algorithm should be optimised to increase computational speed and the selection of an operating system with real time kernel should be ensured.

According to the nature of moving features, the use of different motion models to characterise the feature movement model should be considered since it could bring better results.

An alternative to the use of Kalman filter, such as particle filters should be considered and comparative performance tests must to be carried out.

It's intended the extrapolation of this point tracking framework to other feature representations such as shape or kernel based representations.

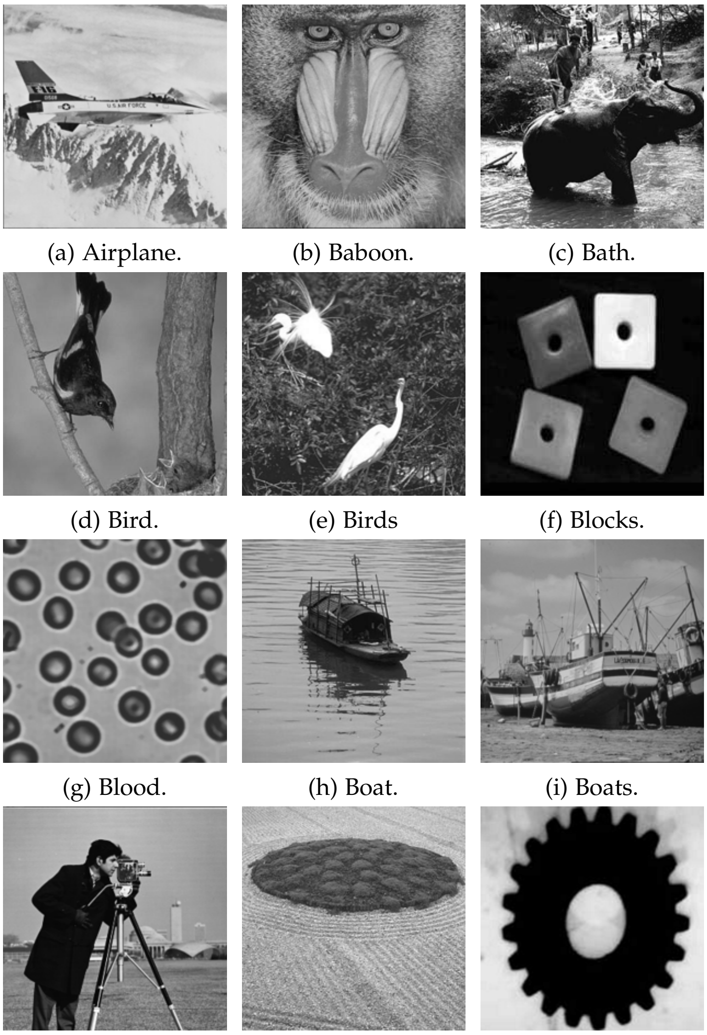
And finally, in order to provide the ability to perform behaviour analysis, an higher level fuzzy model intended to deal with feature behavioural patterns should be developed.

Part IV

APPENDIX

A DATABASE IMAGES FOR SEGMENTATION ALGORITHM

In this Appendix, the images used in the fuzzy segmentation algorithm are presented.



