UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Nicolás Amézquita Gómez

A Probabilistic Integrated Object Recognition and Tracking Framework for Video Sequences

PhD Thesis

A thesis co-directed by:

Francesc Serratosa i Casanelles[*]
René Alquézar Mancho[†]

[*]Department of Computer Engineering and Mathematics (URV)
[†]Institute of Robotics and Industrial Informatics (CSIC-UPC)

UNIVERSITAT ROVIRA I VIRGILI

Tarragona-Spain
2009

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

ACKNOWLEDGEMENTS

I'm deeply grateful to Dr René Alquézar Mancho, not only for accepting me as one of his PhD students in the start but also for his continuous support of my academic activities. I really appreciate his patience and collaboration. I would like to express sincere thanks to Dr Francesc Serratosa, who allows me to follow with this work and becomes advisor of the present thesis.

I would like to thank also to Dr Alberto Sanfeliu and IRI people for their help and to the professor Juan Mauricio Salamanca, who introduced me into the Digital Signal Processing group (DSP) and then object recognition field in my university career final work, in 2004.

Financial support was provided by the Universitat Rovira i Virgili (URV) through a predoctoral research grant and was partially supported by Consolider Ingenio 2010, project CSD2007-00018, and by the CICYT project DPI 2007-61452.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

CONTENTS

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Notation

x	Horizontal coordinate of the pixel.
y	Vertical coordinate of the pixel.
$I^t(x, y)$	2D color image in time t
c	Class to which an objects belongs
N	Number of objects of interest
$N+1$	Class reserved for the background
$p^t(x,y)$	Probabilities of the pixel (x,y) at time t
$p_c^t(x,y)$	Probability that the pixel (x,y) at time t belongs to class c
$p_c^0(x,y)$	Initial probability that the pixel (x,y) belongs to class c
$T_c^t(x,y)$	A posteriori tracking image in time t for class c
$\hat{T}_c^t(x, y)$	A priori prediction tracking image in time t for class c
$T_c^0(x,y)$	Initialization tracking image for class c
$T_c^{t-1}(x,y)$	Previous tracking image in time t for class c
$T_{N+1}^t(x, y)$	A tracking image for the background in time t
$Q_c^t(x,y)$	Estimated probability that the pixel (x,y) belongs to class c given by the static recognition module
r	Recognition function for static class probabilities
f	Update function for the dynamic class probabilities
d	Decision function for tracking
$v(x,y)$	Feature vector for pixel (x,y)
$\vec{m}_c^{t-1}$	Previous movement vector
$\vec{m}_c^t$	Movement weighted average vector for object c
B	Special class $c=N+1$ reserved for the background,
k	Identifier of an object (non-background) class between 1 and N,
H_k	Histogram for class k
C_k	Total count of the histogram for class k
$P(v \mid k)$	Conditional probability of features v given class k
α_c^t	Parameter that weights the influence of the previous probabilities in the dynamic recognition module for class c
A_c^t , A_c^{t-1} , A_c^{t-2}	Areas of the object c
C_c^t, C_c^{t-1} , C_c^{t-2}	Mass centers of object c
$\hat{C}_c^t$	A priori estimate of mass center of object of class c in time t.
$\hat{A}_c^t$	A priori estimate of area of the object of class c in time t.
r_c^{t-1} , r_c^{t-2}	Estimate of the object radius
d_c	Estimate of the displacement of the object
$d_c^{\max}$	Maximum displacement
s_c	Scale change ratio
O_c^t , O_c^{t-1} , O_c^{t-2}	Occlusion flag for the object c
ε , δ	Constant parameters related to uncertainties in movements prediction
ε_c^t , δ_c^t	Adaptive parameters related to uncertainties in movements prediction
$MC(T_c^t)$	Mass center of the 1-valued region in $T_c^t(x,y)$

ISBN:978-84-693-3387-7/DL:T.1001-2010

$MC(\hat{T}_c^t)$	Mass center of the 1-valued region in $\hat{T}_c^t(x,y)$
β	Positive parameter between 0 and 1 for computation of the movement weigthed average vector
$\vec{v}_c^t$	Current movement vector for object *c*
N (p, q)	Normalized central moments of order two
wmc(*o*,*f*)	Weighted mass center
ns(*o*,*f*)	Number of spots classified as object o in frame *f*
p(*o*|*s*)	A-posteriori class probability of object *o* for spot *s*
a(*s*)	Area of *s*.
mc(*s*)	Mass center of *s*.
SO	Spatial overlap
GT_k	Ground truth in frame *k*
ST_k	System track in frame *k*
Dist	Euclidean distance

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

List of Tables

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

List of Figures

ISBN:978-84-693-3387-7/DL:T.1001-2010

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

SUMMARY

Recognition and tracking of multiple objects in video sequences is one of the main challenges in computer vision that currently deserves a lot of attention from researchers. Almost all the reported approaches are very application-dependent and there is a lack of a general methodology for dynamic object recognition and tracking that can be instantiated in particular cases. In this thesis, the work is oriented towards the definition and development of such a methodology which integrates object recognition and tracking from a general perspective using a probabilistic framework called PIORT (probabilistic integrated object recognition and tracking framework). It include some modules for which a variety of techniques and methods can be applied. Some of them are well-known but other methods have been designed, implemented and tested during the development of this thesis.

The first step in the proposed framework is a static recognition module that provides class probabilities for each pixel of the image from a set of local features. These probabilities are updated dynamically and supplied to a tracking decision module capable of handling full and partial occlusions. The two specific methods presented use RGB colour features and differ in the classifier implemented: one is a Bayesian method based on maximum likelihood and the other one is based on a neural network. The experimental results obtained have shown that, on one hand, the neural net based approach performs similarly and sometimes better than the Bayesian approach when they are integrated within the tracking framework. And on the other hand, our PIORT methods have achieved better results when compared to other published tracking methods. All these methods have been tested experimentally in several test video sequences taken with still and moving cameras and including full and partial occlusions of the tracked object in indoor and outdoor scenarios in a variety of cases with different levels of task complexity. This allowed the evaluation of the general methodology and the alternative methods that compose these modules.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

RESUMEN

El reconocimiento y seguimiento de múltiples objetos en secuencias de vídeo es uno de los principales desafios en visión por ordenador que actualmente merece mucha atención de los investigadores. Casi todos los enfoques reportados son muy dependientes de la aplicación y hay carencia de una metodología general para el reconocimiento y seguimiento dinámico de objetos, que pueda ser instanciada en casos particulares. En esta tesis, el trabajo está orientado hacia la definición y desarrollo de tal metodología, la cual integra reconocimiento y seguimiento de objetos desde una perspectiva general usando un marco probabilístico de trabajo llamado PIORT (Probabilistic Integrated Object Recognition and Tracking). Este incluye algunos módulos para los que se puede aplicar una variedad de técnicas y métodos. Algunos de ellos son bien conocidos, pero otros métodos han sido diseñados, implementados y probados durante el desarrollo de esta tesis.

El primer paso en el marco de trabajo propuesto es un módulo estático de reconocimiento que provee probabilidades de clase para cada pixel de la imagen desde un conjunto de características locales. Estas probabilidades son actualizadas dinámicamente y suministradas a un modulo decisión de seguimiento capaz de manejar oclusiones parciales o totales. Se presenta dos métodos específicos usando características de color RGB pero diferentes en la implementacion del clasificador: uno es un metódo Bayesiano basado en la máxima verosimilitud y el otro método está basado en una red neuronal. Los resultados experimentales obtenidos han mostrado que, por una parte, el enfoque basado en la red neuronal funciona similarmente y algunas veces mejor que el enfoque bayesiano cuando son integrados dentro del marco probabilistico de seguimiento. Por otra parte, nuestro método PIORT ha alcanzado mejores resultados comparando con otros métodos de seguimiento publicados. Todos estos métodos han sido probados experimentalmente en varias secuencias de vídeo tomadas con cámaras fijas y moviles incluyendo oclusiones parciales y totales del objeto a seguir, en ambientes interiores y exteriores, en diferentes tareas y niveles de complejidad. Esto ha permitido evaluar tanto la metodología general como los métodos alternativos que componen sus módulos.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

SCOPE OF THIS THESIS

This thesis describes thoroughly and in detail the current state of a *probabilistic integrated object recognition and tracking* (PIORT) methodology that we have developed in the latest years, as well as two particular methods derived from it. It also presents a collection of experimental results in test video sequences obtained by PIORT methods and alternative tracking methods. Previous stages in the development of PIORT, together with preliminary results, have been partially reported elsewhere [Amezquita, 2006, 2007, 2008; Alquézar, Amézquita, 2009].

In this thesis, we define an integrated recognition and tracking model and we develop some algorithms to solve multiple object tracking in video sequences. The main aim of this work is to reduce to the minimum the learning process and to avoid the application dependence, although we know that it may be to the detriment of the effectiveness.

We have initially taken some video sequences in an office, with corridors, doors and windows. Moreover, there are usual objects as chairs, tables, computers and so on. We have used static and moving cameras also showing walking people and mobile robots. We do not use stereo techniques to infer 3D data; just process the sequence taken by a single camera.

We want the learning process to be as easy as possible. For this reason, we have rejected the 3D models that represent very accurately the objects to learn. Some 2D images taken from the objects have to be enough to learn the main features of them. For this reason, it is not possible to model the 2D projection in the image of the 3D object and so, to predict the deformation of the 2D shape from one frame to the other.

We propose a *silhouette shape representation* of the objects and *colour* as the main feature of the object appearance. The silhouette is represented by the inner regions of the object obtained by the recognition and tracking processes. The recognition process is based on a (supervised learnt) *classifier*. We have not used the contour of the object since the objects are modeled by few 2D images and their 3D shape is not available. Then, it is not possible to model the 2D projection of the contour of the object.

This thesis is detailed in the following chapters. In Chapter 1, basics of object representation and recognition and tracking processes are introduced. Chapter 2 describes the objectives and the assumptions and constraints that were imposed. Chapter 3 presents the state of the art in dynamic object recognition and silhouette tracking. In chapter 4, the problem addressed (i.e. recognition and tracking of multiple objects in image sequences) is formally defined and expressed in a probabilistic framework for object recognition and tracking, As shown in Fig. 3, the system is divided in three subproblems or modules that we have named “static recognition”, “dynamic recognition” and “tracking decision module”, respectively. As far as we know, this global approach is original in the field. The implemented methods used for the static recognition module are specified in section 4.1. Neural networks and Bayesian method have been used for static recognition; this is, to recognize objects in single still images. Although this is not new, the use of features extracted from the image regions resulting from an image segmentation process permits to assign a-posteriori class probabilities to

ISBN:978-84-693-3387-7/DL:T.1001-2010

every image pixel using the outputs of a (previously trained) neural net, and this is a novel way to manage the object classification results provided by the net.

The dynamic recognition module is explained in section 4.2 and basically consists of a formula to update the class probabilities of each pixel that combines the static recognition results from the current image with the evidence accumulated in the previous images of the sequence. A slow apparent motion of the objects in the sequence is assumed, but the formula is parameterized to allow different weighting combinations related to different motion speeds. The tracking decision module is described in detail in Section 4.3 to compute the tracking decision for each object in each image, representing the result as a binary image for each object. The current system, that uses the tracking method based on prediction, is able to recognize and track fairly several objects of interest selected in the test sequences, which show some variations in translation, scale and orientation resulting from their motion in the scene or the camera motion. To this end, the system uses very simple features extracted from image segmentation regions and does not involve any geometric model of the objects.

A software system has been designed to implement and test both the global approach presented in chapter 4 and the techniques proposed for each module in sections 4.1 to 4.3. This system has been incrementally developed and applied to several image sequences acquired in both indoor and outdoor environments. The experimental work and the results obtained so far are described in chapter 5, which is divided in two parts: the first part presents the experiments only related with the static recognition module, while the rest include the experiments with dynamic recognition and tracking (in which the static recognition module also is involved). Finally, conclusions and future work are discussed in chapter 6.

As a conclusion, the aim of this thesis is to define a new general methodology for object recognition and tracking that assumes that there is really few information about the objects to be recognized and tracked. With few initial images in the learning step, the object model learnt by a classifier captures the distinguishing features of the object appearance (nowadays, the colour). After the recognition and tracking steps, the system has to obtain the identity and the position of each object, although the 2D shapes vary significantly along the sequence. Finally, we are aware that the results presented in some of the papers reported in the tracking literature, may be better than ours; nevertheless, they usually need a more complicated learning process and also, they are much more application dependent.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 1. Introduction

Object tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around the scene. The estimation of the trajectory of an object is pertinent in the following tasks between others: motion-based recognition (human identity based on gait, automatic object detection), automated surveillance, video indexing, human-computer interaction (gesture recognition, eye gaze tracking), traffic monitoring or vehicle navigation. The process of tracking objects can be very complex and application dependent due to: the projection of the 3D world to a 2D image, noise in images, complex object motion, non-rigid or articulated objects, partial and full object occlusions, complex object shapes, scene illumination changes or real-time processing requirements. Nevertheless, in most of the methods presented elsewhere, the tracking process is simplified by imposing constraints on the motion or appearance of objects. For example, almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. Another usual simplification is to have prior knowledge of about the number and size of the objects or the object appearance and shape.

The location and tracking of objects in indoor and outdoor environments is one of the most challenging problems that a mobile robot has to confront. For this end, the object models have to be defined or learned in conjunction with some associated recognition and tracking procedures. There are several issues that have to be considered while dealing with object locating and tracking which deserve some discussion. The first important issue is to determine the type of object model to learn which usually depends on the application environment. In our case, we want a mobile robot equipped with a camera to locate and track general objects (people, other robots, balls, wastepaper bins …) in both indoor and outdoor environments. A useful object model should be relatively simple and easy to acquire from the result of image processing steps. For instance, the result of a colour image segmentation process, consisting of a set of regions or spots, characterized by simple features related to colour, may be a good starting point to learn the model. Although structured models like attributed graphs or skeletons can be synthesized for each object from several segmented images [Ali, 2001,Foresti, 1999], we have decided to investigate a much simpler approach in which the object is just represented as an unstructured set of pixels.

One of the main drawbacks of structural methods is that the segmented images can be quite different from one frame to the other, and therefore it is difficult to match the structure in the current frame with the previous ones. The starting point of our approach is to accept these differences between segmented images and use a more rudimentary model in which the basic element is not the spot or region of the segmented image but its pixels. An example of structural method was reported in [Foresti, 19992], where the object model was based on the skeleton of the object obtained in the segmented images. Since the skeletons resulting from two almost equal images can be very different, the applicability of such approach is limited. The tracking step was performed in [Foresti, 19992] by an extension of the Kalman filter in which the skeleton and other geometrical features were considered. Numerous approaches for object tracking have been proposed. These primarily differ from each other based on the way they approach the three following questions:

ISBN:978-84-693-3387-7/DL:T.1001-2010

- Which object representation is suitable for the tracking?
- Which image features should be used?
- How should the motion, appearance and shape of the object be modeled?

The answers to these questions depend on the context in which the tracking is performed and the use for which the tracking information is being sought. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios. In the following section, we first describe the object shape representations followed by the appearance representations of the objects. In section 1.2 we comment the main features used to describe the objects. And in section 1.3, we describe the two main processes involved in a tracking system: object detection (or recognition) and object tracking.

1.1 Shape and Appearance Representation

In a tracking scenario, an object is anything that is of interest for our application. For instance, boats on the sea, fish inside an aquarium, vehicles on a road, planes in the air, people walking in the street. Objects can be represented by their shapes and appearances. In this section, we first describe the object shape representations commonly employed for tracking. After that, we address the appearance representations and we finish with the joint shape and appearance representations.

- **Point**. The object is represented by a point (figure 1.a) [Veenman, 2001] or a set of points (figure 1.b) [Serby, 2004]. This representation is useful for objects that occupy small regions in the image.
- **Primitive geometric shapes**. The object is enclosed in a geometric shape as a rectangle or an ellipse (figures 1.c and 1.d) [Comaniciu, 2003]. This representation is useful for simple and rigid objects, although it can be used as an approximation of non-rigid objects.
- **Object contour**. Contour representations define the boundary of an object (figures 1.g and 1.h) [Yilmaz, 2004]. This representation is useful for complex and non-rigid objects that their edges can be easily extracted.
- **Object silhouette**. A silhouette representation defines the region inside the object (figure 1.i) [Haritaoglu 2000; Yilmaz, 2004]. Similarly to contour representations, the silhouette is also useful for non-rigid objects. Besides, it can be used in the applications which it is difficult to extract the edge of the object.
- **Articulated shape models**. Articulated objects are composed of body parts that are held together with joints (figure 1.e). The constituent parts are usually modelled by cylinders or ellipses and the joints or relationships between the parts are governed by kinematics' motion models as joint angles. Thesse models are used for non-rigid objects composed by rigid parts.
- **Skeletal models**. Skeletal models represent the object by their skeleton (figure 1.f). The skeleton of an object is usually extracted applying the medial axis transform to the object silhouette [Ballard, 1982, chapter 8]. This model is used for both rigid and non-rigid objects [Ali, 2001], nevertheless, non-compact objects obtain better representations.

ISBN:978-84-693-3387-7/DL:T.1001-2010

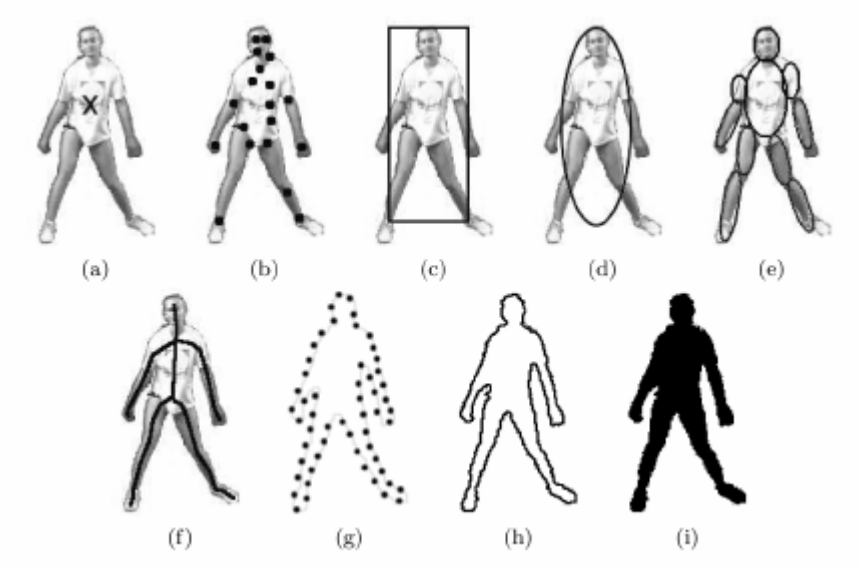


Figure 1 Object representation, (a) Centroid, (b) multiple points, (c) rectangular patch (d) elliptical patch, (e) part-based multiple patch, (f) object skeleton, (g) complete object contour, (h)control points on object contour, (i) object silhouette.

There are few representations that encode only the appearance without considering the shape. The most commonly used is:

- **Probability densities**. In this representation, the probability densities of the object appearance features such as colour or texture are estimated. They can be either parametric, such as Gaussian [Paragios, 2002], or non-parametric such as Parzen windows [Elgammal, 2002] or histograms [Comaniciu, 2003].

In the last methods, shape representations are combined with appearance representations [Cootes, 2001]. Some of them applied on the context of object tracking are:

- **Templates**. The appearance of the object is represented by a set of simple geometric shapes or silhouettes [Fieguth, 1997] that each contain some appearance information. Templates, however, only encode the object appearance generated from a single view. Thus, they are only suitable for tracking objects whose pose does not vary considerably during the course of tracking.
- **Active appearance models**. Active appearance models are generated by simultaneously modeling the object and the appearance [Edwards, 1998]. In general, the object shape is defined by a set of landmarks that reside on the boundary of the region. For each landmark, an appearance vector is stored which is in the form of colour, texture or gradient magnitude. This representation requires a training phase where both the shape and its associated appearance is learned from a set of samples using, for instance, the principal component analysis.
- **Multiview appearance models**. These models encode different views of an object. Usually, the different views of the object are represented by a subspace from the given views. Subspace approaches as Principal Component Analysis or Independent Component Analysis have been used for both shape and appearance representation [Mughadam, 1997; Black, 1998]. Similar to Active appearance models, this representation also requires a training phase. One limitation of multiview appearance models is that the appearances in all the view (or almost similar views) are required ahead of time.

ISBN:978-84-693-3387-7/DL:T.1001-2010

- **Training a set of classifiers**. Another approach to learn the different views of an object is by training a set of classifiers. The most common models are Support vector machines [Avidan, 2001] or Bayesian networks [Park, 2004]. Similar to multiview appearance models, these classifiers require a training phase in which almost similar views are required ahead of time. Nevertheless, they can learn the shape, the appearance or both.

1.2 Feature Selection

Selecting the right features plays a critical role in tracking. In general, the most desirable property of a visual feature is its uniqueness so that the objects can be easily distinguished in the feature space and they do not vary from one image to another. Feature selection is closely related to the object representation. For example, colour is used as a feature for histogram-based appearance representations or object edges are used as features for contour-based representations. The details of common visual features are as follows:

- **Colour**. A variety of colour spaces have been used in tracking due to it is no last word on which colour space is more efficient. To represent the colour, the two most common spaces have been the RGB and HSV. Nevertheless, the differences between the colours in RGB do not correspond to the colour differences perceived by the humans [Paschos, 2001]. In contrast, L*u*v* are perceptually uniform colour spaces while HSV is an approximately uniform colour space but sensitive to noise [Song 1996].
- **Edges**. Edges have been extensively used in tracking due to its simplicity and accuracy. The most common edge extractors are [Canny, 1986; Bowyer, 2001]. An important property of edges is that they are less sensitive to illumination changes compared to colour features. Nevertheless, in some applications, it is not possible to obtain reliable edges due to the nature of the objects and images.
- **Optical flow**. Optical flow is a dense field of displacement vectors which defines the translation of each pixel in a region [Horn, 1981; Barrow, 1994]. It is used as a feature in motion-based segmentation.
- **Texture**. Texture is a measure of the intensity variation of a surface which quantifies properties such as smoothness and regularity [Tomita, 1990]. Compared to colour, textures requires a processing step to generate the descriptors. Similarly to edge features, textures are less sensitive to illumination changes compared to colour. Moreover, textures can be extracted although it is not possible to obtain reliable edges.

1.3 Recognition and Tracking Processes

Every tracking method requires an object detection or recognition mechanism either in every frame or when the object first appears in the video. Two different approaches can be distinguished considering the temporal information. In the static approaches, the object detection and recognition is performed using the information in a single frame. The dynamic approaches make use of the temporal information computed from a sequence of frames to increase the recognition ratio and to reduce the number of false detections. Given the regions that the recognizer has considered that there is an object, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

In the rest of the section, we first depict the most common models for object detection and recognition using the static and dynamic approaches. After that, we comment the most common tracking methods.

ISBN:978-84-693-3387-7/DL:T.1001-2010

1.3.1 Object detection and recognition

In the static models for object detection, the only information considered is the current frame. The most common models are:

- **Interest points**. Interest points in the images are the pixels that have an expressive texture in their respective localities. They have been long used in the context of motion, stereo and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint. For a comparative evaluation of interest point detectors, we refer the reader to the survey [Mikolajczyk, 2003].
- **Segmentation**. The aim of the segmentation algorithms is to partition the image into perceptually similar regions. Every segmentation algorithm addresses two problems, the criteria for a good partition and the method for achieving efficient partitioning [Shi, 2000].
- **Supervised learning**. Object detection can be performed by learning different object views automatically from a set of examples by means of supervised learning mechanism. Learning of different object views waives the requirement of storing a complete set of examples, then, the learning methods generate a function that maps inputs to desired outputs. A standard formulation of supervised learning is the classification problem where the learner approximates the behavior of a function by generating an output in the form of a class label. In the context of object detection, the learning examples are composed of pairs of object features and an associated object class where both of these quantities are manually defined. The learning methods include, but are no limited to, neural networks [Rowley, 1998] adaptive boosting [Viola, 2003], decision trees [Grewe, 1995] and support vector machines [Papageorgiu, 1998].

The dynamic methods make use of the current frame and some previous frames or knowledge taken from them to detect the objects. The principal approach is:

- **Background subtraction**. In this method, object detection is achieved by building a representation of a scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to objects. This process is referred to as the background subtraction [Wren, 1997].

1.3.2 Object tracking

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. The object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing the correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm and then, the tracker corresponds objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. We now briefly introduce the main tracking models.

ISBN:978-84-693-3387-7/DL:T.1001-2010

- **Point tracking**. Objects are represented by *Points*. The recognition algorithm is based on *Interest points*. The detected points in consecutive frames are tracked based on the previous point state which can include point position, speed and acceleration (figure 2.a). There are basically two main categories, deterministic methods [Veenman, 2001] and statistical methods [Streit, 1994].
- **Kernel tracking**. Objects are represented by *Primitive geometric shapes*. The recognition algorithm is based on *Segmentation* or *Supervised learning*. The kernel can be a rectangular or elliptical shape with an associated histogram (figure 2.b). Objects are tracked by computing the motion of the kernel in consecutive frames. This motion is usually in the form of a parametric transformation such as translation, rotation and affine [Schweitzer, 2002].
- **Silhouette tracking**. Objects are represented by the silhouette. The recognition algorithm is based on *Segmentation* or *Supervised learning*. Tracking is performed by estimating the object region in each frame. Silhouettes are tracked by either shape matching or contour evolution (figure 2.c and 2.d). Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames [Huttenlocher, 1993].

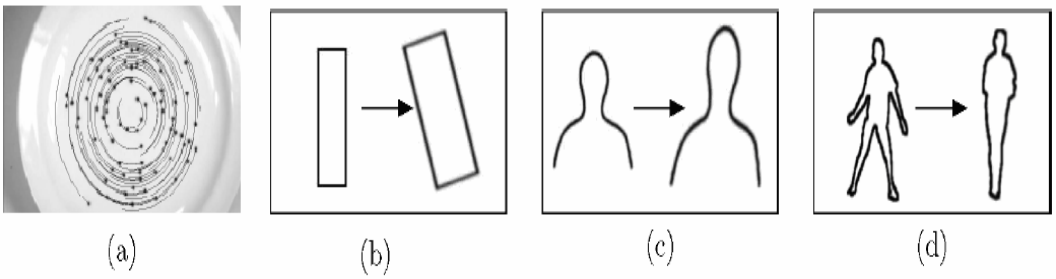


Figure 2 Different tracking approaches. (a) Point tracking (b) Parametric transformation of a rectangular patch, (c, d) Two examples of contour evolution.

ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 2. Objectives

2.1 General objective.

The main objective of this Ph.D. thesis was to design, implement and test a general methodology for dynamic integrated object recognition and tracking in video sequences. This included devising adequate methods for each module in the methodology.

To this end, the following assumptions and constraints were imposed:

- the video sequences to deal with show indoor or outdoor scenes perceived by a color vision system that can move (relatively) slowly through the environment, or can be in fixed position; (this is quite realistic for most applications in practice);

- the objects of interest are 3D and can be rigid (e.g. a static object like a monitor or a wheel-driven mobile robot) or articulated (e.g. a person or a legged mobile robot);

- the objects of interest are defined and learned off-line using some kind of supervised classifier trained from different views of the 3D objects;

- no geometric model of the object 3D shape is built or learned;

- the representation of object appearance and 2D shapes may be distinguished from other objects and the background based on simple features like color. This is internal to the classifier used (for instance, conditional probability densities of the classes given the input features in the case of using a neural network), though these features may experiment slight variations during the image sequence; in fact, this is a requirement of the classifiers we currently use for static recognition and could be relaxed or changed if the classifier in this module were replaced or used a different set of object's appearance features;

- the apparent motion and size of the objects of interest in the sequence (either caused by the motion of the camera or by their own motion) is smooth with no abrupt changes between consecutive frames (non-rigid deformable objects are thus allowed); we think this is not a strong assumption if a typical video acquisition rate is used, as large changes in shape, motion and size are allowed for the whole sequence;

- objects of interest can partially or fully occlude each other during some frames and they can also disappear or enter in the scene from one frame to the next (typically through the image borders, assuming a smooth slow motion), but their motion does not change abruptly during occlusion; this last assumption is certainly stronger and may fail in some cases, but it is caused by the need of predicting an approximate position of the object during occlusion based on its previous trajectory.

ISBN:978-84-693-3387-7/DL:T.1001-2010

- the object 2D shape representation employed for tracking is the object silhouette (the region inside the object), considered as an unstructured set of points.

2.2 Specific objectives.

- To study, review and analyze the state-of-the art in dynamic object recognition and integrated object recognition and tracking in video sequences, with a special attention to the techniques based on silhouette tracking.

- To develop a probabilistic framework for integrated object recognition and tracking in image sequences.

- To design, implement and test static object recognition methods using features extracted from image segmentation regions and from the original images.

- To design, implement and test some dynamic object recognition method using the results of static recognition on the current frame and feed-back from the previous image sequence and results.

- To design, implement and test some silhouette tracking methods in accordance with the imposed assumptions and constraints (that have been listed before).

- To develop a prototype system implementing and integrating all the techniques proposed.

- To devise adequate experiments covering a variety of cases, like partial and full occlusions, objects disappearing and entering in the scene, presence of multiple objects of the same class, noisy backgrounds, illumination changes, etc.

- To evaluate the proposed methodology from these experiments and to compare it with alternative state-of-the-art techniques applicable under the same assumptions and constraints.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 3. State of the art in dynamic object recognition and silhouette tracking

In this section, we comment the most relevant papers that use a framework similar to the one presented in this work. The two main features of the following models are: first, there is a feedback from the tracking to the recognition processes. That is, one of the parameters of the recognition process is the tracking knowledge of objects in the previous frame, i.e., the position, speed, acceleration. Second, the shape representation is based on the silhouette. That is, in the segmentation process, the region inside the object is obtained. Then, in the recognition process, we consider the features of the obtained region; and in the tracking process, we track the object through the dynamic features of this region.

3.1 Dynamic recognition and integrated recognition and tracking

A relevant issue, which generally is not so mentioned, is to integrate the recognition and tracking steps in a common framework that helps to exploit some feedback between them. To the best of our knowledge there are few existing works that combine recognition and tracking in an integrated framework [Tu, 2003, Lee, 2005]. Object recognition and tracking are usually performed sequentially and without any feedback from the tracking to the recognition step [Foresti, 1999]. These tasks often are treated separately and/or sequentially on intermediate representations obtained by the segmentation and grouping algorithms [Zhu, 1996- Tu, 2002]. Sometimes, they are applied in a reverse order, with a first tracking module supplying inputs to the recognition module, as, for instance, in gesture recognition [Chen, 2003].

Tracking and recognizing people through their faces is one of the most important problems solved by the tracking applications. One of the recent works was presented in [Lee, 2005]. In this paper, an integrated framework for tracking and recognizing faces was defined that had the property of tightly coupling the recognition and tracking processes. This was a new framework because, since now, this problem had been solved through a conventional video-based face recognition model plus a tracking module. In contrast, an architecture was defined in [Lee, 2005] that couples these two components within a single framework. The complex and nonlinear appearance manifold of each registered person was partitioned into a collection of sub-manifolds where each one modeled the face appearances of the person in nearby poses. During the training step, a clustering algorithm was used to partition the training images into clusters. The images in each cluster usually came from neighboring poses. The PCA technique was applied to the images in each cluster to yield a low dimensional linear subspace approximation. The recognition module kept a detailed appearance model for each registered individual at each frame. The tracker only used a portion of the appearance model of an individual identified by the recognizer.

Tracking and recognition techniques have been applied to gesture recognition problems [Chen 2003]. In this work, the model has a real-time hand tracking and extraction algorithm to trace the moving hand and extract the hand region. The spatial features are extracted by a Fourier descriptor and the temporal features are obtained by a motion analysis model. The combination of the spatial and temporal features of the input image

ISBN:978-84-693-3387-7/DL:T.1001-2010

sequence is used to extract a feature vector. Then, the gesture is classified by a Hidden Markov Model.

In [Zhou 2004], an observation model and a velocity motion model were defined. The observation model was based on an adaptive appearance model, and the velocity motion model was derived using a first-order linear predictor approximation based on the appearance difference between the incoming observation and the previous particle filter configuration.

3.2 Silhouette tracking

Silhouette tracking is employed when tracking of the complete region of an object is required. In the context of region tracking, the precision and recall measures are defined in terms of the intersection of the hypothesized and correct object regions. The precision is the ratio of the intersection to the hypothesized region and recall is the ratio of the intersection to the ground truth. Important advantage of tracking silhouettes is their flexibility to handle a large variety of object shapes. Silhouettes can be represented in different ways. The most common silhouette representation is in the form of a binary indicator function, which marks the object region by ones and the nonobject regions by zeros. For contour-based methods, the silhouette is represented either explicitly or implicitly. Explicit representation defines the boundary of the silhouette by a set of control points. Implicit representation defines the silhouette by means of a function defined on a grid. The most common implicit contour representation is the level sets representation.

The representations chosen by the silhouette-based object trackers can be in the form of motion models, appearance models, or shape models or a combination of these. Object appearance is usually modeled by parametric or nonparametric density functions such as mixture of Gaussians or histograms. Object shape can be modeled in the form of contour subspace where a subspace is generated from a set of possible object contours obtained from different object poses [Blake and Isard 2000]. Additionally, object shape can be implicitly modeled via a level set function where the grid positions are assigned at the distance generated from different level set functions corresponding to different object poses [Yilmaz et al. 2004]. Appearance-based shape representations are also commonly used by researchers who employ a brute force silhouette search. For edge-based shape representation, Hausdorff distance is the most widely used measure. However, Hausdorff measure is known for its sensitivity to noise. Hence, instead of using the maximum of distances, researchers have considered using an average of the distances [Baddeley 1992].

Silhouette based methods provide an accurate shape description of the objects that have complex shapes, i.e., hands, head and shoulders and that cannot be well described by simple geometric shapes. The goal of a silhouette-based object tracker is to find the object region in each frame by means of an object model generated using the previous frames. This model can be in the form of a colour histogram, object edges or the object contour. There are two categories, shape matching and contour tracking. Shape matching searches the object silhouette in the current frame. Contour tracking evolves an initial contour to its new position in the current frame by either using the state space models or direct minimization of some energy functional.

