Hierarchical Fuzzy Logic Based Approach for Object Tracking

Nuno Vieira Lopes[a,b], Pedro Couto[b], Aranzazu Jurio[c], Pedro Melo-Pinto[b]

[a] *School of Technology and Management, Polytechnic Institute of Leiria, Morro do Lena, Alto do Vieiro, Apartado 4163, 2411-901 Leiria, Portugal*
[b] *CITAB - Centre for the Research and Technology of Agro-Environmental and Biological Sciences, University of Trás-os-Montes e Alto Douro, Quinta de Prados, 5000-911 Vila Real, Portugal*
[c] *Department of Automatic and Computation, Public University of Navarra, Campus Arrosadía, 31006 Pamplona, Spain*

Abstract

In this paper a novel tracking approach based on fuzzy concepts is introduced. A methodology for both single and multiple object tracking is presented. The aim of this methodology is to use these concepts as a tool to, while maintaining the needed accuracy, reduce the complexity usually involved in object tracking problems. Several dynamic fuzzy sets are constructed according to both kinematic and non kinematic properties that distinguish the object to be tracked. Meanwhile kinematic related fuzzy sets model the object's motion pattern, the non kinematic fuzzy sets model the object's appearance. The tracking task is performed through the fusion of these fuzzy models by means of an inference engine. This way, object detection and matching steps are performed exclusively using inference rules on fuzzy sets. In the multiple object methodology, each object is associated with a confidence degree and a hierarchical implementation is performed based on that confidence degree.

Keywords:
Dynamic fuzzy sets, inference engine, hierarchical, multiple object tracking

1. Introduction

Object tracking plays an important role in computer vision. During the last years, extensive research has been conducted in this field and many types and applications of object tracking systems have been proposed in the literature such as automated surveillance, vehicle navigation, human computer interaction and traffic analysis [1, 2, 3, 4, 5, 6, 7]. Tracking is essential to many applications and robust tracking algorithms are still a huge challenge. Difficulties can arise due to noise presence in images, quick changes in lighting conditions, abrupt or complex object motion, changing appearance patterns of the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, camera motion and real time processing requirements. Typically, assumptions are made to constrain the tracking problem in the context of particular applications. For instance, almost all tracking algorithms assume that the object motion is smooth or impose constrains on the object motion to be constant in velocity or acceleration. Multiple view image tracking or prior knowledge about objects, such as size, number or shape, can also be used to simplify the process. In this work, the word "object" refers to the template image pattern being tracked (e.g. person's hair, briefcase, etc.).

Normally, tracking is seen as a main task involving several subtasks such as image segmentation for object detection, object matching and object position estimation. A myriad of algorithms has been developed to implement this subtasks but each one have their strengths and weaknesses and, over the last years, extensive research has been made in this field to find optimal tracking systems for specific applications. Many approaches of tracking techniques have been proposed in the literature, however, they are not completely accurate for all kind of scenarios and just provide good results when a certain number of assumptions are verified. Moreover, tracking methodologies that are not designed for particular applications, where specific and well established assumptions or constrains can easily be imposed, tend to be very complex. These reasons are the motivation to study and implement new tracking approaches where the introduction of soft computing techniques, such as fuzzy logic, is intended for:

- Reducing the tracking task complexity by endowing the methodology with the capability of incorporating reasoning in the same sense that human reasoning simplifies real tracking problems (e.g. most tracking problems are not complex for humans, they are indeed trivial in most situations).
- Endowing the methodology with the needed scalability in order to cope with the specific needs of different tracking problems by easily adding, changing or adapting the used fuzzy sets while maintaining its general framework.

The presented methodology intents to be an ease and feasible general framework for object tracking that can easily be adapted to specific applications or problems.

The remainder of this paper is organized as follows. In Section 2 the definition and a review of object tracking is presented.

Email addresses: nuno.lopes@ipleiria.pt (Nuno Vieira Lopes), pcouto@utad.pt (Pedro Couto), aranzazu.jurio@unavarra.es (Aranzazu Jurio), pmelo@utad.pt (Pedro Melo-Pinto)

A general description of fuzzy set theory is presented in Section 3. The proposed approach is presented in Section 4. A possible implementation of the proposed approach is given at Section 5. Section 6 shows the experimental results to illustrate the effectiveness of the proposed approach and a comparative study with well known tracking approaches is performed. Finally Section 7 presents the final conclusions and future directions.

2. Object Tracking

Object tracking can be described as the problem of estimating the trajectory of an object as it moves around a scene. Although this general concept is almost consensual, the specific definition of tracking can change in the literature. Nevertheless, tracking systems must address two basic processes: figure-ground segmentation and temporal correspondences [8]. Figure-ground segmentation is the process of extracting the objects of interest from the video frame. Segmentation methods are applied as the first step in many tracking systems and therefore they are a crucial task. Object detection can be based on motion [9, 10], appearance [11, 12, 13], etc. Temporal correspondence concerns to the association of the detected objects in the current frame with those in the previous frames defining temporal trajectories [14, 15].

In [16], tracking is described as a motion problem and a matching problem. In this work, the motion problem is related with the prediction of the object location in the next frame. The second step is similar to the explained above. However, [1] and [17] present a wider description of tracking with three steps: detection of interesting objects, tracking such objects and analysis of object tracks to recognize their behavior. In [18] this behavior analysis is seen as a further interpretation of tracking results.

The selection of the most suitable feature to track is a critical role in tracking systems. The uniqueness of such feature provides an easy way to distinguish the object in the scene along time. Properties as intensity, color, gradient, texture or motion are commonly used to perform object tracking.

According to its properties, object tracking could be categorized in three groups: point, kernel and motion based approaches.

2.1. Point based tracking

Point based tracking approaches are suitable for tracking objects that occupy small regions in an image or they can be represented by several distinctive points. These points must be representative of the object and invariant to changes in illumination, object orientation and camera viewpoint. Points denoting significant gradient in intensity are preferred and commonly used by different detectors such as Harris [19], KLT [20] and SIFT [21]. To deal with the point correspondence problem between frames, deterministic constraints such as proximity, maximum velocity and small velocity change could be used. An alternative is to use statistical methods such as Kalman or particle filters. KLT and SIFT approaches provide internal methodologies to address the correspondence problem. Scale-invariant feature transform (or SIFT) is a well-known algorithm for object recognition and tracking. Interesting points are extracted from the object to provide a set of descriptors. These descriptors must be detected on the new image even among clutter, partial occlusion and uniform object scaling and rotation. In order to reduce computational time consumption, a research region in the next frame is defined according the last known location or based in a motion model of the object. However this method would typically not work with deformable or articulated objects since the relative positions between the descriptors differ from the original representation. To overcome this limitation an update scheme could be used and the object descriptors are recomputed after a predefined elapsed time.