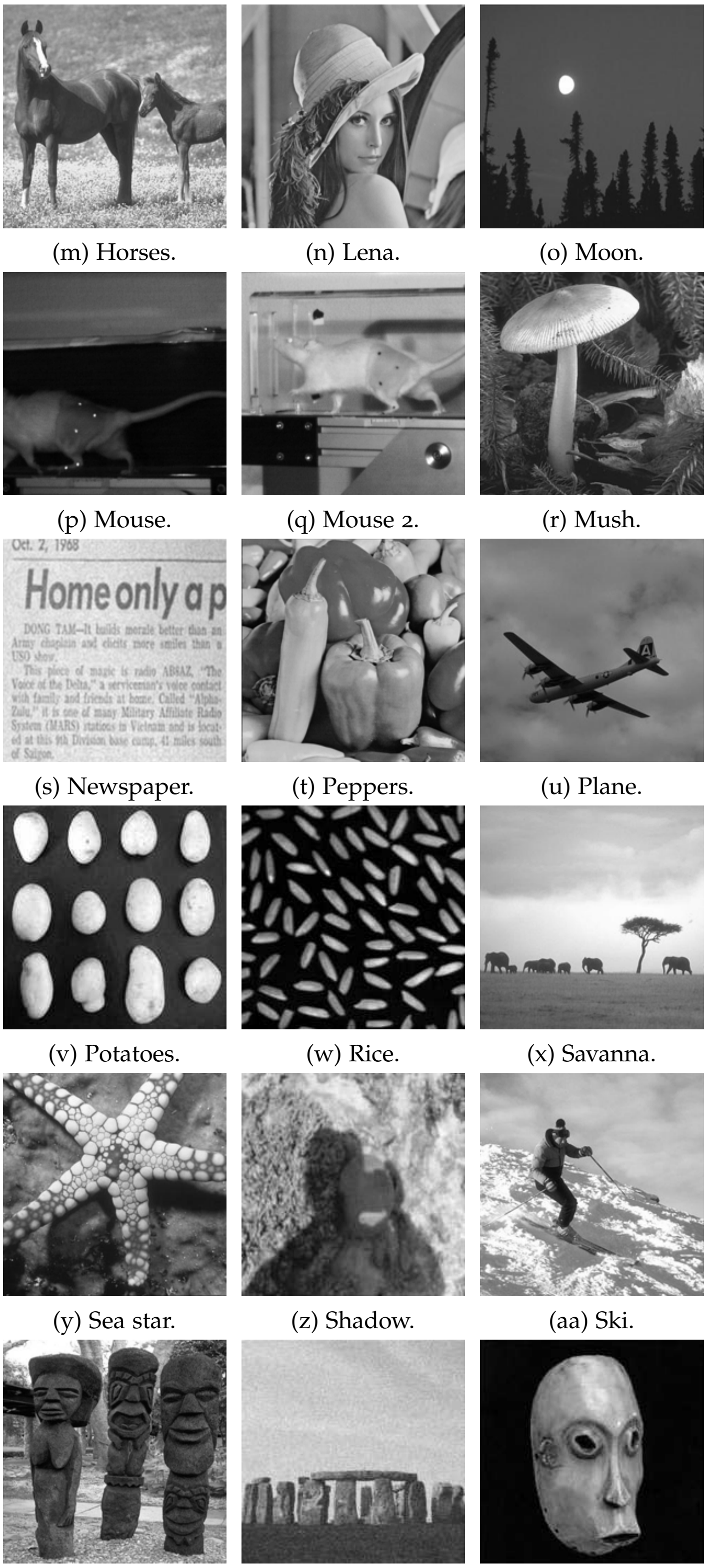
(a) Airplane. (b) Baboon. (c) Bath.

(d) Bird. (e) Birds (f) Blocks.

(g) Blood. (h) Boat. (i) Boats.

(j) Cameraman. (k) Field. (l) Gearwheel.

Figure 76: Database images for fuzzy segmentation algorithm (first images).



Oct. 2, 1968

Home only a p

DONG TAM—It builds morale better than an Army chaplain and elicits more smiles than a USO show.

This piece of magic is radio AB8AZ, "The Voice of the Delta," a serviceman's voice contact with family and friends at home. Called "Alpha-Zulu," it is one of many Military Affiliate Radio System (MARS) stations in Vietnam and is located at this 9th Division base camp, 41 miles south of Saigon.

(m) Horses. (n) Lena. (o) Moon.

(p) Mouse. (q) Mouse 2. (r) Mush.

(s) Newspaper. (t) Peppers. (u) Plane.

(v) Potatoes. (w) Rice. (x) Savanna.

(y) Sea star. (z) Shadow. (aa) Ski.

(ab) Statues. (ac) Stones. (ad) Zimba.

Figure 76: Database images for fuzzy segmentation algorithm (last images).

BIBLIOGRAPHY

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994. ISSN 0162-8828. doi: 10.1109/34.295913. (Cited on page 20.)

[2] Y. S. Akgul, C. Kambhamettu, and M. Stone. Automatic extraction and tracking of the tongue contours. *IEEE Transactions on Medical Imaging*, 18(10):1035–1045, Oct. 1999. doi: 10.1109/42.811315. (Cited on pages 9, 19, and 28.)

[3] S. Amarnag, R. S. Kumaran, and J. N. Gowdy. Real time eye tracking for human computer interfaces. In *Proc. International Conference on Multimedia and Expo ICME '03*, volume 3, pages 557–560, 2003. (Cited on page 9.)

[4] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. P. Papanikolopoulos. A vision-based approach to collision prediction at traffic intersections. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):416–423, Dec. 2005. doi: 10.1109/TITS.2005.858786. (Cited on pages 9 and 16.)

[5] S. Avidan. Support Vector Tracking. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, volume 1, pages 184–191, 2001. doi: 10.1109/CVPR.2001.990474. (Cited on page 27.)

[6] S. Avidan. Support Vector Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.53. (Cited on page 27.)

[7] M. Baklouti, S. Couvet, and E. Monacelli. Intelligent Camera Interface (ICI): A Challenging HMI for Disabled People. In *Proc. First International Conference on Advances in Computer-Human Interaction*, pages 21–25, 10–15 Feb. 2008. doi: 10.1109/ACHI.2008.49. (Cited on page 9.)

[8] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994. (Cited on page 15.)

[9] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466, 1995. ISSN 0360-0300. doi: 10.1145/212094.212141. (Cited on page 15.)

