International Journal of Approximate Reasoning 53 (2012) 693–708

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

A new fuzzy based algorithm for solving stereo vagueness in detecting and tracking people

Rui Paúl [a,*], Eugenio Aguirre [a], Miguel García-Silvente [a], Rafael Muñoz-Salinas [b]

[a] Department of Computer Science and A.I., E.T.S. Ingeniería Informática University of Granada, 18071 Granada, Spain
[b] Department of Computing and Numerical Analysis, E.P.S. University of Cordoba, Cordoba, Spain

ARTICLE INFO

Article history:
Received 5 January 2011
Revised 11 November 2011
Accepted 16 November 2011
Available online 2 December 2011

Keywords:
People detection
People tracking
Fuzzy logic
Particle filtering
Stereo vision
Color information

ABSTRACT

This paper describes a system capable of detecting and tracking various people using a new approach based on color, stereo vision and fuzzy logic. Initially, in the people detection phase, two fuzzy systems are used to filter out false positives of a face detector. Then, in the tracking phase, a new fuzzy logic based particle filter (FLPF) is proposed to fuse stereo and color information assigning different confidence levels to each of these information sources. Information regarding depth and occlusion is used to create these confidence levels. This way, the system is able to keep track of people, in the reference camera image, even when either stereo information or color information is confusing or not reliable. To carry out the tracking, the new FLPF is used, so that several particles are generated while several fuzzy systems compute the possibility that some of the generated particles correspond to the new position of people. Our technique outperforms two well known tracking approaches, one based on the method from Nummiaro et al. [1] and other based on the Kalman/meanshift tracker method in Comaniciu and Ramesh [2]. All these approaches were tested using several color-with-distance sequences simulating real life scenarios. The results show that our system is able to keep track of people in most of the situations where other trackers fail, as well as to determine the size of their projections in the camera image. In addition, the method is fast enough for real time applications.

1. Introduction

People detection and tracking can be done in various ways and with different kinds of hardware. When computer vision is used, the system analyzes the image and searches for cues that provide important information in the detection of people. Those cues could be, for instance, morphological characteristics of the human body [3] or dynamic skin color models [4].

Nowadays, several methods employed for tracking people are based on the color information available from people cloths. Commonly,the first step is to create a color model of the person to be tracked. Then, throughout a sequence of images, the position and size of the image region whose color model best matches the person color model, is considered the new position and size of that person. This technique is called adaptive tracking and it is especially appropriate for tracking non-rigid targets, of which there is no explicit model, or when the background estimation is not possible.

As most of these techniques rely uniquely on color information, they present several drawbacks. The most important is the confusion between two or more areas that have the same color distribution when they are close to each other. Because there is no other information to distinguish them, this issue can cause the system to confuse the objects being tracked. This confusion can also happen with the background, if the tracker does not have information about which parts in the image are part of the background. In case that background, or one part of it, presents a color distribution similar to the person

* Corresponding author.
E-mail address: ruipaul@decsai.ugr.es (R. Paúl)

0888-613X/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2011.11.003

being tracked, the target can be lost. Finally, the situation where the tracker assigns a subregion of the tracked person as the whole region of the tracked person, may also happen. That becomes a problem when determining the appropriate size of that person in the camera image, as only part of the person's body is identified as the whole person.

Some authors have proposed the use of stereo technology which nowadays has been thoroughly studied and has become more common in computer applications [5]. With the development of well consolidated technologies and commercial hardware that deal with stereo computation issues, this technique has turned out to be an important tool when developing computer vision applications such as tracking algorithms. These algorithms can take advantage of pixel distance information for solving problems that non-stereo tracking algorithms present. Firstly, the possibility of knowing the distance from the camera to the person can be of great help when tracking is taking place. Secondly, distance information is less sensitive to illumination changes than information provided by a single camera.

This work proposes a novel approach implementing a particle filter wherein each of its particles is evaluated using a fuzzy logic approach which deals with these sources of information. color, stereo and occlusion information are fused so a final weighting value is computed. Our system also estimates the possibility that a detected face actually corresponds to a face, as well as other features which can tell us about the morphology of a person. By doing so we are able to estimate the area that each person is likely to occupy on the camera and distance images. Finally, we also incorporate confidence measures in our fuzzy system, in order to adjust the importance of each source of information according to its current state. At the same time, confidence measures model the uncertainty whenever it is not possible to obtain the most accurate kind of information.

The use of fuzzy logic to compute the final weight of each particle brings us different benefits compared to the probabilistic approach. First, by using probability models to evaluate particles, it is assumed that variables follow a probability distribution. For example, uncertainty could be modeled in a probabilistic approach, by modifying the probabilistic distribution function by means of some parameter. Those assumptions sometimes are not exactly true or are hard to be modeled. Nevertheless, with fuzzy logic we can achieve the same goal in a more flexible way, without being restricted to particular aspects of the probability distributions. Secondly, fuzzy logic easily allows us to incrementally add other sources of information, in case our system needs so. By using linguistic variables and rules to express relationships the system becomes more understandable and similar to the way humans represent and deal with knowledge.

The remainder of this paper is structured as follows. Section 2 presents some related works while Section 3 introduces both the color modeling and stereo processing algorithms used in this work. Section 4 explains our people detection and tracking approach which fuses different types of information. Section 5 shows the experimentation carried out, and finally Section 6 exposes some conclusions.

2. Related work

The Kalman/mean-shift (described in [2]) is employed in different tracking approaches. In their work, Comaniciu and Ramesh combine the well known mean-shift algorithm with color information to locally move the search region towards the gradient direction of the Bhattacharyya coefficient described in Aherne et al. [6]. The Kalman filter is employed to predict the position of the target in the next frame. Another color-based particle filtering technique that uses this kind of information is the one described by Nummiaro et al. [1], where each particle represents a possible position and size of the tracked object. Sun and Bentabet [7] also present a method where they combine the use of Monte Carlo sequential filtering for tracking and Dezert Smarandache theory (DSmT) to integrate the information provided by different color and position cues.

In several works, stereo vision has been used so that distance information could be extracted from images. In Darrell et al. [8], authors also present a system capable of detecting and tracking several people. Their work is based on a skin detector, a face detector and the disparity map provided by a stereo camera. In the work of Grest and Koch [9] a particle filter [10] is used to estimate the position of the person and to create color histograms of the face and the chest regions of one person and the stereo vision is used to compute its real position. However, stereo and color were not integrated in the tracking process and they use cameras positioned in different parts of a room rather than only one stereo camera. Moreno et al. [11] present a system able to detect and track a single head using the Kalman filter. They combine color and stereo information but head color does not provide enough information to distinguish among different users. In Harville [12] and Muñoz-Salinas et al. [13], authors present an approach to detect and track several people using *plan-view maps*. They use information provided by an *occupancy map* and a *height map* using the Kalman filter.

When merging different unreliable or imprecise sources of information one can choose between using probabilistic/mathematical based models [14–16] or soft computing techniques. An example of a soft computing technique based on fuzzy logic which decomposes the input-output characteristics into noise-free part and probabilistic noise part and identifies them simultaneously can be found in Hong et al. [17]. Other soft computing techniques applied to computer vision have already been used in different works namely the ones from Kil-jae and Bien [18] and Bloch [19]. More information and work on this subject can be found in Solana-Cipres et al. [20], Schultz et al. [21] and Nachtegael et al. [22]. We opted to use fuzzy logic [23] in order to have the possibility of dealing with uncertainty and vagueness in a flexible manner as well as to avoid restrictions when representing imprecision and uncertainty with probabilistic models. Both our people detection and tracking algorithm are based on a fuzzy logic approach. Regarding object detection, different works, as the one from Iqbal et al. [24], are supported by fuzzy logic approaches.

Particle filters are widely used on object tracking algorithms. They can estimate the state of a dynamic system $x(t)$ from sequential observations $z(t)$ as refereed in different works as the ones from Gordon and Salmand [25], Isard and Blake [10] and Kitagawa [26]. They are able to manage multiple hypotheses simultaneously, by dealing naturally with systems where both the posterior density and the observation density are non-Gaussian. However, they may present some problems when used for multi-target tracking. Firstly, the standard version of the particle filter, does not define a method to identify individual targets. Furthermore, particles generated by this kind of filter quickly converge to a single target, discarding the rest of them (also known as the coalescence or particle "hijacking" problem). Another problem is that it can suffer from exponential complexity as the number of targets increases. Vermaak et al. [27] as well as Khan et al. [28] and Okuma et al. [29] propose different approaches to deal with these problems. A MPF (Multi-Particle Filter) consists of employing an independent Particle Filter for each target and an interaction factor which modifies the weight of particles in order to avoid the coalescence problem.

Another way of dealing with uncertainty and vagueness issues susceptible of being found in particle filters are the so called Fuzzy Logic based Particle Filters (FLPF), which is a concept that has been applied by some authors. In Vermaak et al.[30], the ideal number of generated particles is computed using a fuzzy system. In Young-Joong et al. [31], a fuzzy adaptive particle filter for the localization of a mobile robot is proposed, whose basic idea is to generate samples at high-likelihood using a fuzzy logic approach. Shandiz et al. [32] present a particle filtering approach in which particles are weighted using a fuzzy based color model for object which discriminates between background and foreground elements. In this approach, only this information is fuzzified and used to evaluate the particle. Kamel and Badawy [33] present a fuzzy logic-based particle filter algorithm for tracking a maneuvering target. In their work, the nonlinear system which is comprised of two-input and single-output are represented by fuzzy relational equations. In Zheng and Bhandakar [34] a particle filter approach, where face detection information is also used to enhance the performance of the particle filter, is described.