2.2. Kernel based tracking

In this approach it is required a template or an appearance model of the object. Template tracking consists of searching in the current image for a region similar to the object template. The position and, consequently, the object matching between two consecutive frames is achieved by computing a similarity measure such as the cross-correlation. The cross-correlation concept is presented in [13]. Instead of templates, other object representations can be used for matching, for instance, color, color statistics, texture or histogram based information. The mean shift tracking algorithm is an efficient approach to tracking objects whose appearance can be described using histograms [22]. This iterative method maximizes the appearance similarity by comparing the histograms of the object and the region around the predicted object location. The Bhattacharya and Kullback-Leibler distances are commonly employed to measure the similarity between the template and the current target region. It fails in the case of occlusions and quick appearance changes.

2.3. Motion based tracking

This group of approaches perform tracking based on displacement or optical flow of image pixels. The optical flow of a pixel is a motion vector represented by the translation between a pixel in one frame and its corresponding pixel in the following frame. This computation has been proved to be difficult to achieve due to issues such as the brightness constancy assumption and the aperture problem. The classic tracking algorithm Kanade-Lucas-Tomasi (KLT) was firstly proposed by Lucas and Kanade in 1981, being perfected by Tomasi and Kanade in 1991 and explained in detail by Shi and Tomasi in 1994 [20]. The method proposed by Lucas and Kanade computes the optical flow for each pixel of an image, while the method proposed by Tomasi and Kanade known as KLT, extracts optimal points in the image and then computes the optical flow on the subsequent images to only this subset of points. The KLT is a complete method that provides a solution for two problems in computer vision: the problem of optimal selection of suitable points in an image and the problem of determining the correspondence between points in consecutive frames. It has little tolerance in image brightness variation and difficulty in detecting rapid object movements. Tracking moving objects can also be achieved by constructing a reference representation of the environment called background model and then finding deviations between this model and each incoming frame. A significant change between the background model and an image region denotes a moving object. This process is referred as background subtraction and represents a popular method especially under those situations with a relatively static background. An alternative approach to detect changes and, consequently the movement, between two consecutive intensity image frames $I(x, y, t)$ and $I(x, y, t-1)$ taken at times t and $t-1$, respectively, is to perform a pixelwise difference operation. Frame differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, i. e., there may be holes left inside slowly moving objects.

Since the arise of fuzzy logic theory, it has been successfully applied in a large range of areas such as process control systems, automotive navigation systems, information retrieval systems and image processing. As presented beforehand, a tracking system can be seen as a multi-stage process that comprise figure-ground segmentation and temporal correspondences. Hence, fuzzy logic can be used in these two different stages. [23] assigned a membership degree to the pixel using the relationship between its grey value and mean grey value of the region to which it belongs. For each grey level a fuzzy set is constructed and the optimal threshold value is the level of grey associated with the fuzzy set with lowest entropy. [24] proposed a segmentation approach using an extension of fuzzy sets theory, so called the Atanassov's Intuitionistic Fuzzy Sets, for representing the uncertainty of the expert in determining if a pixel belongs to the background or to the object. The optimal threshold value is associated with the intuitionistic fuzzy set of lowest entropy. In [25, 26] an automatic histogram threshold approach based on a fuzziness measures is presented. In [27] an active sonar system to track submarines using a Kalman filter and a posterior fuzzy rule logic association is presented. A fuzzy approach to assign one or several blobs to a track for automatic surveillance in airport areas is described in [28]. The previous work presented in [29, 30, 31], a multi feature tracking approach using dynamic fuzzy sets were introduced, however, no hierarchial scheme was implemented.

3. Fuzzy Sets Theory

In 1965, fuzzy sets were introduced by Zadeh [32] to represent or manipulate data and information containing nonstatistical uncertainties. This theory was specifically created to mathematically represent uncertainty and vagueness and to provide tools for dealing with the imprecision intrinsic to many problems.

A classical (crisp) set is defined as a collection of elements $x \in X$ where each single element can either belong to or not belong to a set A, $A \subseteq X$. However, fuzzy sets have more flexible membership requirements allowing the elements to have partial memberships between 0 and 1 rather than the unique memberships 0 and 1 like in classical sets.

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

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

$$\mu_{\tilde{A}}(x) = 0, \forall x \in X \tag{1}$$

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

$$\mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x), \forall x \in X \tag{2}$$

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

The membership function $\mu_{\tilde{A}}$ is also called grade of membership, degree of compatibility or degree of truth. The range of this function is a subset of the nonnegative real numbers whose supremum is finite, normally 1. The basic operations in fuzzy set theory are the complement, intersection and union. Since the membership function is the crucial component of a fuzzy set, it is therefore not surprising that operations with fuzzy sets are defined via their membership functions. These concepts, firstly suggested in [32], constitute a consistent framework for the theory of fuzzy sets. However, they are not unique since Zadeh and other authors have suggested consistent alternative or additional definitions for fuzzy set operations. The complement of a fuzzy set $\tilde{A}$ in X, written $\neg\tilde{A}$, is the fuzzy set

$$\neg\tilde{A} = \{(x, \mu_{\neg\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x)) | x \in X\}. \tag{3}$$

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

$$\tilde{A} \cap \tilde{B} = \{(x, \mu_{\tilde{A}\cap\tilde{B}}(x) = \wedge(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))) | x \in X\}, \tag{4}$$

where $\wedge$ is the minimum operator.

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

$$\tilde{A} \cup \tilde{B} = \{(x, \mu_{\tilde{A}\cup\tilde{B}}(x) = \vee(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))) | x \in X\}, \tag{5}$$

where $\vee$ is the maximum operator.

When dealing exclusively with fuzzy sets, the symbol $\sim$ could be omitted.

General operators for the intersection and union of fuzzy sets are referred as triangular norms (*t-norms*) and triangular conorms (*t-conorms* or *s-norms*), respectively. A function

$$t : [0, 1] \times [0, 1] \to [0, 1], \tag{6}$$

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

- P1. it has 1 as the unit element: $t(a, 1) = a$;
- P2. it is monotone: $t(a, b) \leq t(c, d)$ if $a \leq c$ and $b \leq d$;
- P3. it is commutative: $t(a, b) = t(b, a)$;
- P4. it is associative: $t[t(a, b), c] = t[a, t(b, c)]$.

is called a *t-norm*.

Some relevant examples of *t-norms* are referred in [33]:

1. the minimum: $t(a, b) = a \wedge b = \min(a, b)$. Which was proposed by [32].