ISBN:978-84-693-3387-7/DL:T.1001-2010

Shape Matching. In shape matching the object silhouette and its associated model is searched in the current frame. The search is performed by computing the similarity of the object with the model generated from the hypothesized object silhouette based on previous frame. Here, the silhouette is assumed to only translate from the current frame to the next, therefore non-rigid object motion is not explicitly handled.
The object model is usually in the form of an edge map, it is reinitialized to handle appearance changes in every frame after the object is located. This update is required to overcome tracking problems related to viewpoint and lighting condition changes as well as non-rigid object motion. In the context of matching using an edge-based model, Hausdorff distance measures the most mismatched edges. Due to this, the method emphasizes parts of the edge map that are not drastically affected by object motion. For instance, in the case of a walking person, the head and the torso do not almost change their shape, whereas the motion of the arms and legs makes the shape change. Therefore, removing the edges of the arms and legs improves the tracking performance. In a similar way, Li et al. [2001] propose using the Hausdorff distance for verification of the trajectories and pose estimation problem.

Another approach to match shapes is to find corresponding silhouettes detected in two consecutive frames and establishing silhouette correspondence. Silhouette matching makes use of object appearance features, whereas point matching uses only motion and position-based features. Silhouette detection is usually carried out by background subtraction. Once the object silhouettes are extracted, matching is performed by computing some distance between the object models associated with each silhouette. Object models are usually in the form of density functions (colour or edge histograms), silhouette boundary (closed or open object contour), object edges or a combination of these models. To match silhouettes in consecutive frames, Haritaoglu et al. [2000] model the object appearance by the edge information obtained inside the object silhouette. In particular, the edge model is used to refine the translation of the object assuming constant velocity. This refinement is carried out by performing binary correlation between the objects edge in the consecutive frames. In contrast to looking for possible silhouette matches in consecutive frames, tracking silhouettes can be performed by computing the flow vectors for each pixel inside the silhouette such that the flow that is dominant over the entire silhouette is used to generate the silhouette trajectory. Following this observation, Sato and Aggarwal [2004] proposed to generate object tracks by applying Hough transform in the velocity space to the object silhouettes in consecutive frames. Binary object silhouettes are detected using background subtraction. Then, from a spatio-temporal window around each moving region pixel, a velocity Hough transform is applied to compute voting matrices for the vertical flow v and the horizontal flow u. These voting matrices provide the so-called Temporal Spatio-Velocity (TSV) image in 4D (x, y, u, v) per frame. TSV image encodes the dominant motion of a moving region pixel and its likelihood in terms of number of votes such that a threshold operation will provide regions with similar motion patterns. In contrast to appearance-based matching of silhouettes, TSV provides a motion-based matching of the object silhouettes and is less sensitive to appearance variations, due to different object views (e.g., front and back of the object may look different).

Contour Tracking. Contour tracking methods evolve an initial contour in the previous frame to its new position in the current frame. This contour evolution requires that some part of the object in the current frame overlap with the object region in the previous frame. Tracking by evolving a contour can be performed using two different

ISBN:978-84-693-3387-7/DL:T.1001-2010

approaches. The first approach uses state space models to model the contour shape and motion. The second approach directly evolves the contour by minimizing the contour energy using direct minimization techniques such as gradient descent.

Tracking Using State Space Models. The object's state is defined in terms of the shape and the motion parameters of the contour. The state is updated at each time instant such that the contour's a posteriori probability is maximized. The posterior probability depends on the prior state and the current likelihood which is usually defined in terms of the distance of the contour from observed edges. Terzopoulos and Szeliski [1992] define the object state by the dynamics of the control points. The dynamics of the control points are modeled in terms of a *spring model*, which moves the control points based on the spring stiffness parameters. The new state (spring parameters) of the contour is predicted using the Kalman filter. The correction step uses the image observations which are defined in terms of the image gradients. In 1998, Isard and Blake defined the object state in terms of spline shape parameters and affine motion parameters. The measurements consist of image edges computed in the normal direction to the contour. The state is updated using a particle filter. In order to obtain initial samples for the filter, they compute the state variables from the contours extracted in consecutive frames during a training phase. During the testing phase, the current state variables are estimated through particle filtering based on the edge observations along normal lines at the control points on the contour.

Tracking by Direct Minimization of Contour Energy Functional. In the context of contour evolution, there is an analogy between the segmentation methods. The contour energy is defined in terms of temporal information in the form of the temporal gradient (optical flow). Bertalmio et al. [2000] computes the flow only on the object boundary, his approach is motivated by computing the flow vector for each pixel inside the complete object region in a circular neighborhood with radius r using a brute force search. Once the flow vectors are computed, the contour energy, which is based on the brightness constancy constraint, is evaluated. This process is iteratively performed until the energy is minimized. In 2003, Cremers and Schnorr also used the optical flow for contour evolution, and constraint such that an object can only have homogeneous flow vectors inside the region. Their energy is a modified form of the common Mumford-Shah energy [Mumford and Shah 1989], which evolves the contour until a region with homogeneous flow vectors is achieved. They also incorporated the shape priors to better estimate the object shape. The shape priors are generated from a set of object contours such that each control point on the contour has an associated Gaussian with a mean and standard deviation of the spatial positions of the corresponding control points on all the contours. An alternative to using the optical flow is to exploit the consistency of the statistics computed inside and outside the object region from one frame to the next. This approach requires initialization of the contour in the current frame with its previous position. In this context, Ronfrad [1994] defines the energy functional governing the contour evolution based on the piecewise stationary image models formulated as Ward distances. Ward distance can be considered as a measure of image contrast [Beaulieu and Goldberg 1989]. However, Ward distance can not be analitycally defined; hence, Ronfard's approach individually evolves each contour point based on its local neighborhood. In a similar vein, Yilmaz and Shah [2004] evolve an object contour using the colour and texture models generated in a band around the object's boundary. The width of the band serves as a means to combine region and boundary-based contour tracking methods into a single framework. In contrast to the aforementioned methods,

ISBN:978-84-693-3387-7/DL:T.1001-2010

Yilmaz et al. [2004] model the object shape and its changes by means of a level setbased shape model. In this model, the grid points of the level set hold the means and the standard deviations of the distances of points from the object boundary. The level set-based shape model resolves the object occlusions during the course of tracking.

3.3 Occlusion handling

Occlusion handling is another important aspect of silhouette tracking methods. Usually methods do not address the occlusion problem explicitly. A common approach is to assume constant motion or constant acceleration where, during occlusion, the object silhouette from the previous frame is translated to its hypothetical new position. Few methods explicitly handle object occlusions by enforcing shape constraints [Mac-Cormick and Blake 2000; Yilmaz et al. 2004]. Another important issue related to silhouette trackers is their capability for dealing with object split and merge. For instance, while tracking a silhouette of a person carrying an object, when the person leaves an object, a part of the person's contour will be placed on the left object (region split). These topology changes of region split or merge can be handled well by implicit contour representations.

A desidered objetive is to provide the tracking procedure with the capacity of determining occlusions and re-emergencies of tracked objects, i.e. occlusion handling. Over recent years, much research has been developed to solve the problem of object tracking under occlusions, because, in real-world tracking, a target being partly or entirely covered by other objects for an uncertain period of time is common. Occlusions pose two main challenges to object tracking systems. The first challenge is how to determine the beginning and the end of an occlusion. The second challenge is how to predict the location of the target during and at the end of the occlusion.

Determining occlusion status is very hard for the trackers where the only knowledge available on the target is its initial appearance. When some parts of an occluder are similar to those of the target, the occluder and the target are mistaken. Various approaches that analyze occlusion situations have been proposed. The most common one is based on background subtraction [Senior, 2006]. Although this method is reliable, yet it only works with a fixed camera and a known background, which is not our case in mobile robotics. Other approaches are based on examining the measurement error for each pixel [Nguyen, 2004]. The pixels that their measurement error exceeds a certain value are considered to be occluded. These methods are not very appropriate in outdoor scenarios, where the variability of the pixel values between adjacent frames may be high. A mixture of distributions is used in [Jepson, 2003] to model the observed value of each pixel, where the occluded pixels are characterised by having an abrupt difference with respect to a uniform distribution. Contextual information is exploited in [Ito, 2001, Hariharakrishnan, 2005]. These methods have better performance in terms of analysing occlusion situations but tracking errors are observed to frequently occur and propagate away. In addition, in the case of using these approaches in a mobile robot application, there is a need of knowing a priori the robot surroundings.

Determining the re-emergence of the target and recapture its position after it is completely occluded for some time is the other main challenge. Setting a similarity threshold is one method, yet the optimal threshold value is difficult to determine. This

ISBN:978-84-693-3387-7/DL:T.1001-2010

problem is circumvented in [Nguyen, 2004], where the image region that matches the best with the template over a prefixed duration is assumed to be the reappearing target. In [Zhou,2004], an observation model and a velocity motion model were defined. The observation model was based on an adaptive appearance model, and the velocity motion model was derived using a first-order linear predictor. Both approaches are defined in the framework of particle filter, with provisions for handling occlusion.

In the scenarios where the motion of the target is not smooth neither predictable most of the aforementioned methods would fail. Recently, new object tracking methods that are robust to occlusion have been reported with very promising results [Zhu, 2008, Pan, 2007]. The method reported in [Zhu,2008] relies on background subtraction (it works only for static cameras) and a *k*-NN classifier to segment foreground regions into multiple objects using on-line samples of object's appearance local features taken before the occlusion. The method described in [Pan,2007] relies on an adaptive template matching but it only handles partial occlusions and the matching process seems to be computationally costly.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 4. A probabilistic framework for integrated object recognition and tracking

Let us assume that we have a sequence of 2D color images $I^t(x,y)$ for $t=1,...,L$, and that there are a maximum of N objects of interest in the sequence of different types (associated with classes $c=1,...,N$, where $N\geq 1$), and that a special class $c=N+1$ is reserved for the background. Furthermore, let us assume that the initial position of each object is known and represented by N binary images, $p_c^0(x,y)$, for $c=1,...,N$, where $p_c^0(x,y)=1$ means that the pixel (x,y) belongs to a region covered by an object of class c in the first image. If less than N objects are actually present, some of these images will be all-zero and they will not be processed further, so, without loss of generality, we consider in the sequel that N is the number of present objects to track.

Hence, we would like to obtain N sequences of binary images $T_c^t(x,y)$, for $c=1,...,N$, that mark the pixels belonging to each object in each image; these images are the desired output of the whole process and can also be regarded as the output of a tracking process for each object. We can initialize these tracking images (for $t=0$) from the given initial positions of each object, this is

$$T_c^0(x,y) = p_c^0(x,y) \tag{1}$$

In our approach, we divide the system in three modules. The first one performs object recognition in the current frame (static recognition) and stores the results in the form of probability images (one probability image per class), that represent for each pixel the probabilities of belonging to each one of the objects of interest or to the background, according only to the information in the current frame. This can be achieved by using a classifier that has been trained previously to classify image regions of the same objects using a different but similar sequence of images, where the objects have been segmented and labeled. Hence, we assume that the classifier is now able to produce a sequence of class probability images $Q_c^t(x,y)$ for $t=1,...,L$ and $c=1,...,N+1$, where the value $Q_c^t(x,y)$ represents the estimated probability that the pixel (x,y) of the image $I^t(x,y)$ belongs to the class c, which has been computed taking into account a local feature vector (see section 4.1). In general, the probability images $Q_c^t(x,y)$ can be regarded as the output of a static recognition module defined by some function r on the current image:

$$Q_c^t(x,y) = r\left(I^t(x,y)\right) \tag{2}$$

In the second module (dynamic recognition), the results of the first module are used to update a second set of probability images, p_c , with a meaning similar to that of Q_c but now taking into account as well both the recognition and tracking results in the previous frames through a dynamic iterative rule. More precisely, we need to store and update $N+1$ probability images $p_c^t(x,y)$, for $c=1,...,N+1$, where the value $p_c^t(x,y)$ represents the probability that the pixel (x,y) in time t belongs to an object of class c (for $c=1,...,N$) or to the background (for $c=N+1$). In general, these dynamic probabilities should be computed as a certain function f of the same probabilities in the previous step, the class

ISBN:978-84-693-3387-7/DL:T.1001-2010

probabilities given by the classifier for the current step (which have been obtained from the actual measurements) and the tracking images resulting from the previous step:

$$p_c^t(x,y) = f\left(p^{t-1}(x,y), Q^t(x,y), T^{t-1}(x,y)\right) \tag{3}$$

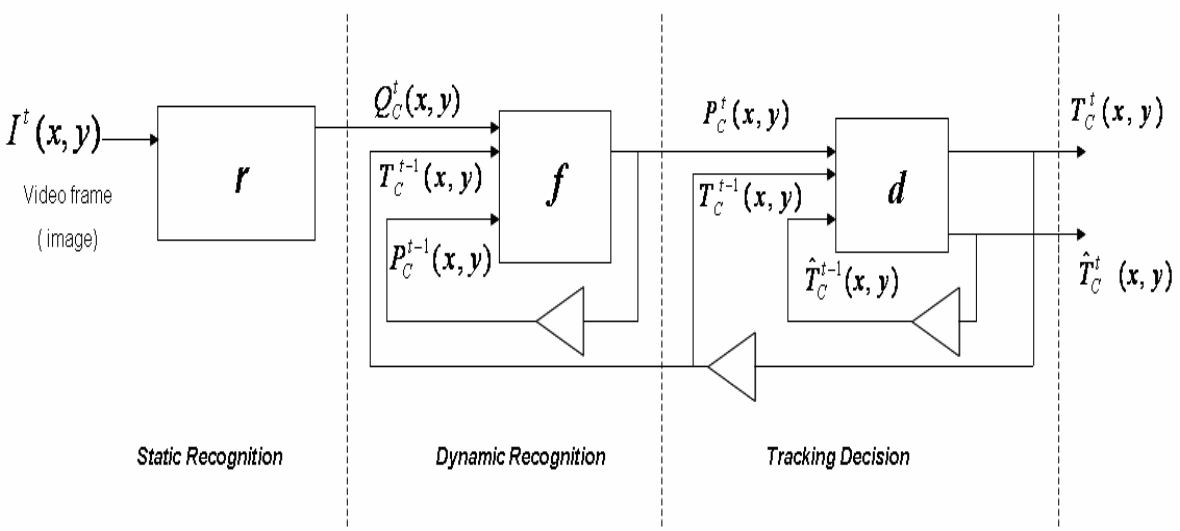


Figure 3 Block diagram of the dynamic object recognition and tracking process.

The update function *f* used in our system is described in section 4.2, which incorporates some additional arguments coming from the tracking module to adapt its parameters.
Finally, in the third module (tracking decision), tracking binary images are determined for each object from the current dynamic recognition probabilities, the previous tracking image of the same object and some other data, which contribute to provide a prediction of the object's apparent motion in terms of translation and scale changes as well as to handle the problem of object occlusion. Formally, the tracking images $T_c^t(x,y)$ for the objects ($1 \leq c \leq N$) can be calculated dynamically using the pixels probabilities $p^t(x,y)$ according to some decision function *d*:

$$T_c^t(x,y) = d\left(p^t(x,y), T_c^{t-1}(x,y)\right) \tag{4}$$

in which some additional arguments and results may be required (see (12) and Section 4.3 for a detailed description of the tracking decision module).

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

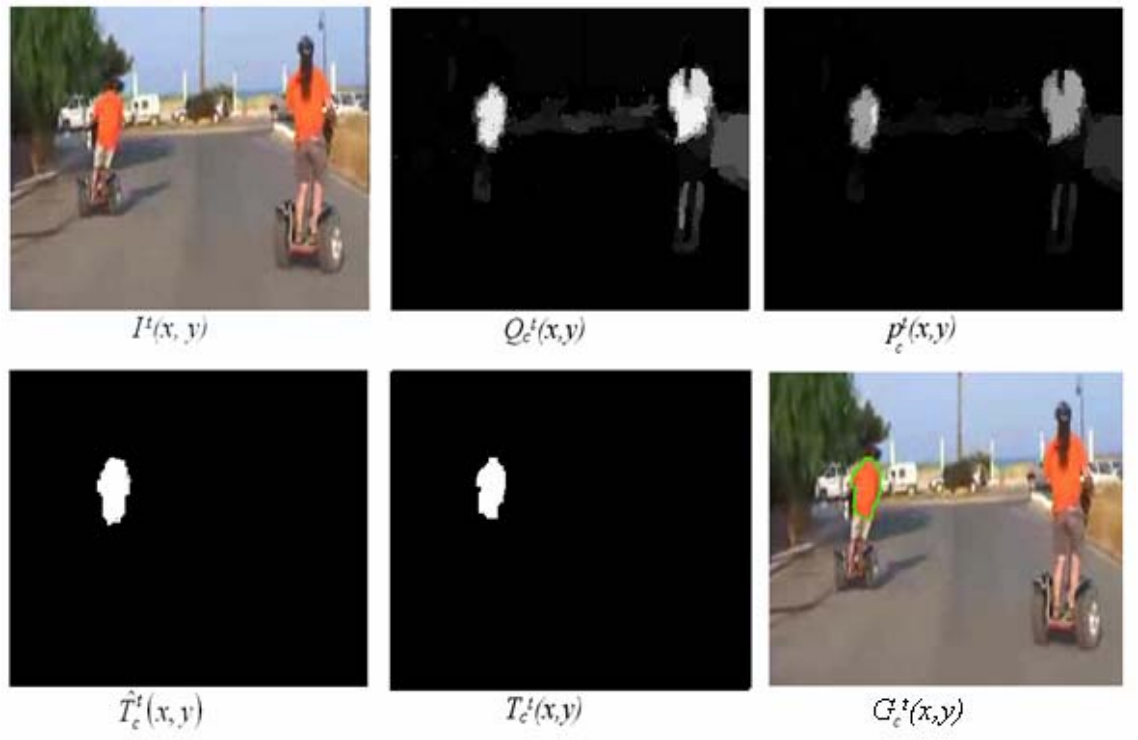


Figure 4 Images of PIORT modules. $I^t(x,y)$ input image, $Q_c^t(x,y)$ static probability image, $P_c^t(x,y)$ dynamic probabilitiy image, $\hat{T}_c^t(x,y)$ a priori tracking image, $T_c^t(x,y)$ a posteriori tracking image, $G_c^t(x,y)$ graphic overly of the contours of the a posteriori tracking image on the input image.

4.1 Static recognition module

In our PIORT (Probabilistic Integrated Object Recognition and Tracking) framework, the static recognition module is based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The local features to be used may be chosen in many different ways. A possible approach consists of first segmenting the given input image $I^t(x,y)$ in homogeneous regions (or spots) and computing some features for each region that are afterwards shared by all its constituent pixels. Hence, the class probabilities $Q_c^t(x,y)$ are actually computed by the classifier once for each spot in the segmented image and then replicated for all the pixels in the spot. For instance, RGB color averages can be extracted for each spot after color segmentation and used as feature vector $v(x,y)$ for a classifier. In the next two subsections we present two specific classifiers that have been implemented and tested within the PIORT framework using this type of information.

4.1.1 A simple Bayesian method based on maximum likelihood and background uniform conditional probability

Let c be an identifier of a class (between 1 and N+1), let B denote the special class c=N+1 reserved for the background, let k be an identifier of an object (non-background) class between 1 and N, and let v represent the value of a feature vector. Bayes theorem establishes that the posterior class probabilities can be computed as

ISBN:978-84-693-3387-7/DL:T.1001-2010

$$P(c \mid v) = \frac{P(v \mid c)P(c)}{P(v)} = \frac{P(v \mid c)P(c)}{P(v \mid B)P(B) + \sum_{k=1}^{N} P(v \mid k)P(k)} \tag{5}$$

Our simple Bayesian method for static recognition is based on imposing the two following assumptions:

a) equal priors: all classes, including B, will have the same prior probability, i.e. $P(B)=1/(N+1)$ and $P(k)=1/(N+1)$ for all k between 1 and N.
b) a uniform conditional probability for the background class, i.e. $P(v|B)=1/M$, where M is the number of values (bins) in which the feature vector v is discretized.

Note that the former assumption is that of a maximum likelihood classifier, whereas the latter assumes no knowledge about the background. After imposing these conditions, equation (5) turns into

$$P(c \mid v) = \frac{P(v \mid c)}{\frac{1}{M} + \sum_{k=1}^{N} P(v \mid k)} \tag{6}$$

and this gives the posterior class probabilities we assign to the static probability images, i.e. $Q_c^t(x,y) = P(c \mid v(x,y))$ for each pixel (x,y) and time t.

It only remains to set a suitable M constant and to estimate the class conditional probabilities $P(v \mid k)$ for all k between 1 and N (object classes). To this end, class histograms H_k are set up using the labeled training data and updated on-line afterwards using the tracking results in the test data.

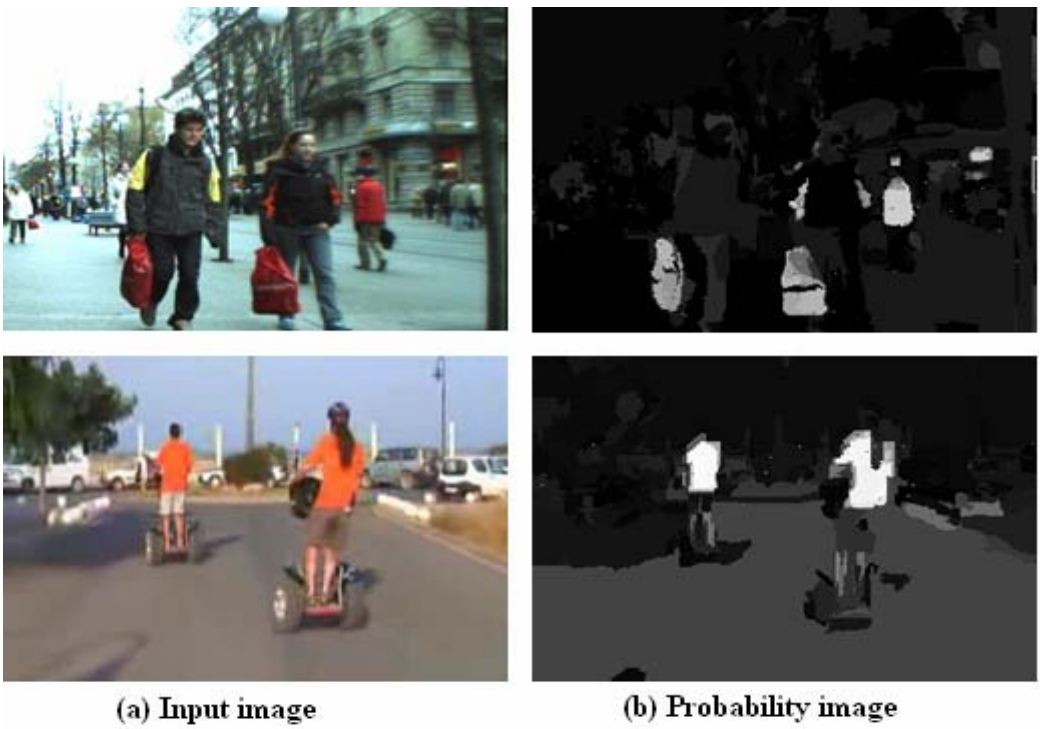


Figure 5 Two examples of probability images given by bayesian method in the static recognition module.

ISBN:978-84-693-3387-7/DL:T.1001-2010

For constructing the histograms, let *v(x,y)* be the feature vector consisting of the original RGB values of a pixel (*x*,*y*) labeled as belonging to class *k*. We uniformly discretize each of the R, G and B channels in 16 levels, so that $M = 16 \times 16 \times 16 = 4096$. Let *b* be the bin in which *v(x,y)* is mapped by this discretization. To reduce discretization effects, a smoothing technique is applied when accumulating counts in the histogram as follows:

$$\begin{aligned} H_k(b) &:= H_k(b) + (10 - \#neighbors(b)) \\ H_k(b') &:= H_k(b') + 1 \quad \text{if } b' \text{ is a neighbor of } b \end{aligned} \tag{7}$$

where the number of neighbors of *b* (using non-diagonal connectivity) varies from 3 to 6, depending on the position of *b* in the RGB space. Hence, the total count C_k of the histogram is increased by ten (instead of one) each time a pixel is counted and the conditional probability is estimated as $P(v \mid k) = H_k(b) / C_k$ where *b* is the bin corresponding to *v*. The above smoothing technique is also applied when updating the histogram from the tracking results; in that case the RGB value *v(x,y)* in the input image $I^t(x,y)$ of a pixel (*x*,*y*) is used to update the histogram H_k (and the associated count C_k) if and only if $T_k^t(x,y)=1$.

4.1.2 A neural net based method

In this method, a neural net classifier (a multilayer perceptron) is trained off-line from the labeled training data. The RGB color averages extracted for each spot after color segmentation are used as feature vector *v(x,y)* and supplied as input to the network in both training and test phases. To the contrary of the Bayesian method described previously, training data for the background class are also provided by selecting some representative background regions in the training image sequence, because the network needs to gather examples for all classes including the background. The network is not retrained on-line using the tracking results in the test phase (this is another difference with respect to the Bayesian method described).

It's well known that using a 1-of-*c* target coding scheme for the classes, the outputs of a network trained by minimizing a sum-of-squares error function approximate the posterior probabilities of class membership (here, $Q_c^t(x,y)$), conditioned on the input feature vector [Bishop, 1995]. Anyway, to guarantee a proper sum to unity of the posterior probabilities, the network outputs (which are always positive values between 0 and 1) are divided by their sum before assigning the posterior probabilities.

In this module we use a feed-forward 2-layer perceptron architecture (i.e. one hidden layer of neurons and an output layer) using standard backpropagation as training algorithm. Hyperbolic tangent and sine functions were used as activation functions in the hidden layer and the output layer, respectively. A modified version of the PDP simulator of Rumelhart and McClelland [Rumelhart and McClelland, 1986] was employed for the experiments, setting a learning rate of 0.003 and a momentum parameter of zero for backpropagation, and a maximum number of 2,000 training epochs for each run.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

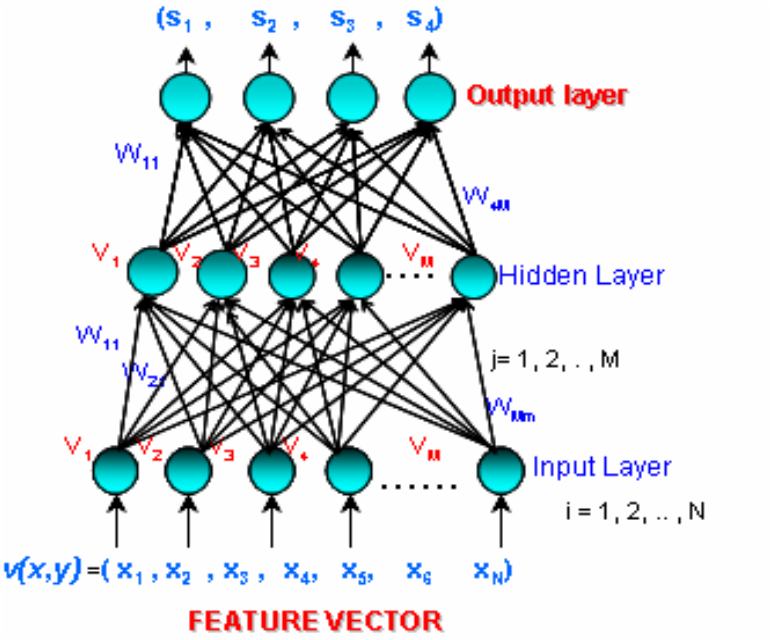


Figure 6 Neural network. Feed-forward 2-layer perceptron.

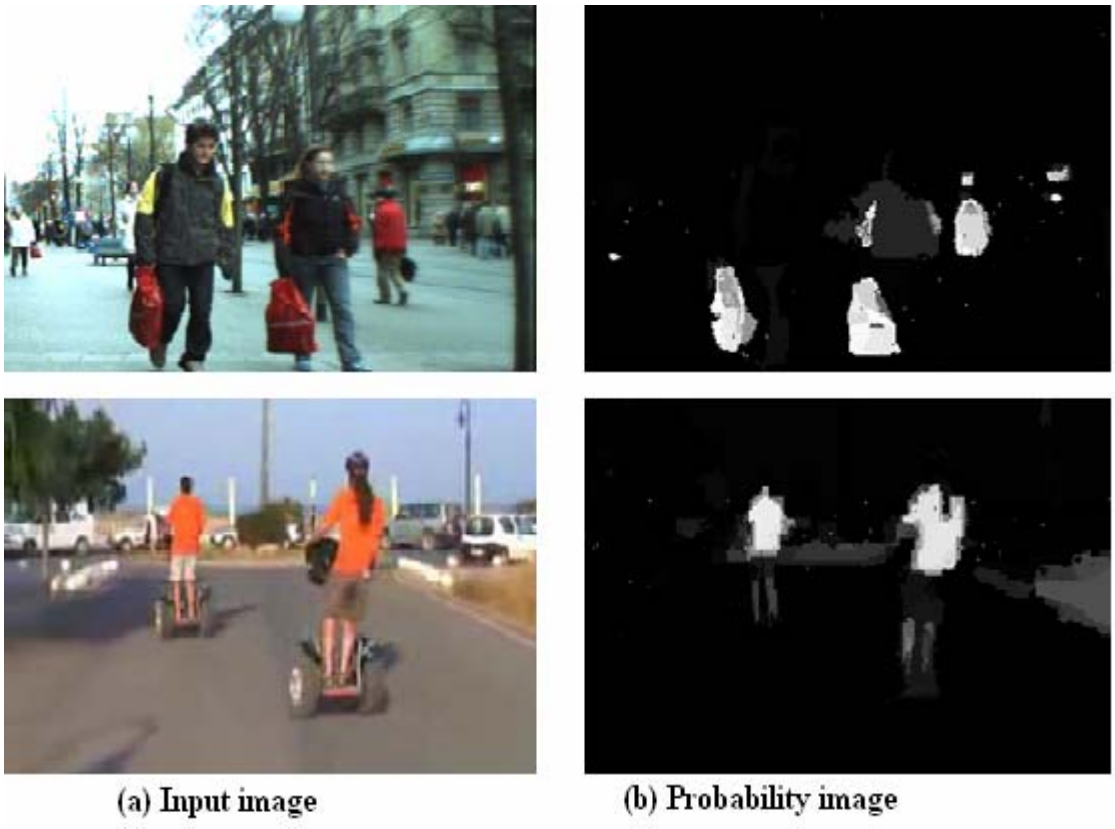


Figure 7 Two examples of probability images given by neural network method in the static recognition module.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

4.2 Dynamic recognition module

Even though the static recognition module can be applied independently to each image in the sequence, this does not exploit the dynamic nature of the problem and the continuity and smoothness properties that are expected in the apparent motion of the objects through the sequence. Hence, a dynamic update of the pixel class probabilities $p_c^t(x,y)$ is desired that takes into account these properties. To this end, not only the previous probabilities $p_c^{t-1}(x,y)$ and the results of the current static recognition $Q_c^t(x,y)$ have to be combined but also the binary results of the tracking decision in the previous step $T_c^{t-1}(x,y)$ have to be considered, since this permits to filter some possible misclassifications made by the static classifier. Typically, some background spots are erroneously classified as part of an object and this can be detected if these spots are situated far from the last known position of the object.

Therefore, the update function f for the dynamic class probabilities can be defined as follows (for some adaptive parameters α_c^t, $0<\alpha_c^t<1$):

$$p_c^t(x,y)=\frac{\alpha_c^t T_c^{t-1}(x,y)p_c^{t-1}(x,y)+(1-\alpha_c^t)Q_c^t(x,y)}{\sum_{k=1}^{N+1}\left(\alpha_k^t T_k^{t-1}(x,y)p_k^{t-1}(x,y)+(1-\alpha_k^t)Q_k^t(x,y)\right)} \tag{8}$$

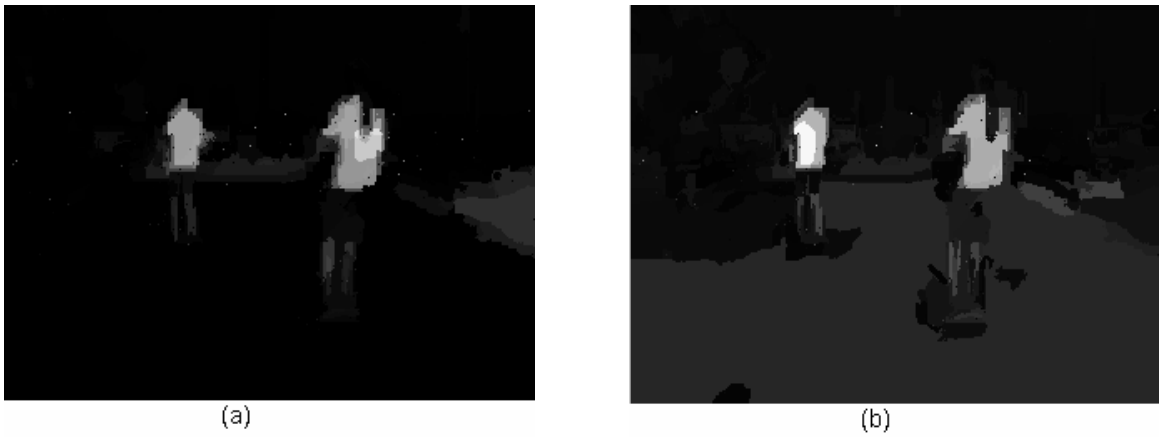


Figure 8 Dynamic probability images coming from (a) neural network, (b) Bayesian method, in the dynamic recognition module.

A tracking image for the background, which is required in the previous equation, can be defined simply as

$$T_{N+1}^t(x,y)=\begin{cases}1 & \text{if } T_c^t(x,y)=0 \quad \forall c: 1\le c\le N \\ 0 & \text{otherwise}\end{cases} \tag{9}$$

and computed after the tracking images for the objects.

