THREE-DIMENSIONAL ADAPTIVE SENSING OF PEOPLE IN A MULTI-CAMERA SETUP

M. Andersen[1], *R. S. Andersen*[1], *N. Katsarakis*[1,2], *A. Pnevmatikakis*[2], *and Z.-H. Tan*[1]

(1) Department of Electronic Systems,
Aalborg University
Fredrik Bajers Vej 7 B, 9220 Aalborg, Denmark
{martina, rasmusan, zt}@es.aau.dk

(2) Autonomic and Grid Computing,
Athens Information Technology
P.O. Box 64, Markopoulou Ave., 19002 Peania, Greece
{nkat, apne}@ait.edu.gr

ABSTRACT

Sensing the presence and state of people is of paramount importance in assistive living environments. In this paper we utilise a set of fixed, calibrated cameras to model the bodies of people directly in three dimensions. An adaptive foreground segmentation algorithm is run per camera, providing evidence to be collected in 3D body blobs. A particle filter tracker allows monitoring the modelled bodies across time, offering estimations of their state by using hot-spots and body posture. We apply our system on fall detection and activity monitoring for the elderly, addressing both emergency and cognitive care.

1. INTRODUCTION

Much interest has in resent years been directed at sensing the presence and state of people. The possible applications include surveillance [1], assistive living environments [2, 3], and human-machine interfaces [4, 5]. In this paper we build a system for tracking the position and posture of human bodies in 3D in real-time. For this, a set of 5 fixed and calibrated cameras is utilized. An adaptive foreground segmentation algorithm runs per camera. The detected 2D foreground masks for each camera are combined into one set of 3D foreground voxels using a hierarchical approach based on octrees [6]. Segmentation separates the voxels into a number of bodies, giving indications of the number and position of persons in the scene. A particle filtering tracker allows monitoring the modeled bodies in time, offering estimations of their state.

One increasingly relevant application is emergency and cognitive care for elderly, including fall detection [7, 8] and activity monitoring [9]. We address these by using the state estimations from the tracker to detect abrupt height changes and position persistence. The former are classified as "person sitting down" or "person falling" and the later are compared against predefined hot-spots, to reason on possible activities like "person at dinner table", "at kitchen", or "by the TV". Also multiple human tracks indicate visits, again classified as "for dinner", "for tea", etc.

The novelty of the proposed system lies partly in the efficient combination of 2D foreground masks into 3D foreground bodies and partly in the utilization of a body measurement likelihood function within the particle filtering framework. From the 3D foreground representation, projections onto the floor plan are obtained by summing the body evidence at all heights for the given position. The resulting 2.5D representation is used to evaluate the measurement likelihood function of the proposed particle filter tracker.

This paper is organized as follows: In Section 2 the proposed tracking system is detailed. Test results of the implemented system are presented in Section 3, based on test video from the setup at the AIT. The performance of the system is evaluated and concluded upon in Section 4.

2. TRACKING SYSTEM

In this section our method for foreground detection in 3D, target management and tracking is detailed. Tracking is done using a particle filter based on an effective likelihood function, and the tracking results are interpreted to determine immobile bodies located near hot spots and the posture of each body.

2.1 Body Detection

Body detection is carried out in three stages: First foreground evidence is collected in 2D per camera. This is then combined to 3D foreground mostly following the approach of [10], and it is finally used to model 3D bodies. Foreground detection in 2D utilises per pixel Gaussian Mixture Models inspired by Stauffer et. al. [11]. The performance of the algorithm at the start-up phase is improved by increasing the learning rate according to a window based approach, inspired by [12]. Robustness of foreground blobs is increased by removing shadows as in [13].

2.1.1 Modelling the space in 3D

The purpose of using several cameras for tracking is partly to be able to track in 3 dimensions, but also to filter out noise. Noise in the 2D foreground exists no matter the method used. By combining information from a number of cameras, this noise can be reduced significantly, thus increasing the robustness of the body detection.

We employ a well-known approach where the 3D space is modelled discretely by spanning a grid of voxels [10, 14, 15]. Information from the different cameras can then be combined for each voxel, instead of for each person or region. The novelty of our 3D body detection system is the speed improvement by using a hierarchy of voxels of different sizes and the efficient implementation using distance transform, both described in the following.