2. the algebraic product: $t(a, b) = a \cdot b$
3. the Lukasiewicz *t-norm*: $t(a, b) = \max(0, a + b - 1)$

A function

$$s : [0, 1] \times [0, 1] \rightarrow [0, 1], \tag{7}$$

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

P1. it has 0 as the unit element: $s(a, 0) = a$;
P2. it is monotone: $s(a, b) \leq s(c, d)$ if $a \leq c$ and $b \leq d$;
P3. it is commutative: $s(a, b) = s(b, a)$;
P4. it is associative: $s[s(a, b), c] = s[a, s(b, c)]$.

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

Some relevant examples of *t-conorms* are also referred in [33]:

1. the maximum: $t(a, b) = a \vee b = \max(a, b)$. Which was proposed by [32].
2. the probabilistic product: $s(a, b) = a + b - ab$
3. the Lukasiewicz *s-norm*: $s(a, b) = \min(a + b, 1)$

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

$$s(a, b) = 1 - t(1 - a, 1 - b). \tag{8}$$

4. Proposed Methodology

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

1. The object has constancy of grey levels intensity;
2. The object presents smooth motion;
3. For sake of simplicity, the motion between two consecutive frames can be described using a linear motion model;
4. The area occupied by the object is small when compared with the total image area;
5. The size of the object is preserved during the sequence.

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

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t), \tag{9}$$

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

Nevertheless, small changes in illumination, camera sensor noise, among other factors that cause variations in the intensity of the object, are tolerated.

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

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

$$\mathbf{p}(t) = \mathbf{p}(t-1) + \mathbf{v}\Delta t + \frac{1}{2}\mathbf{a}\Delta t^2, \tag{10}$$

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

The size of the object is considerably small when compared with the total image area. Assuming this, the object can be represented as a point or by a small $A \times B$ matrix and, similar strategies to the ones used in point correspondence can be developed for object matching. In the examples presented in this work, for simplicity sake, we take $A = B = 3$.

4.1. Single-object Tracking

In this methodology, the visual descriptors used to track objects are divided in two distinct groups: kinematic and non kinematic descriptors. Non kinematic descriptors are used to describe the object's appearance and kinematic properties used to describe the object's motion. At the beginning of the process, both these sources of information are treated separately. At the final stage of the process these sources are combined by a fuzzy inference engine that will ultimately provide the estimated position of the tracked object (Fig. 1).

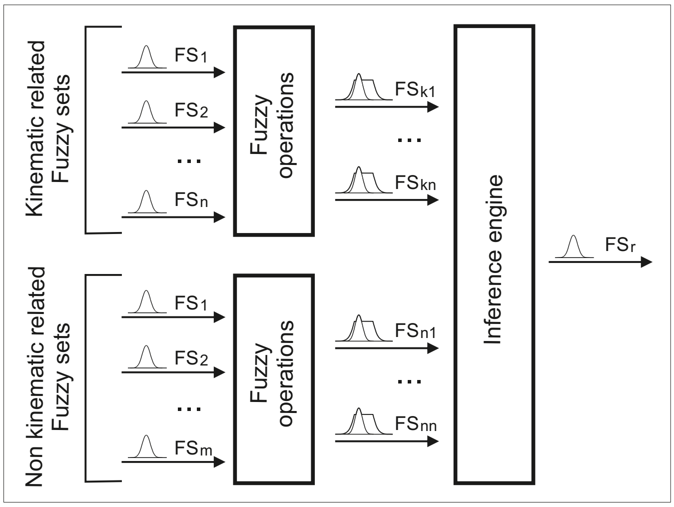


Figure 1: Single-object tracking methodology scheme.

The methodology scheme presented in Fig. 1 can be divided into three main stages:

- Fuzzification of all used descriptors. This way the methodology is able to better deal with the uncertainty and imprecision present in the images (and consequently in the descriptors).
- A fuzzy operations stage where descriptors can be combined according to their precision (higher memberships are more likely to prevail over weaker memberships).

- An inference engine that's responsible for fusing the information provided by all used descriptors (both kinematic and non kinematic). Through this inference engine, the methodology is able to incorporate reasoning, in the same sense as human reasoning, in the tracking process by providing answers based on the existing knowledge base (fuzzified descriptors).

The non kinematic properties used can be chosen according to the specificity of the problem in hand. There's a myriad of possibilities such as color, shape, texture, size, among others, that can be used as visual descriptors of the object appearance. A fuzzy set is constructed to model each one of the chosen properties. These fuzzy sets will represent the membership of all the candidate image positions to the object been tracked. The set of fuzzy operations performed on these fuzzy sets (in their simplest form, fuzzy unions and intersections) are used to combine them and, as a consequence, reduce their number. This dimensionality reduction is an additional advantage since the lower the number of fuzzy sets on the output of this fuzzy operations block (Fig. 1), the less complex the inference engine block will be (i.e. less rules will be necessary).

The kinematic properties will undergo a similar process in the methodology (Fig. 1). As for the non kinematic properties also the kinematic properties can be chosen according to the problem at hand. Kinematic properties such as object velocity, acceleration and other motion patterns can be used as kinematic properties.

The final processing block of the methodology consists in an inference engine which complexity (number of rules) depends on the number of inputs. The design of the inference engine should able the process to model the system in order to achieve a good balance between the information provided by the objects kinematic and non kinematic properties. This way, depending on the problem at hand, through the design of this inference engine, the method is able to incorporate some reasoning valuing either the kinematic or the non kinematic object descriptors depending on their importance. Also, the certainty one has regarding the information provided by each one of these sources (kinematic and non kinematic) can be incorporated in the design of this engine (human reasoning).

The inference engine will present as its output must always be a single fuzzy set. The position of the tracked object is then obtained by maximizing the membership function of this fuzzy set.

4.2. Multi-object Tracking

Based on the previous depicted methodology for single object tracking, a multiple object tracking was developed. For each tracked object, the methodology presented in Fig. 1 is applied and its results (a fuzzy set for each object) are the input of a hierarchical matching system responsible for establishing the correspondences between objects from frame to frame. This hierarchical matching system (Fig. 2) is mainly constituted by a confidence assessment scheme that assigns objects with confidence degrees according the correspondence situation from which the object's position is estimated. To correctly establish these correspondences, there are situations where the input fuzzy set is not sufficient and it is crucial to know the rule used to create this fuzzy set (dashed arrows in Fig. 2). These correspondence situations are depicted in the remainder of this section.