In our approach, the problem of merging different information sources, usually accompanied by vagueness and uncertainty, is solved by using a new approach based on a particle filter which generates particles that are evaluated by means of fuzzy logic. Although we also use stereo and color information as sources of information, they are supplied to several hierarchically sorted out fuzzy systems. This is done by generating different particles in the image and then, using a fuzzy logic approach, computing their likelihood of being the face's central pixel of some previously detected person. Further information on this subject is available in the next Section. In complex applications, containing a large set of variables, it is not appropriate to define the system with a flat set of rules. Among other problems, the number of rules increases exponentially with the number of variables. Thus, fuzzy systems should be organized according to the type of information they cope with and the hierarchical structure assists the reducing of complexity [35]. That is the reason why we opted to use this class of fuzzy system.

3. System description

Our system includes a stereo camera that allows us to extract not only color but also depth information. By combining those two different types of information it is possible to achieve a more robust tracking comparing to cases where only one of them is used. Please note that it is not our purpose to develop or present a new stereo matching algorithm. Instead of it, we use the software that comes with the camera [36], and which deals with lens distortion, to perform the stereo computation. This software supplies an assembly optimized fast-correlation stereo core that performs fast Sum of Absolute Differences (SAD) stereo correlation. This method is known for its speed, simplicity and robustness, and generates dense disparity images. The camera used on this work, is a Bumblebee digital stereo vision camera from Point Grey [37].

Stereo information is used both to detect and to track people through the proposed method in this work. In Fig. 1 we can see an example of a scene captured with our stereo system. While in Fig. 1(a) we can see the left camera image I_l, in Fig. 1(b) we can see the right camera image I_r (defined as the *reference image*). The displacement of the projection in one image in comparison to the other is named disparity and the set of all disparities between two images is the so called *disparity image*. Disparities can only be computed for those points that are registered on both images so it may happen that occlusions or insufficient texture lead to a lack of disparity. Those points whose disparity cannot be calculated are named *unmatched* points.

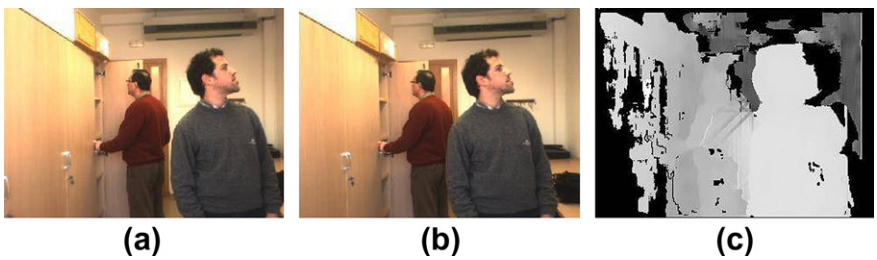

Fig. 1. (a) Image of the left camera I_l captured with the stereo system. (b) Image of the right camera I_r (reference image). (c) Image of distance I_z.

Knowing the intrinsic parameters of the stereo system, such as the focal length (in our case this value is 6 mm), it is possible to reconstruct the three-dimensional structure corresponding to the disparity image and to compute the *distance image* I_z. In Fig. 1(c) it is possible to see the distance image I_z. In this image, brighter pixels indicate lower values of distance Z while darker ones represent farther distances. Black pixels represent unmatched points.

Although we use distance (stereo) information to improve the accuracy of our tracking algorithm, this tracking is done in the reference image, in the 2D domain, where the position of a person is the position of the center of his/her face, which was originally detected by a face detector in the reference image. Thus, the *position of a person* is a (x_p, y_p) pair corresponding to a pixel within the reference image.

Regarding the use of color information for tracking objects, this is a well known problem which has been studied using different approaches: Birchfield [38], Comaniciu and Ramesh [2], Grest and Koch [9] and Nummiaro et al. [1].

The most frequently used method consists of using a histogram to represent a color model $\hat{q}$ where each bin represents a color region. As HSV color space [39] is relatively invariable to illumination changes, it has become a popular approach in this domain. A color histogram $\hat{q}$ comprises $n_h n_s$ bins for the hue and saturation. However, chromatic information cannot be considered reliable when the value component is too small or too big. Therefore, pixels on this situation are not used to describe the chromaticity. Because that these pixels might have important information, the histogram includes also n_v bins to capture its luminance information. Thus, the resulting histogram is composed by $m = n_h n_s + n_v$ bins.

As stated in Birchfield [38], Comaniciu and Ramesh [2] and Nummiaro et al. [1], we consider an elliptical region of the image to create the color model whose horizontal and vertical axis are h_x and h_y respectively. Let p_c be the ellipse center and $\{p_j\}_{j=1,\ldots,n}$ the locations of the interior pixels of the ellipse. Let's also define a function $b : \Re^2 \to 1, \ldots, m$ which associates to the pixel at location p_j the index $b(p_j)$ of the histogram bin corresponding to the color u of that pixel. It is now possible to compute the color density distribution $\hat{q}$ for each elliptical region with:

$$\hat{q}(u) = \frac{1}{n}\sum_{j=1}^{n} k[b(p_j) - u], \tag{1}$$

where the parameter k is the Kronecker delta function. Please notice that the resulting histogram is normalized, i.e., $\sum_{u=1}^{m} \hat{q}(u) = 1$.

After calculating the color model $\hat{q}$, we can compare it with another color model $\hat{q}'$ using the Bhattacharyya coefficient as described in Aherne et al. [6] and Kailath [40]. In the case of a discrete distribution it can be expressed as indicated in Eq. (2). The result expresses the similarity between two color models in the range of [0, 1] where 1 means that they are identical and 0 means that they are completely different. An important feature of ρ is that both color models, $\hat{q}$ and $\hat{q}'$, can be compared even if they have been created using regions of different sizes.

$$\rho(\hat{q}, \hat{q}') = \sum_{u=1}^{m} \sqrt{\hat{q}(u)\hat{q}'(u)}. \tag{2}$$

4. Proposed method

In this section, the process of people detection is first explained just before the description of the people tracker method takes place. Fuzzy logic is used on both phases and a new FLPF is used on the tracking phase.

4.1. People detection

In this subsection we present our approach to detect new people in the environment. Section 4.1.1 presents a face detector tool, Section 4.1.2 the concept of "projection of a person" and Section 4.1.3 the background extraction method and occlusion handling technique used in this work. Finally Sections 4.1.4 and 4.1.5 present an algorithm to detect new people.

4.1.1. Face detection

The people detection process begins with a face detector phase. The system employs the face detector provided by the OpenCV's Library [41]. This face detector is also described in Bradski and Kaehler [42]. It is not in the scope of this paper to develop face detection techniques since there is plenty of literature about it in Yang et al. [43]. The face detector is based on Viola and Jones' method [44] which was later improved by Lienhart and Maydt [45]. The implementation is trained to detect frontal views of human faces and works on gray level images, although it can be trained to detect other perspectives of human faces (for instance, lateral views). This detector is free, fast and able to detect people with different morphological faces. Nevertheless, the problem of face positives should be taken into consideration, no matter the detector chosen for this job.

The classifier outputs the rectangular regions of the frontal faces detected in the camera's reference image. Each detected face's position is firstly compared to the position of each of those people which are already being tracked. If the difference between each of those positions, is higher than *distNewFace*, which was experimentally tuned, the system will initiate a procedure which goal is to reject potential false positives which is described in Sections 4.1.4 and 4.1.5.

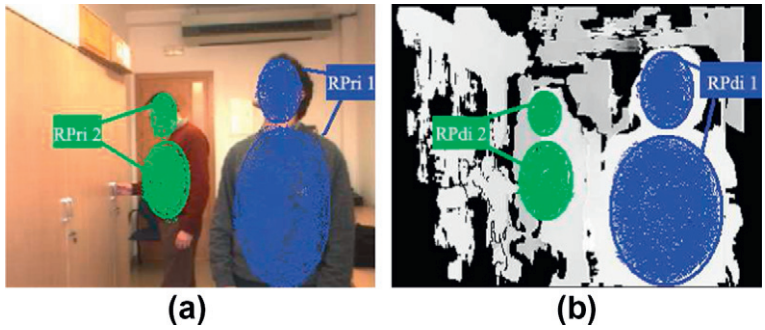

Fig. 2. (a) Projection of 2 people on the reference image. (b) Projection of the 2 people on the distance image.