[10] M. Bertalmío, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):733–737, 2000. ISSN 0162-8828. doi: 10.1109/34.865191. (Cited on page 28.)

[11] S. Beucher and C. Lantuejoul. Use of Watersheds in Contour Detection. In *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France.*, September 1979. (Cited on page 21.)

[12] A. Bevilacqua and S. Vaccari. Real time detection of stopped vehicles in traffic scenes. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS 2007*, pages 266–270, 5–7 Sept. 2007. doi: 10.1109/AVSS.2007.4425321. (Cited on page 9.)

[13] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981. ISBN 0306406713. (Cited on page 47.)

[14] J.C. Bezdek, L.O. Hall, and L.P. Clarke. Review of MR image segmentation techniques using pattern recognition. *Med Phys*, 20(4):1033–1048, July 1993. (Cited on page 47.)

[15] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Proc. Workshop on Motion and Video Computing*, pages 169–174, 5–6 Dec. 2002. doi: 10.1109/MOTION.2002.1182230. (Cited on pages 8, 24, and 29.)

[16] Michael J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. ISSN 1077-3142. doi: 10.1006/cviu.1996.0006. (Cited on page 15.)

[17] Claude R. Brice and Claude L. Fennema. Scene Analysis Using Regions. Technical Report 17, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Apr 1970. (Cited on page 21.)

[18] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851. (Cited on page 17.)

[19] J. Chamorro-Martínez, E. Galán-Perales, D. Sánchez, and J. Soto-Hidalgo. Modelling Coarseness in Texture Images by Means of Fuzzy Sets. In Bogdan Gabrys, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4252 of *Lecture Notes in Computer Science*, pages 355–362. Springer Verlag, 2006. doi: 10.1007/11893004_46. (Cited on page 49.)

[20] Guanrong Chen, Jianrong Wang, and L. S. Shieh. Interval Kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1): 250–259, 1997. ISSN 0018-9251. doi: 10.1109/7.570759. (Cited on page 49.)

[21] Guanrong Chen, Qingxian Xie, and Leang S. Shieh. Fuzzy Kalman filtering. *Information Sciences*, 109(1-4):197–209, August 1998. ISSN 0020-0255. (Cited on page 49.)

[22] Jong-Ho Choi, Kang-Ho Lee, Kuk-Chan Cha, Jun-Sik Kwon, Dong-Wook Kim, and Ho-Keun Song. Vehicle Tracking using Template Matching based on Feature Points. In *Proc. IEEE International Conference on Information Reuse and Integration*, pages 573–577, 16–18 Sept. 2006. doi: 10.1109/IRI.2006.252477. (Cited on page 9.)

[23] Keh Shih Chuang, Hong Long Tzeng, Sharon Chen, Jay Wu, and Tzong Jer Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1): 9–15, 2006. ISSN 0895-6111. doi: 10.1016/j.compmedimag.2005.10.001. (Cited on page 48.)

[24] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003. doi: 10.1109/TPAMI.2003.1195991. (Cited on page 27.)

[25] P. Couto, M. Pagola, H. Bustince, E. Barrenechea, and P. Melo-Pinto. Uncertainty in multilevel image thresholding using Atanassov's intuitionistic fuzzy sets. In *Proc. (IEEE World Congress on Computational Intelligence). IEEE International Conference on Fuzzy Systems FUZZ-IEEE 2008*, pages 330–335, 2008. doi: 10.1109/FUZZY.2008.4630386. (Cited on page 47.)

[26] P. Couto, N.V. Lopes, H. Bustince, and P. Melo-Pinto. Fuzzy dynamic model for feature tracking. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1 –8, Barcelona, Spain, 18-23 July 2010. doi: 10.1109/FUZZY.2010.5583979. (Cited on page 93.)

[27] Ingemar J. Cox. A Review of Statistical Data Association Techniques for Motion Correspondence. *International Journal of Computer Vision*, 10: 53–66, 1993. (Cited on page 25.)

[28] Aldo de Luca and Settimo Termini. A Definition of a Nonprobabilistic Entropy in the Setting of Fuzzy Sets Theory. *Information and Control*, 20(4):301–312, 1972. (Cited on pages 43 and 44.)

[29] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. (Cited on page 18.)

[30] Jianping Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10(10): 1454–1466, 2001. ISSN 1057-7149. doi: 10.1109/83.951532. (Cited on page 21.)

[31] P. Fieguth and D. Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 21–27, 1997. doi: 10.1109/CVPR.1997.609292. (Cited on page 27.)

[32] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI 97)*, pages 175–181, 1997. (Cited on page 15.)

[33] B. Galvin, B. Mccane, K. Novins, D. Mason, and S. Mills. Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. In *British Machine Vision Conference*, pages 195–204, 1998. (Cited on page 15.)