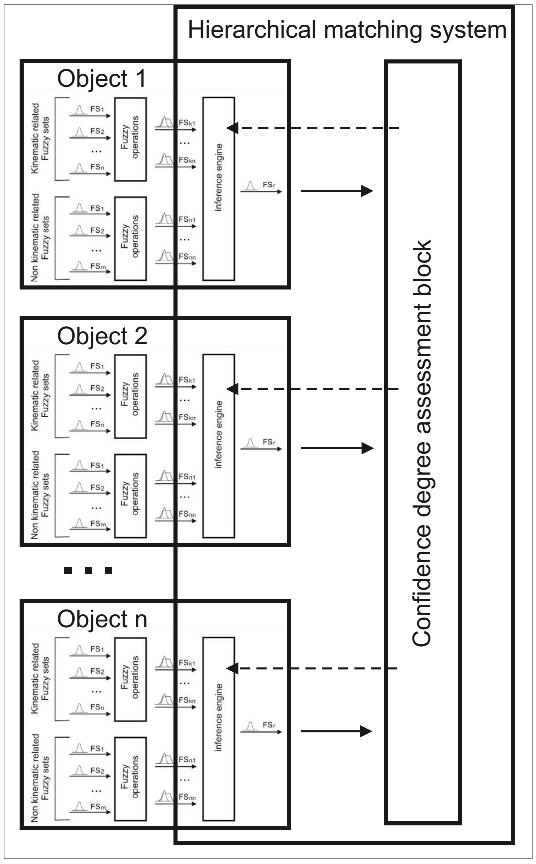


Figure 2: Multi-object tracking methodology scheme.

In multiple object tracking several correspondence situations can occur. Fig. 3 depicts these situations, where $\circ$ denotes the object position at frame $t-1$ and $\times$ denotes the object position at frame t. The question mark (?) represents the absence of matching at frame t. The first situation depicted in Fig. 3a indicates that each object is matched with a different candidate position in the next frame and the current position, at frame t, of the object will be the position of the corresponding candidate position. Sometimes different objects in frame $t-1$ will be assigned to the same point in frame t. When two moving objects pass close each other or even when one object occludes another, or also due to the representation of a 3D world in a 2D plane, they can appear as being just one region in the image. This situation could be seen as a merging of objects or a inter object occlusion case (Fig. 3b). The opposite situation is also considered, i. e., several united objects could have different motion directions and one single region, representing multiple objects, could result in multiple matching positions. It could be seen as a split of objects (Fig. 3c). Finally, if at some instant, the situation depicted at Fig. 3d occurs, i.e., there is no candi-

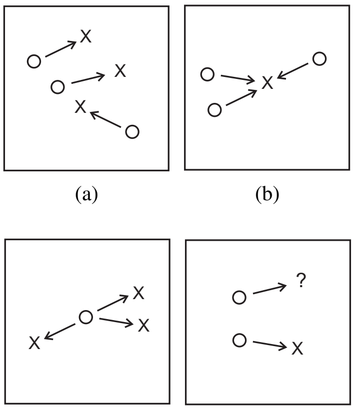


Figure 3: Correspondence situations in multi-object tracking

date position for the object, a predicted position based on the object motion pattern is assumed (Section 5.2). It is a typical case of background occlusion. All these situations must be considered in the tracking algorithm and correct procedures must be applied in each case.

This way, it's possible to discern between inter object occlusion and background occlusion and different actions can be taken for each case.

In this multiple object tracking methodology, the user selects the objects to be tracked and each one of them is associated with a confidence degree that, at the beginning (first frame), assumes its maximum value. This means that the position of each object is known with the highest certainty or confidence. When the algorithm doesn't have reliable information about the object, its confidence degree will decrease. It is considered that situations of inter object or background occlusion and misdetections are examples of such situations. Inter object occlusion occurs when the distance between the objects location is less than the size of one object (a small $A \times B$ matrix with $A = B$).

To deal with all the situations previously described, the following set of rules are defined:

1. When only one object is matched with a single candidate position, its confidence degree increases (Fig. 3a);
2. When several objects are matched with the same candidate position, the object assigned first with that candidate will have its confidence degree increased and all the other will have their confidence degree slowly decreased (Fig. 3b);
3. When there is no candidate position to be matched with an object, the confidence degree of this object decreases (Fig. 3d);
4. Objects with low confidence degree are removed from the list of tracked objects.

All the procedures indicated previously are first applied to objects with higher confidence level and this way a hierarchical matching system is performed. The situation depicted in Fig. 3c falls in the first case since one single object is matched with a candidate and, in this situation, the confidence degree for all the objects increases. To perform the fourth case, a minimum confidence degree is defined and objects denoting a confidence degree below this minimum value will not be tracked.

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

5. Implementation example

5.1. Fuzzy Sets and Fuzzy operations

In this implementation, the constructed fuzzy sets are derived from the initial considerations presented beforehand. Three fuzzy sets related with the first three enumerated assumptions are constructed. Two kinematic related fuzzy sets (Section 5.1.2 and Section 5.1.3) and one non kinematic related fuzzy set (Section 5.1.1). As previously commented, other fuzzy sets could be incorporated in the algorithm but, for the sake of simplicity, to minimize the computational resources and to increase speed, the algorithm is constructed based only on these three fundamental and generic assumptions.

5.1.1. Brightness Constancy

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

Under these conditions, a fuzzy set G is constructed in order to access the certainty of a pixel belonging to the object in such a way that higher similarity in grey levels intensity higher the membership degree to that fuzzy set.

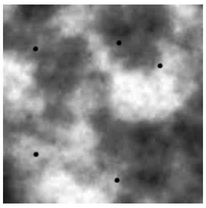
Let I be an image with dimensions $M \times N$, $I(x, y)$ the grey level of the pixel (x, y) so that $0 \leq I(x, y) \leq L$ and $I_f(i, j)$ an intensity matrix of dimensions $A \times B$ representing the original object's grey levels, where $A = 2a+1$, $B = 2b+1$ and $\{a, b\} \in \mathbb{N}$.

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

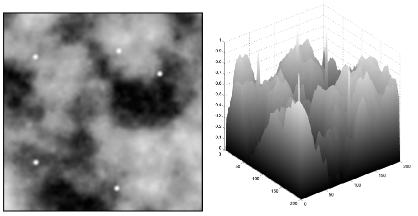
$$\mu_G(x, y) = 1 - d_g, \tag{11}$$

with

$$d_g = \frac{\sum_{i=-a}^{a} \sum_{j=-b}^{b} |I(x + i, y + j) - I_f(i + a, j + b)|}{A \times B \times L}. \tag{12}$$

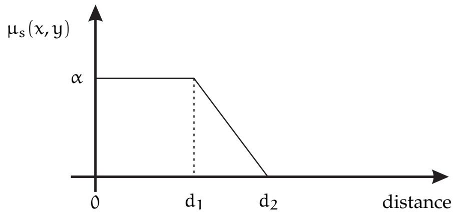


(a) Original image.