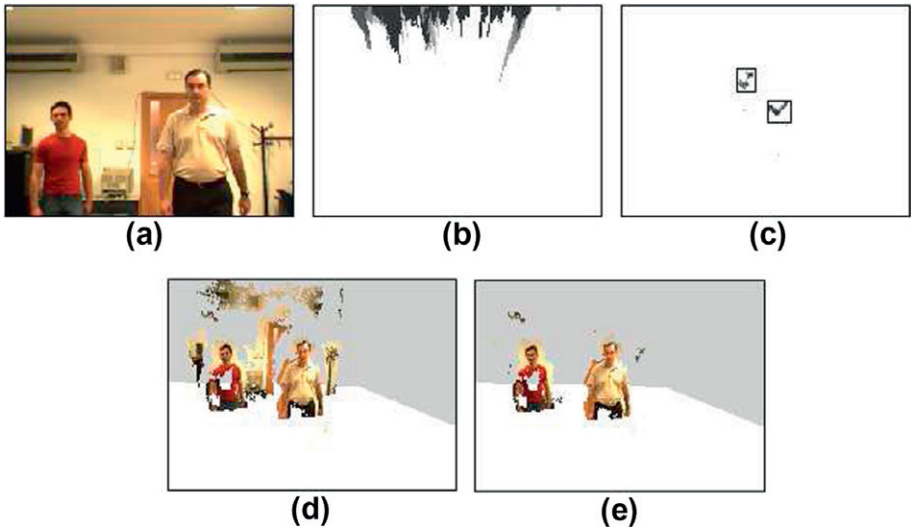

Fig. 3. (a) Reference image. (b) Background: projection of detected static objects in the floor plane after the background extraction initial phase. (c) Projection of detected dynamic objects on the floor plane (corresponding to 2 people) during the experiments. (d) Reconstitution of the scene using distance information (static + dynamic objects). (e) Reconstitution of the scene using distance information and subtracting background.

4.1.2. Projection of a person

We will now present the concept of "projection of a person" which can be interpreted as the 2D region that both face and torso of a standing up and average size person, would occupy either on the reference and on the distance image, if his or her face were approximately on the same 3D location of the detected face.

To explain the concept of "projection of a person" we would like to start by saying that we take into account an assumption regarding various anthropomorphic features of a human being, namely the face and torso approximate size. We consider that a person's face roughly fits inside a 20 x 30 cm ellipse, and his or her torso fits in a 40 x 60 cm ellipse. We also assume that those ellipses' centers are roughly separated by a distance of 45 cm. By assuming these values, we are able to extract from both reference and distance images, the regions occupied by both head and torso of a person. By knowing the intrinsic parameters of the camera we are able to define two elliptical regions (head and torso) in our reference camera image that one denote by RP_{ri} and two elliptical regions in the distance image, denoted by RP_{di} (also head and torso), according to the distance of the detected face's center to the camera (obtained using the stereo algorithm). In our notation, *RP* stands for Region of Projection while *ri* stands for reference image and *di* for distance image. Fig. 2 shows those regions for two different people both in the reference image ($RP_{ri}1$ and $RP_{ri}2$) and in the distance image ($RP_{di}1$ and $RP_{di}2$).

4.1.3. Background extraction and occlusion map

We also use the concepts of background extraction and occlusion map in our algorithm. The first one consists in extracting the previously computed background of the environment in every new frame. To compute the background, an adjustable number of frames are used when the system is initialized. We model the background, by using information about the invariable color and stereo data of the scene, during the initialization of the system, as suggested by Harville in [46]. Fig. 3 exemplifies the background extraction method.

Concerning the concept of occlusion map, and before explaining it more in detail, it is important to have in mind that the system keeps track of the stereo information (distance) of the center of the face of the currently detected/tracked people, in a separate vector. By doing so, it is able to know which people are potentially closer to the camera thus occluding others.

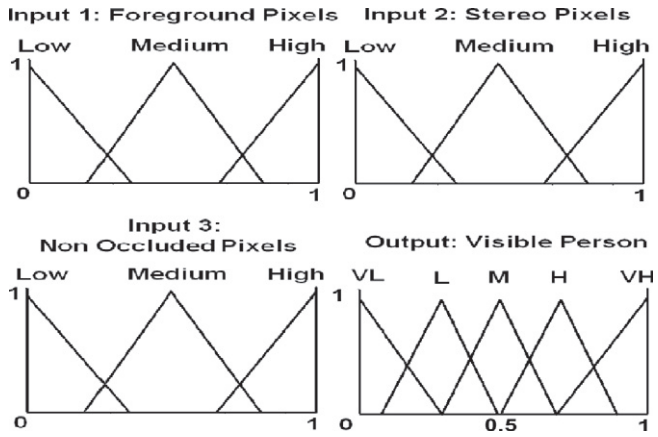

Fig. 4. Fuzzy sets to assess detected faces with variables ForegroundPixels (ratio), StereoPixels (ratio), NonOccludedPixels (ratio) and output variable VisiblePerson.

The occlusion map is a binary image where each pixel of an image is set, every frame, to 0 if it does not belong to a person, and to 1, if it is part of a person. When it is initialized, all pixels of this binary image are set to 0.

During the people detection phase, the system knows which pixels were classified as being part of people in the previous frame, using the occlusion map. By knowing so it is able to determine if a candidate to new person potentially has a part of its body occluded. If so, these projections belong to a region where visual and depth information is not sufficient and consequently not reliable.

The occlusion map is also used in the tracking phase to compute a confidence level to the stereo and color information. The methodology for using occlusion information in the tracking phase is explained in Sections 4.2.1 and 4.2.2.

4.1.4. First test of people detection

Regarding the next two sections, the goal of these tests is to detect false positives, after detecting faces in the reference camera image. We will call $RP_{ri}(DF)$ and $RP_{di}(DF)$ (where DF stands for Detected Face) to the projections of the person whose face belongs to a detected face, on both reference camera image and distance image. It is important to say at this phase that only detected faces, who are located at a distance farther than an experimentally tuned value, with respect to the current position of already tracked people, are going to be evaluated, to avoid the classification of previously detected and tracked people as new people.

The goal of the first test is to check whether inside $RP_{di}(DF)$ there are enough pixels respecting three conditions. First, they belong to the foreground (if they belong to the background they cannot be considered as being part of a person). Second, they have stereo information, ie, they are not unmatched points (if there is a person projected in $RP_{di}(DF)$ then this region should contain a high number of pixels with depth information). Third, they are not occluded. As people moving freely in the environment tend to occlude each other, we take into consideration if most of the pixels inside $RP_{ri}(DF)$ and $RP_{di}(DF)$ are not occluded.

These three measures are fuzzified by three linguistic variables labeled as *ForegroundPixels*, *StereoPixels* and *NonOccluded-Pixels*, respectively (see Fig. 4). Using these three variables as input variables to the Fuzzy System Test 1 (FST1) shown by Table 1, the fuzzy output *VisiblePerson* is computed. FST1 and the rest of the fuzzy systems shown in this work use the Mamdani inference method. The defuzzified value of *VisiblePerson* indicates the possibility, from 0 to 1, whether region $RP_{ri}(DF)$ is likely to contain a visible person whose face is the one detected by the face detector. If this value is higher than α_1, the detected face passes to a second test.

At this stage, it is important to refer that all membership functions and rule bases were created using our expert knowledge and were then experimentally tuned. Rules which were considered irrelevant were eliminated and rules which were similar between themselves were merged. A work presenting automatic learning techniques for the tuning of these fuzzy systems, taking into consideration the error measured on the tracking process, is scheduled to be prepared and presented in the future.

4.1.5. Second test of people detection

The second test also checks whether $RP_{ri}(DF)$ may contain a true positive face. However the idea is different now. If there is a person in that region, then pixels inside $RP_{di}(DF)$ should have approximately the same depth as the center of the face. In case the center of the face is un unmatched point, we will try to find the closest pixel which is able to provide us with stereo information. Therefore the Fuzzy System Test 2 (FST2) receives, as input, the difference between the average depth of $RP_{di}(DF)$ and the depth of the center of the detected face as seen in Eq. (3).

$$d = \left| Z(DF) - \frac{\sum_{j=1}^{n}(z_j)}{n} \right|. \tag{3}$$

Table 1
Rules for Fuzzy System Test 1.

IF			THEN
ForegroundPixels	StereoPixels	NonOccludedPixels	VisiblePerson
High	High	High	Very high
High	High	Medium	High
High	High	Low	Medium
High	Medium	High	High
...	...	...	...
Medium	High	High	Medium
Medium	High	Medium	Medium
Medium	High	Low	Low
...	...	...	...
Low	Medium	Low	Low
Low	Low	High	Very low
Low	Low	Medium	Very low
Low	Low	Low	Very low

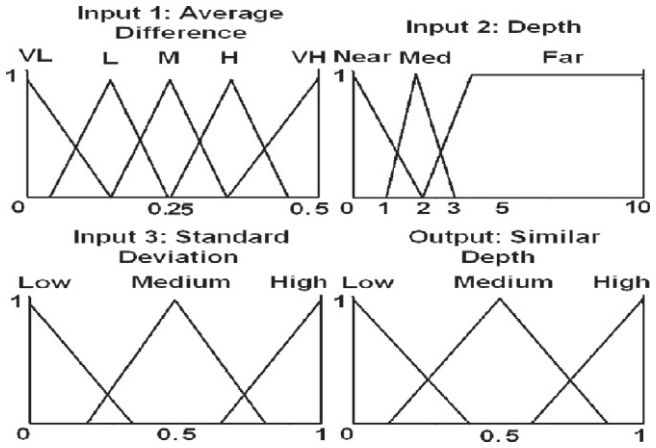

Fig. 5. Fuzzy sets to assess detected faces with variables AverageDifference (m), Depth (m), StandardDeviation (m) and output variable SimilarDepth.

where d is the difference we want to compute, $Z(DF)$ the depth of the center of the detected face (which is considered to be a good approximation of the face distance taking into account the precision of the stereo camera and the expected algorithm's precision), z_j the depth of the j pixel inside $RP_{di}(DF)$ and n the total number of pixels inside $RP_{di}(DF)$. This value is fuzzified by the linguistic variable *AverageDifference*.