[34] Jesús García, José M. Molina, Juan A. Besada, and Javier I. Portillo. A multitarget tracking video system based on fuzzy and neuro-fuzzy techniques. *EURASIP Journal on Applied Signal Processing*, 2005:2341–2358, 2005. ISSN 1110-8657. doi: 10.1155/ASP.2005.2341. (Cited on page 50.)

[35] D. M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999. (Cited on page 12.)

[36] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993. (Cited on pages 18, 22, and 60.)

[37] N. Guil, J. Villalba, and E. L. Zapata. A fast Hough transform for segment detection. *IEEE Transactions on Image Processing*, 4(11):1541–1548, 1995. ISSN 1057-7149. doi: 10.1109/83.469935. (Cited on page 19.)

[38] A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior, and Y. Tian. Smart surveillance: applications, technologies and implications. In *Proc. Joint Conference of the Fourth International Conference on and the Fourth Pacific Rim Conference on Multimedia Information, Communications and Signal Processing*, volume 2, pages 1133–1138, 15–18 Dec. 2003. doi: 10.1109/ICICS.2003.1292637. (Cited on page 8.)

[39] R. M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, 1979. ISSN 0018-9219. (Cited on page 22.)

[40] R. M. Haralick. Ridges and Valleys on Digital Images. *Computer Vision, Graphics and Image Processing*, 22(10):28–38, 1983. (Cited on page 21.)

[41] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, Reading, 1992. (Cited on page 11.)

[42] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973. ISSN 0018-9472. doi: 10.1109/TSMC.1973.4309314. (Cited on page 22.)

[43] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 222, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-8344-9. (Cited on page 15.)

[44] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Aug 2000. ISSN 0162-8828. doi: 10.1109/34.868683. (Cited on pages 15 and 28.)

[45] J. B. Hiley, A. H. Redekopp, and R. Fazel-Rezai. A Low Cost Human Computer Interface based on Eye Tracking. In *Proc. 28th Annual International Conference of the IEEE Engineering in Medicine and Biology*

Society EMBS '06, pages 3226–3229, Aug. 30 2006–Sept. 3 2006. doi: 10.1109/IEMBS.2006.260774. (Cited on page 9.)

[46] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. (Cited on pages 13 and 15.)

[47] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3): 334–352, Aug. 2004. ISSN 1094-6977. doi: 10.1109/TSMCC.2004.829274. (Cited on pages 8 and 10.)

[48] Liang Kai Huang and Mao Jiun J. Wang. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition*, 28(1):41–51, 1995. ISSN 0031-3203. doi: 10.1016/0031-3203(94)E0043-K. (Cited on page 48.)

[49] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. Fourth International Conference on Computer Vision*, pages 93–101, 1993. doi: 10.1109/ICCV.1993.378231. (Cited on page 27.)

[50] Michael Isard and Andrew Blake. CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29:5–28, 1998. (Cited on pages 19 and 29.)

[51] Alejandro Jaimes and Nicu Sebe. Multimodal human-computer interaction: A survey. *Computer Vision and Image Understanding*, 108(1-2): 116–134, 2007. ISSN 1077-3142. doi: 10.1016/j.cviu.2006.10.019. Special Issue on Vision for Human-Computer Interaction. (Cited on page 9.)

[52] C. V. Jawahar, P. K. Biswas, and A. K. Ray. Investigations on fuzzy thresholding based on fuzzy clustering. *Pattern Recognition*, 30(10): 1605–1613, 1997. ISSN 0031-3203. doi: 10.1016/S0031-3203(97)00004-6. (Cited on pages 47 and 63.)

[53] R. K. Justice and E. M. Stokely. 3-D segmentation of MR brain images using seeded region growing. In *Proc. 18th Annual International Conference of the IEEE Bridging Disciplines for Biomedicine Engineering in Medicine and Biology Society*, volume 3, pages 1083–1084 vol.3, 1996. doi: 10.1109/IEMBS.1996.652719. (Cited on page 20.)

[54] Janusz Kacprzyk. *Multistage Fuzzy Control: A Prescriptive Approach*. John Wiley & Sons, Inc., New York, NY, USA, 1997. ISBN 047196347X. (Cited on pages 37 and 39.)

[55] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, June 2000. doi: 10.1109/6979.880968. (Cited on pages 9 and 15.)

[56] Jinman Kang, I. Cohen, and G. Medioni. Object reacquisition using invariant appearance model. In *Pattern Recognition, 2004. ICPR 2004.*

Proceedings of the 17th International Conference on, volume 4, pages 759–762 Vol.4, Aug. 2004. doi: 10.1109/ICPR.2004.1333883. (Cited on page 28.)

[57] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4): 321–331, 1988. doi: 10.1007/BF00133570. (Cited on page 19.)

[58] A. Kaufmann. *Introduction to the theory of fuzzy subsets*, volume I. Academic Press, New York, 1975. (Cited on pages 44 and 57.)