2.1.2 Hierarchical Grid Structure

Foreground in the 3D space will mostly be structured in coherent volumes that indicate the presence of persons. Large areas of the space will be completely without foreground. By dividing the space into hierarchies of voxels, these areas can be ruled out efficiently by only testing very large voxels for foreground. Only if a large voxel contains foreground, is it

 ISSN 2076-1465

1. Span the room with a grid of voxels on N hierarchical levels.
2. Project the centre and corners of each voxel on all levels to the image plane of each camera. Use the corners to determine an enclosing circle C.
3. Let the set S consist of all voxels on the highest hierarchical level.
4. **For** each voxel in S:
   (a) **For** each camera:
      - Test foreground mask for foreground evidence within the enclosing circle C.
   (b) **If** enough cameras detect significant foreground:
      - **If** the voxel has any children, **then** repeat 4 with S consisting of all children of the voxel. **else** mark the voxel as a foreground voxel.

Figure 1: Recursive algorithm for converting the 2D foreground masks to a 3D grid of foreground voxels using distance transforms.

necessary to test smaller voxels it contains (its children) to improve the resolution of the model.

An efficient way to construct hierarchies is to use octrees; that is to divide every voxel on a particular hierarchical level into 8 voxels on a lower level [6]. For the test results in this paper we use a 4-level octree with the following voxel widths: 40 cm, 20 cm, 10 cm and 5 cm. Only if a parent voxel contains foreground are its children voxels tested for foreground. A problem for this approach arises in the border areas of the 3D space of interest, where the larger voxels might not fit very well. If a voxel is partly outside the 3D space but with its centre inside the space, it is used directly. If the centre is outside the room, it cannot be tested for foreground, and must therefore be omitted. Instead, the border region is filled directly with smaller voxels (that have centres inside the 3D space). The algorithm for converting the 2D foreground masks into a grid of foreground voxels is summed up as pseudo-code in Figure 1.

For our system, all of the cameras are stationary. This causes the projection of voxels to the image plane of each camera to be identical for all frames. Therefore, the items 1 and 2 in Figure 1 can be carried out off-line, leaving 3D foreground testing as the only potentially computationally heavy part.

The hierarchical algorithm is a speed optimization of the non-hierarchical version, and has been tested to reduce the computational cost of the algorithm by around 80 % when a distance transform is used to combine the 2D foreground masks into 3D foreground as described in the following section.

2.1.3 Efficient Combination of 2D Foreground Masks into 3D Foreground

To test whether a voxel projected to the image plane of a camera contains foreground, all foreground mask pixels located in that projected voxel should ideally be tested. The percentage of pixels with foreground can then either be compared with a threshold for significant foreground, or used as a non-boolean indication for foreground. This is, however, computationally intensive, since pixels in the 2D foreground masks are included in many voxels, and will thus be tested many times.

The speed of the foreground testing can be increased by making certain simplifications. In many cases, the centre pixel of the projected voxel indicates correctly if the voxel contains foreground. To give some resistance to noise, a blurring kernel can be applied before testing. The hierarchical grid structure causes, however, the voxels to be of very different sizes, which again causes the optimal kernel size to be very different. Therefore it is chosen to apply a distance transform instead, where each pixel gets a value corresponding to the distance to the nearest pixel with foreground. After the distance transform has been applied, the centre pixel can be tested and compared to the radius of the enclosing circle, C, of the projected voxel, calculated off-line. This reduces item 4a in Figure 1 to testing one pixel and comparing to the radius of C. To minimize the computation time of the distance transform an approximating 3×3 kernel is applied following the approach in [16]. This causes the calculated distances to be slightly imprecise, but also enables the time consumption to be comparable to a 3×3 blur kernel. In our implementation, the optimal values 0.95509 and 1.36930 are used for the horizontal/vertical and diagonal entries in the kernel, respectively.

It is worth noting that the results of the hierarchical and non-hierarchical algorithms when based on distance transforms are not completely identical. In some cases, perspective and camera distortion can cause the enclosing circle of a child voxel (C_{child}) to contain an area not included in the enclosing circle of its parent voxel (C_{parent}). If foreground is present in this area, but *not* in the rest of C_{parent}, this will cause the hierarchical structure to sort out the child voxel, even though foreground exists within its enclosing circle. Minor tests have indicated that around 0.1% of the foreground voxels are sorted out for this reason. The issue could easily be avoided by using a circle slightly larger than C_{parent} for parent voxels. However, since this only happens when there is foreground inside the enclosing circle of a child voxels but *not* inside the voxel itself, there is no actual reason to prevent it.