FST2 also receives the standard deviation of the depth of those pixels belonging to $RP_{di}(DF)$, fuzzified by the linguistic variable *StandardDeviation*, and the depth at which the face was detected, fuzzified by the linguistic variable *Depth*. The depth of the detected face is used to compute the confidence that we should assign to the values of the other variables. The farther the distance, the higher the uncertainty, according to the table provided by the manufacturer available online (see [37]). The output variable *SimilarDepth* is computed by FST2 and its defuzzified value is a value between 0 and 1 corresponding to the possibility that $RP_{di}(DF)$ contains pixels with a depth value similar to the depth of the detected face. In Fig. 5 linguistic variables *AverageDifference*, *StandardDeviation*, *Depth* and *SimilarDepth* (output) are shown. In Table 2 it is possible to find examples of the rules defined for FST2.

Finally, if this value is higher than α_2, we assume that a new person was detected and we assign a tracker for him or her. The values for parameters α_1 and α_2 have been experimentally tuned. In our experiments, a value of $\alpha_1 = \alpha_2 = 0.6$ proved to be adequate for a good performance by our system.

The rules and linguistic variables defined for other fuzzy systems in Section 4.2.2 are similar to the ones of Figs. 4, 5 and Tables 1, 2 so that they are omitted in order not to be redundant.

4.2. People tracking

In this subsection we present our approach to track people in the environment. Section 4.2.1 presents the Fuzzy Based Particle Filter, Section 4.2.2 introduces the Observation Model and Section 4.2.3 describes in detail the functioning of the fuzzy systems. Finally Section 4.2.4 explains how changes in the color model, such as illumination changes are handled with a model update phase.

4.2.1. Fuzzy based particle filter

In each frame, the tracking of people is done in depth order, which means that we begin with the closest person to the camera until the farthest. There are as many trackers as people being tracked, and the maximum number of tracked people

Table 2
Rules for Fuzzy System Test 2.

IF			THEN
AverageDifference	StandardDeviation	Depth	SimilarDepth
VL	Low	Far	High
VL	Low	Medium	High
VL	Low	Near	High
L	Medium	Far	Medium
...	...	...	...
M	Medium	Far	Low
M	Medium	Medium	Medium
M	Medium	Close	Medium
...	...	...	...
VH	Medium	Close	Low
VH	High	Far	Low
VH	High	Medium	Low
VH	High	Close	Low

1. **Evaluate** whether there is a Detected Face with high possibility of being the face of the person being tracked:
   $CDF(t) = (x_{CDF}, y_{CDF})$ where CDF stands for Closest Detected Face
   IF $(CDF(t) - PersonPos(t-1) < \beta)$ **AND** $(\pi_0(t) > \gamma)$ **THEN**
   $PersonPos(t) = CDF(t)$ (Go to Step 5)
   ELSE
2. **Compute** a sample set $S(t)$ from $PersonPos(t-1)$ as:
   Set $s_i(t) = PersonPos(t-1) + N(0,1)$ with $i = 1..J$
3. **Measure** and weight each sample in terms of the new observation:
   $\pi_i(t)$ = OutFSC*OutFSPPDI*OutFSTI
   Then normalize so that $\sum_{i=1}^{J} \pi_i(t) = 1$
4. **Estimate** the new state $S(t)$ and calculate its weight:
   $\mathcal{E}_t = \mathcal{E}[S(t)] = \sum_{i=1}^{J} \pi_i(t) s_i(t)$.
5. **Update** the occlusion map

Fig. 6. Algorithm employed for tracking each person.

essentially depends on time processing constraints (one of the goals of this work is to comply with real time constraints) and on the amount of people that "fit" into the camera's field of view. Considering the hardware of our system, this proposal allows up to 4 people to be tracked at the same time. Therefore, a multiple particle filter approach is used on our system.

At the beginning of our tracking algorithm, and before the FLPF is integrally executed, a test is executed to assess the possibility that a previously detected face (by the face's detector) corresponds to the face of the current person being tracked (see Fig. 6(1)). To do so, the position, in the reference camera image, of the closest detected face ($CDF(t) = (x_{CDF}, y_{CDF})$) to the previous position of the person being tracked $PersonPos(t-1)$ is selected. To consider that $CDF(t)$ corresponds to the new position of the person $PersonPos(t)$ it has to comply with two conditions. The first one is that its distance to $PersonPos(t-1)$ is less than an experimentally tuned threshold β. The second is that its evaluation value is above a certain γ threshold, which once again was experimentally tuned and set to 0.8. This evaluation method is described in the next subsection. In the case that the particle complies with these two conditions, $CDF(t)$ is considered to be the new position of the person $PersonPos(t)$. The aim of this procedure is to avoid all the particle filtering process, when we have strong suspects that some specific face could be the face of the person that we are tracking. By adopting this procedure, we avoid time consumption and improve the our algorithm tracking accuracy.

When no face is detected in the "neighborhood" of the tracked person's last position, the FLPF takes place (Fig. 6(2)). Particle filters can estimate the state of a dynamic system $PersonPos(t)$ from sequential observation $z(t)$. We define $PersonPos(t)$ as the position x_p, y_p of the center of the person's face. To achieve that estimation $\mathcal{E}_t$, a weighted set of J particles $S(t) = \{(s_i(t), \pi_i(t))\}$, with $i = 1, \ldots, J$, is computed, where $s_i(t) = (x_{si}, y_{si})$ represents a possible state of the system, and $\pi_i(t)$ is a non-negative numerical factor called importance weight which represents an estimation of the observation density $p(z(t)|s_i(t))$. Our approach is based on the typical structure of the Condensation algorithm [10], which is partially adapted

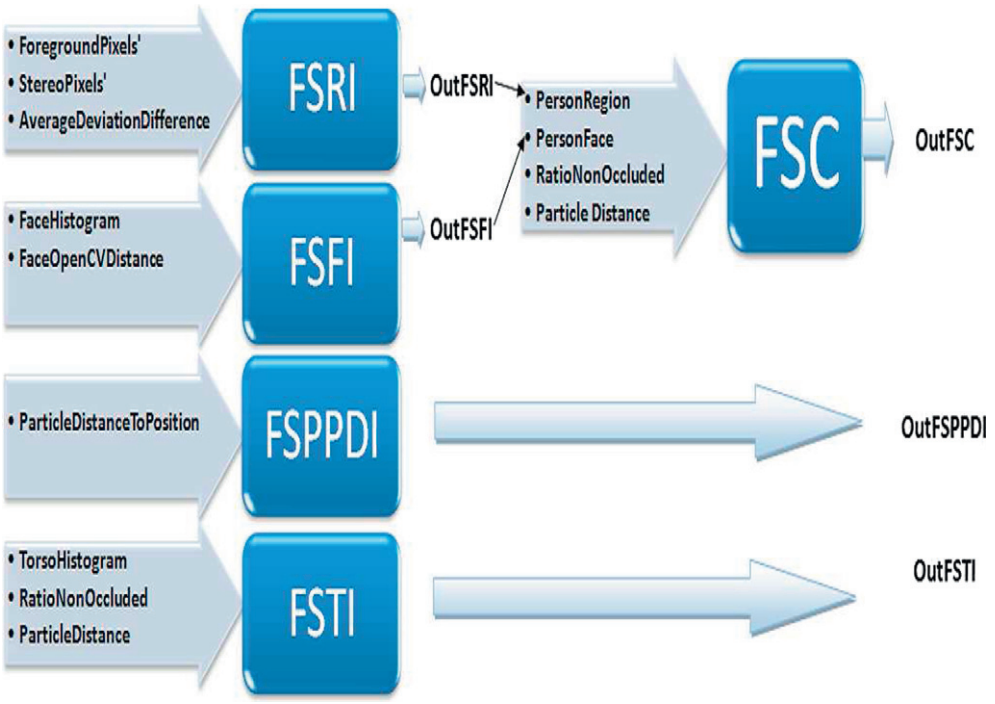

Fig. 7. Fuzzy systems used to evaluate the overall quality of each generated particle. For each fuzzy system, the input linguistic variables are specified.

with new concepts that we will describe. In our system, $\pi_i(t)$ is not computed by means of probabilistic assumptions but using fuzzy logic. This is achieved by combining the output of several hierarchically connected fuzzy systems.

The value of J was experimentally tuned to 50, as lower values might compromise accuracy and higher values might compromise processing time (and real time constraints whenever there are several people being tracked). As we referred before, when no face is detected in the "neighborhood" of the tracked person's last position, the algorithm uses the previous position of the person $PersonPos(t-1)$ to create a set of particles $S(t)$. The propagation model of the particles is based on the previous position of the person plus some δ random gaussian noise with parameters $N(\mu = 0\,px, \sigma = 30\,px)$. The idea is to generate most particles in the surroundings of the previous and a few farther as people are not expected to move fast from frame to frame. The new samples $s_i(t)$ are then weighted.