[59] Won Kim and Ju-Jang Lee. Visual tracking using Snake based on target's contour information. In *Proc. IEEE International Symposium on Industrial Electronics ISIE 2001*, volume 1, pages 176–180 vol.1, 2001. doi: 10.1109/ISIE.2001.931777. (Cited on page 19.)

[60] Won Kim, Sun-Gi Hong, and Ju-Jang Lee. An active contour model using image flow for tracking a moving object. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS '99*, volume 1, pages 216–221 vol.1, 1999. doi: 10.1109/IROS.1999.813007. (Cited on pages 19, 28, and 29.)

[61] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19(1):41–47, 1986. (Cited on page 20.)

[62] J. Lalk. Intelligent adaptation of Kalman filters using fuzzy logic. In *Proc. Third IEEE Conference on Fuzzy Systems IEEE World Congress on Computational Intelligence*, pages 744–749 vol.2, 1994. doi: 10.1109/FUZZY.1994.343829. (Cited on page 49.)

[63] H. Lazoff. Target tracking using fuzzy logic association. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2457–2460 vol.4, 1998. doi: 10.1109/ICASSP.1998.681648. (Cited on page 50.)

[64] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123. Canadian Image Processing and Pattern Recognition Society, 1995. (Cited on page 25.)

[65] Fang Li, M. K. Leung, M. Mangalvedhekar, and M. Balakrishnan. Automated video surveillance and alarm system. In *Proc. International Conference on Machine Learning and Cybernetics*, volume 2, pages 1156–1161, 12–15 July 2008. doi: 10.1109/ICMLC.2008.4620578. (Cited on page 8.)

[66] Dae-Woon Lim, Sung-Hoon Choi, and Joon-Suk Jun. Automated detection of all kinds of violations at a street intersection using real time individual vehicle tracking. In *Proc. Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 126–129, 7–9 April 2002. doi: 10.1109/IAI.2002.999903. (Cited on pages 9, 15, and 24.)

[67] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *Proc. Fourth IEEE Workshop on Applications of Computer Vision WACV '98*, pages 8–14, 1998. doi: 10.1109/ACV.1998.732851. (Cited on page 12.)

[68] Meng Liu, Chengdong Wu, and Yunzhou Zhang. A review of Traffic Visual Tracking technology. In *Proc. International Conference on Audio, Language and Image Processing ICALIP 2008*, pages 1016–1020, 7–9 July 2008. doi: 10.1109/ICALIP.2008.4590198. (Cited on page 9.)

[69] X. Liu, P. H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance AVSS 2005*, pages 306–311, 15–16 Sept. 2005. doi: 10.1109/AVSS.2005.1577286. (Cited on page 8.)

[70] N. V. Lopes, P. Couto, H. Bustince, and P. Melo-Pinto. Fuzzy Dynamic Matching Approach for Multi-Feature Tracking. In *EUROFUSE 2009: Eurofuse Workshop on Preference Modelling and Decision Analysis*, pages 245–250, Pamplona, Spain, 16-18 September 2009. ISBN 978-8497692427. (Cited on page 93.)

[71] N. V. Lopes, P. A. Mogadouro do Couto, H. Bustince, and P. Melo-Pinto. Automatic Histogram Threshold Using Fuzzy Measures. *IEEE Transactions on Image Processing*, 19(1):199–204, 2010. ISSN 1057-7149. doi: 10.1109/TIP.2009.2032349. (Cited on pages 49 and 61.)

[72] Nuno Vieira Lopes, Humberto Bustince, Vítor Filipe, and Pedro Melo-Pinto. Fuzziness Measure Approach to Automatic Histogram Threshold. In João Tavares and Natal Jorge, editors, *Computational Vision and Medical Image Processing: VipIMAGE 2007*, pages 295–299. Taylor and Francis Group, 2007. (Cited on pages 49 and 61.)

[73] A. M. Lopez, F. Lumbreras, J. Serrat, and J. J. Villanueva. Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):327–335, 1999. ISSN 0162-8828. doi: 10.1109/34.761263. (Cited on page 21.)

[74] C. Lopez-Molina, H. Bustince, J. Fernández, E. Barrenechea, P. Couto, and B. Baets. A t-Norm Based Approach to Edge Detection. In *IWANN '09: Proceedings of the 10th International Work-Conference on Artificial Neural Networks*, pages 302–309, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02477-1. doi: 10.1007/978-3-642-02478-8_38. (Cited on page 47.)

[75] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981. (Cited on page 15.)

[76] S. Malassiotis and M. G. Strintzis. Tracking the left ventricle in echocardiographic images by learning heart dynamics. *IEEE Transactions on Medical Imaging*, 18(3):282–290, March 1999. doi: 10.1109/42.764905. (Cited on pages 9 and 19.)

[77] D. Marr and E.C. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London*, B-207(1167):187–217, 1980. (Cited on page 17.)