2.2 Target Management

Target management includes detection and initialization of new targets and destruction of older targets.

2.2.1 Detection of Targets

Target detection is necessary for initialization of new targets. A simple and fast approach which in many cases will work is to do 3D blob analysis of the detected foreground voxels. For our system, additional measures are taken in an attempt to utilise the typical structure of the 3D foreground. These are illustrated in Figure 2. When two individuals are positioned close together, their detections will easily be connected near the ground, e.g. because of shadows. Near their heads they will, however, often be more easily separable, partly because the heads are located farther from the ground, and partly because the head is thinner than the rest of the body. For this reason the height of the connection of two connected blobs are compared with a threshold, τ_1. The blobs are merged only if they are connected above this τ_1. To make the system

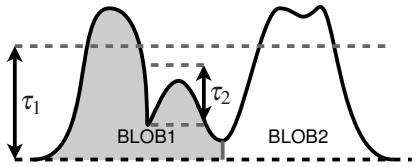


Figure 2: Top/down detection of targets.

robust to people sitting down or falling, an additional threshold τ_2 is used. If the height of one of the blobs relative to the connection point are below τ_2, they are always connected.

2.2.2 Maintenance of Existing Targets

To determine which targets that have significant supporting evidence in the measurements, the position of all targets are associated with the detected blobs using the Munkres or Hungarian algorithm [17]. Non-associated blobs are used to initialize new targets. A variable M for each target is set to 1 if it is associated, and 0 otherwise. The reliability of targets is updated using a simple IIR-filter:

$$r = r + l(M - r) \tag{1}$$

where r is the reliability and l is the learning rate. By comparing r with two thresholds, it can be determined whether the target should be trusted as an individual and (if not) if it should be destroyed. To allow new targets to become reliable relatively fast if they are associated in each frame, while preserving older targets even if they have been unassociated for some frames, the learning rate is adjusted according to the age (given as the number of consecutive frames that the target has existed). The following equation is used:

$$l = \min(l_{\max}, \tfrac{1}{\text{age}} + l_{\min}) \tag{2}$$

It may occasionally happen, that two targets follow the same individual. Therefore targets that are placed very close to one another consecutively for several frames are merged.

2.3 Tracking

The tracking algorithm used in the system is Particle Filtering (PF) [18, 19]. PF's are able to provide a numerical solution to the recursive Bayesian estimation problem when the system dynamics are not linear and/or the noise models are not Gaussian. They hence provide robust solutions to the tracking problem when the object model or the measurement likelihoods are multimodal. This is offered at the expense of additional computational complexity due to their numerical nature. We build a PF that follows the approach for motion tracking described in [19].

Foreground detection is done in 3D and thus tracking should ideally also be done in 3D. To make the tracking algorithm fast enough to allow real-time tracking, we propose what we call a 2.5D approach. All voxels are projected to the floor, and the number of voxels in each column are used to calculate likelihood. The vertical dimension thus provides some additional data for tracking, without itself being part of the target state, hence the term "half dimension". The states related to position can thus be limited to x and y. Note that the vertical position will provide little extra information for tracking since the difference in height between different persons humans are typically small.

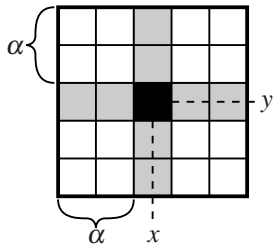


Figure 3: The state space consists of the coordinates x and y on the floor-plan, and the size variable α. The variable α can vary from 0 to the distance between the centre and the boarder of the projection map. Note that the area of a state is given as $(2\alpha+1)^2$.

To reduce the dimensionality of the state-space as much as possible, only a single dimension α is used to determine size. The state space $\mathbb{S}$ is therefore 3 dimensional, and the dimensions are illustrated in Figure 3.