The parameter α_c^t that weights the influence of the previous probabilities must be adapted depending on the apparent motion of the tracked object of class c. If this motion is very slow, α_c^t should reach a maximum α_{max} closer to 1, whereas if the motion is very fast, α_c^t should reach a minimum α_{min} closer to 0. In order to a set a proper value for α_c^t

ISBN:978-84-693-3387-7/DL:T.1001-2010

the areas (A_c^{t-1} and A_c^{t-2}) and mass centers (C_c^{t-1} and C_c^{t-2}) of the object in the two previous tracking images are used in the following way.

Let $r_c^{t-1} = \sqrt{A_c^{t-1}/\pi}$ and $r_c^{t-2} = \sqrt{A_c^{t-2}/\pi}$ be the estimates of the object radius in the two previous frames obtained by imposing a circular area assumption. Let $d_c = \left| C_c^{t-1} - C_c^{t-2} \right|$ be the estimated displacement of the object in the 2D image and let $d_c^{\max} = r_c^{t-1} + r_c^{t-2}$ be the maximum displacement yielding some overlapping between the two former circles.

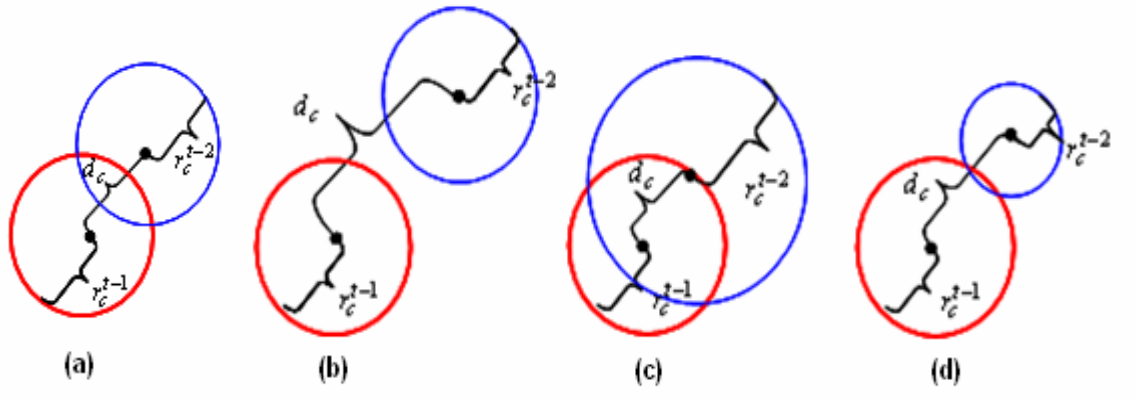


Figure 9 Rough estimate of object's motion using circular area assumption (a) Slow motion, (b) Fast motion, (c) Decreasing size,(d) Growing size.

If $d_c \geq d_c^{\max}$ we would like to set $\alpha_c^t = \alpha_{\min}$, whereas if $d_c = 0$ then the value of α_c^t should be set according to the change of the object apparent size: Let $s_c = \left| r_c^{t-1} - r_c^{t-2} \right| / \max\left(r_c^{t-1}, r_c^{t-2} \right)$ be a scale change ratio. If $s_c = 0$ (unchanged object size) then we would like to set $\alpha_c^t = \alpha_{\max}$ whereas in the extreme case $s_c = 1$ then we would set $\alpha_c^t = \alpha_{\min}$ again. Combining linearly both criteria, displacement and scale change, we define the prior value

$$\hat{\alpha}_c^t = \alpha_{\max} - \left(\alpha_{\max} - \alpha_{\min} \right) d_c / d_c^{\max} - \left(\alpha_{\max} - \alpha_{\min} \right) s_c \tag{10}$$

(which satisfies $\hat{\alpha}_c^t \leq \alpha_{\max}$) and finally the parameter α_c^t is set as follows:

$$\alpha_c^t = \begin{cases} \alpha_{\min} & \text{if } d_c \geq d_c^{\max} \\ \alpha_{\min} & \text{if } d_c < d_c^{\max} \wedge \hat{\alpha}_c^t \leq \alpha_{\min} \\ \hat{\alpha}_c^t & \text{if } d_c < d_c^{\max} \wedge \hat{\alpha}_c^t > \alpha_{\min} \end{cases} \tag{11}$$

The constants α_{min} and α_{max} were set to 0.1 and 0.6, respectively, in our experiments (see Chapter 5).

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

4.3 Tracking decision module

As depicted in Figure 3, the tracking images $T_c^t(x,y)$ for the objects ($1 \le c \le N$) can be calculated dynamically using the pixels probabilities $p^t(x,y)$ according to some decision function d. However, this function involves some additional arguments and results, as explained next.

To give an initial estimate of the foreseen translation and rescaling of the object in the current step, the measurements of both the object mass center and area in the tracking images of the two previous steps are required. Hence, the areas A_c^{t-1} and A_c^{t-2} and the mass centers C_c^{t-1} and C_c^{t-2}, already used in the dynamic recognition module as we have seen, must also be supplied here. The application of the estimated transformation to the previous tracking image $T_c^{t-1}(x,y)$ will serve to reduce the image area to explore using the class probabilities while filtering (blacking) the rest. This strategy alone permits to track visible objects reasonably well [Amézquita, 2006,2007] but it fails completely if the object becomes occluded for some frames [Amézquita , 2009].

In order to cope with occlusion, more information is needed in the decision function d. The key point is to distinguish between the *a posteriori* tracking image $T_c^t(x,y)$ and an *a priori* prediction $\hat{T}_c^t(x,y)$, which could maintain some relevant information of the object before the occlusion such as area and movement. The object mass center C_c^t and area A_c^t needed for tracking should be measured either from $T_c^t(x,y)$ or $\hat{T}_c^t(x,y)$ depending on whether the object is visible or occluded. Hence, an occlusion flag O_c^t has to be determined as an additional result. Moreover, the two previous flags O_c^{t-1} and O_c^{t-2} help to know whether the object is entering or exiting an occlusion. In addition, $\vec{m}_c^t$ is a movement weighted average vector that represents the past trajectory direction of the tracked object, which is useful to solve some ambiguous cases that happen when the object crosses or exits an occlusion by another object with a similar appearance (same-class occlusion). Finally, it should be taken into account that the uncertainty in the prediction $\hat{T}_c^t(x,y)$ grows as the number of consecutive frames the object is occluded increases. In the original method described in [Amézquita, 2007], two constant parameters ε and δ were used to define an uncertainty region around each pixel transformation. Since we want to adjust the level of uncertainty based on the duration of the occlusion, these parameters have to be adaptive for each object, i.e. ε_c^t and δ_c^t. Summarizing, the decision function d involves the following arguments and results:

$$\left\langle T_c^t(x,y), \hat{T}_c^t(x,y), \vec{m}_c^t, O_c^t, C_c^t, A_c^t, \varepsilon_c^t, \delta_c^t \right\rangle = d\begin{pmatrix} p^t(x,y), T_c^{t-1}(x,y), \hat{T}_c^{t-1}(x,y), \\ \vec{m}_c^{t-1}, O_c^{t-1}, O_c^{t-2}, C_c^{t-1}, C_c^{t-2}, \\ A_c^{t-1}, A_c^{t-2}, \varepsilon_c^{t-1}, \delta_c^{t-1} \end{pmatrix} \qquad (12)$$

This function is described in detail in the next subsections, which cover the different independent sub-modules of the tracking decision module. Figure 10 illustrates graphically some of the calculations that are explained in what follows.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

4.3.1 A priori prediction of the tracked objects

The first step is to give *a priori* estimates of the mass center and area of the object in time *t*. The mass center is predicted as follows:

$$\hat{C}_c^t = \begin{cases} C_c^{t-1} & \text{if } O_c^{t-2} \wedge \neg O_c^{t-1} \\ 2C_c^{t-1} - C_c^{t-2} & \text{otherwise} \end{cases} \tag{13}$$

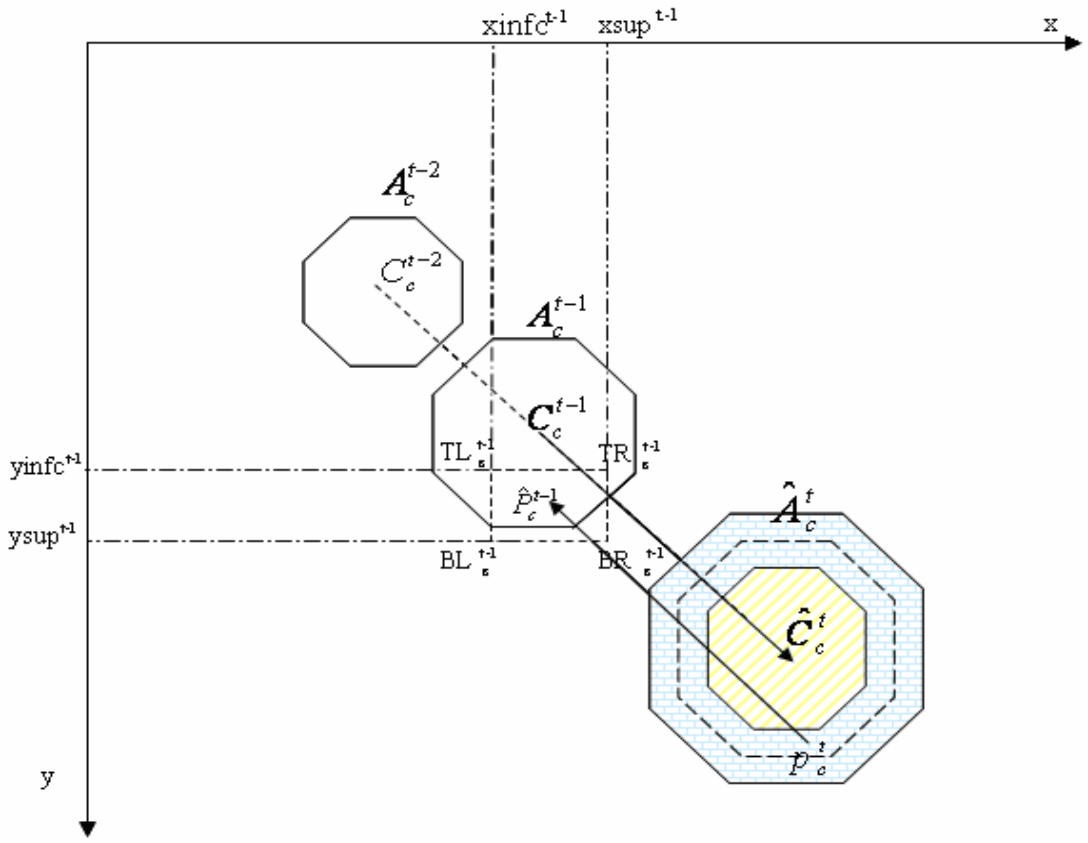


Figure 10 Geometrical illustration of the tracking process. Estimates of object's area and mass center for step t are computed from previous values in t-1 and t-2. For each pixel in step t a rectangular region in step t-1 is determined which allows the assignment to the pixel of one of three labels: "certainly belonging to the object" (yellow diagonal-bar-shaded region), "uncertain" (blue brick-shaded region) and "certainly not belonging to the object" (the rest of the image).

When the object is exiting an occlusion, C_c^{t-2} is not reliable enough to be used together with C_c^{t-1} to predict the next movement; therefore, a conservative estimate is given, just the previous measured value. In the rest of cases (the object is visible, is occluded or is entering an occlusion), a constant rate prediction is used. Note, however, that when the object is occluded, the mass center is not measured on the *a posteriori* tracking image, but on the *a priori* one, as we will see later.

It is interesting to notice that the above constant rate prediction can be proved to be equivalent to the one given by a linear Kalman filter for a particular setting of the filter parameters and equations. Let $w_c^{t+1} = Aw_c^t + Bu_c^t + \omega_c^t$ and $d_c^t = Hw_c^t + v_c^t$ be respectively the state and measurement equations of a linear Kalman filter (KF) for predicting the

ISBN:978-84-693-3387-7/DL:T.1001-2010

mass center of object *c*. If we set *A=I*, *B=I*, *H=I*, $d_c^t = C_c^t$ as the measurement, $u_c^t = C_c^t - C_c^{t-1}$ as the input and $R^t=0$ as the covariance matrix of the measurement noise v_c^t (which is assumed to be zero), then the *a priori* and *a posteriori* estimates of state w_c^t given by the KF are $2C_c^{t-1} - C_c^{t-2}$ and C_c^t respectively.

The *a priori* estimate of the object area is calculated as follows:

$$\hat{A}_c^t = \begin{cases} \left(A_c^{t-1}\right)^2 / A_c^{t-2} & \text{if } \neg O_c^{t-2} \wedge \neg O_c^{t-1} \\ A_c^{t-1} & \text{otherwise} \end{cases} \tag{14}$$

If the object has been visible in the two previous frames, a constant rate of a scale factor is used to predict the area. It can be proved that this prediction is equivalent to the one given by a (non-linear) extended Kalman filter for a particular setting of the filter parameters and equations. Let $w_c^{t+1} = f\left(w_c^t, u_c^t, \omega_c^t\right)$ and $d_c^t = h\left(w_c^t, v_c^t\right)$ be the state and measurement equations, respectively, of an extended Kalman filter (EKF) for predicting the area of object *c*. If we set $f\left(w_c^t, u_c^t, \omega_c^t\right) = w_c^t u_c^t + \omega_c^t$, $h\left(w_c^t, v_c^t\right) = w_c^t + v_c^t$, $d_c^t = A_c^t$ as the measurement, $u_c^t = A_c^t / A_c^{t-1}$ as the input and $R^t=0$ as the covariance matrix of the measurement noise v_c^t (which is assumed again to be zero), then the *a priori* and *a posteriori* estimates of state w_c^t given by the EKF are $(A_c^{t-1})^2 / A_c^{t-2}$ and A_c^t respectively. In the rest of cases (the object is occluded or is entering or exiting an occlusion), the area is supposed to remain constant.

From these predictions, a change of coordinates transformation can also be estimated that maps each pixel $P_c^{t-1} = (x_c^{t-1}, y_c^{t-1})$ of the object *c* in step *t-1* (maybe occluded) into its foreseen position in step *t*:

$$\hat{P}_c^t = \hat{C}_c^t + \left(P_c^{t-1} - C_c^{t-1}\right)\sqrt{\hat{A}_c^t / A_c^{t-1}} \tag{15}$$

Actually, we are interested in applying the transformation in the inverse way, i.e. to know which is the expected corresponding position in time *t*-1, $\hat{P}_c^{t-1} = \left(\hat{x}_c^{t-1}, \hat{y}_c^{t-1}\right)$, of a given pixel $P_c^t = (x_c^t, y_c^t)$ in *t*:

$$\hat{P}_c^{t-1} = C_c^{t-1} + \frac{\left(P_c^t - \hat{C}_c^t\right)}{\sqrt{\hat{A}_c^t / A_c^{t-1}}} \tag{16}$$

This is enough to compute the *a priori* tracking image $\hat{T}_c^t(x,y)$ in time *t*, either from the previous *a posteriori* or *a priori* tracking image, depending on the previous occlusion flag:

$$\hat{T}_c^t(x,y) = \begin{cases} T_c^{t-1}\left(\hat{x}_c^{t-1}, \hat{y}_c^{t-1}\right) & \text{if } \neg O_c^{t-1} \\ \hat{T}_c^{t-1}\left(\hat{x}_c^{t-1}, \hat{y}_c^{t-1}\right) & \text{otherwise} \end{cases} \tag{17}$$

where the values of $\hat{x}_c^{t-1}$, $\hat{y}_c^{t-1}$ are clipped whenever is necessary to keep them within the range of valid coordinates.

ISBN:978-84-693-3387-7/DL:T.1001-2010

4.3.2 First computation of the tracking images

To compute the *a posteriori* tracking image $T_c^t(x,y)$, the pixel class probabilities $p^t(x,y)$ are taking into account only in some image region that is determined from $T_c^{t-1}(x,y)$ or $\hat{T}_c^{t-1}(x,y)$ (depending on O_c^{t-1}) and the tolerance parameters ε_c^t and δ_c^t. Since the estimates of the translation and scale parameters in the coordinate transformation can be inaccurate, we define a rectangular region of possible positions for each pixel by specifying some tolerances in these estimates. To this end, we use the adaptive parameters ε_c^t and δ_c^t, which must be positive values to be set in accordance with our confidence in the translation and scale estimates respectively (the most confidence the smallest tolerance and vice versa), and which are adjusted according to the following rules:

$$\varepsilon_c^t = \begin{cases} \min\left(\varepsilon_{\max}, \varepsilon_c^{t-1} + \varepsilon_{incr}\right) & \text{if } O_c^{t-2} \vee O_c^{t-1} \\ \varepsilon_{ini} & \text{otherwise} \end{cases} \tag{18}$$

$$\delta_c^t = \begin{cases} \min\left(\delta_{\max}, \delta_c^{t-1} + \delta_{incr}\right) & \text{if } O_c^{t-2} \vee O_c^{t-1} \\ \delta_{ini} & \text{otherwise} \end{cases} \tag{19}$$

where ε_{ini}, δ_{ini} are default values, ε_{max}, δ_{max} are the maximal allowed values and ε_{incr}, δ_{incr} are the respective increases for each successive step under occlusion. Note that the tolerances keep on growing when exiting an occlusion until the object has been visible in the two previous frames; this is needed to detect and track the object again.

Let $C_c^{t-1} = \left(x_{Cc}^{t-1}, y_{Cc}^{t-1}\right)$ and $\hat{C}_c^t = \left(\hat{x}_{Cc}^t, \hat{y}_{Cc}^t\right)$ be respectively the previous mass center and the *a priori* estimate of the current mass center. The four vertices of the rectangular uncertainty region centered at $\hat{P}_c^{t-1}$ are denoted (top-left) $TL_c^{t-1} = (xinf_c^{t-1}, yinf_c^{t-1})$, (top-right) $TR_c^{t-1}=(xsup_c^{t-1}, yinf_c^{t-1})$, (bottom-left) $BL_c^{t-1}=(xinf_c^{t-1}, ysup_c^{t-1})$ and (bottom-right) $BR_c^{t-1} = (xsup_c^{t-1}, ysup_c^{t-1})$, where:

$$xinf_c^{t-1} = \begin{cases} x_{Cc}^{t-1} + \dfrac{\left(x_c^t - \hat{x}_{Cc}^t - \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t + \delta_c^t \hat{A}_c^t\right) / A_c^{t-1}}} & \text{if } x_c^t - \hat{x}_{Cc}^t - \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right| \geq 0 \\ x_{Cc}^{t-1} + \dfrac{\left(x_c^t - \hat{x}_{Cc}^t - \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t - \delta_c^t \hat{A}_c^t\right) / A_c^{t-1}}} & \text{otherwise} \end{cases} \tag{20}$$

$$xsup_c^{t-1} = \begin{cases} x_{Cc}^{t-1} + \dfrac{\left(x_c^t - \hat{x}_{Cc}^t + \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t - \delta_c^t \hat{A}_c^t\right) / A_c^{t-1}}} & \text{if } x_c^t - \hat{x}_{Cc}^t + \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right| \geq 0 \\ x_{Cc}^{t-1} + \dfrac{\left(x_c^t - \hat{x}_{Cc}^t + \varepsilon_c^t \left|\hat{x}_{Cc}^t - x_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t + \delta_c^t \hat{A}_c^t\right) / A_c^{t-1}}} & \text{otherwise} \end{cases} \tag{21}$$

ISBN:978-84-693-3387-7/DL:T.1001-2010

$$yinf_c^{t-1}=\begin{cases} y_{Cc}^{t-1}+\dfrac{\left(y_c^t-\hat{y}_{Cc}^t-\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t+\delta_c^t\hat{A}_c^t\right)/A_c^{t-1}}} & \text{if } y_c^t-\hat{y}_{Cc}^t-\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\geq 0 \\ y_{Cc}^{t-1}+\dfrac{\left(y_c^t-\hat{y}_{Cc}^t-\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t-\delta_c^t\hat{A}_c^t\right)/A_c^{t-1}}} & \text{otherwise} \end{cases} \tag{22}$$

$$ysup_c^{t-1}=\begin{cases} y_{Cc}^{t-1}+\dfrac{\left(y_c^t-\hat{y}_{Cc}^t+\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t-\delta_c^t\hat{A}_c^t\right)/A_c^{t-1}}} & \text{if } y_c^t-\hat{y}_{Cc}^t+\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\geq 0 \\ y_{Cc}^{t-1}+\dfrac{\left(y_c^t-\hat{y}_{Cc}^t+\varepsilon_c^t\left|\hat{y}_{Cc}^t-y_{Cc}^{t-1}\right|\right)}{\sqrt{\left(\hat{A}_c^t+\delta_c^t\hat{A}_c^t\right)/A_c^{t-1}}} & \text{otherwise} \end{cases} \tag{23}$$

The values of $xinf_c^{t-1}$, $yinf_c^{t-1}$, $xsup_c^{t-1}$ and $ysup_c^{t-1}$ are clipped whenever is necessary to keep them within the range of valid coordinates.

Now, each pixel $P_c^t = (x_c^t, y_c^t)$ is labeled, with respect to object c, as one of three labels ("certainly belonging to the object c", "certainly not belonging to the object c" or "uncertain") as follows.

If O_c^{t-1} is false then: if all the pixels in the rectangular region delimited by TL_c^{t-1}, TR_c^{t-1}, BL_c^{t-1}, BR_c^{t-1} have a common value of 1 in $T_c^{t-1}(x,y)$, it is assumed that P_c^t is definitely inside and certainly belongs to object c; to the contrary, if they have a common value of 0 in $T_c^{t-1}(x,y)$, it is assumed that P_c^t is clearly outside and certainly does not belong to object c; otherwise, the rectangular region contains both 1 and 0 values, the pixel P_c^t is initially labeled as "uncertain".

However, if O_c^{t-1} is true, $T_c^{t-1}(x,y)$ will represent a totally or partially occluded object and we cannot rely on it, but on the predicted $\hat{T}_c^{t-1}(x,y)$, which is based on information previous to the occlusion. If all the pixels in the rectangular region delimited by TL_c^{t-1}, TR_c^{t-1}, BL_c^{t-1}, BR_c^{t-1} have a common value of 0 in $\hat{T}_c^{t-1}(x,y)$, it is assumed that P_c^t does not belong to object c; otherwise (the rectangular region contains both 1 and 0 values or only 1 values), the pixel P_c^t is labeled as "uncertain".

Only for the uncertain pixels (x,y) the dynamic probabilities $p^t(x,y)$ will be used. Recall that these probabilities will have been updated previously from the object recognition results in time t, $Q^t(x,y)$, also expressed as probabilities. More precisely, we propose the following rule to compute the value of each pixel of the *a posteriori* tracking image for object c in time t:

$$T_c^t(x,y)=\begin{cases} 1 & \begin{array}{l}\text{if } (x,y) \text{ certainly belongs to object } c \\ \textit{or} \text{ it is uncertain and } p_c^t(x,y) \text{ is the} \\ \text{maximum for all } c \text{ between } 1 \text{ and } N+1\end{array} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

4.3.3 Post-processing of the tracking images

Sometimes, the tracking images $T_c^t(x,y)$ obtained by applying eq. (24) contain disconnected regions of 1-valued pixels, or, said in other words, more than one connected component T_{ci}^t, $1 \le i \le I$, $I > 1$. This may be produced by a variety of causes, mainly segmentation or recognition errors, but also may be due to possible partial occlusions of the target object by an object of a different class. In addition, a particular problem that leads to object split occurs immediately after a same-class crossing or occlusion: when the target object has just finished crossing another object or region which is recognized to be in the same class (distracter), then the tracking method is misled to follow both the object and the distracter. It is very difficult to devise a general method that can always distinguish between erroneous components due to noise or distracters and correct object components, especially if separated components are allowed to cope with partial occlusions, but some useful heuristics based on properties such as size, movement or shape may be defined that work reasonably well in a majority of cases.

In order to eliminate noisy regions and to circumvent the same-class crossing problem, while handling partial occlusions at the same time, we propose a post-processing step that removes from $T_c^t(x,y)$ some possible artifacts or distracters (setting some initially 1-valued pixels to zero). In fact, this step is only carried out if $T_c^t(x,y)$ contains more than one component. In such a case, we need to choose which components T_{ci}^t to keep (one or more) and which to discard. To this end, three heuristic filters are applied sequentially, whenever two or more components remain before the filter application.

The first filter is aimed at deleting small noisy regions and is solely based on their size. Let A_{ci}^t be the area of the i-th connected component T_{ci}^t and let $\text{Area}(T_c^t)$ be the total area covered by 1-valued pixels in $T_c^t(x, y)$. The i-th component is removed if the ratio $A_{ci}^t / \text{Area}(T_c^t)$ is below a given threshold κ, e.g. $\kappa = 0.15$.

The next images show an example of application of this first filter, if the ratio $A_{ci}^t / \text{Area}(T_c^t) < 0.15$ delete region A_{ci}^t.

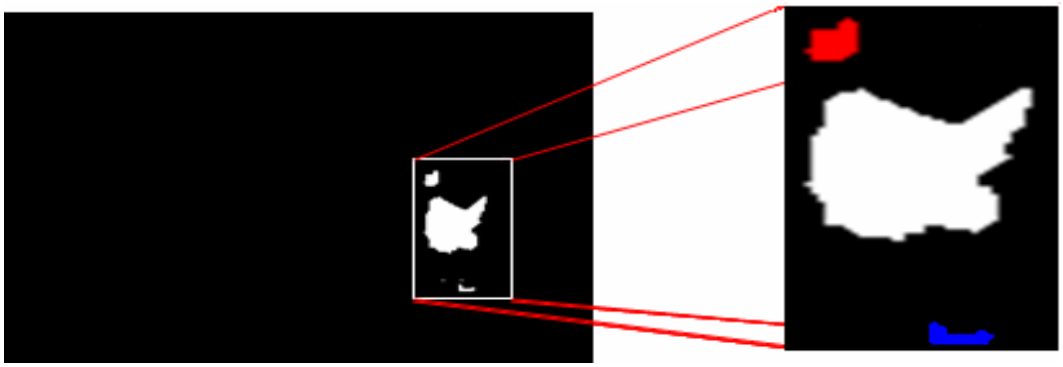


Figure 11 Application of first filter on the intermediate tracking.Region 1 red area =61, Region 2 blue area=23, Region 3 white area=899, $\text{Area}(T_c^t)$ =983; $A_{11}^{76} / \text{Area}(T_c^t)$ =61/983=0,0620; $A_{12}^{76} / \text{Area}(T_c^t)$ =23/983=0,0233; $A_{13}^{76} / \text{Area}(T_c^t)$ =899/983=**0,9145**.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

The following image shows the selected region that passed the area ratio threshold.

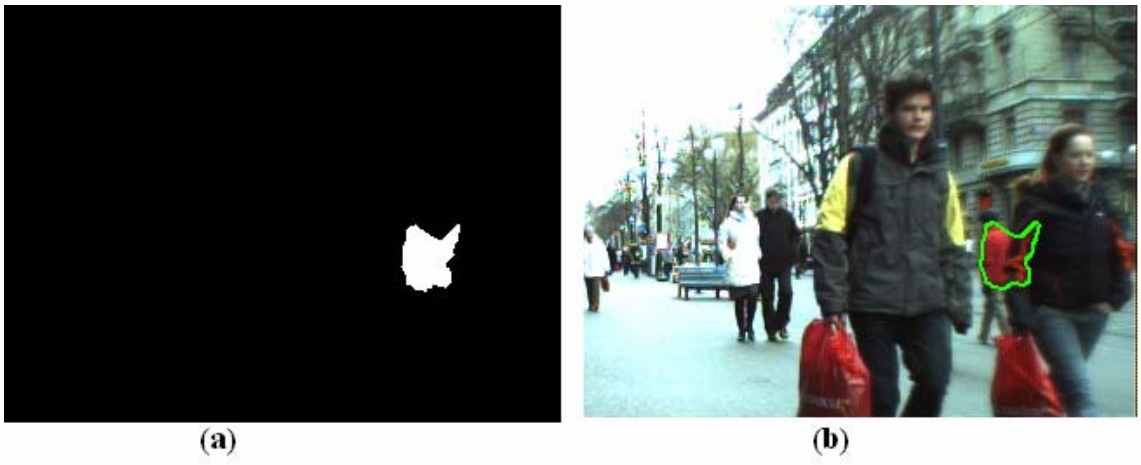


Figure 12 First filter results. (a) Binary final tracking, (b) The final tracking area highlighted in the original image.

The second filter is aimed at deleting distracters, including those appearing after same-class occlusion, and is based on a comparison between the apparent movement of the remaining components and the previous recent trajectory of the tracked object represented by the movement vector $\vec{m}_c^{t-1}$. Let C_{ci}^t be the mass center of the *i*-th connected component T_{ci}^t and define an associated movement vector $\vec{z}_{ci}^t = C_{ci}^t - C_c^{t-1}$ for each component. Then,

$$\cos\theta_{ci} = \frac{\langle \vec{z}_{ci}^t, \vec{m}_c^{t-1} \rangle}{\left\| \vec{z}_{ci}^t \right\| \left\| \vec{m}_c^{t-1} \right\|} \tag{25}$$

is a measure of the alignment between the vectors $\vec{m}_c^{t-1}$ and $\vec{z}_{ci}^t$, which is only reliable for our purposes if both vectors have a sufficiently large norm. Otherwise, the angle θ_{ci} can be considered rather random, since may be affected a lot by adding small perturbations on the vectors. Consequently, abrupt trajectory changes (greater than 90 degrees) are penalized if we remove the *i*-th component T_{ci}^t when the condition $\cos\theta_{ci} < 0 \wedge \left\| \vec{z}_{ci}^t \right\| \geq \lambda \wedge \left\| \vec{m}_c^{t-1} \right\| \geq \lambda$ holds, where λ is another threshold, e.g. $\lambda = 3$. However, to guarantee that at least one component is kept, the remaining component for which $\vec{z}_{ci}^t$ is the most collinear vector with respect to $\vec{m}_c^{t-1}$, i.e. the component *i* such that

$$i = \arg\max \left\{ \frac{\langle \vec{z}_{ci}^t, \vec{m}_c^{t-1} \rangle}{\left\| \vec{z}_{ci}^t \right\|} \right\} \tag{26}$$

is never removed by this second filter.

The following figures show an example of application of this second filter.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

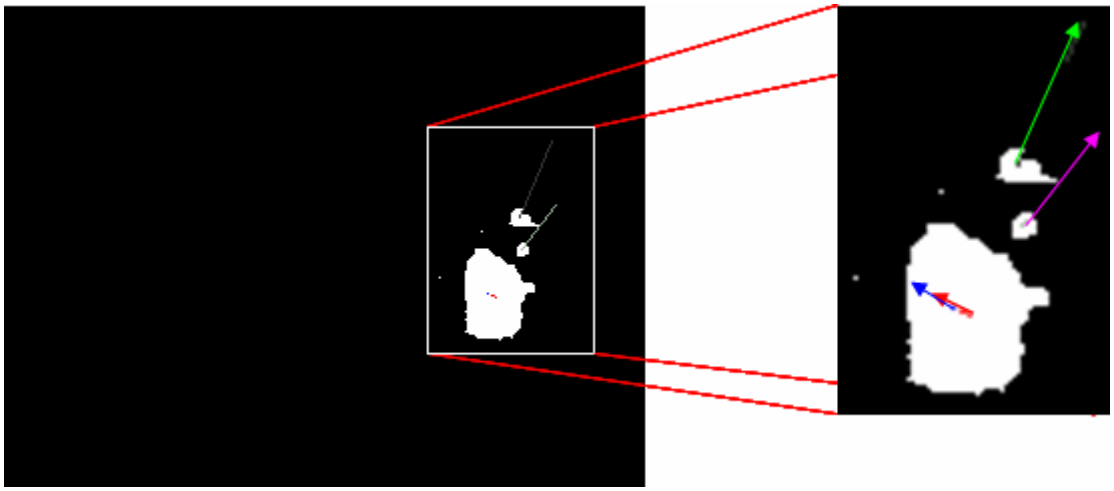


Figure 13 Application of second filter on the intermediate tracking. Blue arrow repesents the $\vec{m}_c^{t-1}$ previous recent trajectory of the tracked object, the red arrow represents the most collinear vector with respect to $\vec{m}_c^{t-1}$, and green, magenta arrow repesent the other $\vec{z}_{ci}^{t}$ vectors.

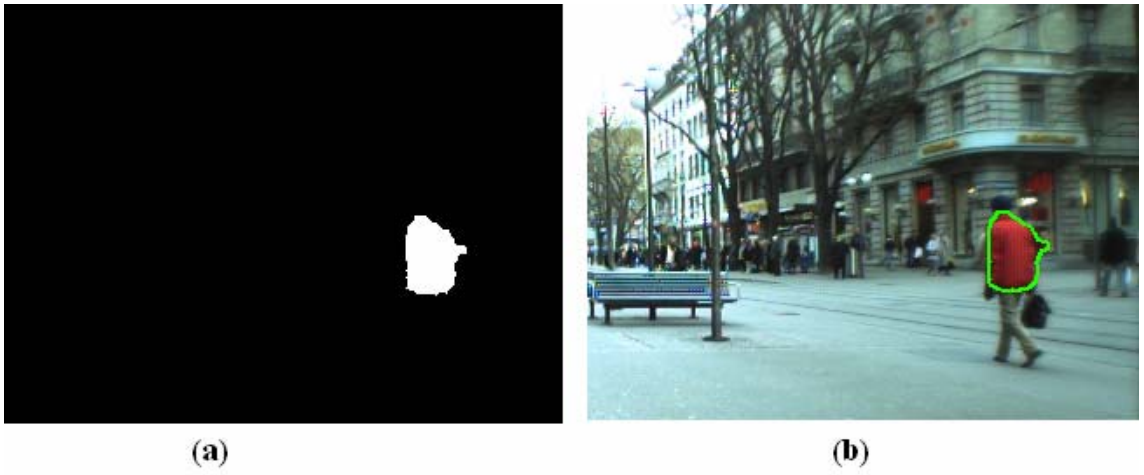


Figure 14 Second filter result. (a) Binary final tracking, (b) Final tracking area highlighted in the original image.