[78] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. (Cited on page 61.)

[79] Fernando Matía, Agustín Jiménez, Basil M. Al-Hadithi, Diego Rodríguez-Losada, and Ramón Galán. The fuzzy Kalman filter: State estimation using possibilistic techniques. *Fuzzy Sets and Systems*, 157 (16):2145–2170, 2006. (Cited on page 50.)

[80] A. H. Mazinan. Specific persons surveillance using satellite technique. In *Proc. International Conference on Intelligent and Advanced Systems ICIAS 2007*, pages 474–478, 25–28 Nov. 2007. doi: 10.1109/ICIAS.2007.4658433. (Cited on page 8.)

[81] Stephen J. McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. Tracking Groups of People. *Computer Vision and Image Understanding: CVIU*, 80(1):42–56, 2000. (Cited on page 15.)

[82] I. Mikic, S. Krucinski, and J. D. Thomas. Segmentation and tracking in echocardiographic sequences: active contours guided by optical flow estimates. *IEEE Transactions on Medical Imaging*, 17(2):274–284, April 1998. doi: 10.1109/42.700739. (Cited on pages 9, 19, and 29.)

[83] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006. ISSN 1077-3142. doi: 10.1016/j.cviu.2006.08.002. (Cited on page 7.)

[84] J. M. Molina, J. García, O. Pérez, J. Carbó, A. Berlanga, and J. I. Portillo. Applying Fuzzy Logic in Video Surveillance Systems. *Mathware and Soft Computing*, 12(3):185–198, 2005. (Cited on page 50.)

[85] Rafael Muñoz-Salinas, Eugenio Aguirre, and Miguel García-Silvente. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25(6):995–1007, 2007. ISSN 0262-8856. doi: 10.1016/j.imavis.2006.07.012. (Cited on pages 8 and 27.)

[86] Wayne Chelliah Naidoo and Jules Raymond Tapamo. Soccer video analysis by ball, player and referee tracking. In *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries*, pages 51–60, Republic of South Africa, 2006. South African Institute for Computer Scientists and Information Technologists. ISBN 1-59593-567-3. doi: 10.1145/1216262.1216268. (Cited on page 10.)

[87] Hoang Thanh Nguyen and B. Bhanu. Multi-object tracking in non-stationary video using bacterial foraging swarms. In *Proc. 16th IEEE Int Image Processing (ICIP) Conf*, pages 877–880, 2009. doi: 10.1109/ICIP.2009.5414276. (Cited on page 27.)

[88] Nobuyuki Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66,

Jan. 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076. (Cited on pages 19, 31, and 63.)

[89] N. Owens, C. Harris, and C. Stennett. Hawk-eye tennis system. In *Proc. International Conference on Visual Information Engineering VIE 2003*, pages 182–185, 2003. (Cited on page 10.)

[90] Nikhil R. Pal and James C. Bezdek. Measuring Fuzzy Uncertainty. In *IEEE Transactions on Fuzzy Systems*, volume 2, 1994. (Cited on pages 43, 45, and 56.)

[91] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung. A Method for Vehicle Count in the Presence of Multiple-Vehicle Occlusions in Traffic Images. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):441–459, 2007. ISSN 1524-9050. doi: 10.1109/TITS.2007.902647. (Cited on page 9.)

[92] Theo Pavlidisn. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, MD, 1982. (Cited on page 21.)

[93] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677–695, Jul 1997. ISSN 0162-8828. doi: 10.1109/34.598226. (Cited on pages 9 and 12.)

[94] A. Perry and D. G. Lowe. Segmentation of textured images. In *Proc. CVPR '89. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 319–325, 1989. doi: 10.1109/CVPR.1989.37867. (Cited on page 22.)

[95] G. S. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 260–265, 1998. doi: 10.1109/CVPR.1998.698618. (Cited on pages 10 and 12.)

[96] F. Porikli, O. Tuzel, and P. Meer. Covariance Tracking using Model Update Based on Lie Algebra. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, pages 728–735, 2006. doi: 10.1109/CVPR.2006.94. (Cited on page 26.)

[97] William K. Pratt. *Digital Image Processing*. John Wiley & Sons, Inc, third edition edition, 2001. ISBN 0-471-22132-5. (Cited on page 21.)

[98] J.M.S. Prewitt. Object Enhancement and Extraction. In B. S. Liplin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 75–149. Academic Press, New York, 1970. (Cited on page 16.)

[99] J. Princen, J. Illingworth, and J. Kittler. A hierarchical approach to line extraction. In *Proc. CVPR '89. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 92–97, 1989. doi: 10.1109/CVPR.1989.37833. (Cited on page 19.)

[100] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005. ISSN 1057-7149. doi: 10.1109/TIP.2004.838698. (Cited on page 12.)