2.3.1 Likelihood Function

The multi hypothetical nature of the particle filter allows tracking of non-global maxima. However, to achieve robust tracking the likelihood function must in as many situations as possible give local maxima close to the correct location and size of the persons in the scene.

To determine a good likelihood function, a number of values can be taken into consideration (where x and y have been left out as function arguments for simplicity):

Volume: A person is expected to constitute a certain volume, which can be given as a number of voxels $N(\alpha)$.

Density: A person is expected to fill most of the volume, V, inside his bounding box. This amount is expressed as:

$$F(\alpha) = \frac{N(\alpha)}{V} = \frac{N(\alpha)}{h \cdot (2\alpha+1)} \tag{3}$$

where α and h are measured in number of voxels. The height h is set to the maximum number of voxels in a single column in that area.

Derivative of density: A good state will be centred close to the centre of a person and include most of that person. This means that $F(\alpha)$ is expected to drop fast if α is increased. This is due to the fact, that most of the area around a person typically is without foreground. The change in $F(\alpha)$ can be measured by its derivative $\frac{\partial F(\alpha)}{\partial \alpha}$, which can be approximated by:

$$\begin{aligned} F_d(\alpha) &= \frac{\Delta F(\alpha)}{\Delta \alpha} \\ &= \frac{F(\alpha+k) - F(\alpha-k)}{2k} \\ &\approx \frac{1}{2kh}\left(\frac{N(\alpha+k)}{(2(\alpha+k)+1)^2} - \frac{N(\alpha-k)}{(2(\alpha-k)+1)^2}\right) \end{aligned} \tag{4}$$

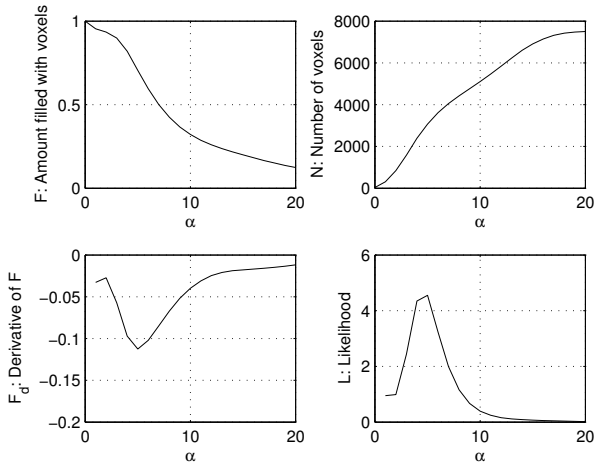


Figure 4: Different values as a function of α.

where h is simplified to be the maximum height of the smaller area (which in most cases is identical to that of the larger area).

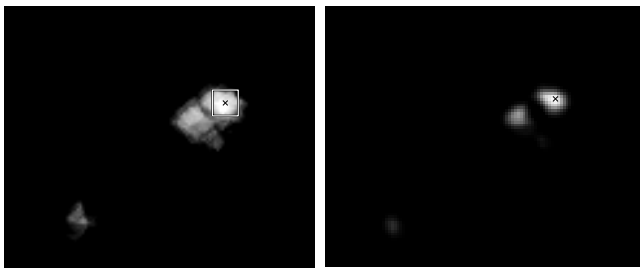
Figure 4 shows $F(\alpha)$, $F_d(\alpha)$ and $N(\alpha)$ along with the final likelihood function when α is varied in the example shown in Figure 5a. In this example, the maximum of $-F_d(\alpha)$ is located at the $\alpha = 5$, which Figure 5a proves is a good result. This is not sufficient for the likelihood function, however, since F_d reacts equally strongly on few voxels of noise and a real person. To counter this effect, the likelihood function could be chosen to $L(\alpha) = -F_d(\alpha) \cdot N(\alpha)$. $N(\alpha)$ biases towards larger areas. A problem with this approach is apparent by comparing Figure 5a and $N(\alpha)$ in Figure 4. When α grows to include both persons, $N(\alpha)$ just keeps growing. To avoid including multiple persons, $F(\alpha)$ is also included to give the final likelihood function:

$$L(\alpha) = -F(\alpha - k)^2 \cdot \sqrt{N(\alpha + k)} \cdot F_d(\alpha) \qquad (5)$$