The third filter is also aimed at deleting distracters and is based on a comparison between the shapes of the components and that of the *a priori* prediction of the target object (represented by the 1-valued region in $\hat{T}_c^t(x,y)$). For each remaining component T_{ci}^t, the *a priori* prediction of the target object is moved from its original center $\hat{C}_c^t$ to the component center C_{ci}^t, thus resulting in a translated copy $\hat{T}_{ci}^t(x,y)$, and the spatial overlap between both shapes is then measured as follows:

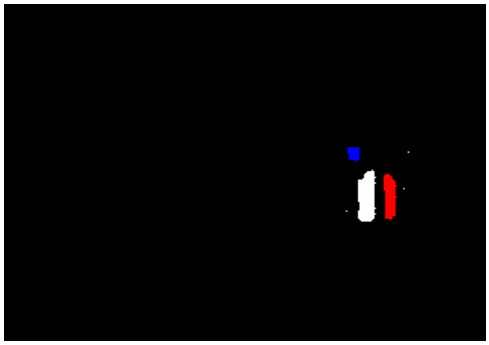
$$SO_{ci} = \frac{\text{Area}\left(T_{ci}^t \cap \hat{T}_{ci}^t\right)}{\text{Area}\left(T_{ci}^t \cup \hat{T}_{ci}^t\right)} \tag{27}$$

The components having a spatial overlap $SO_{ci} < 0.243$ (which is the overlap obtained between two circles of the same size when one of the centers is located in the border of

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

the other circle) are deleted in this third filter, unless SO_{ci} is the maximum spatial overlap of the remaining components. This exception guarantees the persistence of at least one component in the final tracking image.

The following figures show an example of how the spatial overlap between T_{ci}^{t} and $\hat{T}_{ci}^{t}$ is calculed for one of the regions in the intermediate tracking image.

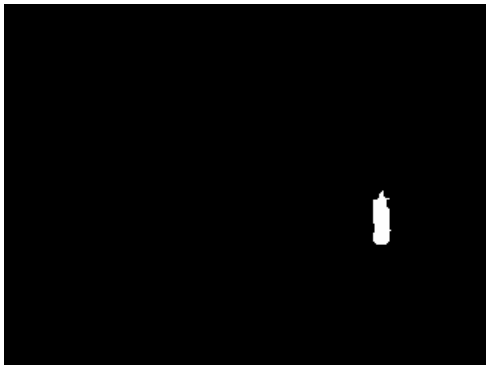


(a) Intermediate tracking. Blue color R1 White color R2, Red color R3.

(b) A priori prediction of the target object.

(c) Tracking Rregion R2 displaced.

(d) Region R2 of $\hat{T}_{ci}^{t}$ A priori prediction.



(e) AND image between R2 region of $\hat{T}_{ci}^{t}$ and R2 region of T_{ci}^{t}

(f) OR image between R2 region of $\hat{T}_{ci}^{t}$ and R2 region of T_{ci}^{t}

Figure 15 Application of third filter on the intermediate tracking.

In the example above, the overlap resutls for each region of T_{ci}^{t} are

$SO_{11} = 70/433 = 0.161663$

$SO_{12} = 329/467 =$ **0.704497**

$SO_{13} = 218/437 =$ **0.498856**

ISBN:978-84-693-3387-7/DL:T.1001-2010

The region R1 of T_{ci}^t is deleted because $SO_{ci} < 0.243$ while R2 and R3 remain.

Figure 16. Final tracking. R2 and R3 regions selected area highlighted in the original image

As a result of the post-processing, the pixels of all the components T_{ci}^t removed by any of the three filters are set to zero in the final tracking image $T_c^t(x, y)$.

4.3.4 Determination of occlusion and geometric measurements

Once both $T_c^t(x, y)$ and $\hat{T}_c^t(x, y)$ have been determined, it is possible to detect the occurrence of an occlusion (i.e. to set the current occlusion flag) in the following way. Let Area (T_c^t) be the measured area of the 1-valued region in the final $T_c^t(x,y)$ and let Area($\hat{T}_c^t$) be the measured area of the 1-valued region in $\hat{T}_c^t(x, y)$. Then,

$$O_c^t = \begin{cases} \text{Area}\left(T_c^t\right) / \text{Area}\left(\hat{T}_c^t\right) < r_1 & \text{if } O_c^{t-1} \\ \text{Area}\left(T_c^t\right) / \text{Area}\left(\hat{T}_c^t\right) < r_2 & \text{otherwise} \end{cases} \tag{28}$$

where $0 < r_1 < r_2 < 1$ (for instance, r_1=0.5, r_2=0.75). Note that the condition for remaining in occlusion mode is harder than the condition for initiating an occlusion. This facilitates the recovery of the object track when exiting an occlusion or when a false occlusion has been detected.

Next, the *a posteriori* estimates of the object mass center and area are selected between those of the *a priori* and *a posteriori* tracking images based on the value of the occlusion flag:

$$C_c^t = \begin{cases} MC\left(T_c^t\right) & \text{if } \neg O_c^t \\ MC\left(\hat{T}_c^t\right) & \text{otherwise} \end{cases} \tag{29}$$

where $MC(T_c^t)$ is the measured mass center of the 1-valued region in the final $T_c^t(x,y)$ and $MC(\hat{T}_c^t)$ is the measured mass center of the 1-valued region in $\hat{T}_c^t(x, y)$, and

ISBN:978-84-693-3387-7/DL:T.1001-2010

$$A_c^t = \begin{cases} \text{Area}\left(T_c^t\right) & \text{if } \neg O_c^t \\ \text{Area}\left(\hat{T}_c^t\right) & \text{otherwise} \end{cases} \tag{30}$$

Finally, the movement weighted average vector $\vec{m}_c^t$ is updated afterwards as follows:

$$\vec{m}_c^t = \begin{cases} \left(\vec{v}_c^t + (t-1)\vec{m}_c^{t-1}\right)/t & \text{if } t < 1/\beta \\ \beta\,\vec{v}_c^t + (1-\beta)\vec{m}_c^{t-1} & \text{if } t \geq 1/\beta \end{cases} \tag{31}$$

where β is a positive parameter between 0 and 1, e.g. β=0.2, and $\vec{v}_c^t$ is the current movement defined by $\vec{v}_c^t = C_c^t - C_c^{t-1}$. Note that the second row in (31) is a typical moving average computation, while the first row denotes a simple average for the starting steps, and both give the same result for t=1/β.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 5. Experimental work and results

The experiments that have been performed are presented next divided in two sections. The first section presents the preliminary experiments that were realized without object occlusion. These include both the feature selection and static recognition results and the intial experiments on dynamic recognition and tracking (in which the static recognition module also is involved).

In the second section, a set of comparative tracking experiments including object occlusions are presented. These are also divided in two parts: the experimental results on video sequences taken with a still camera and the experimental results on video sequences taken with a moving camera.

5.1 Preliminary experiments without object occlusion

Two video sequences of 88 color images each, that correspond to the left and right sequences of a stereo vision system installed on the MARCO mobile robot at the IRI[1] research centre, were employed for an initial validation of the proposed approach. They show an indoor scene with slight changes in perspective and scale caused by the robot movement. In this scene we selected N=3 objects of interest: a box, a chair and a pair of identical wastebaskets put together side by side; and the objective was to discriminate them from the rest of the scene (background) and locate them in the images. Figure 17 displays three frames of the right sequence; the whole right sequence can be obtained in http://www.lsi.upc.edu/~alquezar/ris.avi. The slow relative motion of the objects in the sequences is due to the slow motion of the mobile robot during its navigation in an indoor environment, and this small displacement of the objects is an expected characteristic of the video sequences we wish to deal with.

Figure 17 Three consecutive frames of the right sequence.

Before segmentation, the images in the sequences were preprocessed by applying a median filter on the RGB planes to smooth the image and reduce some illumination reflectance effects and noise. Then, all images in both sequences were segmented independently using the Felzenszwalb-Huttenlocher algorithm [Felzenszwalb, 98], which is a pixel merge method based on sorted edge weights and minimum spanning tree. Figure 18 displays three frames of the segmented right sequence.

[1] IRI is an acronym for the *Institute of Robotics and Industrial Informatics*, CSIC-UPC, in Barcelona.

ISBN:978-84-693-3387-7/DL:T.1001-2010

Figure 18 Three consecutive frames of the segmented right sequence.

The output of the segmentation process for each image consists of a list of regions (spots) that partition the image in homogeneous pieces, where each region is defined by the set of coordinates of the pixels it contains. For each spot, its mass center and several features listed in the following subsection were computed.

The experimental work carried out with these image sequences can be split in two parts: the former covers the experiments related only to feature selection and the performance of the static recognition module (subsection 5.1.1), whereas the latter includes the experiments related mainly to dynamic recognition and tracking (subsection 5.1.2). This presentation order also coincides with the temporal order in which the experiments were performed. This is relevant because the static recognition module is also involved in the dynamic recognition and tracking results, so that the results of the first experiments were taken into account for the subsequent work.

5.1.1 Feature selection and static recognition results.

In order to be processed as a pattern by a neural network, a spot must be described by a feature vector. Table 1 displays the 14 variables that were initially considered to form the feature vector for training and testing the networks. In this subsection we will present classification results obtained from several different subsets of these 14 variables.

Two types of information were extracted from the spots: color and geometry. With regards to color, average and variance values for each one of the three RGB bands were calculated for each spot on the basis of the corresponding intensity values of the spot pixels in the original image (not in the segmented image, for which spot color variance would be zero). This is, the result of the segmentation algorithm served to identify the pixels of every spot, but the color characteristics of these pixels were taken from the original RGB image.

The geometrical information might include features related to position, orientation, size and shape. Because of the robot movement, we were mainly interested in shape descriptors that were invariant to translation and scale, and to this end, we decided to use the seven invariant geometric moments defined by Hu [Hu, 62]. In addition and since the range of variation of the objects' size was rather limited in the video sequence, we also calculated and used the size of each spot, i.e. its area measured in number of pixels.

For the calculation of the moments corresponding to a spot, all the pixels that form the spot are involved (not only its boundary pixels). More precisely, the seven invariant

ISBN:978-84-693-3387-7/DL:T.1001-2010

moments, independent of position and size of the region, that we used are defined by the following equations:

$$I_1=N(2,0)+N(0,2) \tag{32}$$

$$I_2=(N(2,0)-N(0,2))^2+4(N(1,1))^2 \tag{33}$$

$$I_3=(N(3,0)-3N(1,2))^2 +(3N(2,1)-N(0,3))^2 \tag{34}$$

$$I_4=(N(3,0)+N(1,2))^2+(N(2,1)+N(0,3))2 \tag{35}$$

$$I_5=(N(3,0)-3N(1,2))(N(3,0)+N(1,2))[(N(3,0)+N(1,2))^2 - 3(N(2,1)+N(0,3))^2] +(3N(2,1) – N(0,3))(N(2,1)+N(0,3))[3(N(3,0)+N(1,2))^2-(N(2,1)+N(0,3))^2] \tag{36}$$

$$I_6=(N(2,0) – N(0,2)) [(N(0,3)+N(1,2))^2-(N(2,1)+N(0,3))^2] +4N(1,1)(N(3,0)+N(1,2))(N(2,1)+N(0,3)) \tag{37}$$

$$I_7=(3N(2,1)-N(0,3)) (N(3,0)+N(1,2))[(N(3,0)+N(1,2))^2- 3(N(2,1)+N(0,3))^2] +(3N(1,2)-N(3,0))(N(2,1)+N(0,3))[3(N(3,0)+N(1,2)^2)-(N(2,1)+N(0,3))^2] \tag{38}$$

where N (p, q) are the normalized central moments of order two, which are given by:

$$N(p, q) = MC(p, q) / MC^{\beta}(0, 0) \;\; ; \beta = ((p + q) / 2) + 1 \tag{39}$$

$$MC(p,q) = \sum\sum (x-X)^{p} (y-Y)^{q} f(x,y) \tag{40}$$

$$MC(0,0)= \sum\sum f(x,y) \tag{41}$$

where f (x,y) is the intensity value of the pixel (x,y) in the segmented image, as given by the average of the three planes RGB, and (X,Y) are the mean coordinates of the spot. It must be noted that, in this case, all pixels in the same spot share the same value f(x,y), which depends on the color assigned to the spot as result of the segmentation process.

Number of variable	Feature
1	Size of spot
2	Average red plane
3	Average green plane
4	Average blue plane
5	I1RGB invariant moment
6	I2RGB invariant moment
7	I3RGB invariant moment
8	I4RGB invariant moment
9	I5RGB invariant moment
10	I6RGB invariant moment
11	I7RGB invariant moment
12	Variance red plane
13	Variance green plane
14	Variance blue plane

Table 1. Initial set of variables that form the feature vector for every spot.

ISBN:978-84-693-3387-7/DL:T.1001-2010

For object learning, spots selected through ROI (region-of-interest) windows in the left image sequence were collected to train the neural networks. These windows were manually marked on the images with a graphics device to encompass the three objects of interest and a large region on the floor. Figure 19 shows one of the images and its segmentation together with the ROI windows on them.

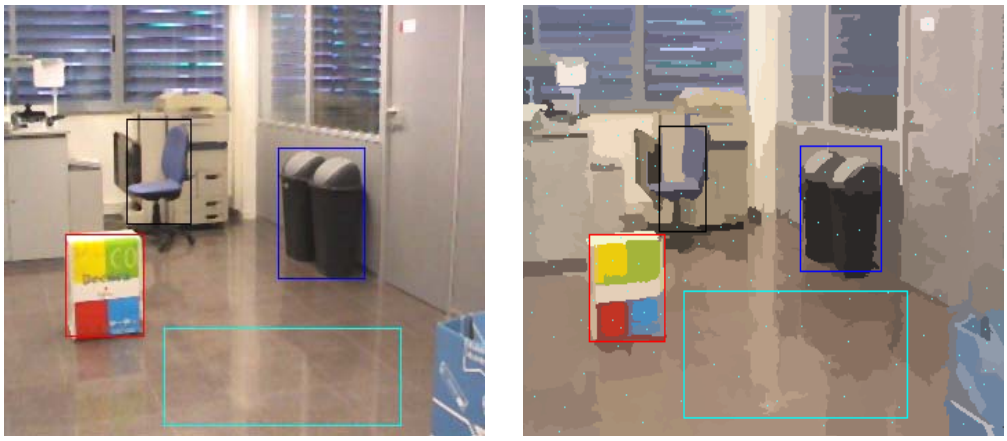


Figure 19 ROI windows. One of the original images (left) and the corresponding segmented image (right), with four boxes marked on them.Spot mass centers are also displayed in the right image.

The remaining set of spots, those with its mass center inside the ROI windows, was further filtered by removing all the spots with a size lower than 100 pixels, with the purpose of eliminating small noisy regions caused by segmentation defects. Hence, from the 88 images, a total number of 3,411 spots were finally chosen.

The inputs of the neural nets are the spot features and the target is the class that we impose to the spot. In order to assign a class label to each spot, to be used as target for the spot pattern in the neural network training and test processes, a simple decision was made: each one of the four ROI windows constituted a class and all the spots in a window were assigned the same class label. Note that this is a rough decision, since several background spots may be included in the ROI windows of the objects (especially in the case of the chair) and therefore are not correctly labeled really. Incorrectly labeled patterns are a clear source of error that puts some bounds on the level of classification accuracy that the learning system, in this case a neural net, may reach. However, we preferred to carry out this simple but more practical approach instead of manually labeling each spot, which is obviously a very tedious task.

For illustrative purposes, the spots of figure 19 that were assigned to the three classes associated with the objects of interest are displayed in figure 20; for each class, the union of selected spots is shown in the left and isolated spots that belong to the class are shown in the right.

ISBN:978-84-693-3387-7/DL:T.1001-2010

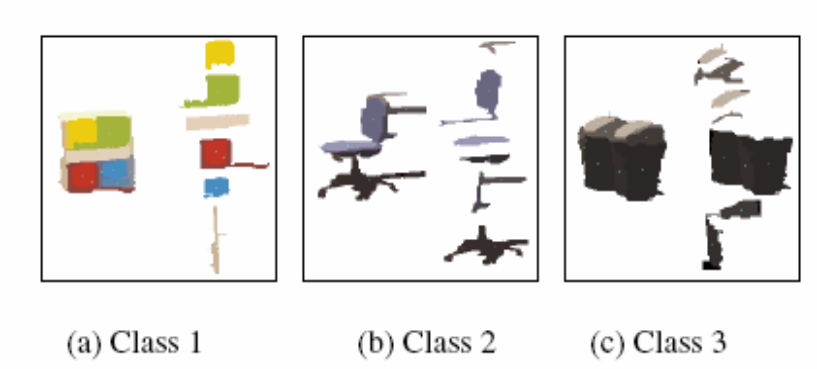


Figure 20 Selected spots.Labeling of the object spots from the segmented image in figure 19

In all the experiments we used neural nets with a feed-forward two-layer perceptron architecture using standard gradient descent through backpropagation as training algorithm. After some preliminary experiments, we set the number of hidden units to 180, although we observed that the results were not very sensitive to this choice. Hyperbolic tangent and sine functions were used as activation functions in the hidden layer and the output layer, respectively. For backpropagation, we set a learning rate of 0.003, a momentum parameter of zero and a maximum number of 500 training epochs for each run.

As mentioned before, a dataset containing 3,411 labeled patterns (spots) was available after the image segmentation and feature calculation processes described previously. For each subset of features that was tried, a double cross-validation procedure was carried out that generated 90 different partitions of the dataset, each including 80% of the patterns for the training set, 10% for the validation set and 10% for the test set. The validation sets were used for early stopping the training phase. Actually, the network chosen at the end of the training phase was the one that yielded the best classification result on the validation set among the networks obtained after each training epoch.

The results of the double cross-validation procedure obtained for different subsets of features are displayed in Table 2, ordered decreasingly by test classification performance. For each one of the three sets (training, validation and test set), the classification performance is measured as the average percentage of correctly classified patterns in the 90 cross-validation partitions, evaluated in the networks selected after training (the ones that maximize the performance on the validation set). It can be noted that similar results are obtained if the average RGB color features are taken into account, but the performance falls down dramatically when they are not used. The best result was 92.93% test classification performance for a subset comprising color features (both RGB averages and variances) and spot size (and without the shape invariant moments). Using only the seven invariant moments, the performance was almost as poor as that of a random classification decision rule.

In addition, starting from the architecture selected for the full set of features, a sequential backward selection method [Romero, 03] was applied trying to determine a good subset of input features by eliminating variables one by one and retraining the network each time a variable is temporarily removed. In this case, each partition of the cross-validation procedure divided the dataset in 60% of patterns for training and 40% for test (no validation set) and the training stop criterion was to obtain the best result in the training set for a maximum of 2,000 epochs. The results of the sequential backward

ISBN:978-84-693-3387-7/DL:T.1001-2010

feature selection clearly confirmed that invariant moments and RGB color variances were practically useless (since they were the first features removed without significance performance degradation) and that RGB color averages provided almost all the relevant information to classify the spots.

Classification performance (with feature subsets)

Feature subsets	Training	Validation	Test
spot size, average and variance RGB	94.20	93.38	92.93
spot size and average RGB	93.29	93.26	92.75
all 14 features	94.58	93.19	92.72
spot size, average RGB and three first moments	93.47	92.92	92.22
average RGB and three first invariant moments	92.11	92.37	91.93
average RGB and the seven invariant moments	92.04	92.35	91.60
spot size and variance RGB	62.12	63.54	63.06
spot size, variance RGB and three first moments	62.46	63.52	62.79
seven invariant moments and variance RGB	55.98	57.48	57.33
seven invariant moments	32.29	32.49	32.38

Table 2 classification results. It presents the classification results for several groups of selected variables to assess the relative importance of the different types of features (size, color averages, color variances and shape invariant moments).

The following experiment consisted in adding a reclassification module based on clustering to check whether the classification performance could be improved using spot spatial distributions in the image (see figure 21). Hence, a final step for spot classification involved the detection and reclassification of possibly misclassified spots based on the context information provided by the mass centers of the spots classified as the same class (or object) in the same frame.

For each one of the classes (or objects) o and for each frame f, a weighted mass center $wmc(o,f)$ was computed as

$$wmc(o,f) = \frac{\sum_{s=1}^{ns(o,f)} p(o/s)a(s)mc(s)}{\sum_{s=1}^{ns(o,f)} p(o/s)a(s)} \tag{42}$$

where $ns(o,f)$ is the number of spots classified as object o in frame f, $p(o|s)$ is the a-posteriori class probability of object o for spot s given by the net, and $a(s)$ and $mc(s)$ are respectively the area and mass center of s. Then, for every spot in the segmented image classified by the net as an object, the distance between its mass center and the weighted mass center of the assigned object was computed. If this distance exceeded a given threshold, the spot was marked as possibly misclassified and it was optionally reclassified to the object with the nearest mass center. Note that this step is a kind of spot clustering process that is inspired in both the dynamic and k-means clustering algorithms, but starting from the clusters (class assignments) given by the neural network.

ISBN:978-84-693-3387-7/DL:T.1001-2010

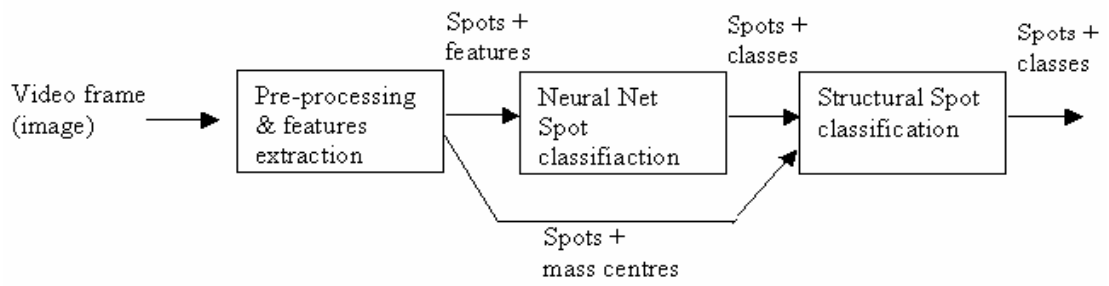


Figure 21 Classification process based on three main steps: first, the extraction of the spots and features; second, spot classification based on a neural net; and third, reclassification of the spots based on their distances with respect the other spots within the same class.

Using spot size and RGB averages and variances as features, the network (and associated dataset partition) that gave the best result in the training set (97.25%) was selected for computing the weighted mass centers and to assess the effect of the clustering process on the spot classification performance. Table 3 compares the results obtained without clustering with those obtained after clustering and reclassification to the nearest object. A 78.8% of the spots misclassified by the network were correctly reclassified by the clustering and only a 0.1% of the correctly classified spots were incorrectly reclassified.

Classification performance (with the best feature subsets)

Classifier	Training	Validation	Test
Only the neural network	97.25	95.01	96.18
Combining the neural net and the clustering	99.34	98.53	99.71

Table 3 Spot classification. Spot classification results before and after clustering using the net that maximized the result in the training set.

Figure 22 displays an example of the beneficial effects of performing the structural reclassification. In the left hand image, there are two spots that were misclassified by the net, one in the chair was classified as wastebasket and one in the wastebasket was classified as chair. These spots could be correctly reclassified after the structural reclassification, as shown in the right hand image.

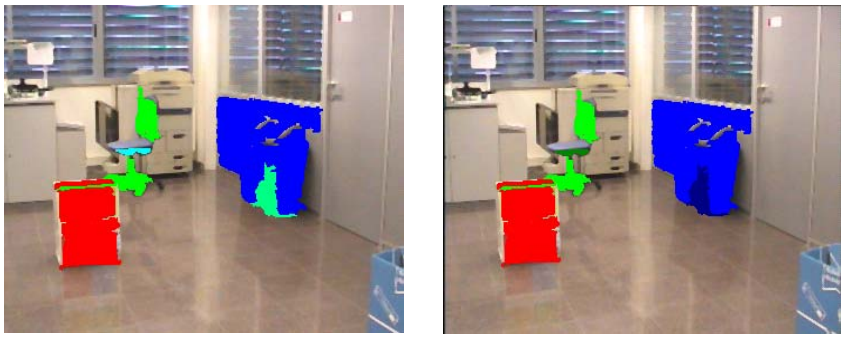


Figure 22 Spots classified as belonging to the three objects by the net (left) and the result of the reclassification after the clustering (right)

ISBN:978-84-693-3387-7/DL:T.1001-2010

Finally, the network selected in the previous experiment (trained from just the left image sequence) was applied to classify the spots in the right image sequence but only those within the same ROI windows, giving a 90% of correctly classified spots.

However, for the test phase, it is somewhat tricky to restrict the object recognition to predefined ROI windows, since we cannot rely on having the ROI windows marked on every frame in a realistic experimental scenario. Hence, in the next experiments reported in the following subsection, the same neural network trained from selected ROI windows in the left sequence was used, but the whole right sequence including all spots was taken for testing both object recognition and tracking. ROI windows for each object were only defined in the first frame to initialize the tracking images. To the contrary of the experiments that have been reported in this subsection, in the following experiments we were not so interested in achieving a high spot classification ratio but a sequence of tracking images of good quality for each object of interest, as a first validation of the methodology proposed in chapter 4.

5.1.2 Dynamic recognition and tracking results

In the next experiment, dynamic object recognition was achieved through the use of probability images and the update function given by equation (8) (see Section 4.2) and we employed the tracking decision function based on a simple predictive model that has been described in chapter 4 (except that occlusion flags were not computed). A block diagram of the whole process has been shown in figure 3.

For static object recognition, the trained neural network was applied to estimate the class probabilities for all the spots in the right image sequence. As mentioned before, the spot class probabilities were replicated for all the pixels in the same spot. In this case, the class assignments for each spot ($C^t(x,y) = \arg\max_c (Q_c^t(x,y))$ were not taken into account.

For object tracking in the right sequence, ROI windows for each one of the three objects were only marked in the first image to initialize the tracking process and the dynamic class probabilities. The α parameter in the probability update function was fixed to 0.5 for the three clsses. The uncertainty parameters ε and δ used en the computation in the tracking image were also fixed to 1 and 0.25, respectively, for the three classes.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

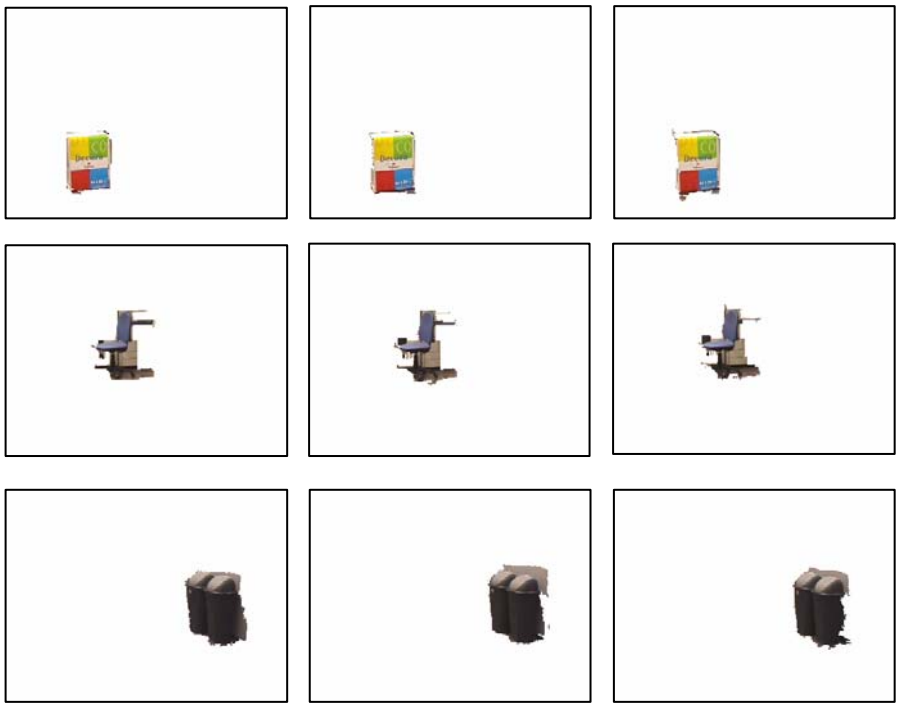


Figure 23 Tracking of the three objects of interest in the second experiment

Figures 23 and 24 show the results of tracking the three objects of interest in three consecutive frames that belong to the right sequence. In Figure 23, the corresponding binary images of tracking each object are applied as a transparency mask to the original images in order to visualize only the pixels considered to belong to the object (the rest of pixels are set to white). Similar results for the whole sequence can be seen in http://www.lsi.upc.edu/~alquezar/box.avi, chair.avi and basket.avi, respectively.

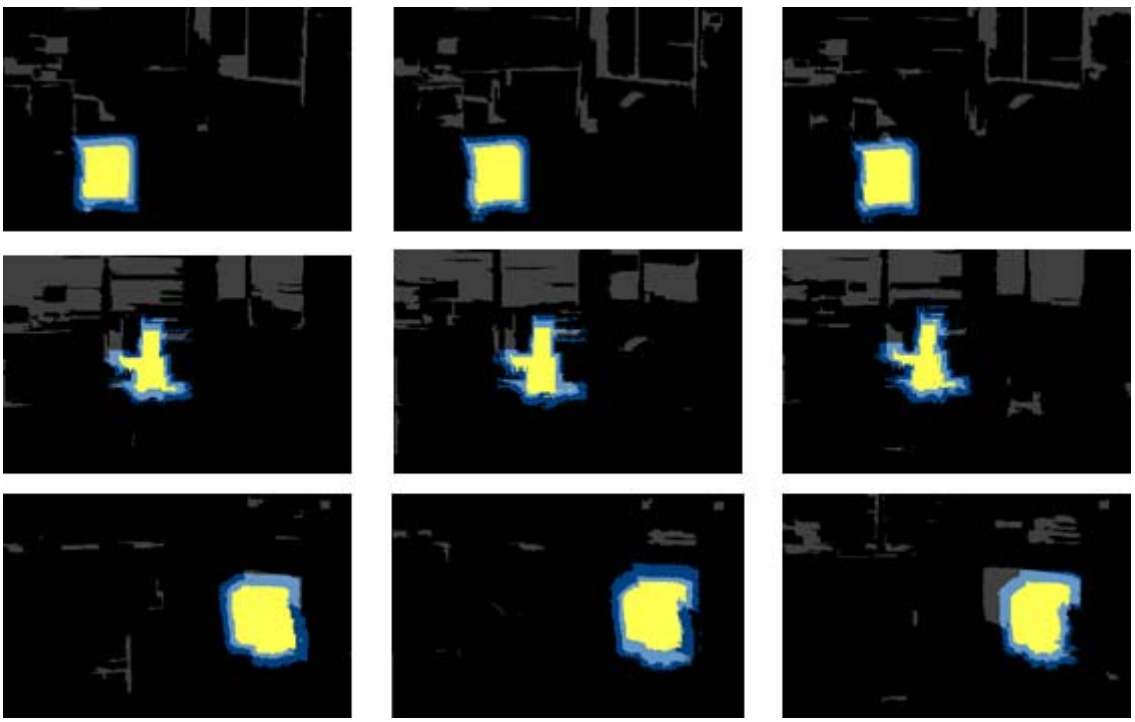


Figure 24 Analysis of tracking process.

ISBN:978-84-693-3387-7/DL:T.1001-2010

Figure 24 displays a more informative picture of the tracking process in the same frames shown in fig.23. The one-valued pixels of the tracking images are divided in two groups: those colored in yellow correspond to pixels labeled as "certainly belonging to the object" by the tracking method, and those colored in light blue correspond to pixels initially labeled as "uncertain" but with the largest dynamic recognition probability for the object class (recall equation (24)). The zero-valued pixels of the tracking images are divided in three groups: those colored in dark blue correspond to pixels labeled as "uncertain" and with a low probability, those shown in dark grey correspond to pixels labeled as "certainly not belonging to the object" but with a high probability for the object class (which are mostly recognition mistakes that are ignored thanks to the tracking prediction) and the rest are black pixels with both a low probability and a "certainly not belonging to the object" label. Similar results for each object in the whole right sequence can be observed in http://www.lsi.upc.edu/~alquezar/box_track.avi, chair_track.avi and basket_track.avi, respectively. The preliminar tracking results can be considered quite satisfactory for the three objects, especially if we note that numerous spots are incorrectly classified by the neural network within the static recognition module. The proposed tracking method allows a reasonable recovery of these recognition errors without relying on any contour detection and tracking procedure.

5.2 Comparative tracking experiments including object occlusion

We were interested in testing both PIORT approaches in video sequences including object occlusions and taken with a moving camera. Nevertheless, we also performed a first set of validation experiments in video sequences taken with a still camera. In all tests we defined N=1 objects of interest to track.

All images in the video sequences were segmented independently using the EDISON implementation of the mean-shift segmentation algorithm, figure 33, code available at http://www.caip.rutgers.edu/riul/research/code.html. The local features extracted for each spot were the RGB colour averages. For object learning, spots selected through ROI (region-of-interest) windows in the training sequence were collected to train a two-layer perceptron using backpropagation and to build the target class histogram. When using the neural net in the test phase the class probabilities for all the spots in the test sequences were estimated from the net outputs.

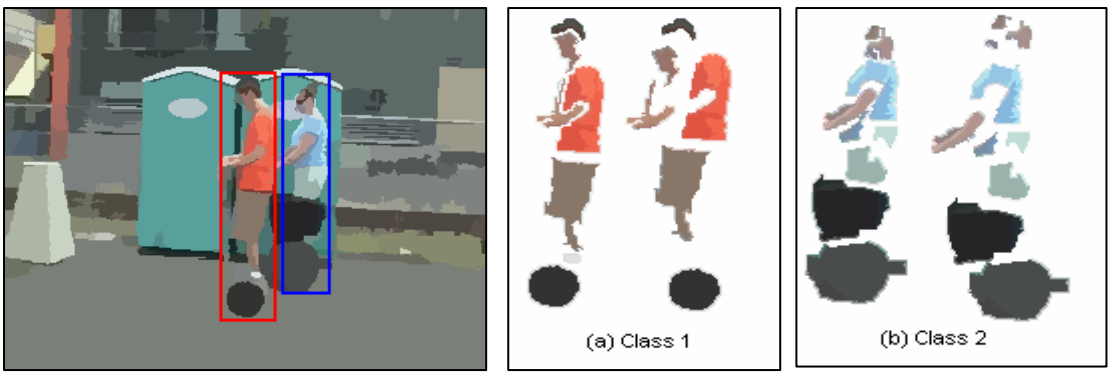


Figure 25 Selected spots. Labeling of the object spots,(a) (b) from Edison segmented image in the left figure

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

When using the histogram, the spot class probabilities were estimated according to equation (6). In both cases, the spot class probabilities were replicated for all the pixels in the same spot. For object tracking in the test sequences, ROI windows for the target object were only marked in the first image to initialise the tracking process.