[101] T. Randen and J. H. Husoy. Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999. ISSN 0162-8828. doi: 10.1109/34.761261. (Cited on page 22.)

[102] K. Rangarajan and M. Shah. Establishing motion correspondence. In *Proc. CVPR '91. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 103–108, 1991. doi: 10.1109/CVPR.1991.139669. (Cited on page 24.)

[103] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In *Proceedings of International Conference on recent Advances in Mechatronics, ICRAM'95*, pages 193–199. UNESCO Chair on Mechatronics, 1995. (Cited on page 15.)

[104] T. Ridler and S. Calvard. Picture Thresholding Using an Iterative Selection Method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8):630–632, Aug. 1978. ISSN 0018-9472. doi: 10.1109/TSMC.1978.4310039. (Cited on page 19.)

[105] Lawrence G. Roberts. Machine Perception of Three-Dimensional Solids. In J. T. Tippet, editor, *Optical and Electro-Optical Information Processing*, volume chapter 9, pages 159–197. MIT Press, Cambridge, MA, 1965. (Cited on page 16.)

[106] A. Rosenfeld and P. de la Torre. Histogram Concavity Analysis as an Aid in Threshold Selection. *SMC*, 13(3):231–235, March 1983. (Cited on page 19.)

[107] P. Rosin. Thresholding for change detection. In *Proc. Sixth International Conference on Computer Vision*, pages 274–279, 1998. doi: 10.1109/ICCV.1998.710730. (Cited on page 12.)

[108] Timothy J. Ross. *Fuzzy Logic with Engineering Applications, 2nd Edition*. John Wiley & Sons, Ltd, University of New Mexico, USA, June 2004. (Cited on pages 36 and 38.)

[109] Supatra Sahaphong and Nualsawat Hiransakolwong. Unsupervised Image Segmentation Using Automated Fuzzy c-Means. *7th IEEE International Conference on Computer and Information Technology, CIT*, pages 690–694, Oct. 2007. doi: 10.1109/CIT.2007.144. (Cited on page 48.)

[110] V. Salari and I. K. Sethi. Feature Point Correspondence in the Presence of Occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):87–91, 1990. ISSN 0162-8828. doi: 10.1109/34.41387. (Cited on page 23.)

[111] Ishwar K. Sethi and Ramesh Jain. Finding Trajectories of Feature Points in a Monocular Image Sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:56–73, January 1987. ISSN 0162-8828. doi: 10.1109/TPAMI.1987.4767872. (Cited on pages 23 and 24.)

[112] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, January 2004. (Cited on pages 20 and 65.)

[113] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *Proc. Ninth IEEE International Conference on Computer Vision*, pages 110–115 vol.1, 2003. doi: 10.1109/ICCV.2003.1238321. (Cited on page 24.)

[114] Mubarak Shah. Motion-Based Recognition: A Survey. *Image and Vision Computing*, 13:129–155, 1995. (Cited on page 12.)

[115] Yu Shi, P. Raniga, and I. Mohamed. A Smart Camera for Multimodal Human Computer Interaction. In *Proc. IEEE Tenth International Symposium on Consumer Electronics ISCE '06*, pages 1–6, 2006. doi: 10.1109/ISCE.2006.1689443. (Cited on page 9.)

[116] Yu Shi, R. Taib, and S. Lichman. GestureCam: A Smart Camera for Gesture Recognition and Gesture-Controlled Web Navigation. In *Proc. 9th International Conference on Control, Automation, Robotics and Vision ICARCV '06*, pages 1–6, 5–8 Dec. 2006. doi: 10.1109/ICARCV.2006.345267. (Cited on page 9.)

[117] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999. doi: 10.1109/CVPR.1999.784637. (Cited on page 16.)

[118] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, 2000. ISSN 0162-8828. doi: 10.1109/34.868677. (Cited on page 16.)

[119] S. L. Stoev and W. Strasser. Extracting regions of interest applying a local watershed transformation. In *Proc. Visualization 2000*, pages 21–28, 2000. doi: 10.1109/VISUAL.2000.885672. (Cited on page 21.)

[120] Richard Szeliski and James Coughlan. Spline-Based Image Registration. *Int. J. Comput. Vision*, 22(3):199–218, 1997. ISSN 0920-5691. doi: 10.1023/A:1007996332012. (Cited on page 15.)

[121] Chin-Wang Tao, Wiley E. Thompson, and J.S. Taur. A Fuzzy If-Then Approach to Edge Detection. *Second IEEE International Conference on Fuzzy Systems*, 2:1356–1360, 1993. (Cited on page 47.)

[122] Orlando J. Tobias and Rui Seara. Image Segmentation by Histogram Thresholding Using Fuzzy Sets. In *IEEE Transactions on Image Processing*, volume 11, 2002. (Cited on pages 49, 55, 56, and 59.)