The weight $\pi_i(t)$ of each particle is computed based on the new observations obtained from our fuzzy systems, as described in Section 4.2.2.

4.2.2. Observation models

After generating the set of particles we begin the process of evaluating the possibility $\pi_i(t)$ that each particle corresponds to the tracked person (Fig. 6(3)). The observation model for each particle is based on the output of different fuzzy systems as shown in Fig. 7. There are 5 fuzzy systems, which we call FSRI (Fuzzy System Region Information), FSFI (Fuzzy System Face Information), FSC (Fuzzy System Confidence), FSPPDI, (Fuzzy System Particle to Person Distance Information) and FSTI (Fuzzy System Torso Information). They are sorted out according the type of information which each of their variables represent. The whole system is structured in a hierarchical way, which is one alternative presented in the literature [35] to overcome the problem of reducing the complexity of rule understanding, when several variables are used on fuzzy systems. Therefore, we use a two layer fuzzy system approach which takes into account the confidence level of the outputs of some of the fuzzy systems. The overall result for each particle is given by $\pi_i(t) = OutFSC * OutFSPPDI * OutFSTI$ where each parcel corresponds to the defuzzified output of a fuzzy system and is a value between 0 and 1 (see Fig. 7).

The new position of the tracked person, $PersonPos(t)$, is equal to the final state estimation $\mathcal{E}_t = \mathcal{E}[S(t)]$ which is obtained from the mean of the state $S(t)$ by weighting all particles $s_i(t)$ (see Fig. 6(4)).

For better understanding of our algorithm, a detailed description about the functioning of the FLPF algorithm is shown in Fig. 6.

4.2.3. Fuzzy systems description

In the next paragraphs we will describe each of the fuzzy systems used to compute the value of $\pi_i(t)$. We would like to highlight the fact that, because of space constraints, we are not able to exemplify each of the labels and rule bases of each fuzzy system. We believe that those examples presented on Section 4.1 provide a good understanding of our system.

Table 3
Rules for Fuzzy System Confidence (FSC).

IF			ParticleDistance	THEN
PersonRegion	PersonFace	RatioNonOccluded		Output FSC
VH	VH	High	Close	VH
VH	VH	High	Medium	VH
VH	VH	High	Far	H
VH	VH	Medium	Close	H
...	...	...	...	...
M	M	High	Close	M
M	M	High	Medium	M
M	M	High	Far	L
...	...	...	...	...
VL	M	Medium	Close	L
VL	M	Medium	Medium	VL
VL	M	Medium	Far	VL
VL	M	Low	Close	VL
VL	L	Low	Medium	VL
VL	VL	Low	Far	VL

In the evaluation process of $S(t)$ we also use the concept of "projection of a person" presented in Section 4.1.2. In this case, we use (x_{si}, y_{si}), the position of the particle currently being evaluated, as the center of the face to compute the projection of the person $RP_{ri}(s_i)$ in the reference camera image and $RP_{di}(s_i)$ in the distance camera image.

The goal of FSRI is to evaluate the region $RP_{di}(s_i)$ (see Fig. 2). This evaluation will take into consideration only aspects related to the possibility that some object, similar to a person, is projected in that region. The first step is to compute the area of $RP_{di}(s_i)$. After obtaining this information we define three linguistic variables: *ForegroundPixels*′, *StereoPixels*′ and *AverageDeviationDifference*. *ForegroundPixels*′ and *StereoPixels*′ are defined in a similar way to *ForegroundPixels*, *StereoPixels* at Section 4.1. *AverageDeviationDifference* provides information about the difference between the depth of s_i and the average depth of all pixels inside $RP_{di}(s_i)$. This value is also fused with the standard deviation of the depth of those pixels. The reason for defining this variable is that, all pixels inside $RP_{di}(s_i)$, should have approximately the same depth as s_i and should have approximately the same depth between them, as long as they belong to some person or object. We would like to highlight the fact that only pixels that are considered as not being occluded by other person are taken into consideration. To know which are these pixels, we use the occlusion map that it is updated at the end, for each tracker. These values will be the input to FSRI that will output a deffuzified value between 0 and 1. The higher amount of foreground, stereo pixels and lower difference in average and standard deviation, the closer the output is to 1. A value closer to 1 means that, in the area represented by $RP_{ri}(s_i)$, it is likely to have some object that could hypothetically be a person.

The scope of FSFI is to evaluate face issues related to the person being tracked. We define two linguistic variables called *FaceHistogram* and *FaceOpenCVDistance*. The first one contains information about the similarity between the face region of $RP_{ri}(s_i)$ and the face histogram of the person being tracked. As people from frame to frame (at a 15 fps frame rate) do not tend to move or rotate their face so abruptly, those histograms should be similar. We use the elliptical region of the face to create a color model [38]. We then measure the difference between the face histogram of region of $RP_{ri}(s_i)$ and the face histogram of the person being tracked. This difference is based on a popular measure between two color distributions: the Bhattacharyya coefficient [6]. Once again, only pixels that are not occluded are used in this process. This method gives us the similarity measure of two color models in the range [0, 1]. Values close to 1 mean that both color models are identical. Values near 0 indicate that the distributions are different. An important feature of this method is that two color models can be compared even if they have been created using a different number of pixels. The second linguistic variable measures the distance between s_i and the position of the nearest face to s_i detected by the OpenCV face detector. Although OpenCV is not 100% accurate, most of time this information can be worth as it can tell if there is really a face near s_i. The deffuzified output of this fuzzy system is also a number between 0 and 1 where 1 is an optimal value.

The deffuzified outputs of FSRI and FSFI are then provided as input of FSC. The aim of this fuzzy system is to measure the confidence of the outputs of FSRI and FSFI based on occlusion and depth information. As including new variables in FSRI and FSFI would make it more difficult to define rules and better understand the whole system, we opted to create a hierarchical fuzzy system structure, that allows us to measure the confidence of the mentioned outputs. Thus, for FSC, we define four linguistic variables called *PersonRegion*, *PersonFace*, *RatioNonOccluded* and *ParticleDistance* to compute its final output as it is possible to see in Fig. 8. *PersonRegion* and *PersonFace* have five linguistic labels Very Low, Low, Medium, High and Very High distributed in a uniform way into the interval [0, 1] and its inputs are the defuzzified outputs of FSRI and FSFI respectively. *RatioNonOccluded* contains information about the ratio of non occluded pixels inside $RP_{ri}(s_i)$. The higher the number of non occluded pixels, the more confidence we have on the output values. In other words, the more pixels we can use from $RP_{ri}(s_i)$ and $RP_{di}(s_i)$ to compute foreground, depth, average information and histogram the more trustable the outputs of FSRI and FSFI. Finally *ParticleDistance* has information about the distance of the particle evaluated (s_i). As errors in stereo information increase with distance, the farther the particle is located, the less trustable it is in means of depth information. The defuzzified output of FSC (*OutFSC*) is also a number between 0 and 1. Higher values indicate a region with higher possibility to contain a person. Rules for this fuzzy system can be seen in Table 3.

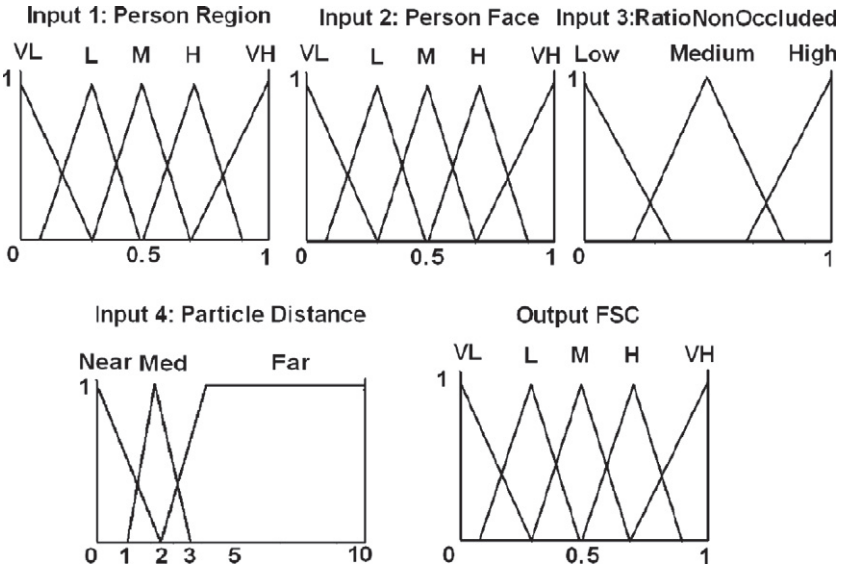

Fig. 8. Variables for fuzzy system FSC, PersonRegion, PersonFace, RatioNonOccluded, ParticleDistance (m) and OutputFSC.

With respect to FSPPDI, its goal is to evaluate whether s_i is likely to be the person being followed taking into consideration the distance to the previous location of the person (in the frame before). Due to the frame rate used, people from frame to frame are not expected to move significantly. Therefore, we define only one variable called *ParticleDistanceToPosition* that contains information about the distance in pixels between the position of s_i and the position of the currently tracked person ($PersonPos(t-1)$). The deffuzified output will be, once again, a value between 0 and 1 represented by *OutFSPPDI*. An output equal to 1 means that s_i is located exactly in the same place where $PersonPos(t-1)$ was located.