Figure 26. Layout results of 2 x 3 images for one of the freames in sequence S8 using Bayesian method in static recognition module

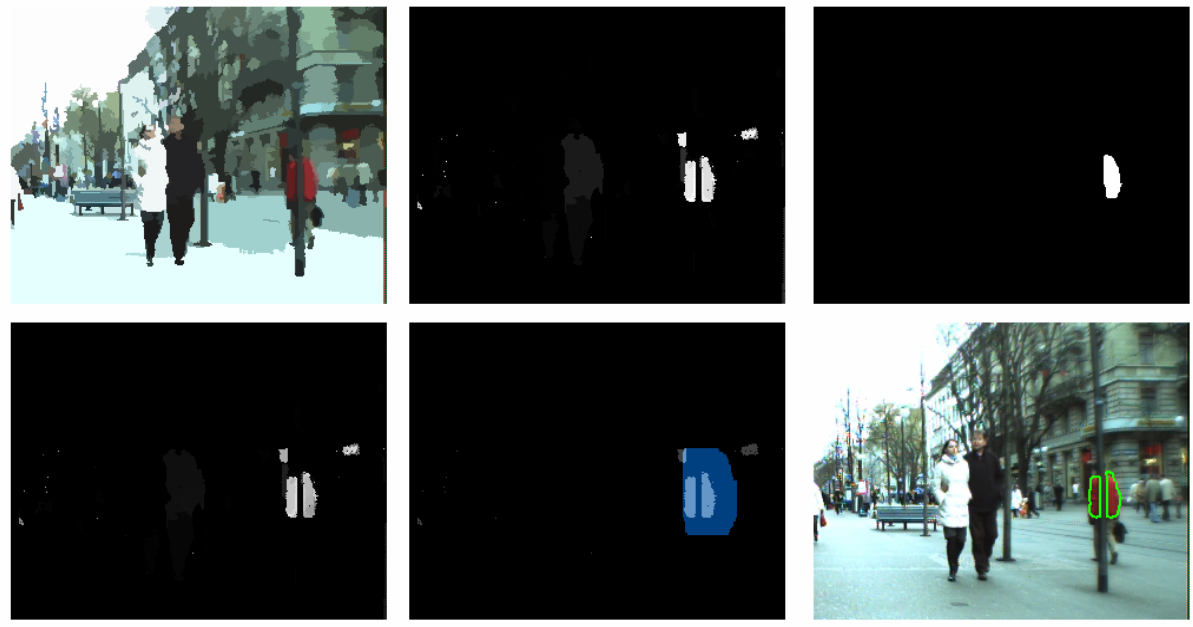


Figure 27. Layout results of 2 x 3 images for one of the freames in sequence S8 using Neural Net in static recognition module

The recognition and tracking results for the test sequences of our PIORT approaches were stored in videos where each frame (figure 26,27) has a layout of 2 x 3 images with the following contents: the top left is the image segmented by EDISON; the top middle

ISBN:978-84-693-3387-7/DL:T.1001-2010

is the image of probabilities given by the static recognition module for the current frame; the top right is the *a priori* prediction of the tracking image; the bottom left is the image of dynamic probabilities; the bottom right is the original image with a graphic overlay that represents the boundaries of the *a posteriori* binary tracking image (the final result for the frame); and the bottom middle is an intermediate image labelled by the tracking module where yellow pixels correspond to pixels labelled as "certainly belonging to the object", light blue pixels correspond to pixels initially labelled as "uncertain" but with a high dynamic probability, dark blue pixels correspond to pixels labelled as "uncertain" and with a low probability, dark grey pixels are pixels labelled as "certainly not belonging to the object" but with a high probability and the rest are black pixels with both a low probability and a "certainly not belonging to the object" label.

In the experimental evaluation carried out, PIORT methods have been compared to six state-of-the-art tracking methods of which we were able to get and apply their program codes to the test video sequences:

- Template Match by Correlation (TMC) [Comaniciu, 2003];
- Basic Meanshift (BM) [Comaniciu, 2002];
- Histogram Ratio Shift (HRS) [Collins, 2005];
- Variance Ratio Feature Shift (VRFS) [Collins, 2005];
- Peak Difference Feature Shift (PDFS) [Collins, 2005]; and
- Graph-Cut Based Tracker (GCBT) [Bugeau, 2008; Boykov, 2001].

Their codes were downloaded from the VIVID tracking evaluation web site www.vividevaluation.ri.cmu.edu, which unfortunately seems not to be accessible anymore. All these methods only requiere the ROI window mark in the first frame of the sequence. We briefly summarise these methods next.

In the TMC method [Comaniciu, 2003], the features of the target object are represented by histograms. These histograms are regularised by an isotropic kernel which produce spatially smooth functions suitable for gradient-based optimisation. The metric used to compare these functions is based on the Bhattacharyya distance and the optimisation is performed by the mean-shift procedure.

In [Comaniciu, 2002] a general non-parametric framework is presented for the analysis of a multimodal feature space and to separate clusters. The mean-shift procedure (localisation of stationary points in the distributions) is used to obtain the clusters. Throughout this framework, a segmentation application is described.

In [Collins, 2005] three different tracking methods are presented. They are based on the hypothesis that the best feature values to track an object are the ones that best discriminate between the object and the present background. Therefore, with several sample densities of the object and also of the background, the system computes the separability of both classes and obtains new features. The feature evaluation mechanism is embedded in a mean-shift tracking system that adaptively selects the top-ranked discriminative features for tracking. In the first method, Histogram Ratio Shift (HRS), the weights applied to each feature are dynamically updated depending on the histograms of the target and also of the background. In the second one, Variance Ratio Feature Shift (VRFS), the ratio between the variance of the target and the surrounding

ISBN:978-84-693-3387-7/DL:T.1001-2010

background is computed and considered for selecting the features. Finally, the Peak Difference Feature Shift (PDFS) softens the histogram of the features by a Gaussian kernel; moreover, it considers possible distracter objects near the target and dynamically changes the feature selection.

And finally, in [Bugeau, 2008;Boykov, 2001], a method for direct detection and segmentation of foreground moving objects is presented called Graph-Cut Based Tracker (GCBT). The method first obtains several groups of pixels with similar motion and photometric features. The mean-shift procedure is used to validate the motion and bandwidth. And then, the system segments the objects based on a MAP framework.

From the tracking results of all the tested methods, two evaluation metrics were computed for each frame: the **spatial overlap** and the **centroid distance** [Yin,2007]. The spatial overlap $SO(GT_k,ST_k)$ between the ground truth GT_k and the system track ST_k in a specific frame k is defined as the ratio

$$SO\left(GT_k,\, ST_k\right)=\frac{\text{Area}\left(GT_k \cap ST_k\right)}{\text{Area}\left(GT_k \cup ST_k\right)} \tag{43}$$

and $Dist(GTC_k,\ STC_k)$ refers to the Euclidean distance between the centroids of the ground truth (GTC_k) and the system track (STC_k) in frame k. Naturally, the larger the overlap and the smaller the distance, the better performance of the system track.

Since the centroid distance can only be computed if both GT_k and ST_k are non-null, a **failure ratio** was measured as the number of frames in which either GT_k or ST_k was null (but not both) divided by the total number of frames. Finally, an **accuracy** measure was computed as the number of good matches divided by the total number of frames, where a good match is either a true negative or a true positive with a spatial overlap above a threshold of 0.243 (which is the overlap obtained between two circles of the same size when one of the centres is located in the border of the other circle).

5.2.1 Experimental results on video sequences taken with a still camera

The first set of comparative experiments comprised three test video sequences taken with a still camera that show indoor office scenes where the target to track is a blue ball moving on a table. A similar but different sequence was used for training a neural network to discriminate between blue balls and typical sample regions in the background and for constructing the class histogram of the blue ball (this training sequence is available at http://www-iri.upc.es/people/ralqueza/bluetraining.avi).

In the first test sequence, http://www-iri.upc.es/people/ralqueza/S1S2.avi, two blue balls are moving on the table and one occludes temporally the other one during some frames. Two experiments were performed on this test sequence depending on the initialisation of the tracking. In test S1, the tracking was initialised at the right ball and in test S2, the tracking was initialised at the left ball. The static recognition module considers that both balls belong to the same class. In both tests, the temporal overlapping was correctly managed by our methods since the tracked ball is well relocated after exiting the occlusion. The corresponding videos displaying the results of PIORT methods (in the layout described above) are at http://www-iri.upc.es/people/ralqueza/S1_NN.mpg and S2_NN.mpg for the PIORT-Neural net method and at S1_Bayes.mpg and

ISBN:978-84-693-3387-7/DL:T.1001-2010

S2_Bayes.mpg for the PIORT-Bayesian method.

In the second test sequence (test S3), http://www-iri.upc.es/people/ralqueza/S3.avi, the tracked blue ball is occluded twice by a box during 5 and 12 frames, respectively. Recognition and tracking results for the whole sequence using the PIORT-Neural Net and Bayesian methods are at http://www-iri.upc.es/people/ralqueza/S3_NN.mpg and S3_Bayes.mpg, respectively. The tracking of the blue ball is quite satisfactory since both occlusions are correctly detected and the ball is correctly relocated when exiting the occlusion.

In the last test sequence of this group, http://www-iri.upc.es/people/ralqueza/S4.avi, there are again two blue balls and the target moving ball crosses twice, once in front of and once behind, the second ball, which does not move. As the recognition module classifies both balls in the same class, the same-class occlusion is not detected as an occlusion (the two balls are merged into a single blue object), but anyway the target ball is well tracked after the two crossings. The videos displaying the results of the PIORT-Neural Net and Bayesian methods for this sequence are at http://www-iri.upc.es/people/ralqueza/S4_NN.mpg and S4_Bayes.mpg, respectively.

Table 4 presents the results (mean ± std. deviation) of the spatial overlap (SO) and centroid distance (CD) measures together with the failure ratio (FR) and accuracy (Acc) of each tracking method for the four tests S1 to S4, emphasizing the best values for each measure and test in bold. Our PIORT tracking methods worked fine in the four tests obtaining the best values of the four measures (except in the Accuracy measure for test S4, where the HRS method gave a slightly superior performance). All methods performed quite well in S1; only PDFS method performed comparably to PIORT approaches in S2; only PIORT methods worked in S3 while the rest failed; and only BM and HRS methods performed comparably to PIORT approaches in S4.

ISBN:978-84-693-3387-7/DL:T.1001-2010

Video Sequence	Tracking method	SO	CD	FR	Acc
S1 Blue balls crossed (Right ball)	TMC	0.56 ± 0.10	5.07 ± 2.07	**0**	0.98
	BM	0.60 ± 0.06	3.19 ± 1.21	**0**	**1.00**
	HRS	0.46 ± 0.11	6.03 ± 2.05	**0**	**1.00**
	VRFS	0.66 ± 0.07	1.15 ± 0.47	**0**	**1.00**
	PDFS	0.63 ± 0.10	2.01 ± 0.94	**0**	**1.00**
	GCBT	0.64 ± 0.18	13.20 ± 52.52	0.05	0.94
	PIORT-Neural Net	**0.84** ± 0.09	1.38 ± 1.39	**0**	**1.00**
	PIORT-Bayesian	0.80 ± 0.07	**0.75** ± 0.76	**0**	**1.00**
S2 Blue balls crossed (Left ball)	TMC	0.22 ± 0.27	44.34 ± 52.24	**0**	0.41
	BM	0.23 ± 0.29	42.51 ± 50.42	**0**	0.36
	HRS	0.25 ± 0.31	44.93 ± 51.96	**0**	0.41
	VRFS	0.28 ± 0.35	42.82 ± 52.62	**0**	0.41
	PDFS	0.50 ± 0.30	36.27 ± 86.95	0.14	0.77
	GCBT	0.20 ± 0.27	70.69 ± 68.80	**0**	0.36
	PIORT-Neural Net	**0.60** ± 0.23	**3.94** ± 4.98	**0**	**0.91**
	PIORT-Bayesian	0.46 ± 0.25	15.04 ± 52.64	0.05	0.73
S3 Blue ball moving occluded by box	TMC	0.01 ± 0.04	173.40 ± 68.71	0.22	0
	BM	0.01 ± 0.07	182.54 ± 68.14	0.22	0
	HRS	0	187.85 ± 67.96	0.25	0
	VRFS	0.02 ± 0.18	140.14 ± 93.44	0.20	0.17
	PDFS	0.13 ± 0.41	131.07 ± 106.1	0.42	0.02
	GCBT	0	237.02 ± 134.6	0.74	0.22
	PIORT-Neural Net	**0.81** ± 0.42	**0.47** ± 0.38	**0**	**1.00**
	PIORT-Bayesian	0.53 ± 0.37	8.39 ± 48.61	0.03	0.95
S4 Blue ball moving around still blue ball	TMC	0.35 ± 0.22	13.10 ± 32.38	**0.01**	0.75
	BM	0.56 ± 0.15	7.39 ± 29.05	**0.01**	0.93
	HRS	0.60 ± 0.13	6.21 ± 29.16	**0.01**	**0.96**
	VRFS	0.10 ± 0.62	74.68 ± 45.00	**0.01**	0.14
	PDFS	0.13 ± 0.43	44.39 ± 36.14	**0.01**	0.17
	GCBT	0.10 ± 0.53	201.60 ± 98.35	0.80	0.18
	PIORT-Neural Net	**0.74** ± 0.21	5.90 ± 29.33	**0.01**	0.94
	PIORT-Bayesian	0.72 ± 0.20	**5.58** ± 29.38	**0.01**	0.94

SO: Spatial Overlap; CD: Centroid Distance; FR: Failure Ratio; Acc: Accuracy

Table 4. Results of ball tracking on video sequences taken with a still camera.

ISBN:978-84-693-3387-7/DL:T.1001-2010

5.2.2 Experimental results on video sequences taken with a moving camera

The second set of comparative experiments comprised another three test video sequences where the target is a ball, but this time taken with a moving camera. The first of them (test S5) again shows an indoor office scene where a blue ball is moving on a table and is temporally occluded, while other blue objects appear in the scene. This test sequence can be downloaded at http://www-iri.upc.es/people/ralqueza/S5.avi. The other two test sequences in this group show outdoor scenes in which a Segway robot tries to follow an orange ball that is being kicked by a person. Both include multiple occlusions of the tracked orange ball and differ in the surface over which the ball runs, which is pavement in the case of test S6 and grass in test S7 (see http://www-iri.upc.es/people/ralqueza/S6.avi and S7.avi, respectively). A similar but different sequence was used for training a neural network to discriminate between orange balls and typical sample regions in the background and for constructing the class histogram of the orange ball (this training sequence is available at http://www-iri.upc.es/people/ralqueza/orangetraining.avi).

Video Sequence	Tracking method	SO	CD	FR	Acc
S5 Blue bouncing ball on table	TMC	0.28 ± 0.48	74.65 ± 91.53	0.19	0.43
	BM	0.23 ± 0.52	78.40 ± 90.33	0.19	0.37
	HRS	0.16 ± 0.45	125.88 ± 11.80	0.43	0.30
	VRFS	0.20 ± 0.38	96.72 ± 134.84	0.39	0.60
	PDFS	0.28 ± 0.57	103.60 ± 36.77	0.41	0.59
	GCBT	0.01 ± 0.29	188.79 ± 18.13	0.75	0.21
	PIORT-Neural Net	**0.60** ± 0.40	12.53 ± 59.38	**0.05**	**0.95**
	PIORT-Bayesian	0.59 ± 0.39	**12.46** ± 59.40	**0.05**	**0.95**
S6 Segway - Orange ball on pavement	TMC	0.06 ± 0.40	146.35 ± 81.83	0.03	0.14
	BM	0.09 ± 0.43	110.94 ± 76.70	0.03	0.19
	HRS	0.09 ± 0.38	156.99 ± 103.80	0.41	0.21
	VRFS	0.16 ± 0.68	70.46 ± 49.17	0.03	0.21
	PDFS	0.14 ± 0.59	117.09 ± 81.43	0.03	0.21
	GCBT	0.01 ± 0.34	233.56 ± 62.12	0.93	0.06
	PIORT-Neural Net	**0.72** ± 0.20	**2.67** ± 19.21	**0.01**	**0.98**
	PIORT-Bayesian	0.13 ± 0.73	202.14 ± 99.35	0.81	0.19
S7 Segway - Orange ball on grass	TMC	0.02 ± 0.29	137.93 ± 84.53	0.04	0.04
	BM	0.15 ± 0.27	125.13 ± 116.14	0.34	0.35
	HRS	0.03 ± 0.33	190.63 ± 89.72	0.54	0.08
	VRFS	**0.59** ± 0.21	**7.93** ± 38.85	**0.02**	**0.95**
	PDFS	0.33 ± 0.50	121.46 ± 125.91	0.48	0.51
	GCBT	0.01 ± 0.37	208.39 ± 83.88	0.79	0.04
	PIORT-Neural Net	0.47 ± 0.23	17.02 ± 60.98	0.06	0.88
	PIORT-Bayesian	0.25 ± 0.49	133.43 ± 126.22	0.53	0.42

SO: Spatial Overlap; CD: Centroid Distance; FR: Failure Ratio; Acc: Accuracy

Table 5. Results of ball tracking on video sequences taken with a mobile camera.

ISBN:978-84-693-3387-7/DL:T.1001-2010

The tracking results videos for the above test sequences are attainable at http://www-iri.upc.es/people/ralqueza/S5_NN.mpg, S5_Bayes.mpg, S6_NN.mpg, S6_Bayes.mpg, S7_NN.mpg and S7_Bayes.mpg.

Table 5 presents the results (mean ± std. deviation) of the spatial overlap (SO) and centroid distance (CD) measures together with the failure ratio (FR) and accuracy (Acc) of each tracking method for the three tests S5 to S7, emphasizing the best values for each measure and test in bold. Our PIORT-Neural net method worked fine in the three tests obtaining the best values of spatial overlap and accuracy measures in tests S5 and S6 and yielding results a little bit under the performance of the VRFS method in test S7, in which the VRFS method gave the best values of the four measures. Our PIORT-Bayesian method worked well in test S5 but failed to track the orange ball correctly in tests S6 and S7. Only both PIORT methods performed well in S5; only PIORT-Neural net method worked in S6 while the rest failed; and only VRFS and PIORT-Neural net methods obtained satisfactory results in S7.

The last set of experiments comprised another three test video sequences, taken with a moving camera in outdoor environments, where the targets are humans, more precisely, some part of their clothing. The first sequence in this group (test S8) is a long sequence taken on a street where the aim is to track a pedestrian wearing a red jacket (see http://www-iri.upc.es/people/ralqueza/S8.avi) and includes total and partial occlusions of the followed person by other walking people and objects on the street. In this case, a short sequence of the scene taken with a moving camera located in a different position (http://www-iri.upc.es/people/ralqueza/redpedestrian_training.avi) was used as training sequence. The other two test sequences in this group, tests S9 and S10, show outdoor scenes in which humans riding Segway robots and wearing orange T-shirts are followed. In test S9 a single riding guy is followed, whereas in test S10, two men are riding two Segway robots simultaneously and crossing each other. These test sequences are at http://www-iri.upc.es/people/ralqueza/S9.avi and S10.avi and the training sequence associated with them is at http://www-iri.upc.es/people/ralqueza/T-shirt_training.avi.

The tracking results videos for the above test sequences are attainable at http://www-iri.upc.es/people/ralqueza/S8_NN.mpg, S8_Bayes.mpg, S9_NN.mpg, S9_Bayes.mpg, S10_NN.mpg and S10_Bayes.mpg.

Table 6 presents the results of the evaluation measures of each tracking method for the three tests S8 to S10, emphasizing the best values for each measure and test in bold. Both PIORT methods gave the best results, very similar between them, in tests S8 and S9, and PIORT-Neural net method performed clearly the best in test S10. Note that in the pedestrian sequence (S8), an occlusion by people carrying red bags distracted the attention of the PIORT tracking module and caused a momentarily impairment in performance, especially for the centroid distance measure, but the tracker was able to recover correctly the target after that occlusion. In this sequence S8, only the PDFS method performed comparably to PIORT approaches in terms of accuracy and centroid distance, although it achieved a rather lower spatial overlap. In test S9, the HRS, VRFS and PDFS methods obtained similar and reasonably well results, but not as good as those of PIORT methods. Finally, only the PIORT-Neural net method worked well in

ISBN:978-84-693-3387-7/DL:T.1001-2010

test S10, where the PIORT-Bayesian method performed poorly because it followed the other Segway-riding man after a crossing between both men.

Video Sequence	Tracking method	SO	CD	FR	Acc
S8 Pedestrian with red jacket	TMC	0.44 ± 0.31	25.25 ± 61.10	0.07	0.77
	BM	0.24 ± 0.58	72.08 ± 64.33	0.07	0.34
	HRS	0.35 ± 0.24	13.49 ± 38.27	**0.02**	0.64
	VRFS	0.45 ± 0.32	34.27 ± 81.13	0.12	0.82
	PDFS	0.50 ± 0.20	11.42 ± 45.11	0.03	0.95
	GCBT	0.04 ± 0.32	194.7 ± 105.3	0.77	0.16
	PIORT-Neural Net	**0.79** ± 0.24	11.90 ± 50.87	0.04	**0.96**
	PIORT-Bayesian	0.74 ± 0.24	**11.15** ± 48.14	0.04	0.95
S9 Guy on Segway with orange T-shirt	TMC	0.10 ± 0.53	130.3 ± 69.75	**0.00**	0.15
	BM	0.22 ± 0.13	41.30 ± 58.70	0.01	0.40
	HRS	0.53 ± 0.25	22.83 ± 58.43	0.05	0.86
	VRFS	0.69 ± 0.25	27.69 ± 75.15	0.10	0.90
	PDFS	0.56 ± 0.21	29.19 ± 74.65	0.10	0.90
	GCBT	0.14 ± 0.22	101.6 ± 112.7	0.36	0.19
	PIORT-Neural Net	0.73 ± 0.16	**3.40** ± 14.78	**0.00**	0.97
	PIORT-Bayesian	**0.74** ± 0.13	3.70 ± 14.61	**0.00**	**0.98**
S10 Men on Segway with orange T-shirts	TMC	0.06 ± 0.39	104.3 ± 83.15	**0.03**	0.10
	BM	0.29 ± 0.28	42.10 ± 59.06	**0.03**	0.59
	HRS	0.28 ± 0.30	38.72 ± 65.09	0.06	0.58
	VRFS	0.38 ± 0.34	36.81 ± 64.53	0.06	0.61
	PDFS	0.32 ± 0.36	91.14 ± 119.4	0.35	0.56
	GCBT	0.04 ± 0.31	187.1 ± 103.1	0.72	0.08
	PIORT-Neural Net	**0.73** ± 0.18	**8.37** ± 40.74	**0.03**	**0.96**
	PIORT-Bayesian	0.16 ± 0.58	81.36 ± 62.93	**0.03**	0.22

SO: Spatial Overlap; CD: Centroid Distance; FR: Failure Ratio; Acc: Accuracy

Table 6. Results of human tracking on video sequences taken with a mobile camera.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Chapter 6. Conclusions and future work

In this thesis we have described an updated version of the *probabilistic integrated object recognition and tracking* (PIORT) methodology that we have developed in the latest years, partially reported in [Amézquita, 2006,2007,2008; Alquézar, Amézquita, 2009], and presented a collection of experimental results in test video sequences, with the aim of comparing two particular approaches derived from PIORT, based on Bayesian and neural net methods, respectively, with some state-of-the-art tracking methods proposed by other authors.

An improved method for object tracking, capable of dealing with rather long occlusions and same-class object crossing, has been proposed to be included within our probabilistic framework that integrates recognition and tracking of objects in image sequences. PIORT does not use any contour information but the results of an iterative and dynamic probabilistic approach for object recognition. These recognition results are represented at pixel level as probability images and are obtained through the use of a classifier (e.g. a neural network) from region-based features.

The PIORT framework is divided in three parts: a static recognition module, where the classifier is applied to single-frame images, a dynamic recognition module that updates the object probabilities using previous recognition and tracking results, and a tracking decision module, where tracking binary images are determined for each object. This third module combines the recognition probabilities with a model that predicts the object's apparent motion in terms of translation and scale changes, while coping with the problems of occlusion and re-emergence detection. Moreover, the tracking module can deal with object splitting, either due to partial occlusions or same-class object crossing, and, in most cases, is able to select and track only the target object after it crosses or is occluded by another object which is recognised as belonging to the same class, i.e. it is able to re-establish the identity of the target object.

Although the comparative experimental work has been focused on the case of single object tracking, the PIORT system is capable of tracking multiple objects of different classes simultaneously (as shown in section 5.1.2) and, as demonstrated in the experiments, it can be applied to video sequences acquired either by a fixed or a moving camera. The size, shape and movement of the target objects can vary softly along the sequence, but the appearance features used by the classifier (up to now, colour features) should remain rather stable for a successful tracking. It must be taken into account that the global performance of the system depends not only on the ability of the tracking method but also on the quality of the object recognition probabilities provided by the trained classifier.

We have presented two static recognition methods that can be embedded in the first module of PIORT, giving rise to two different instances of the methodology. Both methods are based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The first classifier is based on a maximum likelihood Bayesian method in which the conditional probabilities for object classes are obtained from the information of the class histograms (for discretized RGB values) and a uniform conditional probability is assumed for the

ISBN:978-84-693-3387-7/DL:T.1001-2010

background. The second classifier is based on a neural net which is trained with the RGB colour averages extracted for each spot of the segmented images.

Even though the characteristics of these two classifiers are quite different, the recognition and tracking results of PIORT using both approaches were excellent and very similar in five of the ten test sequences, which might mean that the good ability of PIORT to track the objects is mostly due to a smart cooperation of the three inner modules and is not very dependent on the specific method used for object recognition. However, in the remaining five test sequences, the tracking method based on a neural net classifier clearly outperformed the one based on a simple Bayesian classifier, which failed in three of these test sequences.

In the experimental comparison with other six methods proposed in the literature for object tracking, a PIORT method obtained the best results in nine of the ten test sequences and only a slightly inferior performance with respect to best method in the other one (VRFS). Except for the case of the first test sequence S1, where all methods worked fine, the six alternative methods tested mostly failed to track the target objects correctly in the test sequences, due to the difficult instances of occlusions and object crossings they contain.

Although further experimental work is needed, the tracking module included in PIORT has demonstrated by now to be effective under several-frames occlusions produced by an object of a class different to that of the target object. If the occluding and the target objects are recognised as belonging to the same class, then the occlusion is not detected as such, both objects are merged temporarily, but despite this behaviour, the tracking method is able in most cases to recover and track the original target when the same-class object occlusion or crossing ends. However, as observed in some of the test sequences, still there are cases where the behaviour of the tracking decision module of PIORT should be improved, particularly in the step of object re-emergence after occlusion and when other objects of similar appearance are next to the target. The upgrade of this tracking module will be subject of future research.

Our PIORT methodology is based on the iterative and adaptive processing of consecutive frames by a system that integrates recognition and tracking in a probabilistic framework. The system uses object recognition results provided by a classifier, e.g. a Bayesian classifier or a neural net, which are computed from colour features of image regions for each frame. The location of tracked objects is represented through probability images that are updated dynamically using both recognition and tracking results. The tracking procedure is capable of handling quite long occlusions. In particular, object occlusion is detected automatically and the prediction of the object's apparent motion and size takes into account the cases of occlusion entering, full occlusion and occlusion exiting. In contrast with [Zhu, 2008], our tracking method does not rely on background subtraction and a fixed camera and, to the contrary of [Pan, 2007], it can cope with complete occlusion and it does not involve any template to match and update.

We think that PIORT approaches for object tracking are especially suitable in noisy environments where segmented images vary so much in successive frames that it is very hard to match the corresponding regions or contours of consecutive images. The empirical results presented are quite satisfactory, despite the numerous mistakes made

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

by the static recognition module, which can be mostly ignored thanks to the integration with the proposed tracking decision module.

As future work, we want to extend the experimental validation of PIORT by applying it to new and more difficult image sequences. In addition, we are interested in implementing and testing new classifiers in the static recognition module, which could exploit other features completely different to the basic colour features used up to now. For instance, an SVM classifier could be applied to a set of features formed by Gabor filter responses, provided that class probability values were estimated from margin values.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

ISBN:978-84-693-3387-7/DL:T.1001-2010

Publications derived from this thesis

2009

N. Amézquita Gómez, R. Alquézar, F. Serratosa,"A Probabilistic Integrated Object Recognition and Tracking Framework" submitted to Journal on Image and Vision Computing, JIVC , ISSN: 0262-8856 ELSEVIER, Chief K.D. Baker, M. Pantic Editors

"Experimental Assessment of Probabilistic Integrated Object Recognition and Tracking Methods"Proc.14th Iberoamerican Congress on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, México, Springer,LNCS-5856.

R. Alquézar, N. Amézquita Gómez, F. Serratosa, "Tracking deformable objects and dealing with same class object occlusion", in: *Proc. Fourth Int. Conf. On Computer Vision Theory and Applications* (VISAPP 2009), Lisboa, Portugal.

2008

N. Amézquita Gómez, R. Alquézar, F. Serratosa, "Dealing with occlusion in a probabilistic object tracking method", in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2008), Anchorage, Alaska, June 2008.

2007

Nicolás Amézquita Gómez, René Alquézar, Francesc Serratosa: "A New Method for Object Tracking Based on Regions Instead of Contours". Proc. 4th IEEE Computer Vision and Pattern Recognition, CVPR2007, Minneapolis, USA, pp. 1 -8, 2007

2006

Francesc Serratosa, Nicolás Amézquita Gómez, René Alquézar: Combining Neural Networks and Clustering Techniques for Object Recognition in Indoor Video Sequences. Proc. 11th Iberoamerican Congress on Pattern Recognition, CIARP 2006, Cancún, México, Springer LNCS-4225, pp. 399-405

Nicolás Amézquita Gómez, René Alquézar, Francesc Serratosa: Object Recognition and Tracking in Video Sequences: A New Integrated Methodology. Proc. 11th Iberoamerican Congress on Pattern Recognition, CIARP 2006, Cancún, México, Springer LNCS-4225, pp. 481-490

2005

Nicolás Amézquita Gómez, René Alquézar: Object Recognition in Indoor Video Sequences by Classifying Image Segmentation Regions Using Neural Networks. Proc. 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005, La Habana, Cuba, Springer LNCS-3773, pp. 93-102

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

ISBN:978-84-693-3387-7/DL:T.1001-2010

Bibliographical references

[Ali, 2001] Ali, A. and Aggarwal, J. 2001. Segmentation and recognition of continuous human activity. In *IEEEWorkshop on Detection and Recognition of Events in Video*. 28–35.

[Alquézar, 2009] R. Alquézar, N. Amézquita Gómez, F. Serratosa, "Tracking deformable objects and dealing with same class object occlusion", in: *Proc. Fourth Int. Conf. on Computer Vision Theory and Applications* (VISAPP 2009), Lisboa, Portugal.

[Amézquita; and Alquézar,2005] Nicolás Amézquita Gómez, René Alquézar: Object Recognition in Indoor Video Sequences by Classifying Image Segmentation Regions Using Neural Networks. Proc. 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005, La Habana, Cuba, Springer LNCS-3773, pp. 93-102

[Amézquita, 2006] Amézquita Gómez N., R Alquézar. and F Serratosa., Object Recognition and Tracking in Video Sequences: A New Integrated Methodology, *Proc. 11th Iberoamerican Congress on Pattern Recognition, CIARP 2006*, LNCS 4225, pp. 481 – 490, 2006.

[Amézquita, 2007] N. Amézquita Gómez, R. Alquézar, F. Serratosa, "A new method for object tracking based on regions instead of contours", in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2007), Minneapolis, Minnesota, June 2007.

[Amézquita, 2008] N. Amézquita Gómez, R. Alquézar, F. Serratosa,"Dealing with occlusion in a probabilistic object tracking method", in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2008), Anchorage, Alaska, June 2008.

[Avidan, 2001] Avidan, S. 2001. Support vector tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 184–191.

[Baddeley 1992]. Baddeley, A. 1992. Errors in binary images and an l version of the haus- dorff metric. *Nieuw Archief voor Wiskunde 10*, 157–183.

[Ballard, 1982] Ballard D.H. and Brown C.M., *Computer Vision*, Prentice Hall, New Jersey, 1982.

[Barrow, 1994] Barron, J., Fleet, D., and Beauchemin, S. 1994. Performance of optical flow techniques. *Int. J. Comput.Vision 12*, 43–77

[Bar-Shalom, 1988] Bar-Shalom Y. and Foreman T., *Tracking and Data Association*, Academic Press Inc., 1988.

[Beaulieu and Goldberg, 1989]. Beaulieu, J. and Goldberg, M. 1989. Hierarchy in picture image segmentation: A step wise optimization approach. *IEEE Trans. Patt. Analy. Mach. Intell. 11*, 150–163.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Bertalmio, 2000] Bertalmio, M., Sapiro, G., and Randall, G. 2000. Morphing active contours. *IEEE Trans. Patt. Analy. Mach. Intell. 22*, 7, 733–737

[Bishop, 95] Bishop C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[Black, 1998]. Black, M. and Jepson, A. 1998. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Comput. Vision 26*, 1, 63–84.

[Blake and Isard, 2000]. Blake, A. and Isard,M. 2000. *Active Contours:The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*.

[Bowyer, 2001] Bowyer, K., Kranenburg, C., and Dougherty, S. 2001. Edge detector evaluation using empirical roc curve. *Comput. Vision Image Understand. 10*, 77–103.

