ADAPTIVE PARTICLE FILTERING APPROACH TO AUDIO-VISUAL TRACKING

Volkan Kılıç, Mark Barnard, Wenwu Wang, and Josef Kittler

Centre for Vision, Speech and Signal Processing, University of Surrey, UK
Emails:{v.kilic, mark.barnard, w.wang, j.kittler}@surrey.ac.uk

ABSTRACT

Particle filtering has emerged as a useful tool for tracking problems. However, the efficiency and accuracy of the filter usually depend on the number of particles and noise variance used in the estimation and propagation functions for re-allocating these particles at each iteration. Both of these parameters are specified beforehand and are kept fixed in the regular implementation of the filter which makes the tracker unstable in practice. In this paper we are interested in the design of a particle filtering algorithm which is able to adapt the number of particles and noise variance. The new filter, which is based on audio-visual (AV) tracking, uses information from the tracking errors to modify the number of particles and noise variance used. Its performance is compared with a previously proposed audio-visual particle filtering algorithm with a fixed number of particles and an existing adaptive particle filtering algorithm, using the $AV16.3$ dataset with single and multi-speaker sequences. Our proposed approach demonstrates good tracking performance with a significantly reduced number of particles.

Index Terms**—** Adaptive particle filter, tracking.

1. INTRODUCTION

The problem of tracking and localization of speakers in indoor environments using AV information has received much interest in the last few decades. Many approaches have been proposed by researchers and among these the particle filter (PF) is a popular one. The PF became widely used in tracking after being proposed by Isard and Blake [1]. Speaker tracking may be achieved in a single modality domain through video or audio. However, it has been shown in [2], [3] and [4] that using both video and audio data in tracking gives more reliable results than using each modality individually, as is also confirmed in our recent study [5].

The PF also has some limitations. The samples (particles) used in PF are weighted in order to approximate the filtering distributions. The quality of the sample based representation rises with the number of particles, N. A key question is: How many particles should be used for a specific estimation problem. In most cases the choice is made experimentally and users tend to choose N as large as possible to get accurate results, leading to an increased computational cost.

Adaptive particle filtering (APF) approaches have therefore been proposed in [6], [7] and [8] to address these problems and to find the optimal N for the PF to use. An early and popular approach, i.e. KLD-Sampling was proposed by Fox in [8]. This approach aims to bound the error introduced by the sample-based representations of the PF using the Kullback-Leibler divergence between Maximum Likelihood estimates (MLE) of states and the underlying distribution to optimize the number of particles. Specifically, this method is applied to mobile robot localization problem, where the initial set of particles is generally very large. Also, it is not clear how to apply this approach to more general particle filters that provide posterior-based estimates rather than MLE. Moreover, It assumes that the true posterior is given by a discrete constant piecewise distribution such as a multi-dimensional histogram bins, but characteristics of robot localization (e.g., binning of the state space) might not be valid in other situations.

Another parameter in a PF that is often fixed is the noise variance (Q) which has a critical role in the distribution of particles. This role makes the determination of Q crucial so it should not be chosen randomly or empirically since incorrectly chosen Q may lead to the use of a greater N than actually needed. Therefore, we intend to design an APF approach that involves dynamic estimation of Q in order to find the optimal N. In this paper, we propose an AV-APF algorithm which is based on our previous AV-PF algorithm. We show the efficiency of our algorithm in single and multi speaker tracking in comparison with the KLD-Sampling algorithm.

The next section introduces related works. Our proposed APF algorithm is given in Section 3, and experimental results are presented in Section 4, followed by the conclusions.

2. RELATED WORKS

The AV-APF algorithm presented in this paper is based on our recent work in [5], and the KLD-Sampling algorithm [8].

2.1. Audio-Visual Particle Filtering Algorithm

The AV-PF algorithm that we presented in [5] is created by combining a standard sampling importance resampling (SIR) PF based visual tracker with the direction of arrival (DOA) information estimated from audio measurements. In the DOA estimation process, a parametric approach [9] is used for localization, and the location parameters are optimized with respect to a cost function such as SRP-PHAT [10]. Then, a third-order AR model is performed to reduce the estimation noise in the DOA azimuth θ_k.

$$\theta_k = \sum_{i=1}^{3} \varphi_i \theta_{k-i} + \varepsilon_k \tag{1}$$

where φ_i is the parameter of the AR model and ε_k is white noise at time frame $k = 1, ..., K$.