The last fuzzy system FSTI is related with torso information. Identically to FSFI we also define a variable that translates the similarity between the torso histogram information of $RP_{ri}(s_i)$ and the histogram information of the torso of the person being tracked. This variable is called *TorsoHistogram*. Similarly to FSFI, only pixels that are considered as non occluded are taken into consideration. We also define for this fuzzy system, the variables *RatioNonOccluded* and *ParticleDistance* analogously to FSC. When doing this, we are adding a measure of confidence for the output which, after its deffuzification, is called *OutFSTI* and has a value between 0 and 1.

As said before, all these outputs are multiplied and the result is a value between 0 and 1. Then a weighted average of the position in the reference image $PersonPos(t)$ is computed by taking into consideration all the possibility values for the set of particles. A particle that has a possibility value closer to 1 weights much more than one with a possibility value of 0. Its region of projection is also added to the occlusion map, so the following trackers and the people detection's algorithm know, that there is already a person occupying that region.

4.2.4. Model update

Changes in the illumination conditions and person's different perspectives might alter the observed color distribution of the tracked region. Therefore, it is necessary to update the head and torso's color models to achieve robust tracking. For that purpose, after the tracking process is concluded, the projection of the person on the reference camera image RP_{ri} is used to update his or her color model. The pixels of RP_{ri} are employed for creating the new observed color model as:

$$\hat{q}_E(t) = (1-\alpha)\hat{q}_E(t-1) + \alpha\hat{q}_E a(t) \tag{4}$$

where the parameter $\hat{q}_E a(t)$ refers to the observed color model for the current estimated projection and $\alpha \in [0, 1]$ determines the contribution of the observed color model to the updated target model. In order to avoid the inclusion of pixels from the background or from occluding objects as part of the updated model, only pixels that are part of the foreground, that are not occluded, and belong to RP_{ri} are employed. Finally, we have opted to set $\alpha = 1 - \rho(\hat{q}_E(t), \hat{q}_E a(t))$. In that way, the model is automatically updated accordingly to its difference to the actual observed color model. The higher the difference between them the higher the value employed for α. This is done both for the head and torso's color models independently.

5. Experimental results

This section shows the experiments carried out in order to validate our proposal. The system was tested in an Intel Core i5 2.67 GHz Processor. The achieved operation frequency of our system depends on the number of people being tracked and the number of particles that were used by the Particle Filter algorithm. As each tracked person implies a new tracker,

processing time increases for each added tracker. We consider up to 4 people for the system to perform in real time with this kind of camera and processor.

For this experimentation, different color-with-depth sequences have been recorded using a Bumblebee 1 stereo camera by Point Grey Research (see [37]). We used a stereo correspondence algorithm provided by the manufacturer of the camera (see [36]). The stereo camera allows us to record sequences of 320 x 240 pixels size at a 15 fps frame rate. Videos were recorded in different rooms with different illuminance conditions so we could have diversity regarding background scenarios. Several people participated in the recording and they were instructed to move freely and to simulate different interaction situations either with other people or with the camera.

We tested our algorithm in real time situations where one, two or more people interacted freely, without having any kind of restrictions. The aim was to check whether our algorithm was able to keep track of different people in several situations that are part of our daily life. After performing several experiments and concluding that 50 particles were an optimal number of particles for achieving accurate results without compromising real time performance, we compare our algorithm with an adaptation of Nummiaro's algorithm [1] which is a particle filter approach that uses the Bhattacharyya coefficient to compare two color regions. We also compare it with the Kalman/meanshift tracker proposed by Comaniciu and Ramesh [2], which is implemented in the OpenCV library. We would like to underline the fact that this version of the Comaniciu's algorithm, is able to track only one person at a time. Nevertheless this feature is enough for testing accuracy and executing times, which are our main concerns.

The comparison of our approach with the Comaniciu's and the Nummiaro's based algorithms is made by measuring the distance error between the indicated position provided by all the three algorithms and the manually defined position, on the reference image, of the person being tracked. We also measured the error concerning the size of the indicated face rectangle. We compared the projected size of the face in the camera image and the manually defined size of the face. To do so, we used both differences between the equivalent sides of both rectangles. For approaches that output an ellipse, we compare the longest axis of the ellipse to the longest size of the manually determined rectangle, and the shortest axis of the ellipse with the shortest size of the manually determined rectangle.

In total, more than 5000 frames have been manually annotated. These frames correspond to 8 videos that last between 40 and 60 seconds each. We present statistical information indicating the error values for different algorithms. We present the RMSE (Root Mean Square Error) between the manually determined positions and the position indicated by the tracker as well as the RMSE between the manually determined rectangle sizes of the faces and the ones indicated by the trackers. Please note that because of the stochastic nature of the Particle Filter algorithms used, results are affected by the initialization of the random number generator. To avoid this problem, each experiment has been repeated 30 times with different initialization seeds. We present the values of the mean values of the RMSE for the set of frames concerning the 30 runs on Table 4.

Concerning the processing time, we achieved an average of 22 ms for the 50 particle version of the algorithm, for one tracking cycle per person. Despite the high amount of data involved, we can thus conclude that our algorithm can be used in real time environments while achieving a more accurate and robust tracking than other traditional algorithms.

By looking at Table 4 it is possible to see that, once again, our algorithm outperforms other algorithms in accuracy, without compromising real time performance. Depending on the color of the target and the background, other algorithms can vary their accuracy while our algorithm generally does not lose track of its targets. We would like to highlight the fact that methods based only on color perform very poorly when the background of the scene presents a color model very similar to the color of the skin or clothes. In those cases, the algorithm simply does not work, sometimes detecting the whole background and/or image as the initial person being tracked. In Fig. 9(b) and (c) and specially in Fig. 10(b) and (c) examples that illustrate these remarks can be found.

We will now take a deeper look at Figs. 9 and 10, where we can observe different aspects regarding the tested algorithms. Fig. 9 represents a scene with two people, slightly moving forward and backward, partially or totally occluding their face. Fig. 10 help us to understand how using only color information on a tracking algorithm approach, can critically downgrade the accuracy of those algorithms. We chose 5 frames from each video to exemplify how different algorithms track both of them.

In Fig. 9(a) we can observe that our proposed algorithm manages to keep track of those two people without ever losing their track. We can see that in frames number 205 and 275, sometimes, the square of the face is not totally centered, but the error is not substantial. This was one of our main goals, ie, to acquire a reasonable approximation of each face's region. In Fig. 10(a) we can also state that our algorithm keeps track of people in different situations, even when they cross their paths. Below, we will analyze it deeper in Fig. 11.

If we trust only in color information, as exemplified in Figs. 9(b) and 10(b) it is very common that the tracking algorithm starts to assume that neighbor regions are part of the head and starts to slide from the target person (Fig. 10(b) frame 185 on both people) to similar color objects. In Fig. 10(b), we can see that the algorithm also loses its target. The new squares

Table 4
Comparison between approaches.

	Our approach	Nummiaro based	Comaniciu based
RMSE position	8.85 px	35.99 px	58.49 px
RMSE rectangle size	4.88 px	61.39 px	220.87 px
Processing time per cycle and person	22.64 ms	12.62 ms	17.65 ms

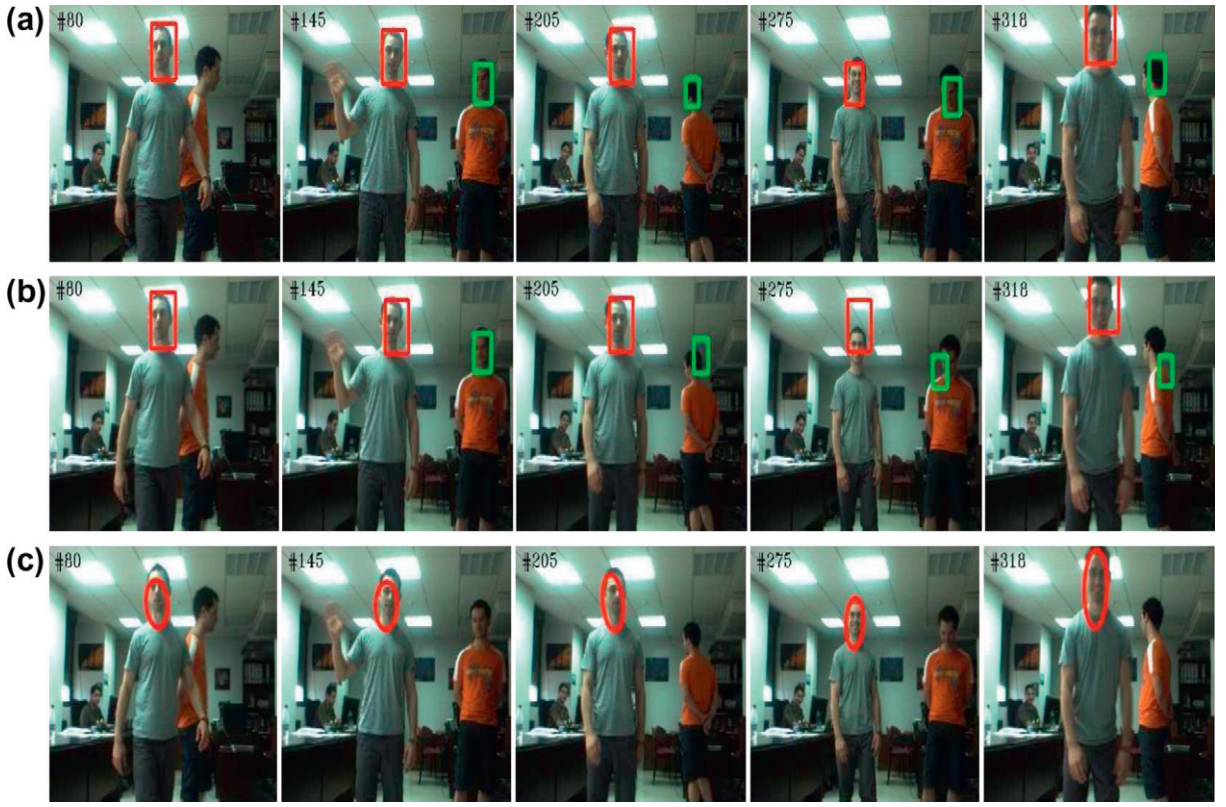

Fig. 9. (a) Proposed algorithm; (b) Nummiaro based algorithm; (c) Comaniciu algorithm.

observed in frames 343 and 399 correspond to new trackers, as the system detects faces that are faraway from the previously tracked people.