Instead of $F(\alpha)$ and $N(\alpha)$, $F(\alpha - k)$ and $N(\alpha + k)$ are used to avoid calculating $N(\alpha)$. The functions are weighted by squaring $F(\alpha - k)$ and taking the square root of $N(\alpha + k)$ to bias towards single coherent persons. Figure 5 illustrates the performance of the likelihood for a particular situation, where two persons are located close to one another. The projection of the voxels to the floor is shown in Figure 5b, where a brighter colour correspond to more voxels in the same column. Figure 5a illustrates the likelihood for all values of x and y with α fixed to 4. The set (x_i, y_i) that satisfy $(x_i, y_i) = \text{argmax}_\alpha(L(x, y, 4))$ is marked, and α is adjusted at that location to satisfy $\alpha_i = \text{argmax}_\alpha(L(x_i, y_i, \alpha))$. The state $\mathbb{S}(x_i, y_i, \alpha_i)$ is shown as a box in Figure 5a.

2.3.2 Body Posture and Hot Spots

When noise is present in the detected 3D foreground, it is mostly located close to the floor. This is partly due to shadows and partly due to the fact, that other kinds of noise in the 2D foreground detections in most cases are filtered out by the combination of the cameras. This means, that the height of the persons can be accurately estimated by taking the maximum vertical position of the voxels located within



(a) Projection of voxels to the floor. L is optimised with respect to α with fixed x and y. The optimal value is found to be $\alpha = 5$.

(b) Likelihood $L((x, y, 4)|\vec{y})$.

Figure 5: Illustration of likelihood function.

2D-position of the tracked target. By comparing the height with different thresholds, the body posture is identified as either standing, sitting or fallen. FIR-filters are applied to ensure robustness to noise.

People staying near hot spots are detected by analysing the movement of the targets over a predetermined period of time. The variance in the distance from the mean 2D location in the period under consideration is calculated and compared to a threshold.

3. RESULTS

The system is tested on a setup of 5 calibrated cameras available at AIT. Four cameras are placed in the corners of a room and one camera with a fish-eye lens is placed in the ceiling. Using this setup, qualitative tests of the systems ability to detect people falling, sitting, and spending time on hot spots are carried out.

In a test sequence, up to four people move around in the area under surveillance for 6:23 min. At 3 occasions in total a person falls and at 6 occasions a person sits down. All of these events are detected correctly and there are no standing/sitting persons that are falsely detected as fallen. When a person kneels or bows he can be classified as sitting but not as fallen.

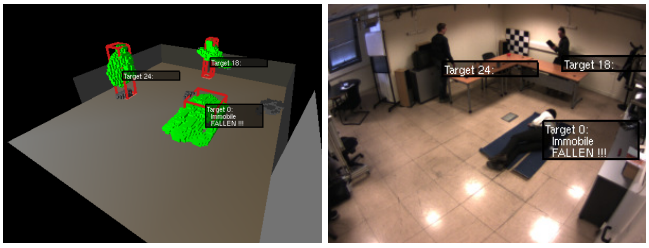
An image from one of the corner cameras from the test sequence is shown and compared with the detected foreground in Figure 6 and the complete test videos of both detected foreground and images from the camera are available on our website[1].

With a recorded set of test videos, a single dual-core 2.2 GHz computer is capable of processing a frame in approximately 1/8 second excluding the time required to load the images from the hard drive. To make the system run in real time, a distributed version has been developed, which enables the whole system to run in real-time on five 3.0 GHz dual-core computers when the cameras are recording at 15 fps.

4. DISCUSSION AND CONCLUSION

We have in this paper presented an adaptive 3D approach for sensing people in a multi camera setup. Foreground is found per camera using a per pixel Gaussian Mixture Model and combined to a discrete 3D foreground. For this, a novel

[1] http://kom.aau.dk/~zt/online/3DSensing/



(a) Detected foreground is shown as green voxels, targets are shown using wire frame boxes, and info boxes are shown for each target.

(b) Frame from one of the corner cameras with targets and info superimposed.

Figure 6: Detected 3D foreground are shown in (a) and reasoning results are shown both on top of the foreground and in (b) superimposed on a frame from a corner camera. The person lying in the floor is marked as "Fallen".

approach is used to determine the 3D foreground that combines a hierarchical octree structure with distance transforms to computational cost of the algorithm.