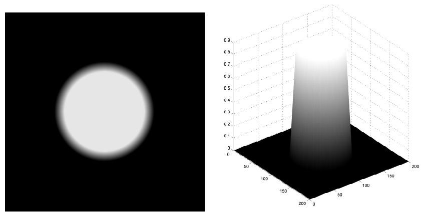


(b) Fuzzy set G. (c) 3D view.

Figure 4: Construction of fuzzy set G.



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



(b) Fuzzy set S. (c) 3D view.

Figure 5: Membership function $\mu_S(x, y)$.

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

Assuming a dark object, the resulting fuzzy set G using a test image is illustrated in Fig. 4.

In order to reduce processing time and increase computational speed, this membership function is applied locally in a neighborhood centered in the previous position of the object. The interest area (ROI: region of interest) can be seen as a square region whose sides have a length l defined as

$$l = 2K_G.d_2, \tag{13}$$

where K_G is a positive constant and d_2 is the parameter defined in Equation 17.

5.1.2. Smooth Motion

Assuming the smoothness of the object movement, i. e., assuming that the object position do not drastically change between two consecutive frames, it is plausible to consider that the next location of the object lies in a neighborhood centered in its previous location.

Therefore, a fuzzy set S associated with each image pixel (x, y) by means of this proximity assumption related to the object position in the previous frame is constructed. The membership function $\mu_S(x, y) \in [0, \alpha]$ can be graphically depicted as illustrated in Fig. 5a, where the horizontal axis represents the Euclidian distance between the image pixels position and the previous location of the object. In Fig. 5b, a pictorial description of the fuzzy set S assuming a previous position $(x, y) = (100, 100)$, with $d_1 = 40$, $d_2 = 50$ and a maximum value $\alpha = 0.9$, is depicted.

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

$$f_d(t) = \sqrt{(\mathbf{p}(t) - \mathbf{p}(t-1))^2}, \tag{14}$$

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

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

$$\Delta d = A_f f_d(t) + (1 - A_f) f_d(t-1), \tag{15}$$

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

$$d_1 = M_1 + \frac{\Delta d}{2}, \tag{16}$$

$$d_2 = d_1 + M_2 + \frac{\Delta d}{2}, \tag{17}$$

where M_1 and M_2 are two positive constants in order to deal with objects denoting zero velocity. The values of M_1 and M_2 are directly proportional to the dimensions of the object. Parameter d_1 has a minimum value of M_1 and parameter d_2 has a minimum value of $M_1 + M_2$.

5.1.3. Linear Motion Model

Another membership function is constructed based on the assumption that the object motion between two consecutive frames can be described using a linear motion model with constant acceleration. An object can increase/decrease its velocity between two consecutive frames. Several motion models are discussed in the literature, however, the selected motion model is a compromise between the proximity with the real motion performed by the object and computer processing requirements. The used motion model follows the restrictions imposed by Equation 10.

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

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

$$\begin{aligned} \mathbf{x}(t) &= A\mathbf{x}(t-1) + B\mathbf{u}(t), \\ \mathbf{y}(t) &= C\mathbf{x}(t), \end{aligned} \tag{18}$$

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

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

$$\mathbf{x}(t) = \begin{bmatrix} x & v_x & a_x & y & v_y & a_y \end{bmatrix}', \tag{19}$$

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

Matrix A is defined as

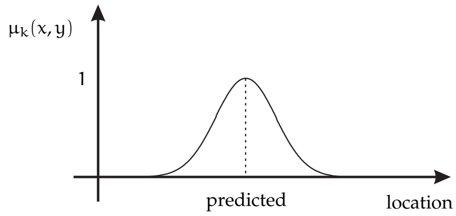
$$A = \begin{bmatrix} 1 & t_{aq} & \frac{t_{aq}^2}{2} & 0 & 0 & 0 \\ 0 & 1 & t_{aq} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & t_{aq} & \frac{t_{aq}^2}{2} \\ 0 & 0 & 0 & 0 & 1 & t_{aq} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

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

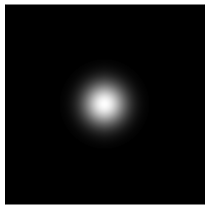
Matrix C is defined as follows

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{21}$$

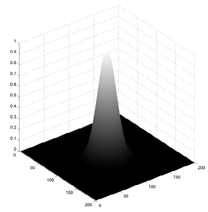
The Kalman filter is used to estimate the state vector of the object, i. e., its position, velocity and acceleration. Using information provided by the state vector of the object it is possible to predict the object position (x, y) in the next frame. The estimated position leads to the development of another membership function, $\mu_K(x, y)$. This membership function assigns a higher membership degree to pixels near the predicted location and its value decreases for locations far from this predicted point. To implement such behavior, a gaussian shape function is used (Fig. 6a) to ensure a higher decay in the membership degrees and, this way, to give more prominence for locations near the predicted one. The gaussian function shape can be changed through the standard deviation parameter σ defined as follows



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



(b) Fuzzy set K.



(c) 3D view.

Figure 6: Membership function $\mu_K(x, y)$.

$$\sigma = \frac{3M_\sigma + \Delta d}{3}, \tag{22}$$

where M_σ is a minimum value to deal with still objects and Δd is the previous observed displacement of the object. This parameter is changed according to the velocity of the object where higher velocities give rise to higher standard deviation values.

This membership function $\mu_K(x, y)$ is applied in a circular neighborhood with radius equal to 3σ. For locations whose distance to the predicted position is greater than 3σ the membership degree is zero. In Fig. 6b, the resulting fuzzy set K assuming a predicted position $(x, y) = (100, 100)$ and $\sigma = 15$ is depicted. The corresponding 3D view is also illustrated in Fig. 6c.