[Boykov, 2001] Y. Boykov, O. Veksler, R. Zabih, "Fast approximate energy minimization via graph cuts", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23 (2001) 1222-1239.

[Bugeau, 2008] A. Bugeau, P. Pérez, "Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts", in: *Proc. Third Int. Conf. on Computer Vision Theory and Applications* (VISAPP 2008), Funchal, Madeira,Portugal.

[Canny, 1986] Canny, J. 1986. A computational approach to edge detection. *IEEE Trans. Patt. Analy. Mach. Intell. 8*, 6, 679–698.

[Chen 2003] F-S. Chen, C-M. Fu and C-L. Huang, Hand Gesture recognition using a real-time tracking method and Hidden Markov Models, *Image and Vision Computing*, vol. 21, pp: 745-758, 2003.

[Cremers, and Schnorr, 2003] Cremers, D. and Schnorr, C. 2003. Statistical shape knowledge in variational motion segmentation. *I.Srael Nent. Cap. J. 21*, 77–86.

[Comaniciu, 2002] Comaniciu D. and Meer P., Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Patt. Analy. Mach. Intell. 24*, 5, 603–619, 2002.

[Comaniciu, 2003] Comaniciu, D., Ramesh, V., and Meer, P. 2003. Kernel-based object tracking. *IEEE Trans. Patt. Analy. Mach. Intell. 25*, 564–575

[Cootes, 2001] Cootes, T., Edwards, G., and Taylor, C. 2001. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. Patt. Analy. Mach. Intell. 23*, 6, 681–685.

[Comaniciu, 2002] D. Comaniciu, P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24 (2002) 603-619.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Collins, 2005] R. Collins, Y. Liu, "On-line selection of discriminative tracking features", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27 (2005), 1631-1643.

[Edwards, 1998] Edwards, G., Taylor, C., and Cootes, T. 1998. Interpreting face images using active appearance models. In *International Conference on Face and Gesture Recognition*. 300–305.

[Elgammal, 2002] Elgammal, A.,Duraiswami, R.,Harwood, D., and Davis, L. 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of IEEE 90*, 7, 1151–1163.

[Felzenszwalb, 1998] Felzenszwalb P. and Huttenlocher D., Efficiently computing a good segmentation, *Proceedings CVPR'98*, pp. 98-104, 1998.

[Fieguth, 1997] Fieguth, P. and Terzopoulos, D. 1997. Color-based tracking of heads and other mobile objects at video frame rates. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 21–27.

[Fiesler, 1997] Fiesler E. and Beale R. (eds.), *Handbook of Neural Computation*, IOP Publishing Ltd and Oxford University Press, 1997.

[Fiesler E ,1997]. Fiesler E. and Beale R. (eds.), Handbook of Neural Computation, IOP Publishing Ltd and Oxford University Press, 1997.

[Foresti ,1999]. G. L. Foresti, "Object recognition and tracking for remote video surveillance", *IEEE Trans. on Circuits and Systems for Video Technology* 9 (1999) 1045-1062.

[Gerke, 2001] Gerke M., Heipke C., Straub B.M., Building extraction from aerial imagery using a generic scene model and invariant geometric moments, in *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop*, pp.85-89, 2001.

[Grewe, 1995] Grewe, L. and Kak, A. 1995. Interactive learning of a multi-attribute hash table classifier for fast object recognition. *Comput. Vision Image Understand. 61*, 3, 387–416.

[Hariharakrishnan,2005] K. Hariharakrishnan, D. Schonfeld, "Fast object tracking using adaptive block matching", *IEEE Trans. Multimedia* 7 (2005) 853-859.

[Haritaoglu, 2000] Haritaoglu, I., Harwood, D., and Davis, L. 2000. W4: real-time surveillance of people and their activities. *IEEE Trans. Patt. Analy. Mach. Intell. 22*, 8, 809–830.

[Hoffmann, 1989] Hoffmann C., *Geometric and Solid Modeling*, Morgan-Kaufmann, San Mateo, CA, 1989.

[Horn, 1981] Horn, B. and Schunk, B. 1981. Determining optical flow. *Artific. Intell. 17*, 185–203.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Hu, 1962] Hu M.K., Visual pattern recognition by moment invariants, *IRE Trans. on Information Theory*, Vol. 8 (2), pp.179-187, 1962.

[Huttenlocher, 1993] Huttenlocher, D., Noh, J., and Rucklidge, W. 1993. Tracking nonrigid objects in complex scenes. In *IEEE International Conference on Computer Vision (ICCV)*. 93–101.

[Ito,2001] K. Ito, S. Sakane, "Robust view-based visual tracking with detection of occlusions", in: *Proc. Int. Conf. Robotics Automation*, 2001, vol. 2, pp. 1207-1213.

[Jepson,2003] A.D. Jepson, D.J. Fleet, T.F. EI-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2003) 1296-1311

[Kerner 2004] B.S. Kerner, H. Rehborn, M. Aleksic and A. Haug, Recognition and Tracking of spatial-temporal congested traffic patterns on freeways, vol 12, pp: 369-400, 2004.

[Lee, 2005] K-C. Lee, J. Ho, M-H. Yang, D. Kriegman, Visual Tracking and Recognition using Probabilistic appearance manifolds, Computer Vision and Image Understanding, vol. 99, pp: 303-331, 2005.

[Li , 2001] Li, B., Chellappa, R., Zheng, Q., and Der, S. 2001. Model-based temporal object verification using video. *IEEE Trans. Image Process. 10*, 6, 897–908.

[Mac-Cormick and Blake, 2000] Maccormick, J. and Blake, A. 2000. Probabilistic exclusion and partitioned sampling for multiple object tracking. *Int. J. Comput. Vision 39*, 1, 57–71.

[Malik,2001] J. Malik, S. Belongie, T. Leung, J. Shi, "Contour and texture analysis for image segmentation", *Int. J. Computer Vision*, 43 (2001) 7-27.

[Mikolajczyk, 2003] Mikolajczyk, K. and Schmid, C. 2003. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1615–1630.

[Mughadam, 1997] Mughadam, B. and Pentland, A. 1997. Probabilistic visual learning for object representation. *IEEE Trans. Patt. Analy. Mach. Intell. 19*, 7, 696–710.

[Mumford and Shah, 1989] Mumford, D. and Shah, J. 1989. Optimal approximations by piecewise smooth functions and variational problems. *Comm. Pure Appl. Mathemat. 42*, 5, 677–685.

[Mutch, 2006] Mutch J. and Lowe D.G., Multiclass Object Recognition with Sparse, Localized Features, *Proceedings CVPR'06,* New York*,* pp. 11 – 18, June 2006.

[Nguyen,2004] H.T. Nguyen, A.W.M. Smeulders, "Fast occluded object tracking by a robust appearance filter", *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2004) 1099-1104.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Paragios, 2002] Paragios, N. and Deriche, R. 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. Patt. Analy. Mach. Intell. 22*, 3, 266–280.

[Pan,2007] J. Pan, B. Hu, "Robust occlusion handling in object tracking", in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2007), Minneapolis, Minnesota, June 2007.

[Papageorgiu, 1998] Papageorgiou, C., Oren, M., and Poggio, T. 1998. A general framework for object detection. In *IEEE International Conference on Computer Vision (ICCV)*. 555–562.

[Park, 2004] Park, S. and Aggarwal, J. K. 2004. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimed. Syst. 10*, 2, 164–179.

[Paschos, 2001] Paschos, G. 2001. Perceptually uniform color spaces for color texture analysis: an empirical evaluation. *IEEE Trans. Image Process. 10*, 932–937

[Romero, 2003] Romero E., Sopena J.M., Navarrete G., Alquézar R., Feature selection forcing overtraining may help to improve performance, *Proc. Int. Joint Conference on Neural Networks, IJCNN-2003*, Portland, Oregon, Vol.3, pp.2181-2186, 2003.

[Ronfrad, 1994] Ronfard, R. 1994. Region based strategies for active contour models. *Int. J. Comput. Vision 13*, 2, 229–251.

[Rowley, 1998] Rowley, H., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. *IEEE Trans. Patt. Analy. Mach. Intell. 20*, 1, 23–38.

[Rumelhart and McClelland , 1986] Rumelhart D.E., McClelland J.L. and the PDP Research Group (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, 1986.

[Sato and Aggarwal, 2004] Sato, K. and Aggarwal, J. 2004. Temporal spatio-velocity transform and its application to tracking and interaction. *Comput. Vision Image Understand. 96*, 2, 100–128.

[Sanfeliu, 2004]. A. Sanfeliu, F. Serratosa, R. Alquézar, "Second-order random graphs for modeling sets of attributed graphs and their application to object learning and recognition", *Int. Journal of Pattern Recognition and Artificial Intelligence* 18 (2004) 375-396.

[Senior,2006]A. Senior, A. Hampapur, Y-L. Tian, L. Brown, S. Pankanti, R. Bolle,"Appearance models for occlusion handling", *Image and Vision Computing* 24 (2006) 1233-1243.

[Schweitzer, 2002] Schweitzer, H., Bell, J. W., and Wu, F. 2002. Very fast template matching. In *European Conference on Computer Vision (ECCV)*. 358–372.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Serby, 2004] Serby, D., Koller-Meier, S., and Gool, L. V. 2004. Probabilistic object tracking using multiple features. In *IEEE International Conference of Pattern Recognition (ICPR)*. 184–187.

[Serre, 2005] Serre T., Wolf L. and Poggio T., Object recognition with features inspired by visual cortex, *Proceedings CVPR'05,* Volume 2, pp. 994 – 1000, 2005.

[Shi, 2000] Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Patt. Analy. Mach. Intell. 22*, 8, 888–905.

[Song, 1996] Song, K. Y., Kittler, J., and Petrou, M. 1996. Defect detection in random color textures. *Israel Verj. Cap J. 14*, 9, 667–683.

[Streit, 1994] Streit, R. L. and Luginbuhl, T. E. 1994. Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proceedings of the International Society for Optical Engineering (SPIE.)* vol. 2235. 394–405.

[Tanizaki, 1987] Tanizaki H., Non-gaussian state-space modeling of nonstationary time series. *J. Amer. Statist. Assoc.* 82, pp. 1032-1063, 1987.

[Terzopoulos and Szeliski, 1992] Terzopoulos, D. and Szeliski, R. 1992. Tracking with kalman snakes. In *Active Vision*, A. Blake and A. Yuille, Eds. MIT Press.

[Tomita, 1990] F. Tomita, S. Tsuji, Computer Analysis of Visual Textures. Ed. Kluwer Academic Publishers. 1990.

[Tu,2003] Z. Tu, X. Chen, A.L. Yuille, S.C. Zhu, "Image parsing: unifying segmentation, detection, and recognition", in: *Proc. Ninth IEEE Int. Conf. on Computer Vision*, 2003, pp 18- 25.

[Tu,2002] Z. Tu, S.C. Zhu, "Image segmentation by data driven Markov chain Monte Carlo", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24 (2002) 657-673.

[Veenman, 2001] Veenman, C., Reinders, M., and Backer, E. 2001. Resolving motion correspondence for densely moving points. *IEEE Trans. Patt. Analy. Mach. Intell. 23*, 1, 54–72.

[Viola, 2003] Viola, P., Jones, M., and Snow, D. 2003. Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision (ICCV)*. 734–741.

[Wren, 1997] Wren, C., Azarbayejani, A., and Pentland, A. 1997. Pfinder: Real-time tracking of the human body. *IEEE Trans. Patt. Analy. Mach. Intell. 19*, 7, 780–785.

[Yilmaz, 2004] Yilmaz, A., Li, X., and Shah, M. 2004. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Patt. Analy. Mach. Intell. 26*, 11, 1531–1536.

[Yilmaz, 2006] Yilmaz A., Javed O. and Shah M., Object Tracking: A Survey. *ACM Computing Surveys* 38 (4), Article 13, 2006.

ISBN:978-84-693-3387-7/DL:T.1001-2010

[Yin, 2007]F. Yin, D. Makris, S.A. Velastin, "Performance evaluation of object tracking algorithms", in: *Proc. 10th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance* (PETS'2007).

[Zhu,2008] L. Zhu, J.Zhou, J. Song, "Tracking multiple objects through occlusion with online sampling and position", *Pattern Recognition* 41 (2008) 2447-2460.

[Zhou, 2004] S.K. Zhou, R. Chellamppa and B. Moghaddam, Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters, *IEEE Trans. Image Process.* vol. 13. (11), pp:1491-1506, 2004.

[Zhu, 1996] S.C. Zhu, A. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18 (1996) 884-900.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

ISBN:978-84-693-3387-7/DL:T.1001-2010

Appendix A. Detailed results of the comparative tracking experiments

Frame-based Metrics

Starting with the first frame of the test sequence, frame-based metrics are computed for every frame in the sequence. From each frame in the video sequence, first the following quantities are computed:

***True Negative, TN**:* Number of frames where both ground truth and system results agree on the absence of any object.

***True Positive, TP**:* Number of frames where both ground truth and system results agree on the presence of one or more objects, and the bounding box of at least one or more objects coincides among ground truth and tracker results.

***False Negative, FN**:* Number of frames where ground truth contains at least one object, while system either does not contain any object or none of the system's objects fall within the bounding box of any ground truth object.

***False Positive, FP**:* Number of frames where system results contain at least one object, while ground truth either does not contain any object or none of the ground truth's objects fall within the bounding box of any system object.

In the above definitions, the two bounding boxes are said to be *coincident* if the centroid of one of the boxes lies inside the other box. Also, TF is the total number of frames in the video sequence. Once the above defined quantities are calculated for all the frames in the test sequence, in the second step, the following metrics are computed:

$$\text{False Alarm Rate (FAR)} = \frac{FP}{TP+FP} \tag{A.1}$$

$$\text{Detection Rate} = \frac{TP}{TP+FN} \tag{A.2}$$

$$\text{Specificity} = \frac{TN}{FP+TN} \tag{A.3}$$

$$\text{Accuracy} = \frac{TP+TN}{TF} \tag{A.4}$$

$$\text{Positive Prediction} = \frac{TP}{TP+FP} \tag{A.5}$$

$$\text{Negative Prediction} = \frac{TN}{FN+TN} \tag{A.6}$$

$$\text{False Negative Rate} = \frac{FN}{FN+TP} \tag{A.7}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \tag{A.8}$$

ISBN:978-84-693-3387-7/DL:T.1001-2010

Image tracking results for sequences S1- S10

S1 Blue balls crossed. Right ball tracking- 68 frames

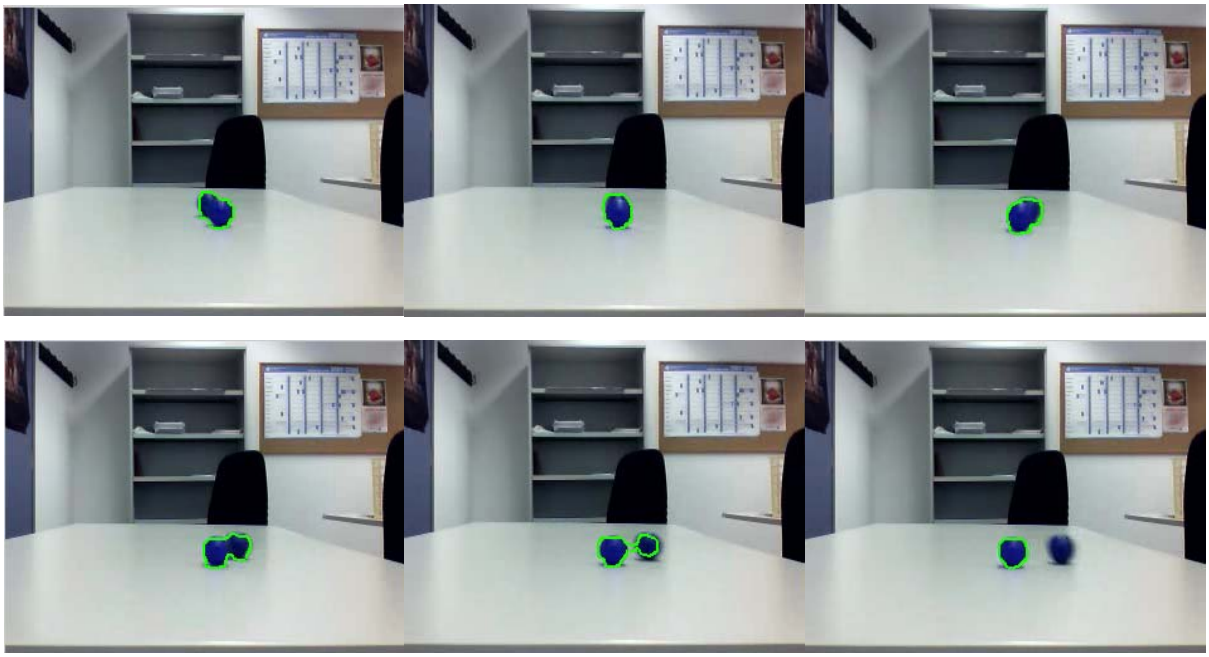


Figure A1 tracking for some consecutive frames in S1.

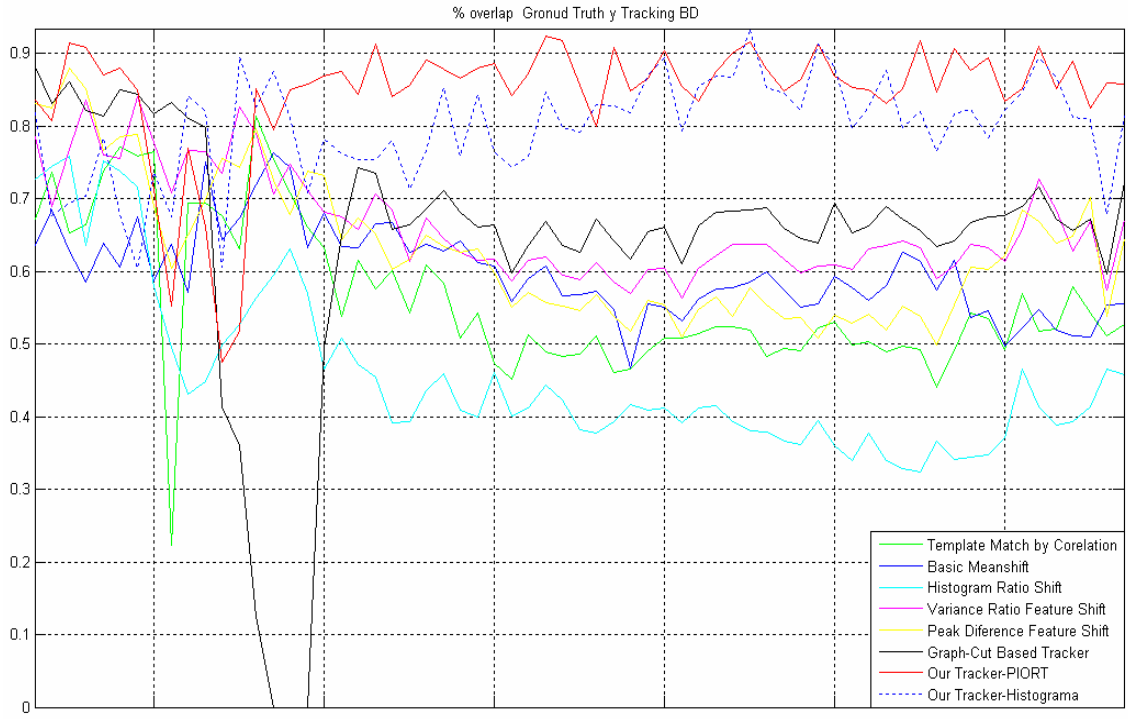


Figure A2 Overlap between ground truth and tracking results for S1.

ISBN:978-84-693-3387-7/DL:T.1001-2010

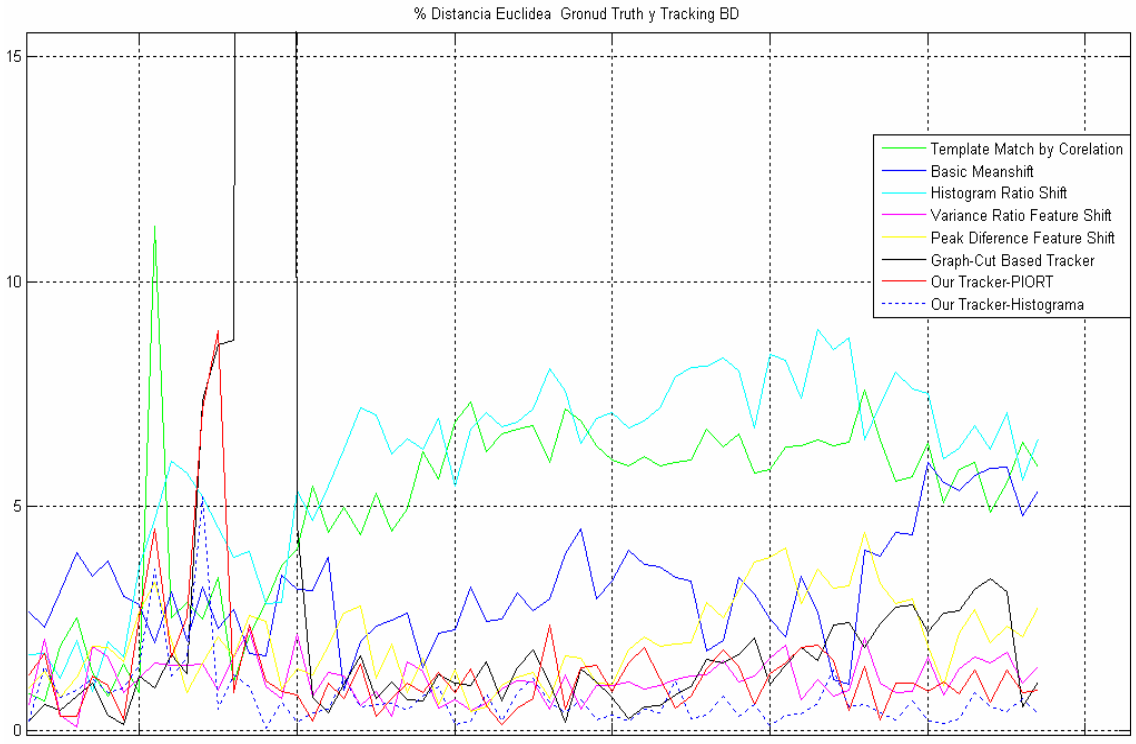


Figure A3 Euclidean distance between ground truth and tracking result centers for S1

S1 Blue balls crossed Right ball Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.0154	0.9846	NaN	0.9846	NaN	0.0154	NaN
BM	0	1.0000	NaN	1.0000	NaN	0	NaN
HRS	0	1.0000	NaN	1.0000	NaN	0	NaN
VRFS	0	1.0000	NaN	1.0000	NaN	0	NaN
PDFS	0	1.0000	NaN	1.0000	NaN	0	NaN
GCBT	0.0161	0.9385	NaN	0.9839	0	0.0615	NaN
PIORT-Neural Net	0	1.0000	NaN	1.0000	NaN	0	NaN
PIORT-Bayesian	0	1.0000	NaN	1.0000	NaN	0	NaN

Table A1 Tracking metrics S1

ISBN:978-84-693-3387-7/DL:T.1001-2010

S2 Blue balls crossed. Left ball tracking - 67 frames

Figure A4 Tracking for some consecutive frames in S2.

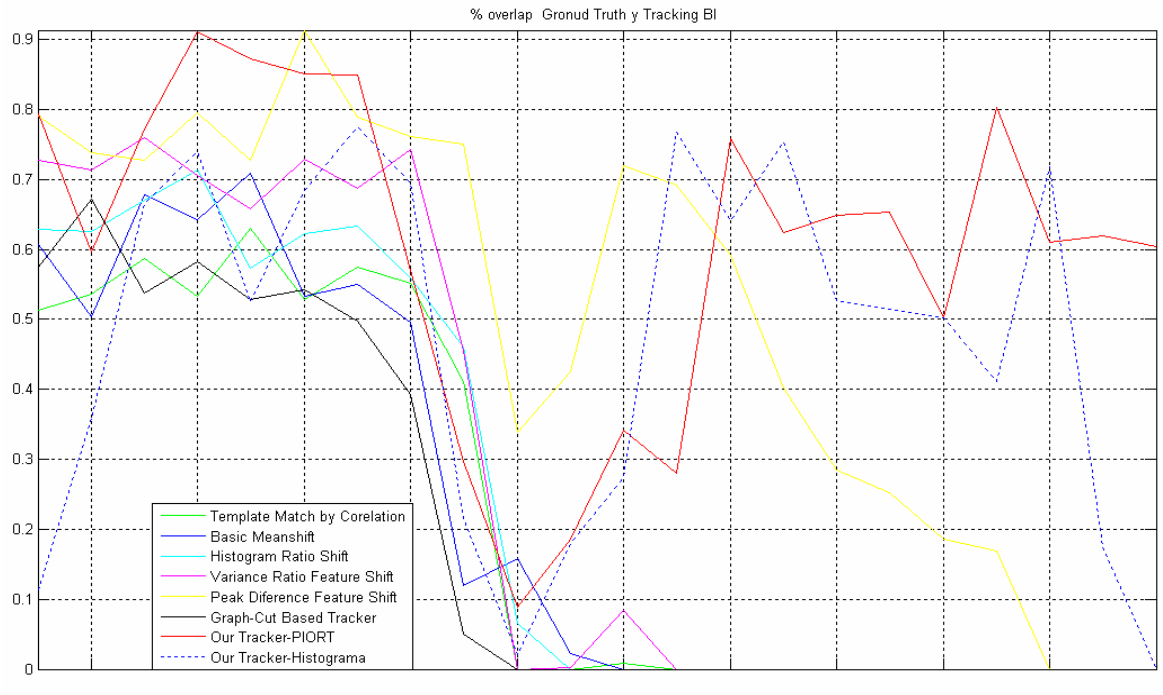


Figure A5 Overlap between ground truth and tracking results for S2.

ISBN:978-84-693-3387-7/DL:T.1001-2010

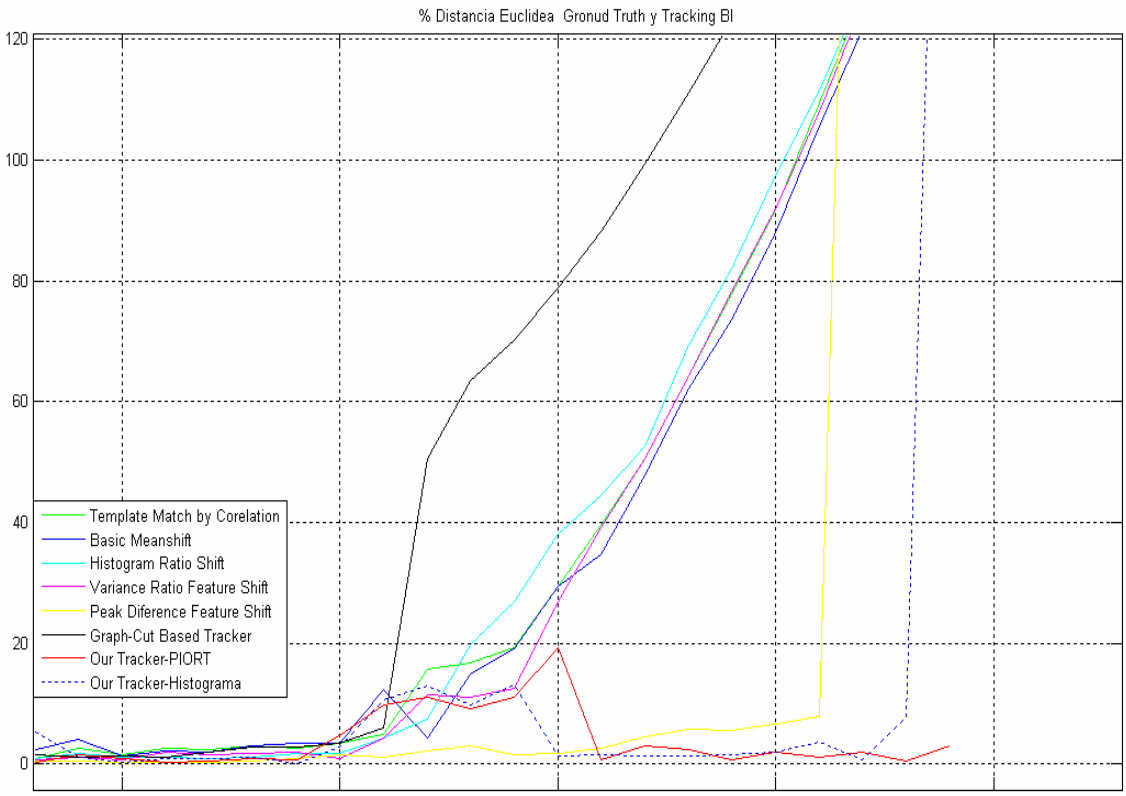


Figure A6 Euclidean distance between ground truth and tracking result centers for S2.

S2 Blue balls crossed Left ball Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.5909	0.4091	NaN	0.4091	NaN	0.5909	NaN
BM	0.6364	0.3636	NaN	0.3636	NaN	0.6364	NaN
HRS	0.5909	0.4091	NaN	0.4091	NaN	0.5909	NaN
VRFS	0.5909	0.4091	NaN	0.4091	NaN	0.5909	NaN
PDFS	0.1053	0.7727	NaN	0.8947	0	0.2273	NaN
GCBT	0.6364	0.3636	NaN	0.3636	NaN	0.6364	NaN
PIORT-Neural Net	0.0909	0.9091	NaN	0.9091	NaN	0.0909	NaN
PIORT-Bayesian	0.2381	0.7273	NaN	0.7619	0	0.2727	NaN

Table A2 Tracking metrics S2

ISBN:978-84-693-3387-7/DL:T.1001-2010

S3 Blue ball moving occluded by box - 75 frames

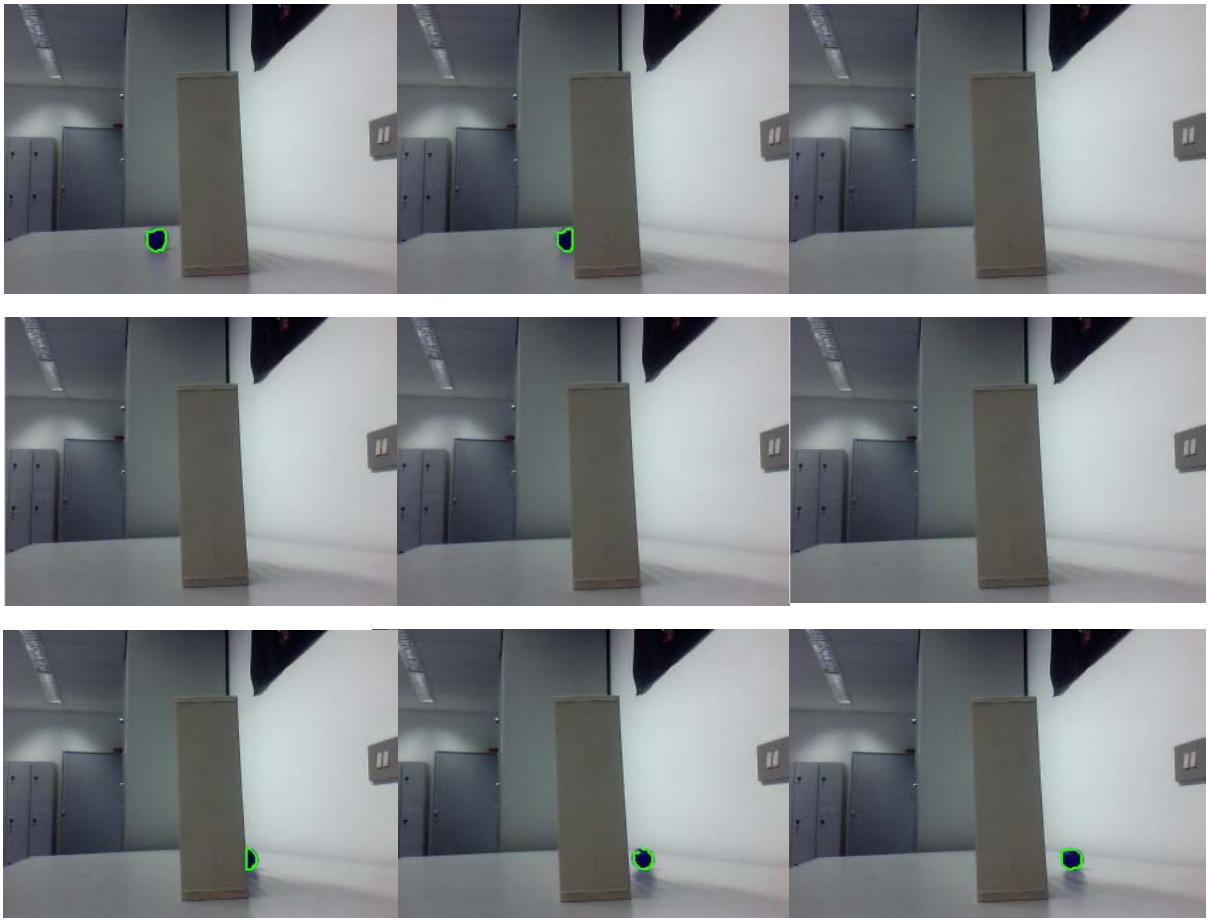


Figure A7 Tracking for some consecutive frames in S3.