The DOAs are then used to constrain the propagation of the particles and the weights in the observation model of the visual tracker. To do this, a DOA line is drawn from the centre of the microphone array to the coordinates of speaker's head. The Euclidean distances $\mathbf{d}_k = \left[d_k^{(1)} \quad ... \quad d_k^{(N)}\right]^T$ of the particles to the DOA line are calculated and used to derive the movement distances $\hat{\mathbf{d}}_k$ which guide by what distance the particles should be moved towards the DOA line,

$$\hat{\mathbf{d}}_k = \frac{\mathbf{d}_k \odot \mathbf{d}_k}{\|\mathbf{d}_k\|_1} \tag{2}$$

where $\hat{\mathbf{d}}_k = \left[\hat{d}_k^{(1)} \quad ... \quad \hat{d}_k^{(N)}\right]^T$ and $\odot$ is the dot (element-wise) product and $\|.\|_1$ is the ℓ_1 norm.

The SIR PF has five steps. First, the particles are initialized by $\mathbf{x}_0^{(n)} \sim p(\mathbf{x}_0)$, $w_0^{(n)} = \frac{1}{N}$ for $n = 1, ..., N$. Here $w_0^{(n)}$ is the initial weights of the particles. The state vector is defined as $\mathbf{x} = \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 & s \end{bmatrix}^T$, where x_1 and x_2 are respectively the horizontal and vertical position of the rectangle centred around the face, $\dot{x}_1$ and $\dot{x}_2$ are respectively the horizontal and vertical velocity, and s is the scale of the rectangle centred around (x_1, x_2). The particles are propagated in the second step by a dynamic model,

$$\mathbf{x}_k^{(n)} = \mathbf{F}\mathbf{x}_{k-1}^{(n)} + \mathbf{q}_k^{(n)} \tag{3}$$

where $\mathbf{x}_k^{(n)}$ is the state of n^{th} particle and $\mathbf{q}_k^{(n)}$ is the zero-mean Gaussian noise with covariance $\mathbf{Q}$, $\mathbf{q}_k^{(n)} \sim \mathcal{N}(0, \mathbf{Q})$ for each particle. $\mathbf{F}$ is the linear motion model. In the third step, the particles are weighted by the observation model,

$$w_k^{(n)} = p(\mathbf{y}_k^{(n)} | \mathbf{x}_k^{(n)}) = e^{-\lambda (D^{(n)})^2} \tag{4}$$

where $\mathbf{y}_k^{(n)}$ is the observation. The observation $\mathbf{y}_k^{(n)}$ is obtained for each state estimate $\mathbf{x}_k^{(n)}$ by the design parameter λ and $D^{(n)}$ which is the Bhattacharyya distance,

$$D^{(n)} = \sqrt{1 - \sum_{u=1}^{U} \sqrt{r(u) q^{(n)}(u)}} \tag{5}$$

where, U is the number of bins used by the histogram, $r(u)$ is the Hue histogram of the reference image and $q^{(n)}(u)$ is the Hue histogram extracted from the rectangle centred on the position of the n^{th} particle. After the weights are normalized, the position of the speaker is estimated in the fourth step by:

$$\tilde{\mathbf{x}}_k = \sum_{n=1}^{N} w_k^{(n)} \mathbf{x}_k^{(n)} \tag{6}$$

The audio information is fused with video in the particle propagation and importance weighting steps where a weighting parameter γ_k is also used to balance the potential adverse effect of estimation noise within the DOA. We choose the image patch $q(u)$ centred on the estimated position and calculate γ_k as the distance between $q(u)$ and the reference image patch $r(u)$, by substituting $q(u)$ for $q^{(n)}(u)$ in (5). The dynamic model given in (3) is then revised to

$$\hat{\mathbf{x}}_k^{(n)} = \mathbf{x}_k^{(n)} \oplus \hat{d}_k^{(n)} tan(\theta_k) \gamma_k \tag{7}$$

where $\oplus$ is the element-wise addition. The importance weights are also adapted using $\hat{d}_k^{(n)}$ and γ_k as follows:

$$\hat{w}_k^{(n)} = (e^{-\lambda (D^{(n)})^2}) \frac{\|\mathbf{d}_k\|_1}{d_k^{(n)}} \gamma_k \tag{8}$$

After the weights are normalized, position estimation follows and it is calculated using (6). Then the resampling step is performed to generate the new particles.