5.1.4. Fuzzy operations

Finally, the motion is modeled by a fuzzy set M constructed from the fuzzy union between fuzzy sets K and S, using the maximum operator as follows:

$$\mu_M(x, y) = \vee(\mu_K(x, y), \mu_S(x, y)) \tag{23}$$

5.2. Inference engine

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

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

$$\mu_E(x,y) = \vee(\mu_G(x,y), \mu_S(x,y)). \tag{24}$$

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

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

$$\mu_{G'_i}(x,y) = \psi_i \mu_G(x,y), \forall i = 1,\ldots,n, \tag{25}$$

with

$$\psi_i = 1 - \frac{d_i}{d_{MAX}}, \forall i = 1,\ldots,n, \tag{26}$$

and

$$d_i = \sqrt{(x_i - x_K)^2 + (y_i - y_K)^2}, \forall i = 1,\ldots,n, \tag{27}$$

$$d_{MAX} = max\{d_i\}, \forall i = 1,\ldots,n, \tag{28}$$

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

Finally, fuzzy set E is constructed using the maximum operator

$$\mu_E(x,y) = \vee(\mu_{G'_1}(x,y),\ldots,\mu_{G'_n}(x,y), \mu_S(x,y)). \tag{29}$$

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

$$\mu_E(x,y) = \mu_K(x,y). \tag{30}$$

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

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

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

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

5.3. Hierarchical matching system

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

6. Results

In order to assess its performance, the proposed method was applied in two sequences. These sequences were used because they denote difficult tracking situations such as occlusions, rapid object motion and appearance and/or illumination variations. The first sequence is a test case scenario used in CAVIAR[1] project. The second was obtained from PETS[2]. The first tested sequence comprises 150 frames starting at frame number 420. At this starting frame, three people are present

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
[2] http://www.cvg.rdg.ac.uk/PETS2010/a.html

in the scene: one person will walk straight on the corridor, one will go inside a store and the other will wait outside. The selected objects to track are the heads of the people. The result of the proposed methodology is illustrated in Fig. 7. In this sequence there are no occlusions or misdetections and therefore the confidence level of each object remains at its maximum.

In the second sequence, the white pixels of a mark of concrete placed in the floor and the white paper sheets carried by a person are selected to be tracked. In this sequence both objects suffer occlusions and their confidence level change accordingly. The result of the proposed methodology between frames 50 and 150 is illustrated in Fig. 8. Despite of the static behavior of the concrete mark it suffers occlusions by the pedestrians in such a way that its confidence level decreases below the predefined minimum value of 6. When it happens this object is no longer tracked. To ensure that objects that don't provide information to the method are still tracked for a long period of time, the difference between the maximum and minimum values of the confidence level must be higher.

Experimental positions of each object in the sequences computed by the proposed methodology are compared with a ground-truth position generated manually. The evolution of the Euclidian distance between them, for all frames of the sequence, can be observed in Figs. 9 and 10, respectively, for CAVIAR and PETS sequences.

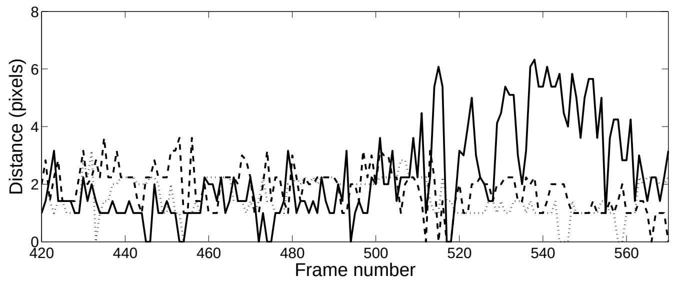


Figure 9: Distance between experimental and ground-truth positions. (Object number 1, 2 and 3 represented, respectively, by solid, dotted and dashed lines)

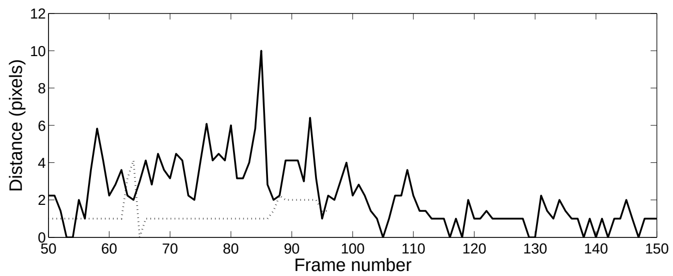


Figure 10: Distance between experimental and ground-truth positions. (Object number 1 and 2 represented, respectively, by solid and dotted lines)

The average and standard deviation of the Euclidian distance between experimental and ground-truth positions for all objects presented in these two sequences are presented in Table 1.

In the sequence obtained from CAVIAR, the distance error for object number 1, depicted in Fig. 9, assumes higher values

Table 1: Average distance error and standard deviation.

Sequence	Object	Mean	Standard Deviation
CAVIAR	1	2.3208	1.6405
	2	1.5577	0.6460
	3	1.8297	0.7723
PETS	1	2.2577	1.7355
	2	1.2716	0.6595

at the end of the sequence. This fact is explained since the long hair of the woman becomes more exposed in the image plane and it also suffers changes in intensity. These two factors lead the method to provide positions not equal to the ground-truth. The error values observed in the other two objects are considerable lower and don't denote a clear variation.

In object number 1, related with the sequence obtained from PETS, higher error values at the beginning of the sequence could be explained by the large size, intensity changes and the swinging pattern of the object observed in the image plane. Another issue is related with occlusion situations that increase the errors between the ground-truth positions since the method doesn't have any new information about the location of the object. An interesting analysis concerning object number 2 can be made: since it is a landmark and the camera is static, its position should remain constant along time, but, observing Fig. 10, the error between experimental and ground-truth positions changes over the sequence. Moreover, after frame number 96 the approach stops to track this object. Error variations are explained by partial occlusions made by pedestrians passing across the concrete mark that introduce an apparent displacement. After frame 96 the confidence degree of this object becomes lower than the established minimum value and it is removed from the tracking process because this object suffers several consecutive total occlusions leading a fast decreasing on its confidence degree.

Despite these errors, object tracking is performed successfully for all objects in both sequences. Having in mind that tracking is performed successfully, the error values presented in Figs. 9 and 10, are considered not significant since they only occur because they are calculated as the distance between the tracked objects computed position and the center of the objects (depending on the homogeneity of the objects grey levels, the computed position may not coincide with the objects center).

A comparative study was also performed by applying the same sequences to other commonly used object tracking approaches such as SIFT with descriptors updating, mean shift and KLT.