In Fig. 9(c), although only one person is detected in the available version of the Comaniciu's algorithm, it is easy to observe that this algorithm works quite well, although there are some issues that we would like to point out. Indeed, we observe that Comaniciu's algorithm works fine in this scene but, as it makes the tracking based on the skin color model of the face, when a person turns back towards the camera, it detects his neck as the whole head area. Furthermore, when a person is facing the camera, it includes the neck as part of the person's face. This aspect could turn out to be a problem, for applications that make use of face features. Concerning Fig. 10(c), as it can be observed, background presents a very similar to face color model which turns out to be a reason for Comaniciu's algorithm rapidly lose its tracked person. In this kind of situation, we can say that the algorithm fails completely.

Despite the good results achieved by our proposal, we would like to mention that, in scenarios where there are two people dressed with the same colors and located near to each other, our system might lose track of them. Nevertheless, this hypothetical scenario affects all the analyzed algorithms in this section. This issue can be solved in a near future by providing more information sources to the system which, in our approach, should be as simple as adding new fuzzy systems or rules to the existing ones. We consider this issue an important advantage with respect to other approaches.

5.1. Behavior results with two people interacting

Finally, in Fig. 11 we have four frames taken from one of those videos, with both reference image and distance image shown for each frame. The aim of this example is to show how our algorithm behaves during a natural interaction between 2 people. In the distance image, lighter areas represent shorter distances to the camera. In Fig. 11(a) it is possible to see that the system detected person A (square 1) but person B was not detected (due to the fact that the employed face detector only detects frontal faces). In Fig. 11(b) we can see that person B was detected (square 2) as his head was now facing the camera. We would like to underline the fact that the stereo camera sometimes produces errors that tend to decrease the accuracy of the stereo part of the algorithm. For example, sometimes the region of the face has the double of its size, when we look at the distance image (in Fig. 11(c), we can see that the size of the head of the person A in the distance image is much bigger that its actual size). In this experiment, people cross their trajectories achieving similar values for their positions. However, the system could still keep an accurate track for each of the people. The reason for achieving this accuracy relies on color information that compensated the similarity of position information. Finally in Fig. 11(d) it is possible to see that, for person

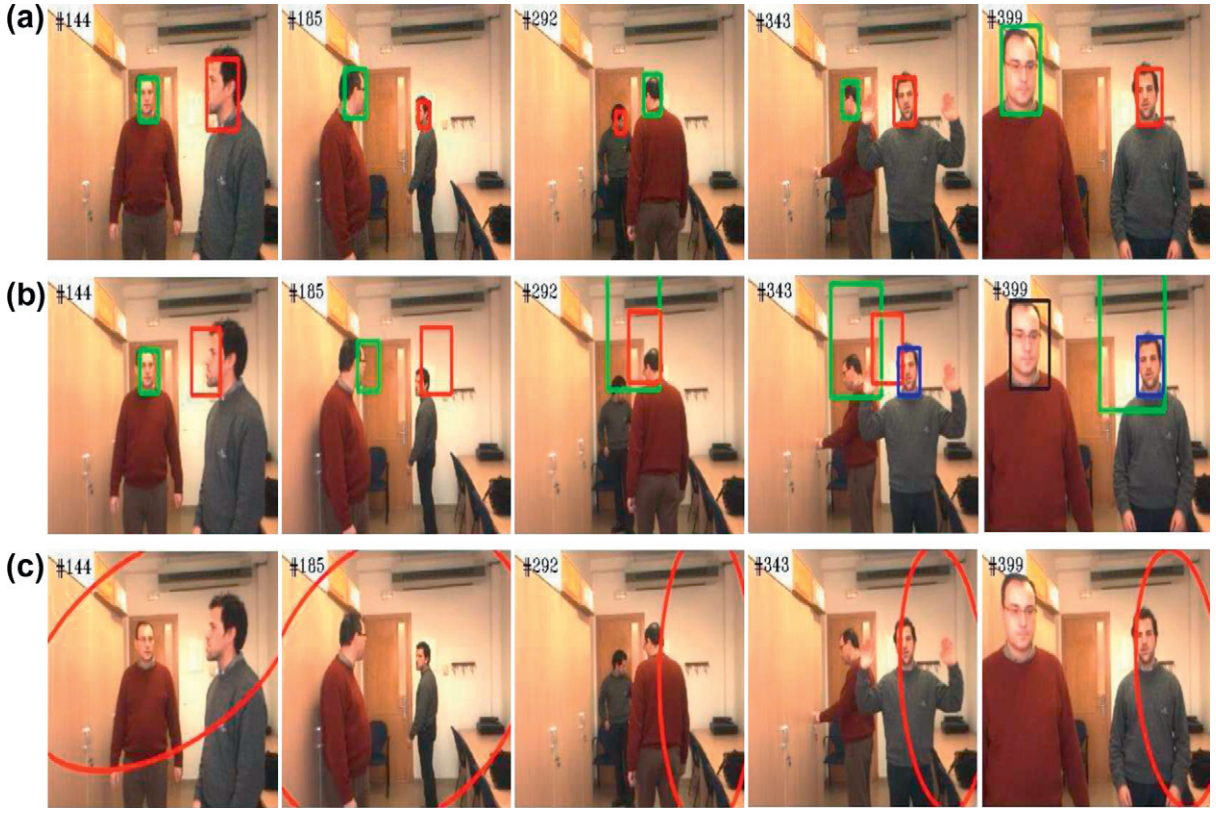

Fig. 10. (a) Proposed algorithm; (b) Nummiaro based algorithm; (c) Comaniciu algorithm.

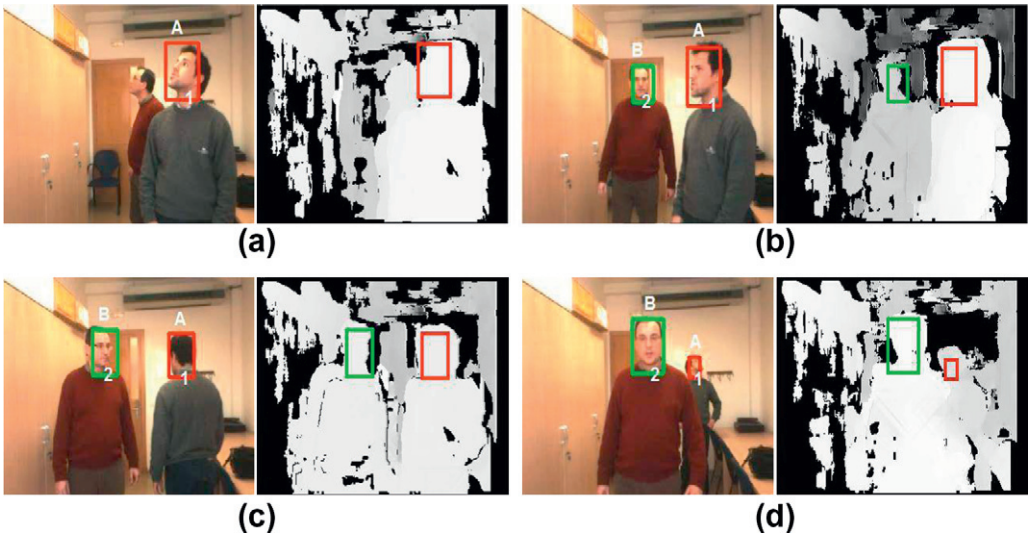

Fig. 11. Different frames taken from a video with 2 people being tracked.

A, although part of his body was occluded, the system could still achieve an accurate tracking, based on stereo information rather than color information.

6. Conclusions and future work

A system able to detect and track various people simultaneously using a new approach based on both color and stereo information handled by means of fuzzy logic has been presented. The results showed that our system managed to keep track

of people, in the reference image, in most of the situations where other trackers fail. It was tested in simulated complex real life situations, where people were interacting freely and occluding each other sometimes. The method proved to be fast enough for detecting and tracking people simultaneously and therefore adequate to be used in real time applications.