Our tracker is based on a 2.5D particle filter that provides fast tracking with relatively little computational cost. A likelihood function is developed that uses the amount of foreground, the density of the foreground, and the approximate derivative of the density with respect to the size of the target.

The system has been tested on a test video and is able to detect all occasions where a person falls or sits down. It should be noted that the system can fail in detecting a fall if another person is standing close by. This is, however, not critical for monitoring elderly since the fallen can get help from the other person.

It is possible for the system to run in real-time using 5 cameras at 15 fps when distributed to 5 computers. Using 5 computers might not be optimal in a real-life implementation. The major reason is that the computers use USB 2.0, whose bandwidth prevents more cameras from running simultaneously. However, when USB 3.0 gets available one computer will be able to handle a much larger data flow than our test computers.

One major limitation in our system is that tracking is based solely on foreground estimation which again is based solely on motion. Therefore, if a person stays immobile for a long duration, the associated target will eventually be lost. This can be avoided by adding additional modalities to the tracker such as colour, faces, or even sound [14, 19].

ACKNOWLEDGEMENTS

This work has been partly funded by the HERMES Specific Targeted Research Project (Contract No: FP7-216709).

REFERENCES

[1] D. Moellman and P. Matthews, "Video Analysis and Content Extraction." http://videorecognition.com/vt4ns/vace_brochure.pdf.

[2] R. Stiefelhagen, R. Bowers, and J. Fiscus, eds., *Multimodal Technologies for Perception of Humans*. Springer Berlin / Heidelberg, 2008.

[3] "Cognitive Care and Guidance for Active Aging: HERMES Technology." http://www.fp7-hermes.eu/.

[4] A. Waibel and R. Stiefelhagen, eds., *Computers in the Human Interaction Loop*. Springer-Verlang London, 2009.

[5] A. Pnevmatikakis, J. Soldatos, F. Talantzis, and L. Polymenakos, "Robust multimodal audio–visual processing for advanced context awareness in smart spaces," *Personal Ubiquitous Comput.*, vol. 13, no. 1, 2009.

[6] C. Ericson, *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology)*. Morgan Kaufmann, January 2005.

[7] B. U. Treyin, Y. Dedeolu, and A. E. etin, "Hmm based falling person detection using both audio and video," in *IEEE International Workshop on Human-Computer Interaction*, pp. 211–220, Springer-Verlag GmbH, 2005.

[8] A. Leone et. al., "A multi-sensor approach for people fall detection in home environment," in *Proc. 10th ECCV*, 2008.

[9] V. Libal et. al., "Multimodal classification of activities of daily living inside smart homes," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009.

[10] C. Canton-Ferrer, J. Salvador, J. R. Casas, and M. Pardàs, "Multi-person Tracking Strategies Based on Voxel Analysis," in *Multimodal Technologies for Perception of Humans*, (Berlin, Heidelberg), pp. 91–103, Springer-Verlag, 2008.

[11] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, 2000.

[12] P. Kaewtrakulpong and R. Bowden, "An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection," in *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, September 2001.

[13] T. Horprasert, D. Harwood, and L. S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection," in *ICCV Frame-Rate WS*, 1999.

[14] N. Katsarakis, F. Talantzis, A. Pnevmatikakis, and L. Polymenakos, "The AIT 3D Audio / Visual Person Tracker for CLEAR 2007," in *Multimodal Technologies for Perception of Humans*, (Berlin, Heidelberg), pp. 35–46, Springer-Verlag, 2008.

[15] A. Koutsia, et. al., "Traffic Monitoring using Multiple Cameras, Homographies and Multi-Hypothesis Tracking," in *3DTV Conference, 2007*, pp. 1–4, May 2007.

[16] G. Borgefors, "Distance transformations in digital images," *Comput. Vision Graph. Image Process.*, vol. 34, no. 3, pp. 344–371, 1986.

[17] S. S. Blackman, *Multiple-target tracking with radar applications*. Dedham, MA, Artech House, Inc., 1986.

[18] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, 1998.

[19] P. Perez, J. Vermaak, and A. Blake, "Data Fusion for Visual Tracking with Particles," *Proceedings of the IEEE*, vol. 92, pp. 495–513, Mar 2004.