After the selection of the features is done, as illustrated in Fig. 11, application of the SIFT approach to both sequences allow to verified that the features could not be tracked properly. The selected regions to be tracked don't provide any or enough descriptors necessary for this method. After one or two frames, no features are tracked due to the lack of descriptor matching because some features became occluded, changed its shape or

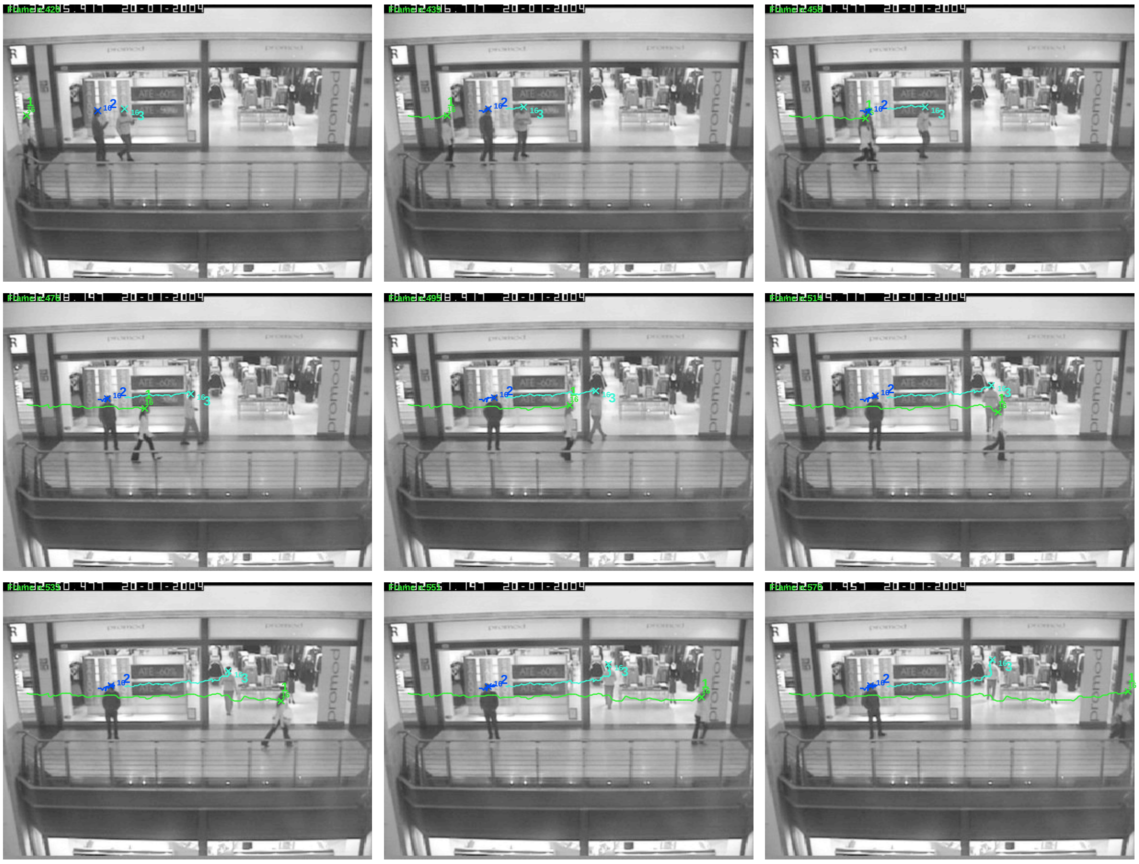


Figure 7: Estimated trajectories between frames 420 and 570.

no similar descriptors were found due to illumination variation. The update rate is very important to deal with object shape variations, but the higher the update frequency the higher the computational time. However, even with higher update rates, this approach did not perform correctly.

Using the mean shift method the observed results were different for each sequence. This approach is not suitable to track any selected features in the CAVIAR sequence. The reduced size of the selected regions results in a non discriminative feature histogram area leads the method to converge to background regions with similar histograms. An example is depicted in Fig. 12: at frame number 420 a region is selected to be tracked but after seven frames the method fails.

A different region initialization was performed by increasing the selected area in order to generate a more representative histogram. Therefore, the torso of the third person was selected and the person is correctly tracked during 31 frames, as observed in Fig. 13. After frame number 451 the tracking fails since the background denotes higher histogram similarity than the feature histogram due to illumination and object pose changes.

Using the PETS sequence with the same method we obtained better results. The selected feature was successfully tracked until the occlusion since this approach don't take into account the kinematic information and therefore is unable to deal with feature occlusions or multi feature crossings. These results are depicted in Fig. 14.

Figure 13: Results at frame number 451.

KLT method also performs dissimilarly for these two test sequences. As presented in Fig. 15, in the CAVIAR sequence the selected feature is tracked until the tracked region became similar with the background. At this moment the approach, destituted from any kinematic information, assumes the background region as being the feature.

A better response is given with the PETS sequence. The method performs a correct feature tracking until the feature disappears. The problem is the same as previously.

In order to be able to present a comparison with a quantitative performance measure, the average and standard deviation of the Euclidian distance between ground-truth and experimental po-

Figure 8: Estimated trajectories between frames 50 and 150.

sitions given by the methods is presented in Fig. 2. To ensure reliable conclusions, the comparative study does not incorporate the SIFT method because experimental results showed that it was incapable to track any features and the remaining methods were applied from frame number 50 to frame 88 of the PETS sequence because they were functional in this frame interval. These methods were implemented in MATLAB® and tested in a laptop computer equipped with a 1.8GHz Intel ® CPU and 1.5GB of RAM under Windows ® XP. The elapsed computational time to process the tracking task within this interval is also indicated.

Table 2: Tracking performance and computational costs.

Method	Elapsed time	Mean error	Standard deviation
Mean shift	11.9660	10.3596	12.7284
KLT	14.7197	14.6424	7.9479
Proposed	42.5574	3.3723	1.8244

It is clearly verified a tradeoff between tracking accuracy and computational time. The proposed methodology incorporates hierarchical matching schemes to deal with multi feature tracking and Kalman filters to incorporate the kinematic feature model that increase the processing time. Moreover, the code is not optimized for real time purposes. However, with an optimized algorithm, real time processing requirements are expected to be achieved.

7. Conclusion

In this work, a new fuzzy tracking system is introduced. Both single and multiple tracking methodologies are presented. Possible implementations of such methodologies are presented. Trough the construction of three fuzzy sets related either with kinematic and non kinematic properties of the object and with a construction of an inference engine with three rules the proposed fuzzy single object tracking methodology is implemented. For multiple object tracking, is presented a hierarchical implementation where objects that provided more information to the method are assigned first than objects with low confidence degree. Experimental and comparative tests to evaluate the performance of the approach are also presented. Although the presented hierarchical matching approach for multiple object tracking has provided encouraging results, these results lead us to further work with intend to improve robustness, introduce new capabilities and achieve computational efficiency

(a) (b)

Figure 11: Selected features for SIFT tracking. Descriptors are represented by a × mark.

(a) Selected feature in the initial frame. (b) Estimated trajectory between frames 420 and 427.