The system uses Fuzzy Logic in order to integrate information to detect and track people managing the vagueness of the data provided by the sensors. Fuzzy Logic is an interesting tool that has a proved efficacy for treating uncertain and vague information as well as noisy data from different sources. A modified particle filter is used to generate particles that are evaluated using fuzzy logic instead of probabilistic methods. As we know, information supplied by sensors is commonly affected by errors, and therefore the use of fuzzy systems help us to deal with this problem. By setting up linguistic variables and rules that deal with this problem we achieved an efficient way of solving it.

Both fuzzy systems used for people detection and the hierarchical fuzzy system used for the tracking process, deal with several sources of information as color, position in the reference image, depth, occlusion and other data obtained from the stereo vision. In this sense, information regarding depth and occlusion is used to create confidence levels to fuse, in an appropriate way, both color and stereo information. Furthermore, the advantage of using several sources of information relies on the fact that these sources complement each other. Thus, when information about the position of people is not enough to identify them, color as well as other information sources can be used to identify them. On the other hand, when the color information extracted from a person is similar to the color information extracted from another person or similar to the color of the background, the stereo data is useful to identify each of them. Overall, the people detection and tracking processes achieve very good results thanks to the fusion of these kinds of information.

Also, when fuzzy systems are used to represent knowledge, the complexity in understanding the system is substantially lower as this kind of knowledge representation is similar to the way the human being uses to represent its own knowledge. Furthermore, it allows an easy way of adding new features, just by adding more variables or fuzzy systems. Thus, it will be easy to expand the system in the future, when new sources of information are available.

In this work, rules and linguistic variables are defined after testing different values in different experiments. As a future work, we would like to build a system capable of learning and therefore adjusting these parameters automatically.

Acknowledgments

This work has been partially funded by the FCT Scholarship SFRH/BD/22359/2005, POPH/FSE (Programa Operacional Potencial Humano do Fundo Social Europeu), the Spanish MCI Project TIN2007-66367 and the Andalusian Regional Government project P09-TIC-04813.

References

[1] K. Nummiaro, E. Koller-Meier, L.V. Gool, An adaptive color-based particle filter, Image and Vision Computing 21 (2003) 99–110.
[2] D. Comaniciu, V. Ramesh, Mean shift and optimal prediction for efficient object tracking, in: IEEE International Conference on Image Processing, vol. 3, 2000, pp. 70–73.
[3] N. Hirai, H. Mizoguchi, Visual tracking of human back and shoulder for person following robot, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol. 1, 2003, pp. 527–532.
[4] L. Sigal, S. Sclaroff, V. Athitsos, Skin color-based video segmentation under time-varying illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 862–877.
[5] M. Brown, D. Burschka, G. Hager, Advances in computational stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 993–1008.
[6] F. Aherne, N. Thacker, P. Rockett, The Bhattacharyya metric as an absolute similarity measure for frequency coded data, Kybernetica 32 (1997) 1–7.
[7] Y. Sun, L. Bentabet, A particle filtering and dsmt based approach for conflict resolving in case of target tracking with multiple cues, Journal of Mathematical Imaging and Vision 36 (2010) 159–167.
[8] T. Darrell, G. Gordon, M. Harville, J. Woodfill, Integrated person tracking using stereo, color, and pattern detection, International Journal of Computer Vision 37 (2000) 175–185.
[9] D. Grest, R. Koch, Realtime multi-camera person tracking for immersive environments, in: IEEE Sixth Workshop on Multimedia Signal Processing, 2004, pp. 387–390.
[10] M. Isard, A. Blake, Condensation-conditional density propagation for visual trackings, International Journal of Computer Vision 29 (1998) 5–28.
[11] F. Moreno, A. Tarrida, J. Andrade-Cetto, A. Sanfeliu, 3d real-time head tracking fusing color histograms and stereovision, in: International Conference on Pattern Recognition, 2002, pp. 368–371.
[12] M. Harville, Stereo person tracking with adaptive plan-view templates of height and occupancy statistics, Image and Vision Computing 2 (2004) 127–142.
[13] R. Muñoz-Salinas, E. Aguirre, M. Garcfa-Silvente, People detection and tracking using stereo vision and color, Image and Vision Computing 25 (2007) 995–1007.
[14] R. Muñoz-Salinas, R. Medina-Carnicer, F. Madrid-Cuevas, A. Carmona-Poyato, Multi-camera people tracking using evidential filters, International Journal of Approximate Reasoning 50 (2009) 732–749.
[15] T. Lukasiewicz, U. Straccia, Description logic programs under probabilistic uncertainty and fuzzy vagueness, International Journal of Approximate Reasoning 50 (2009) 837–853.
[16] J. Klein, C. Lecomte, P. MichT, Hierarchical and conditional combination of belief functions induced by visual tracking, International Journal of Approximate Reasoning 51 (2010) 410–428.
[17] S. Hong, H. Lee, E. Kim, A new probabilistic fuzzy model: fuzzification–maximization (fm) approach, International Journal of Approximate Reasoning 50 (2009) 1129–1147.
[18] L. Kil-jae, Z. Bien, A model-based machine vision system using fuzzy logic, International Journal of Approximate Reasoning 16 (1997) 119–135.
[19] I. Bloch, Defining belief functions using mathematical morphology – application to image fusion under imprecision, International Journal of Approximate Reasoning 48 (2008) 437–465.
[20] C. Solana-Cipres, G. Fernandez-Escribano, L. Rodriguez-Benitez, J. Moreno-Garcia, L. Jimenez-Linares, Real-time moving object segmentation in h.264 compressed domain based on approximate reasoning, International Journal of Approximate Reasoning 51 (2009) 99–114.

[21] R. Schultz, T. Centeno, G. Selleron, M. Delgado, A soft computing-based approach to spatio-temporal prediction, International Journal of Approximate Reasoning 50 (2009) 3–20.
[22] M. Nachtegael, E. Kerre, S. Damas, D.V. der Weken, Special issue on recent advances in soft computing in image processing, International Journal of Approximate Reasoning 50 (2009) 1–2.
[23] R. Yager, D. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, Inc., 1994.
[24] R. Iqbal, C. Barbu, F. Petry, Fuzzy component based object detection, International Journal of Approximate Reasoning 45 (2006) 546–563.
[25] N. Gordon, D. Salmand, Bayesian state estimation for tracking and guidance using the bottstrap filter, Journal of Guidance, Control and Dynamics 18 (1995) 1434–1443.
[26] G. Kitagawa, Monte carlo filter and smoother for non-gaussian nonlinear state space models, Journal of Computational and Graphical Statistics 5 (1996) 1–25.
[27] J. Vermaak, S. Godsill, P. Perez, Monte carlo filtering for multi-target tracking and data association, IEEE Transactions on Aerospace and Electronic Systems 41 (2005) 309–332.
[28] Z. Khan, T. Balch, F. Dellaert, Mcmc-based particle filtering for tracking a variable number of interacting targets, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1805–1819.
[29] K. Okuma, A. Taleghani, D. de Freitas, J. Little, D. Lowe, A boosted particle filter: multi target detection and tracking, Lectures Notes in Computer Science 3021 (2004) 28–39.
[30] C. Zhenjiang, L. Zongli, Fuzzy particle filter used for tracking of leukocytes, in: Proceedings of the 2008 International Symposium on Intelligent Information Technology Application Workshops, 2008, pp. 562–565.
[31] K. Young-Joong, W. Jung-Min, L. Myo-Taeg, Fuzzy adaptive particle filter for localization of a mobile robot, in: Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems, 2007, pp. 41–48.
[32] H. Shandiz, S. Mirhassani, B. Yousefi, M. Fatemi, Fuzzy based foreground background discrimination for probabilistic color based object tracking, International Journal of Computer Science and Network Security 10 (2010) 120–125.
[33] H. Kamel, W. Badawy, Fuzzy-logic-based particle filter for tracking a maneuverable target, in: 48th Midwest Symposium on Circuits and Systems, vol. 2, 2005, pp. 1537–1540.
[34] W. Zheng, S. Bhandakar, Face detection and tracking using a boosted adaptive particle filter, Journal of Visual Communication and Image Representation 2 (2009) 9–27.
[35] V. Torra, A review of the construction of hierarchical fuzzy systems, International Journal of Intelligent Systems 17 (2002) 531–543.
[36] Triclops sdk, <http://www.ptgrey.com/products/triclopssdk/index.asp>, 2010.
[37] Point grey research, bumblebee 1 stereo camera, <http://www.ptgrey.com/products/bumblebee/bumblebee.pdf>, 2005.
[38] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 232–237.
[39] J. Foley, A.V. Dam, Fundamentals of Interactive Computer Graphics, Addison Wesley, 1982.
[40] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Transactions on Communication Technologies 15 (1967) 52–60.
[41] Sourceforge, Opencv, intel: open source computer vision library, 2010.
[42] G. Bradski, A. Kaehler, Learning OpenCV, Computer Vision with the OpenCV, O'Reilly, 2008
[43] M.H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 34–58.
[44] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. 511–518.
[45] R. Lienhart, J. Maydt, An extended set of haar-like features for rapid object detection, in: IEEE Conference on Image Processing, 2002, pp. 900–903.
[46] M. Harville, G. Gordon, J. Woodfill, Foreground segmentation using adaptive mixture models in color and depth, in: IEEE Workshop on Detection and Recognition of Events in Video, 2001, pp. 3–11.