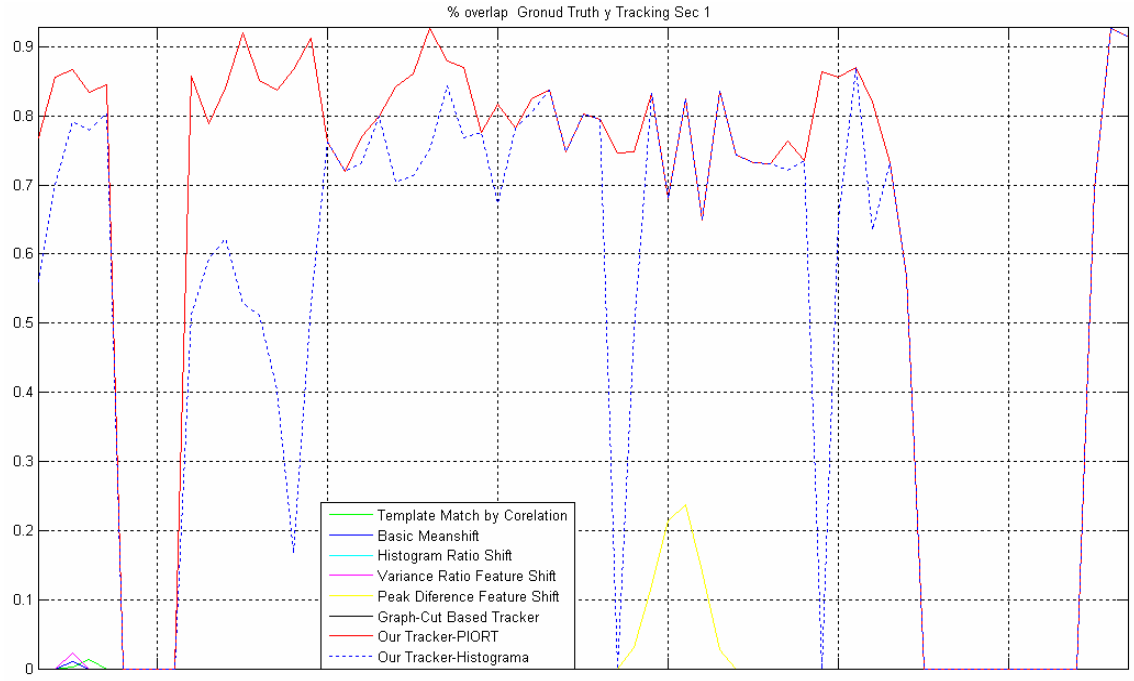


Figure A8 Overlap between ground truth and tracking results for S3

ISBN:978-84-693-3387-7/DL:T.1001-2010

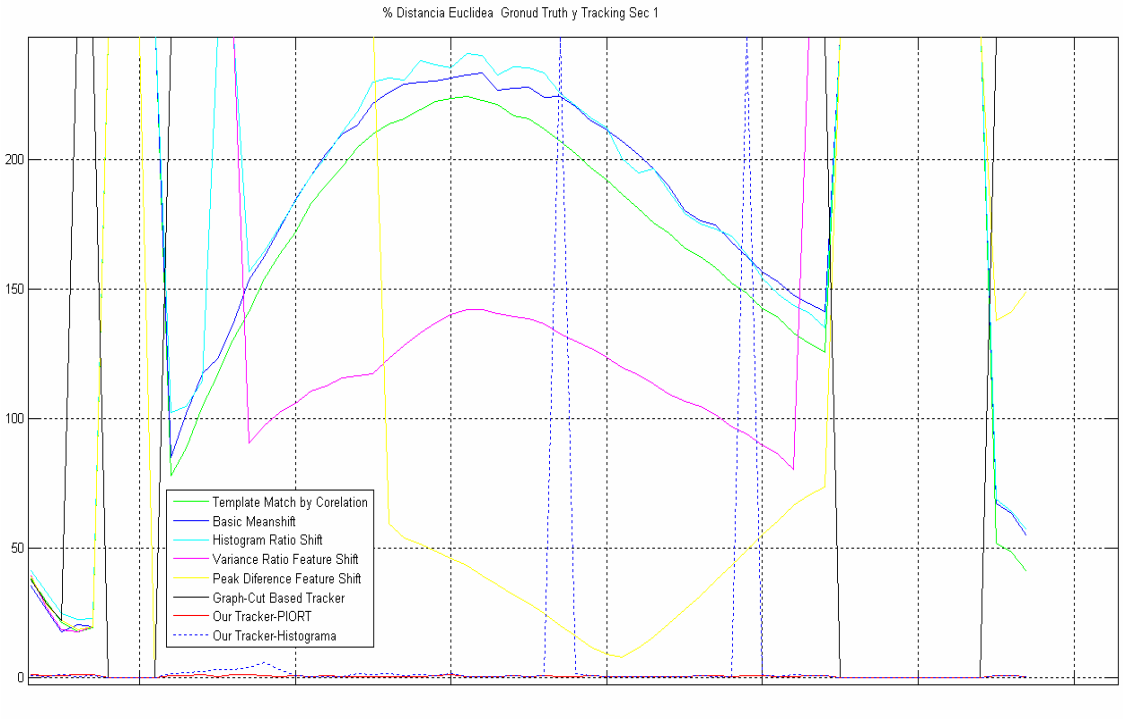


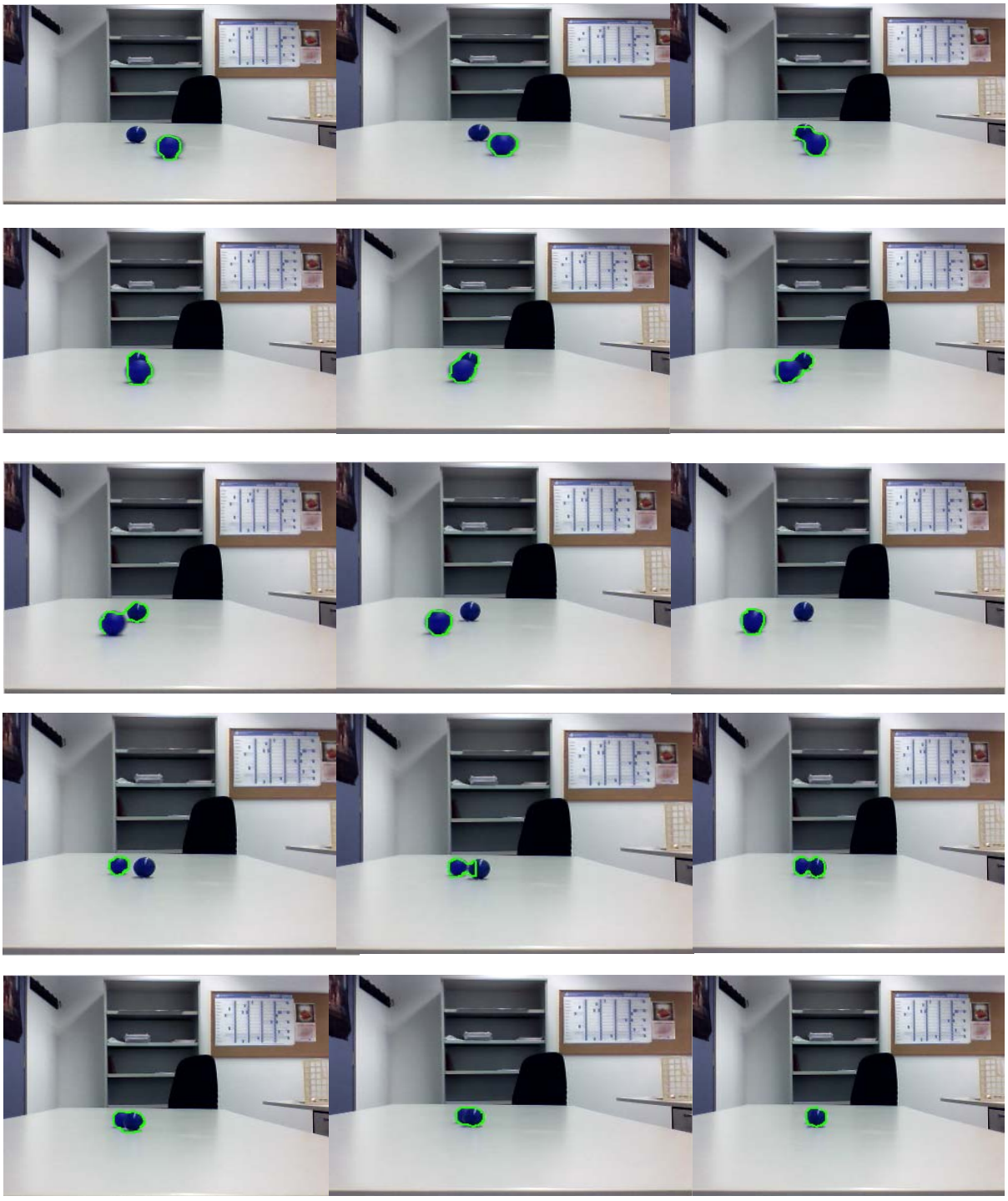
Figure A9 Euclidean distance between ground truth and tracking result centers for S3.

S3 Blue ball moving occluded by box Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	1	0	0	0	NaN	1	1.0000
BM	1	0	0	0	NaN	1	1.0000
HRS	1	0	0	0	0	1	1.0000
VRFS	1	0	0.7857	0	0.5238	1	0.2143
PDFS	1	0	0.0714	0	0.0667	1	0.9286
GCBT	1	0	1.0000	0	0.2258	1	0
PIORT-Neural Net	0	1	1.0000	1	1.0000	0	0
PIORT-Bayesian	0.0204	0.9412	1.0000	0.9796	0.8750	0.0588	0

Table A3 Tracking metrics S3

ISBN:978-84-693-3387-7/DL:T.1001-2010

S4 Blue ball moving around still blue ball -75 frames



ISBN:978-84-693-3387-7/DL:T.1001-2010

Figure A10 Tracking for some consecutive frames in S4.

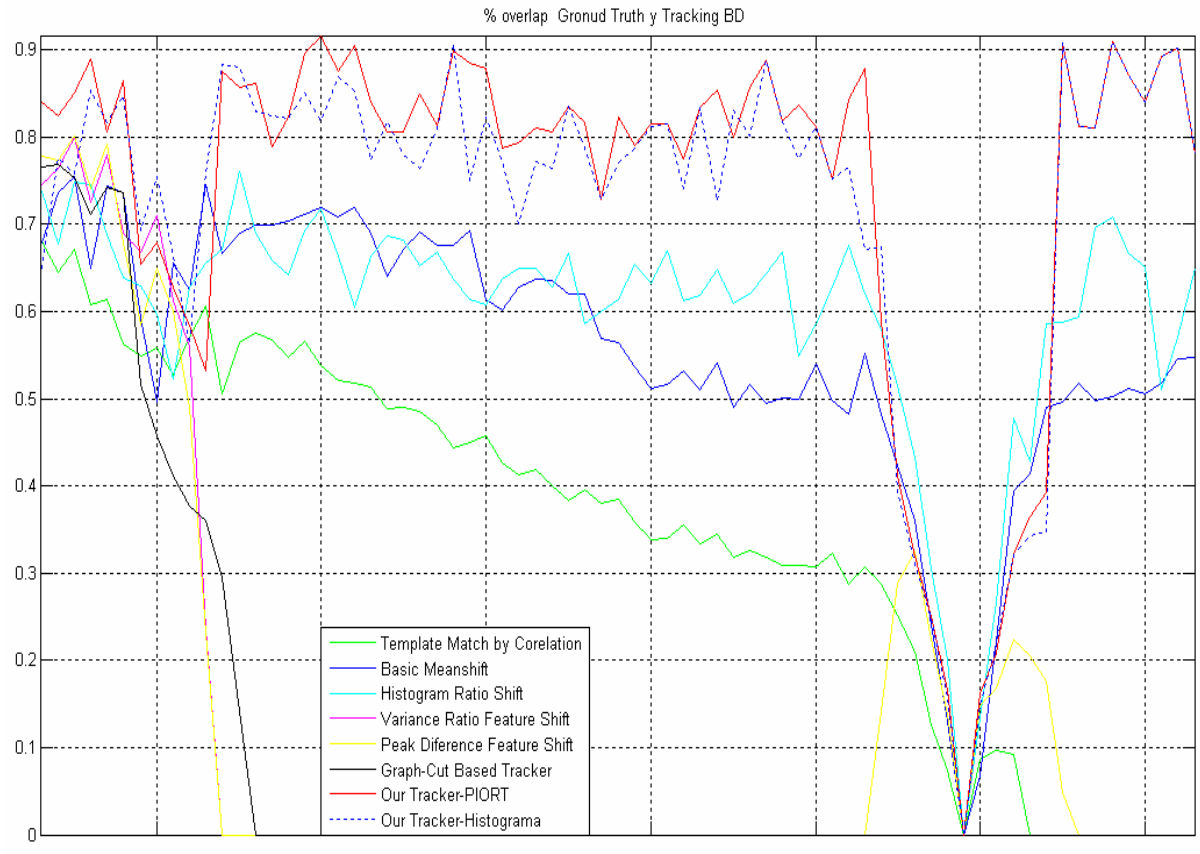


Figure A11 Overlap between ground truth and tracking results for S4

ISBN:978-84-693-3387-7/DL:T.1001-2010

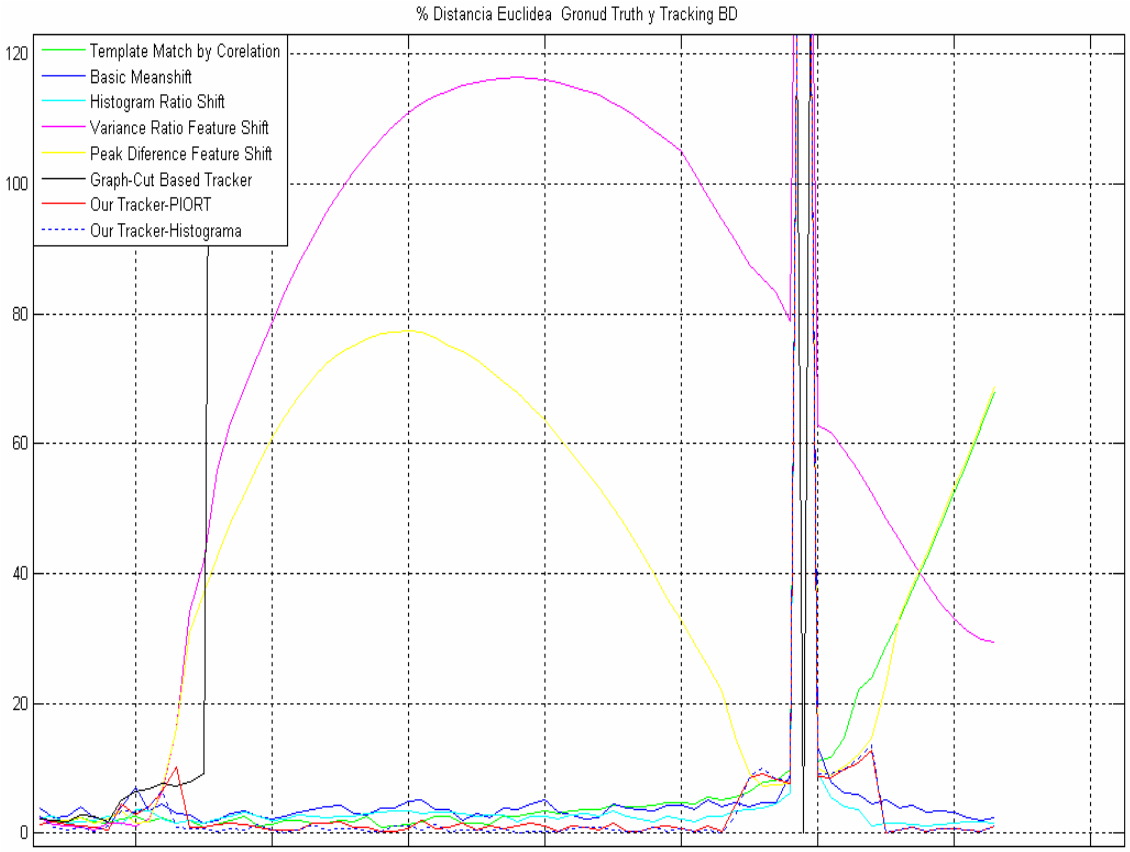


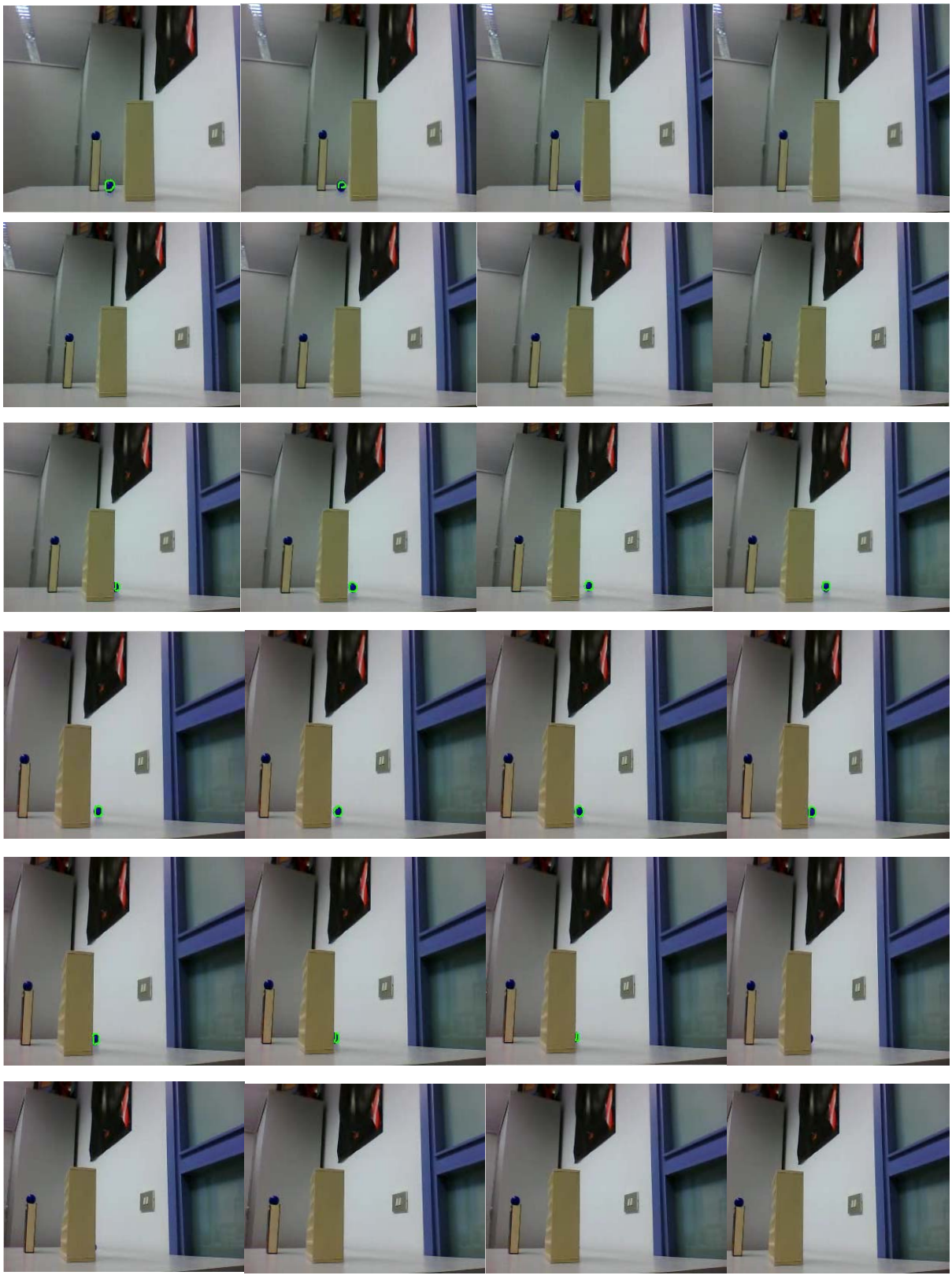
Figure A12 Euclidean distance between ground truth and tracking result centers for S4.

S4 Blue ball moving around still blue ball Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.2535	0.7571	0	0.7465	NaN	0.2429	1
BM	0.0704	0.9429	0	0.9296	NaN	0.0571	1
HRS	0.0423	0.9714	0	0.9577	NaN	0.0286	1
VRFS	0.8592	0.1429	0	0.1408	NaN	0.8571	1
PDFS	0.8310	0.1714	0	0.1690	NaN	0.8286	1
GCBT	0.0769	0.1714	1	0.9231	0.0172	0.8286	0
PIORT-Neural Net	0.0563	0.9571	0	0.9437	NaN	0.0429	1
PIORT-Bayesian	0.0563	0.9571	0	0.9437	NaN	0.0429	1

Table A4 Tracking metrics S4

ISBN:978-84-693-3387-7/DL:T.1001-2010

S5 Blue bouncing ball on table - 106 frames



ISBN:978-84-693-3387-7/DL:T.1001-2010

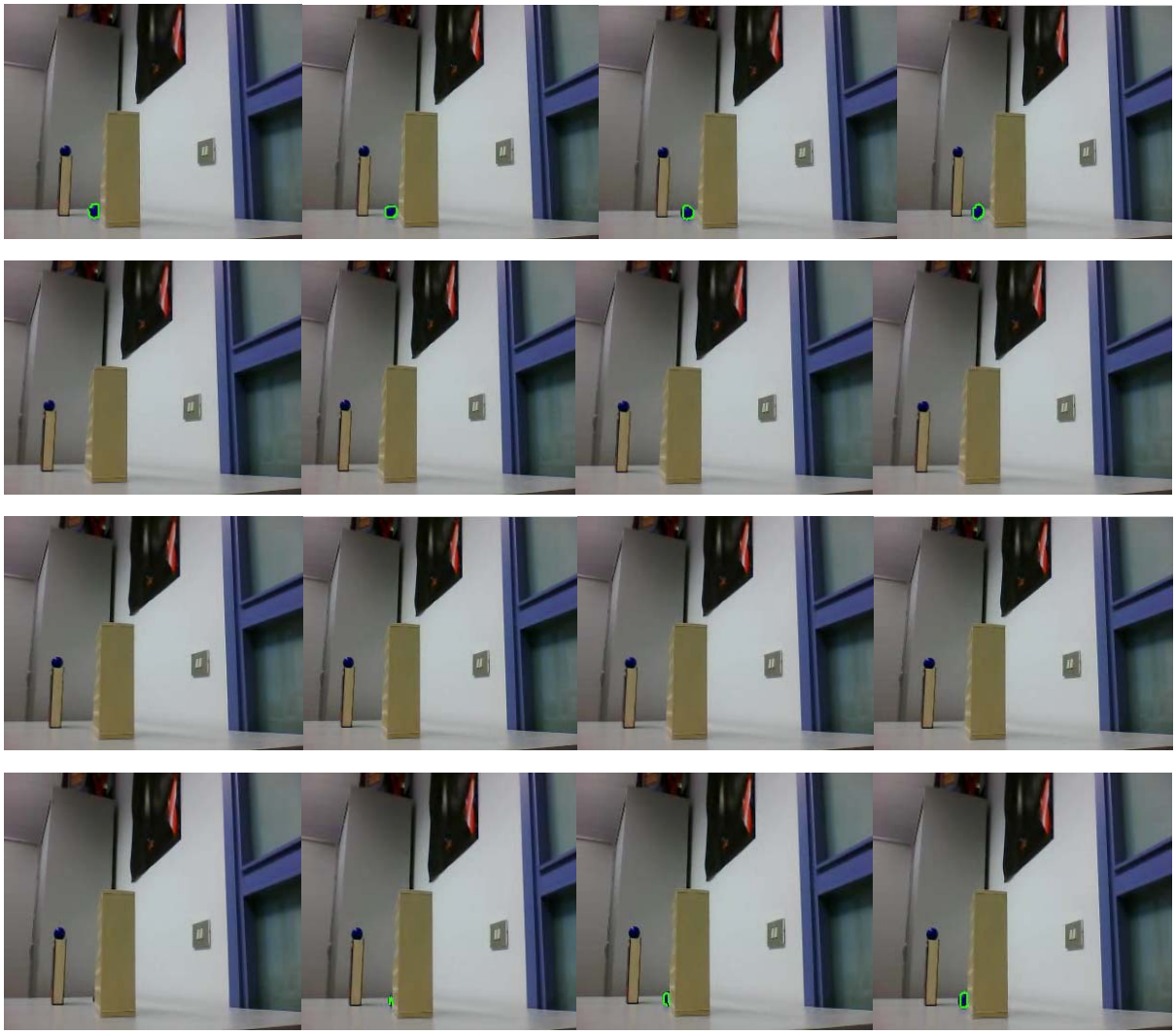


Figure A13 Tracking for some consecutive frames in S5.

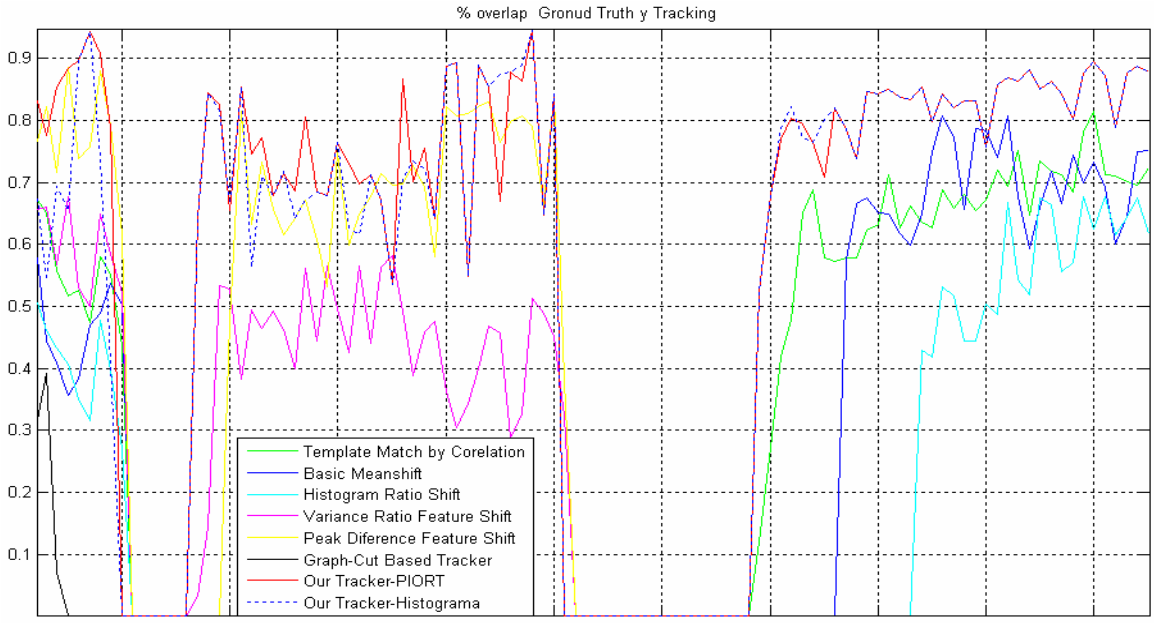


Figure A14 Overlap between ground truth and tracking results for S5.

ISBN:978-84-693-3387-7/DL:T.1001-2010

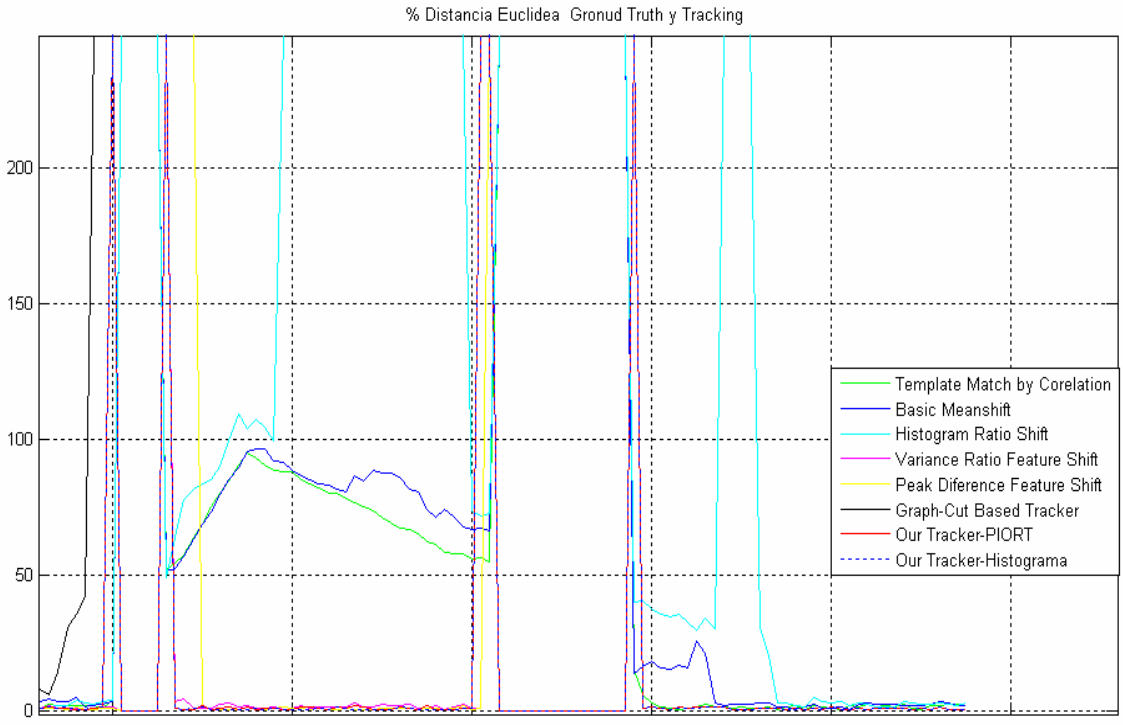


Figure A15 Euclidean distance between ground truth and tracking result centers for S5.

S5 Blue bouncing ball on table Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.5673	0.5357	0	0.4327	NaN	0.4643	1
BM	0.6346	0.4524	0	0.3654	NaN	0.5476	1
HRS	0.6076	0.3690	0	0.3924	0	0.6310	1
VRFS	0.0455	0.5000	1	0.9545	0.3333	0.5000	0
PDFS	0	0.4881	1	1.0000	0.3175	0.5119	0
GCBT	0.6667	0.0238	1	0.3333	0.2041	0.9762	0
PIORT-Neural Net	0	0.9405	1	1.0000	0.8000	0.0595	0
PIORT-Bayesian	0	0.9405	1	1.0000	0.8000	0.0595	0

Table A5 Tracking metrics S5

ISBN:978-84-693-3387-7/DL:T.1001-2010

S6 Segway - Orange ball on pavement - 342 frames

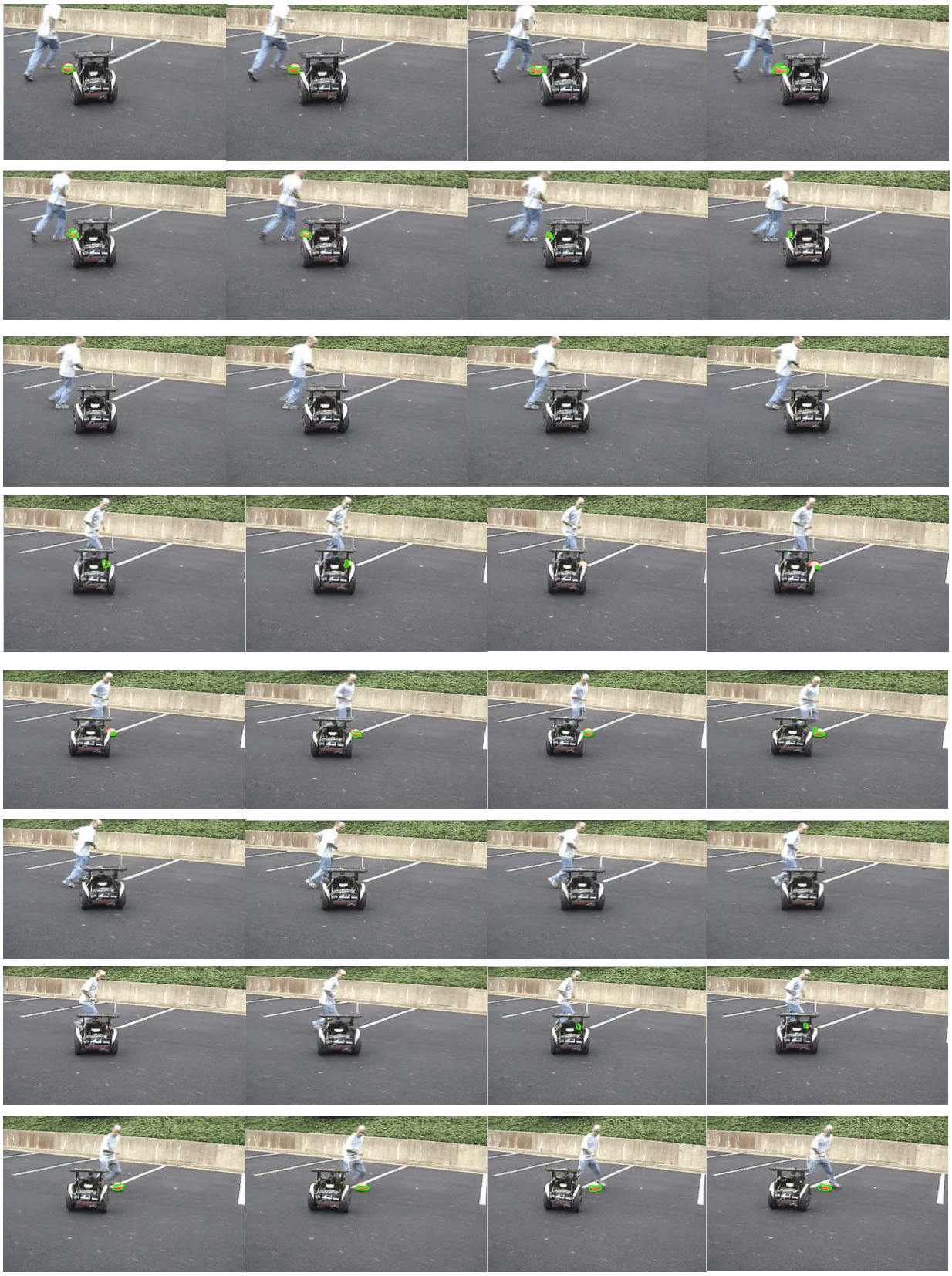


Figure A16 Tracking for some consecutive frames in S6.