Figure 12: Selected feature for mean shift tracking.

over different image sequences. Further work is intended on the introduction and performance evaluation of different distinctive object properties such as shape, texture and other object descriptors, in order to construct suitable fuzzy sets and introduce new rules in the inference engine. Since the proposed methodology is intended to be a general framework, SIFT, KLT or mean shift based methodologies could be incorporated and used as inputs. This way, it's expectable an increase on the robustness of the estimated trajectories. The introduction of an automatic capability to deal with entries and exits of objects over the sequence is also an important issue to be studied and tested.

Acknowledgments

This work is supported by European Union Funds (FEDER/COMPETE - Operational Competitiveness Programme) and by national funds (FCT - Portuguese Foundation for Science and Technology) under the project FCOMP-01-0124-FEDER-022692.

References

[1] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34 (Aug. 2004) 334–352.

[2] A. Jaimes, N. Sebe, Multimodal human-computer interaction: A survey, Computer Vision and Image Understanding 108 (2007) 116 –134. Special Issue on Vision for Human-Computer Interaction.

[3] M. Liu, C. Wu, Y. Zhang, A review of Traffic Visual Tracking technology, in: Proc. International Conference on Audio, Language and Image Processing ICALIP 2008, pp. 1016–1020.

[4] R. Gopalan, T. Hong, M. Shneier, R. Chellappa, A Learning Approach Towards Detection and Tracking of Lane Markings., IEEE Transactions on Intelligent Transportation Systems 13 (2012) 1088–1098.

[5] C. Conati, C. Merten, Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation, Knowledge-Based Systems 20 (2007) 557–574. Special Issue On Intelligent User Interfaces.

[6] L. Bai, Y. Wang, R. d. Michael Fairhurst, Computer vision, C. f. tracking, An extended hyperbola model for road tracking for video-based personal navigation, Knowledge-Based Systems 21 (2008) 265–272. The 27th SGAI International Conference on Artificial Intelligence.

[7] H. Moon, R. Chellappa, A. Rosenfeld, Optimal edge-based shape detection., IEEE Transactions on Image Processing 11 (2002) 1209–1227.

[8] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (2006) 90–126.

[9] R. J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: a systematic survey, IEEE Transactions on Image Processing 14 (2005) 294–307.

[10] J. L. Barron, D. J. Fleet, S. S. Beauchemin, Performance of optical flow techniques, International Journal of Computer Vision 12 (1994) 43–77.

[11] N. Otsu, A threshold selection method from gray level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1979) 62–66.

[12] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (1986) 679–698.

[13] J. P. Lewis, Fast normalized cross-correlation, in: Vision Interface, Canadian Image Processing and Pattern Recognition Society, 1995, pp. 120–123.

[14] K. Shafique, M. Shah, A non-iterative greedy algorithm for multi-frame point correspondence, in: Proc. Ninth IEEE International Conference on Computer Vision, 2003, volume 1, pp. 110–115.

[15] I. J. Cox, A Review of Statistical Data Association Techniques for Motion

(a) Selected feature in the initial frame. (b) Estimated trajectory between frames 50 and 88.

Figure 14: Selected feature for mean shift tracking.

(a) Selected feature in the initial frame. (b) Estimated trajectory between frames 420 and 430.

Figure 15: Selected feature for KLT tracking.

Correspondence, International Journal of Computer Vision 10 (1993) 53–66.

[16] E. Trucco, K. Plakas, Video Tracking: A Concise Survey, Oceanic Engineering, IEEE Journal of 31 (April 2006) 520–529.

[17] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Comput. Surv. 38 (2006) 13.

[18] J. J. Wang, S. Singh, Video analysis of human dynamics - a survey, Real-Time Imaging 9 (2003) 321–346.

[19] C. Harris, M. Stephens, A combined corner and edge detector, in: Proceedings of the Fourth Alvey Vision Conference, pp. 147–151.

[20] J. Shi, C. Tomasi, Good Features to Track, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), Seattle, pp. 593–600.

[21] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, Int. J. Comput. Vision 60 (2004) 91–110.

[22] D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 603–619.

[23] L. K. Huang, M. J. J. Wang, Image thresholding by minimizing the measures of fuzziness, Pattern Recognition 28 (1995) 41–51.

[24] P. Couto, M. Pagola, H. Bustince, E. Barrenechea, P. Melo-Pinto, Uncertainty in multilevel image thresholding using Atanassov's intuitionistic fuzzy sets, in: Proc. (IEEE World Congress on Computational Intelligence). IEEE International Conference on Fuzzy Systems FUZZ-IEEE 2008, pp. 330–335.

[25] N. V. Lopes, H. Bustince, V. Filipe, P. Melo-Pinto, Fuzziness Measure Approach to Automatic Histogram Threshold, in: J. Tavares, N. Jorge (Eds.), Computational Vision and Medical Image Processing: VipIMAGE 2007, Taylor and Francis Group, 2007, pp. 295–299.

[26] N. V. Lopes, P. A. Mogadouro do Couto, H. Bustince, P. Melo-Pinto, Automatic Histogram Threshold Using Fuzzy Measures, IEEE Transactions on Image Processing 19 (2010) 199–204.

[27] H. Lazoff, Target tracking using fuzzy logic association, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 1998, volume 4, pp. 2457–2460.

[28] J. García, J. M. Molina, J. A. Besada, J. I. Portillo, A multitarget tracking video system based on fuzzy and neuro-fuzzy techniques, EURASIP Journal on Applied Signal Processing 2005 (2005) 2341–2358.

[29] N. V. Lopes, P. Couto, H. Bustince, P. Melo-Pinto, Fuzzy Dynamic Matching Approach for Multi-Feature Tracking, in: EUROFUSE 2009, Pamplona, Spain, pp. 245–250.

[30] P. Couto, N. Lopes, H. Bustince, P. Melo-Pinto, Fuzzy dynamic model for feature tracking, in: Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, Barcelona, Spain, pp. 1–8.

[31] N. V. Lopes, P. Couto, P. Melo-Pinto, Multi-feature Tracking Approach Using Dynamic Fuzzy Sets, in: P. Melo-Pinto, P. Couto, C. Serôdio, J. Fodor, B. De Baets (Eds.), Eurofuse 2011: Workshop on Fuzzy Methods for Knowledge-Based Systems, volume 107 of *Advances in Intelligent and Soft Computing*, Springer-Verlag Berlin and Heidelberg GmbH, 2011, pp. 389–400.

[32] L. A. Zadeh, Fuzzy Sets, Information Control 8 (1965) 338–353.

[33] J. Kacprzyk, Multistage Fuzzy Control: A Prescriptive Approach, John Wiley & Sons, Inc., New York, NY, USA, 1997.

(a) Selected feature in the initial frame.

(b) Estimated trajectory between frames 50 and 94.

Figure 16: Selected feature for KLT tracking.