[123] Orlando J. Tobias, Rui Seara, and Flávio A. P. Soares. Automatic image segmentation using fuzzy sets. In *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, volume 2, pages 921–924, 1996. (Cited on pages 49 and 55.)

[124] Vincent Torre and Tomaso A. Poggio. On Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):147–163, 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767769. (Cited on page 17.)

[125] E. Trucco and K. Plakas. Video Tracking: A Concise Survey. *IEEE Journal of Oceanic Engineering*, 31(2):520–529, 2006. ISSN 0364-9059. doi: 10.1109/JOE.2004.839933. (Cited on pages 7 and 10.)

[126] Shung-Tsang Tseng and Kai-Tai Song. Real-time image tracking for traffic monitoring. In *Proc. IEEE 5th International Conference on Intelligent Transportation Systems*, pages 1–6, 2002. doi: 10.1109/ITSC.2002.1041179. (Cited on pages 9, 19, and 24.)

[127] Y. Tuduki, K. Murase, M. Izumida, H. Miki, K. Kikuchi, K. Murakami, and J. Ikezoe. Automated seeded region growing algorithm for extraction of cerebral blood vessels from magnetic resonance angiographic data. In *Proc. 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 3, pages 1756–1759 vol.3, 2000. doi: 10.1109/IEMBS.2000.900424. (Cited on page 21.)

[128] M. Valera and S. A. Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings -Vision, Image and Signal Processing*, 152 (2):192–204, 8 April 2005. doi: 10.1049/ip-vis:20041147. (Cited on page 8.)

[129] Cor J. Veenman, Marcel J.T. Reinders, and Eric Backer. Resolving Motion Correspondence for Densely Moving Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001. ISSN 0162-8828. doi: 10.1109/34.899946. (Cited on pages 23 and 24.)

[130] J. F. Vega-Riveros and K. Jabbour. Review of motion analysis techniques. *IEE Proceedings I Communications, Speech and Vision*, 136(6):397–404, Dec 1989. ISSN 0956-3776. (Cited on page 15.)

[131] Luc Vincent and Pierre Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991. (Cited on page 21.)

[132] J. R. Wang and N. Parameswaran. Survey of sports video analysis: research issues and applications. In *VIP '05: Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 87–90, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc. ISBN 1-920682-18-X. (Cited on page 10.)

[133] Jessica JunLin Wang and Sameer Singh. Video analysis of human dynamics - a survey. *Real-Time Imaging*, 9(5):321–346, 2003. ISSN 1077-2014. doi: 10.1016/j.rti.2003.08.001. (Cited on pages 8 and 12.)

[134] Liang Wang. Abnormal Walking Gait Analysis Using Silhouette-Masked Flow Histograms. In *Proc. 18th International Conference on Pattern Recognition ICPR 2006*, volume 3, pages 473–476, 2006. doi: 10.1109/ICPR.2006.199. (Cited on pages 9 and 15.)

[135] J. S. Weszka. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, 7(2):259–265, 1978. (Cited on page 20.)

[136] Joan S. Weszka and Azriel Rosenfeld. Threshold Evaluation Techniques. *IEEE Transactions on Systems, Man and Cybernetics*, 8(8):622–629, 1978. ISSN 0018-9472. doi: 10.1109/TSMC.1978.4310038. (Cited on page 20.)

[137] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, 1997. (Cited on page 15.)

[138] Yuan Xiong, Yi Zhang, and Kan Jiang. Tracking athletes in diving videos based on pose learning. In *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 538–541, 20–22 Oct. 2004. doi: 10.1109/ISIMP.2004.1434120. (Cited on page 10.)

[139] Ronald R. Yager. On measures of fuzziness and negation. Part I: Membership in unit interval. *International Journal of General Systems*, 5: 221–229, 1979. (Cited on page 44.)

[140] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006. ISSN 0360-0300. doi: 10.1145/1177352.1177355. (Cited on pages 8, 10, and 29.)

[141] Lotfi A. Zadeh. Fuzzy Sets. *Information Control*, 8:338–353, 1965. (Cited on pages 34, 36, and 39.)

[142] Lotfi A. Zadeh. Knowledge representation in fuzzy logic. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):89–100, 1989. ISSN 1041-4347. doi: 10.1109/69.43406. (Cited on page 34.)

[143] Ying Zhang, Shengjin Wang, and Xiaoqing Ding. Traffic Flow Surveillance System for Urban Intersections. In *Proc. International Conference on MultiMedia and Information Technology MMIT '08*, pages 405–408, 30–31 Dec. 2008. doi: 10.1109/MMIT.2008.44. (Cited on pages 9, 16, 24, and 29.)

[144] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation–A survey. *Biomedical Signal Processing and Control*, 3(1): 1–18, 2008. ISSN 1746-8094. doi: 10.1016/j.bspc.2007.09.001. (Cited on page 10.)