ISBN:978-84-693-3387-7/DL:T.1001-2010

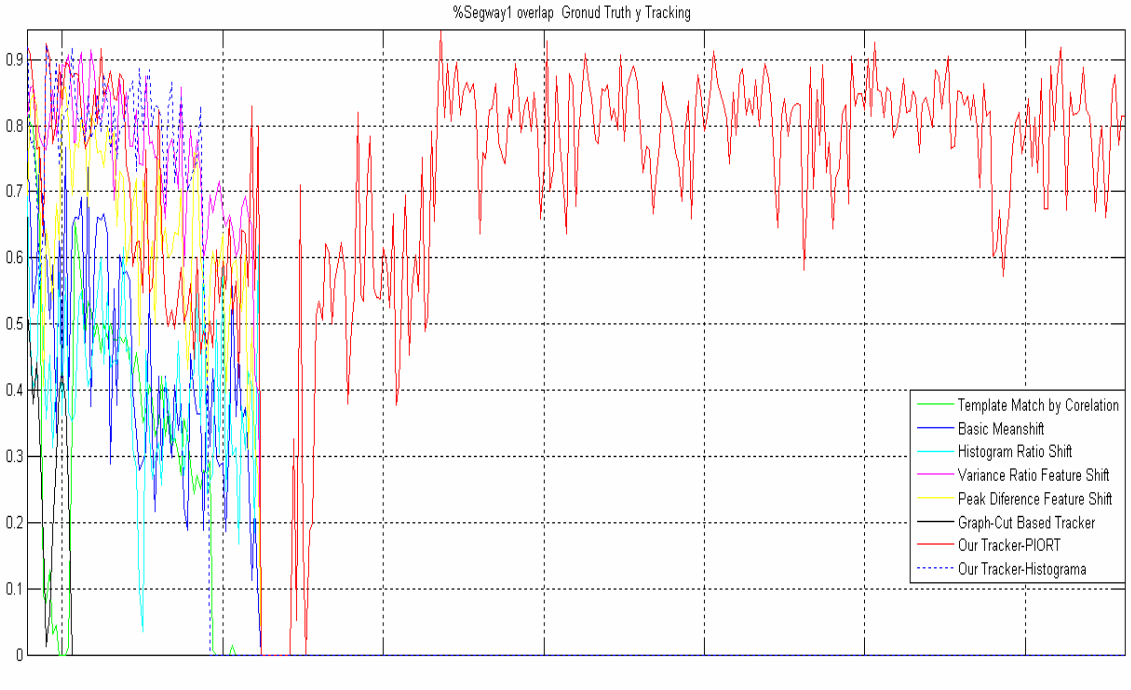


Figure A17 Overlap between ground truth and tracking results for S6

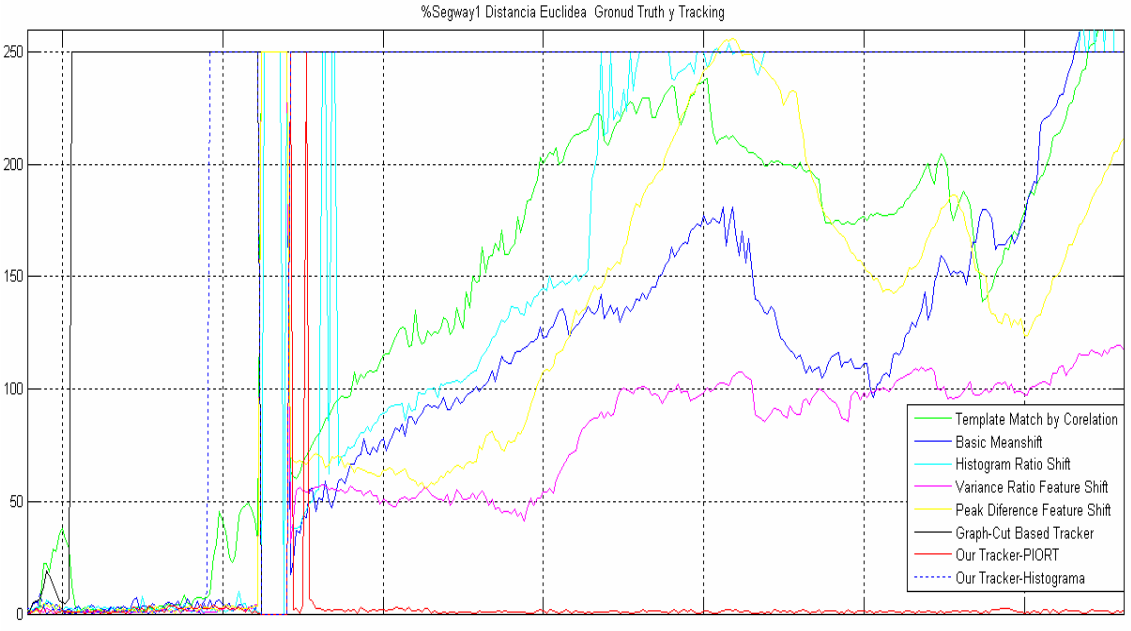


Figure A18 Euclidean distance between ground truth and tracking result centers for S6.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

S6 Segway - Orange ball on pavement Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.8601	0.1437	0	0.1399	NaN	0.8563	1.0000
BM	0.8105	0.1946	0	0.1895	NaN	0.8054	1.0000
HRS	0.6650	0.2066	0.2222	0.3350	0.0146	0.7934	0.7778
VRFS	0.7872	0.2186	0	0.2128	NaN	0.7814	1.0000
PDFS	0.7872	0.2186	0	0.2128	NaN	0.7814	1.0000
GCBT	0.2857	0.0299	1.0000	0.7143	0.0274	0.9701	0
PIORT-Neural Net	0.0120	0.9820	1.0000	0.9880	0.8182	0.0180	0
PIORT-Bayesian	0	0.1707	1.0000	1.0000	0.0315	0.8293	0

Table A6 Tracking metrics S6

S7 Segway - Orange ball on grass - 407 frames

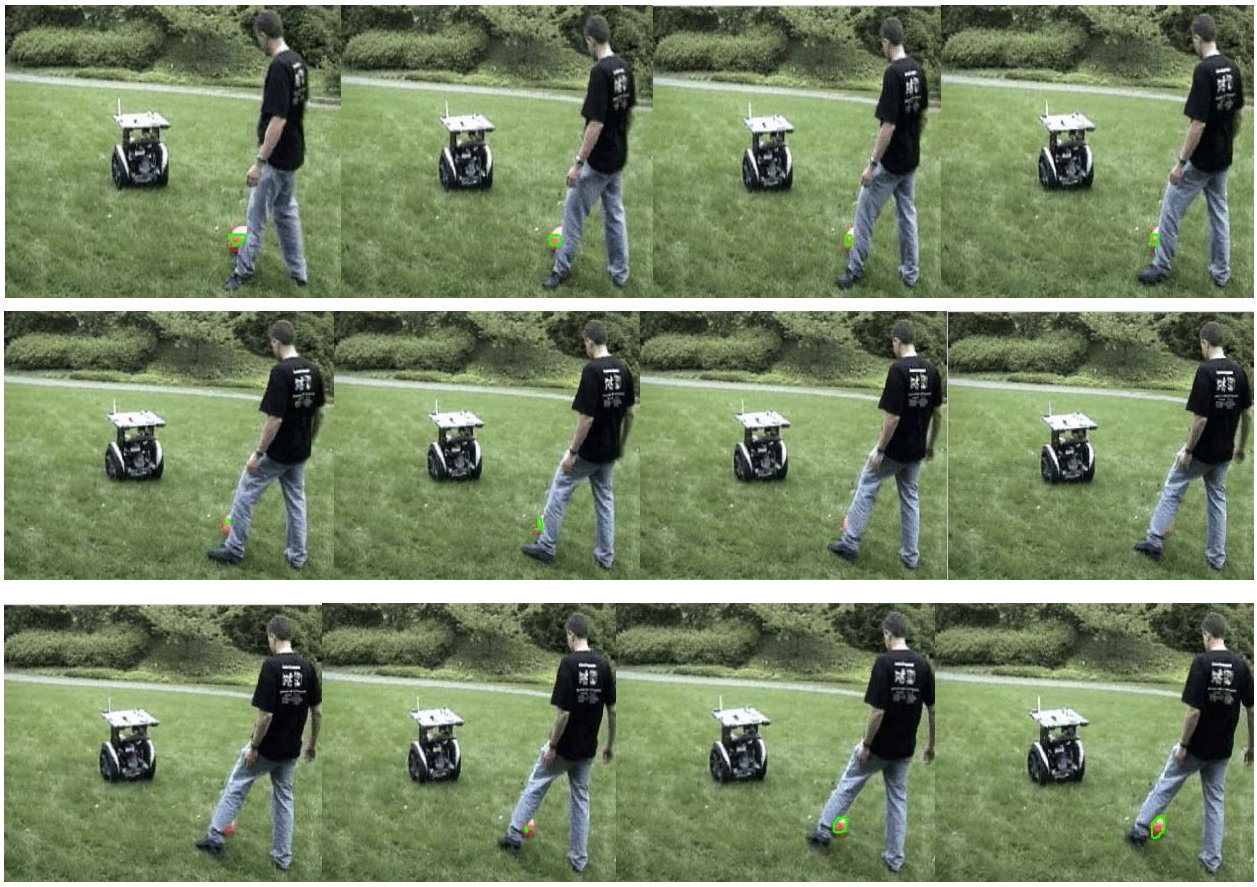


Figure A19 Tracking for some consecutive frames in S7.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

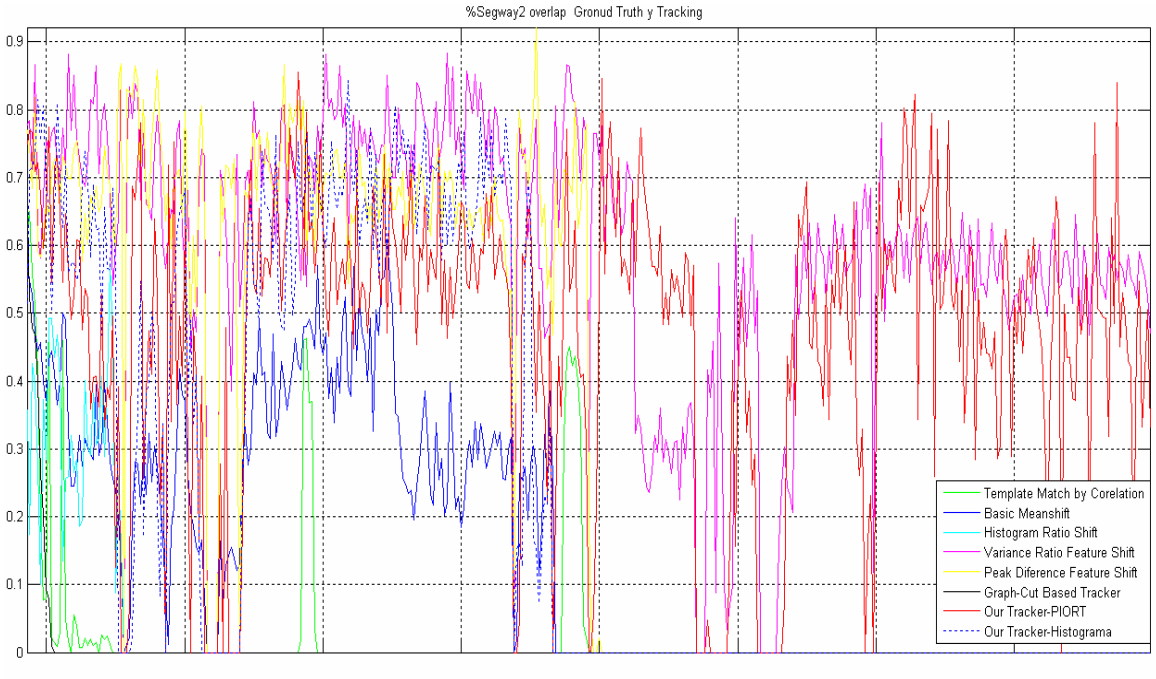


Figure A20 Overlap between ground truth and tracking results for S7.

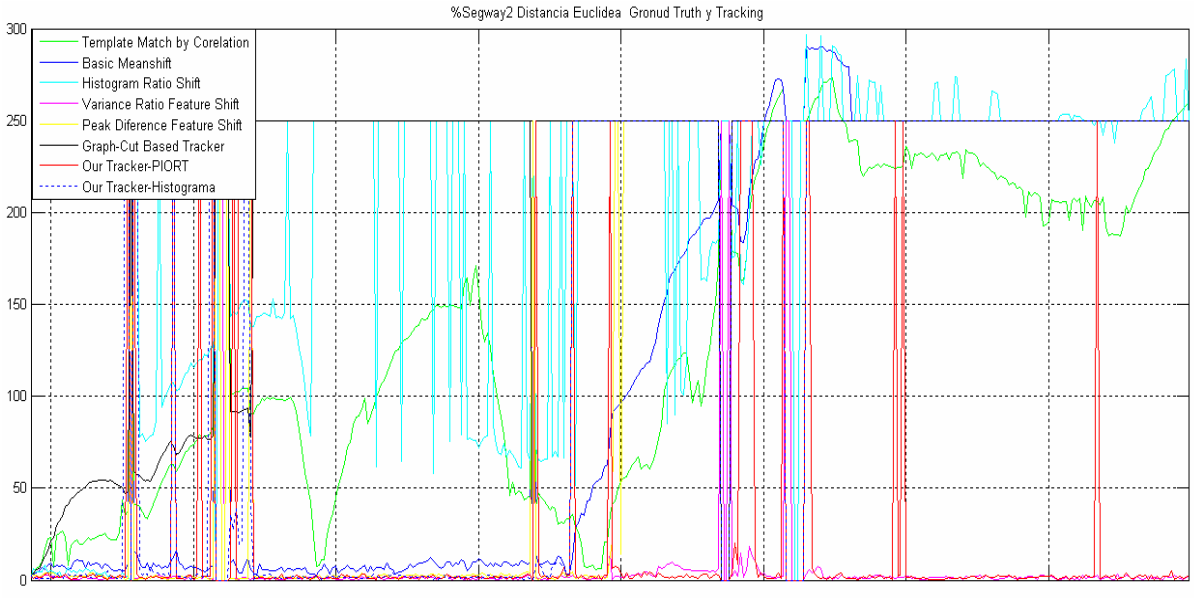


Figure A21 Euclidean distance between ground truth and tracking result centers for S7.

ISBN:978-84-693-3387-7/DL:T.1001-2010

S7 Segway - Orange ball on grass Tracking Algoritm	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.0463	0	0.0442	NaN	0.9537	1.0000
BM	0.3625	0	0.4896	0	0.6375	1.0000
HRS	0.0720	0.2222	0.1429	0.0190	0.9280	0.7778
VRFS	0.9717	0.4444	0.9474	1.0000	0.0283	0.5556
PDFS	0.5013	0.7222	0.9606	0.0637	0.4987	0.2778
GCBT	0.0154	0.6667	0.0769	0.0365	0.9846	0.3333
PIORT-Neural Net	0.8792	1.0000	0.9396	0.4186	0.1208	0
PIORT-Bayesian	0.3933	1.0000	0.8793	0.0773	0.6067	0

Table A7 Tracking metrics S7.

S8 Pedestrian with red jacket - 215 frames

ISBN:978-84-693-3387-7/DL:T.1001-2010

Figure A22 Tracking for some consecutive frames in S8.

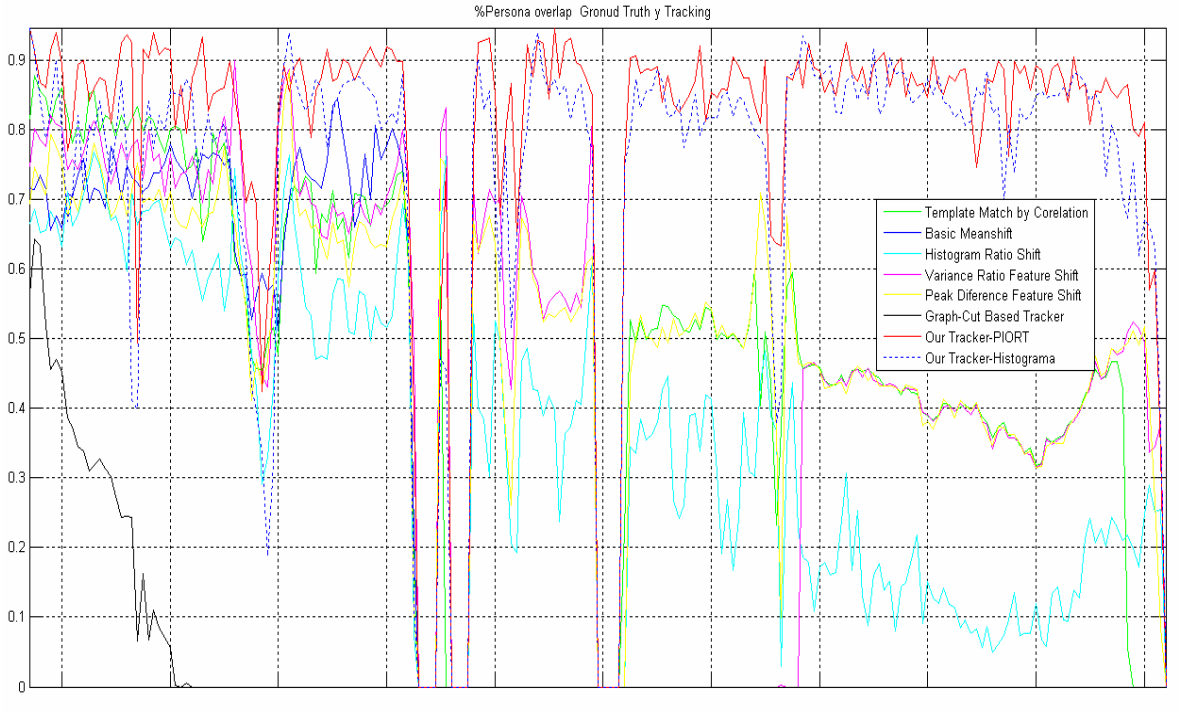


Figure A23 Overlap between ground truth and tracking results for S8

ISBN:978-84-693-3387-7/DL:T.1001-2010

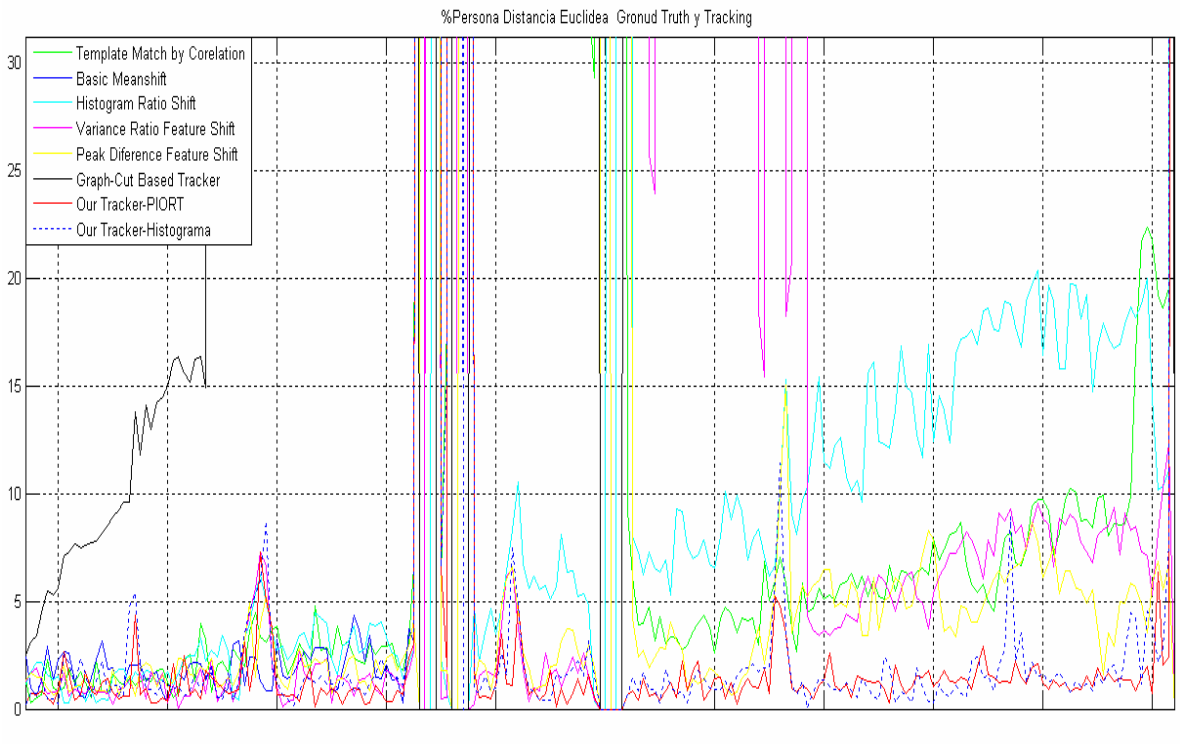


Figure A24 Euclidean distance between ground truth and tracking result centers for S8.

S8 Pedestrian with red jacket Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.2275	0.8274	0	0.7725	NaN	0.1726	1.0000
BM	0.6635	0.3604	0	0.3365	NaN	0.6396	1.0000
HRS	0.3700	0.6396	0.7143	0.6300	0.9091	0.3604	0.2857
VRFS	0.0989	0.8325	0.6429	0.9011	0.3103	0.1675	0.3571
PDFS	0.0446	0.9797	0.5714	0.9554	0.8889	0.0203	0.4286
GCBT	0.4118	0.1015	1.0000	0.5882	0.0791	0.8985	0
PIORT-Neural Net	0.0437	1.0000	0.3571	0.9563	1.0000	0	0.6429
PIORT-Bayesian	0.0488	0.9898	0.4286	0.9512	1.0000	0.0102	0.5714

Table A8 Tracking metrics S8

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

S9 Guy on Segway with orange T-shirt - 297 frames

Figure A25 Tracking for some consecutive frames in S9.

ISBN:978-84-693-3387-7/DL:T.1001-2010

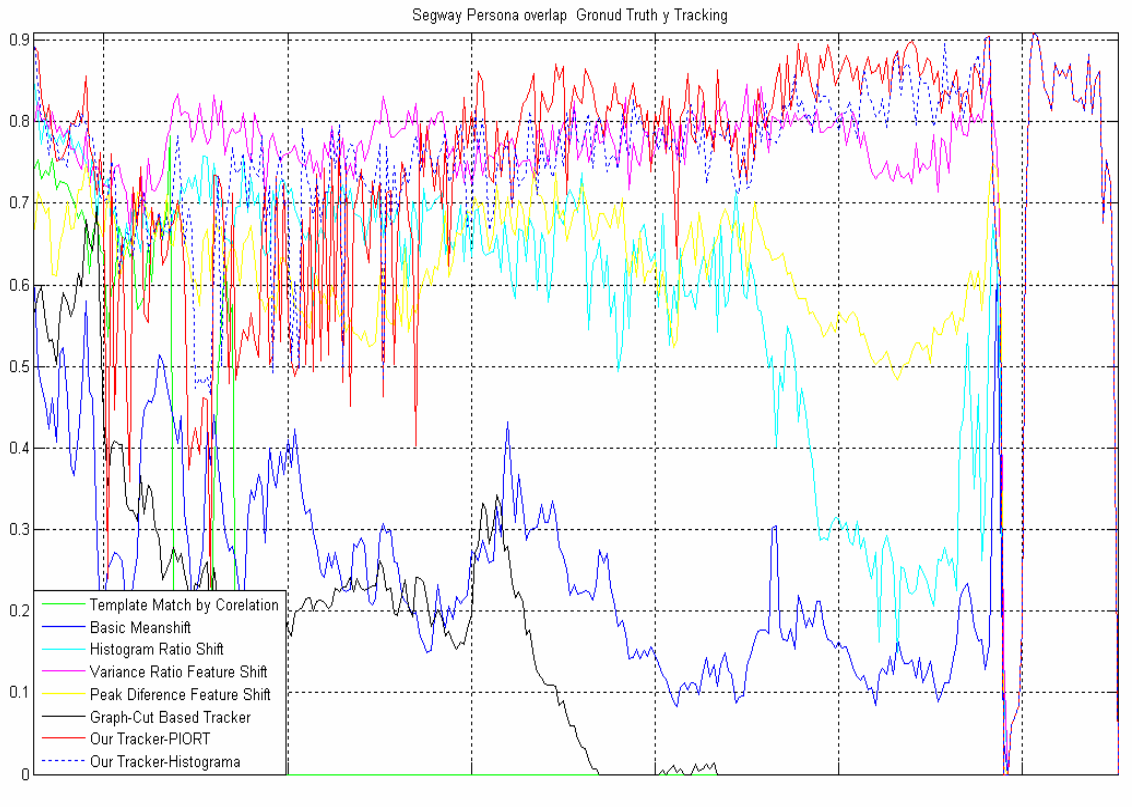


Figure A26 Overlap between ground truth and tracking results for S9.

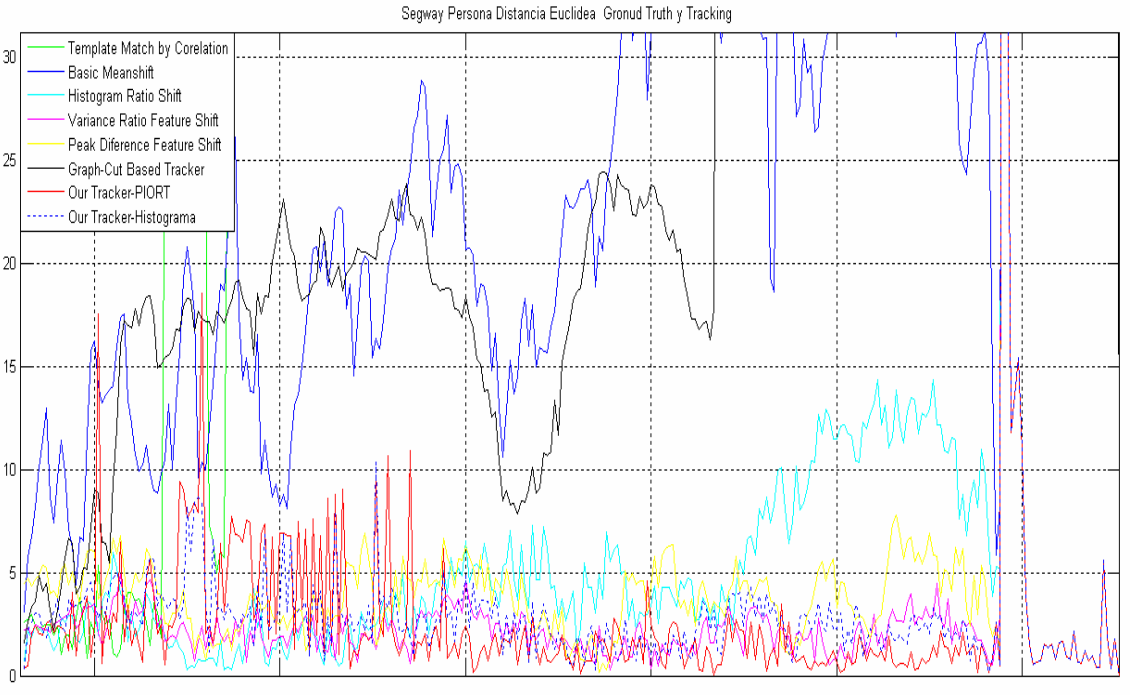


Figure A27 Euclidean distance between ground truth and tracking result centers for S9.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

S9 Guy on Segway with orange T-shirt Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.8514	0.1492	0	0.1486	NaN	0.8508	1
BM	0.5952	0.4034	0	0.4048	0	0.5966	1
HRS	0.0896	0.8610	1	0.9104	0.0588	0.1390	0
VRFS	0.0038	0.8949	1	0.9962	0.0323	0.1051	0
PDFS	0.0038	0.8949	1	0.9962	0.0323	0.1051	0
GCBT	0.7005	0.1898	1	0.2995	0.0092	0.8102	0
PIORT-Neural Net	0.0238	0.9729	1	0.9762	0.5000	0.0271	0
PIORT-Bayesian	0.0170	0.9797	1	0.9830	0.5000	0.0203	0

Table A9 Tracking metrics S9

S10 Men on segway with orange T-shirts - 256 frames

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

Figure A28 Tracking for some consecutive frames in S10.

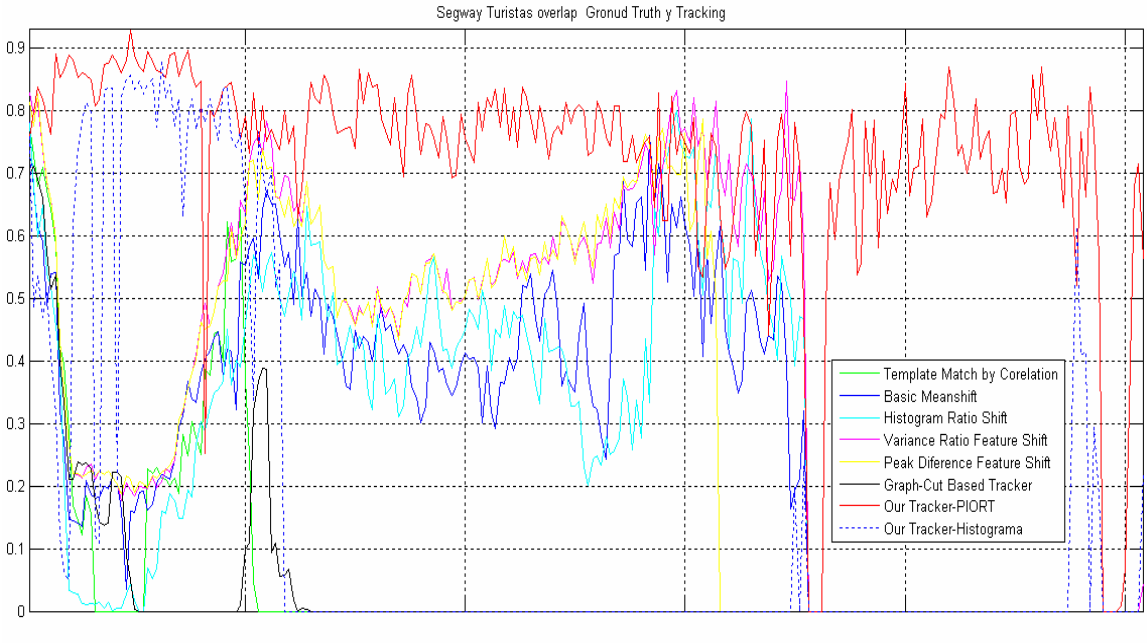


Figure A29 Overlap between ground truth and tracking results for S10.

UNIVERSITAT ROVIRA I VIRGILI
A PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING FRAMEWORK FOR VIDEO SEQUENCES
Nicolás Amézquita Gómez
ISBN:978-84-693-3387-7/DL:T.1001-2010

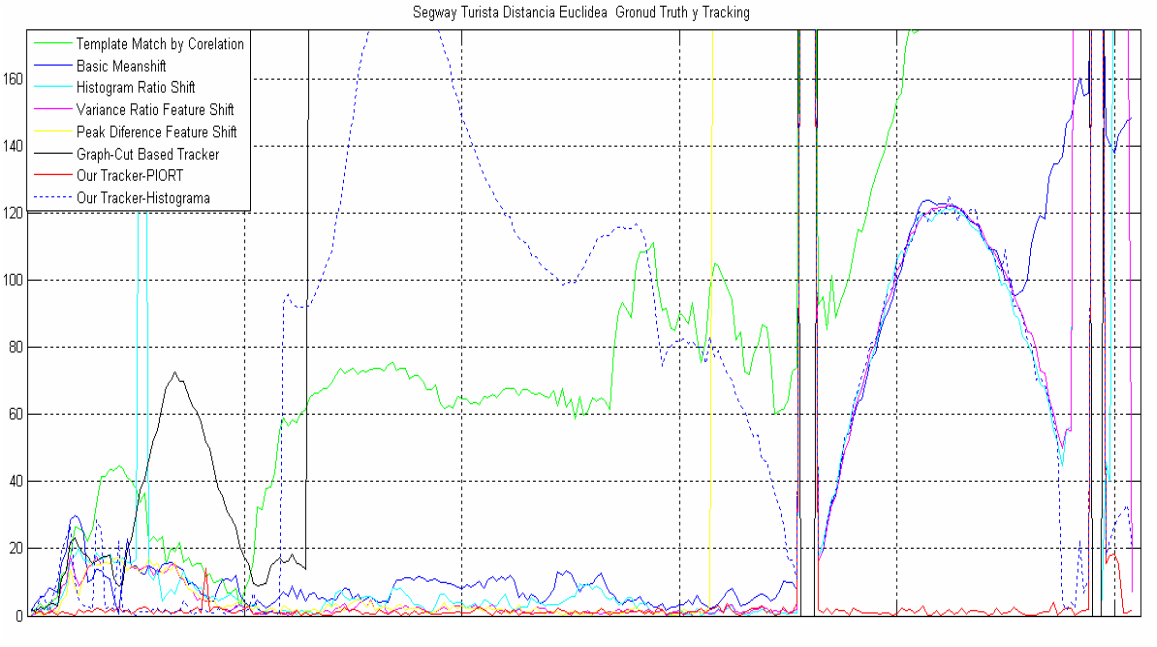


Figure A30 Euclidean distance between ground truth and tracking result centers for S10.

S10 Men on segway with orange T-shirts Tracking Algoritm	False Alarm Rate	Detection Rate	Specificity	Positive Prediction	Negative Prediction	False Negative Rate	False Positive Rate
TMC	0.8976	0.1053	0	0.1024	NaN	0.8947	1.0000
BM	0.4134	0.6032	0	0.5866	NaN	0.3968	1.0000
HRS	0.4000	0.5830	0.4286	0.6000	0.2143	0.4170	0.5714
VRFS	0.3610	0.6235	0.4286	0.6390	0.2308	0.3765	0.5714
PDFS	0.1465	0.5425	1.0000	0.8535	0.0722	0.4575	0
GCBT	0.7969	0.0526	1.0000	0.2031	0.0368	0.9474	0
PIORT-Neural Net	0.0394	0.9879	0	0.9606	NaN	0.0121	1.0000
PIORT-Bayesian	0.7756	0.2308	0	0.2244	NaN	0.7692	1.0000

Table A10 Tracking metrics S10