2.2. Existing APF Schemes

The key idea behind the KLD-Sampling algorithm [8] is to estimate the number of particles adaptively to bound the error of the particle filter. To measure the error, the Kullback-Leibler (KL) divergence between the empirical distribution and the true posterior distribution, known as nonparametric maximum likelihood estimate, is used. KLD-Sampling assumes that the true posterior can be represented by a discrete piecewise constant distribution consisting of a set of multidimensional bins. This assumption allows the use of a chi-square (χ^2) distribution in the convergence of the likelihood ratio statistic to find a bound for the number of particles, N:

$$N = \frac{1}{2\epsilon} \chi^2_{m-1, 1-\delta} \tag{9}$$

where $(1 - \delta)$ is the quantile of χ^2 distribution with $m - 1$ degrees of freedom, m is the number of bins and ϵ is the upper bound for the error given by the KL-divergence. In order to determine N according to (9), a Wilson-Hilferty transformation [8] is applied to compute the quantiles of the chi-square distribution, which yields

$$\begin{aligned} N &= \frac{1}{2\epsilon} \chi^2_{m-1, 1-\delta} \\ &\doteq \frac{m-1}{\epsilon} \left\{1 - \frac{2}{9(m-1)} + \sqrt{\frac{2}{9(m-1)}} z_{1-\delta}\right\}^3 \end{aligned} \tag{10}$$

where $z_{1-\delta}$ is the upper $1 - \delta$ quantile of the standard normal $N(0, 1)$ distribution.

Incorporation of KLD-Sampling into the PF algorithm is done by estimating m in the sampling step by incrementally checking for each generated sample whether it falls into an empty bin. The bin size is specified initially, depending on the application where PF is used, and kept constant during implementation. At the beginning of sampling, m goes up with almost every new sample since virtually all bins are empty. After each sample, N is updated by equation (10) required for the current estimate of m. Eventually, more and more bins become non-empty and once N remains unchanged, the update stops.

Detailed information about the KLD-Sampling algorithm can be found in [8].

3. PROPOSED APF APPROACH

The adaptive approach brings flexibility on the determination of optimal parameters and potentially improves the performance of the tracking system. The KLD-Sampling approach introduced in Section 2.2, builds on the assumption that the distribution consists of a set of multidimensional bin sizes. There is no certain way to estimate that size and it may easily cause deviation in the estimation of N if the size is not selected properly. Another problem is the fixed parameter Q which needs to be found empirically. Its selection affects the distribution of the particles, causing the tracker to be potentially unstable. Therefore we introduce a new algorithm which is able to adapt both N and Q dynamically in a simple way and can be applied to any implementation.

Our proposed algorithm is based on the area occupied by the rectangles centred on the positions of the particles in order to detect the face of the speakers. The rectangles occupy an area on the image frame, and size of the area can be defined as below:

$$A = f(N, Q, d) \tag{11}$$

where A is the area, d contains the horizontal and vertical distances from the center of the rectangle. The A depends on N, dimension of the rectangle d and the overlap between the rectangles. The overlap is highly related to the distance between the particles, namely Q, which affects the distribution of the particles. To be able to calculate A, we need to identify a mapping function f which is analytically challenging and intractable. We develop another solution by creating a mapping table using extensive tests under fixed d (in our application $d = (15\mathbf{x}22)$) by changing Q from 10 to 150 with 10 steps and N from 5 to 100 with 5 steps to calculate the occupied area in pixels. For every point (for example, $N = 10$, $Q = 50$), it is repeated 100 times and the average of the occupied area is taken as shown in Figure 1. Illustration of the occupied area estimation is given in Figure 2. After particle distribution, rectangles are drawn centred on the position of particles. For $N = 5$ and $Q = 50$, the occupied area inside the blue line of Figure 2 is calculated as 2000 in pixel units for the point stressed by red cross in Figure 1.

Then, every corresponding point (A, Q) is interpolated using the N lines in the mapping table. So, given A and Q we can use the mapping table to find N. From the observation of the mapping table, A and Q are approximated empirically as:

$$\begin{aligned} A_k &= A_{k-1} * e^{\gamma_k - \gamma_{threshold}} \\ Q_k &= \sqrt{A_k/\pi} * 2 \end{aligned} \tag{12}$$

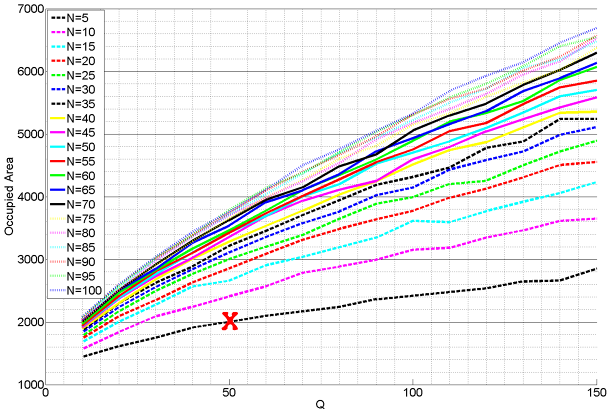


Fig. 1. Mapping between N and A, Q.

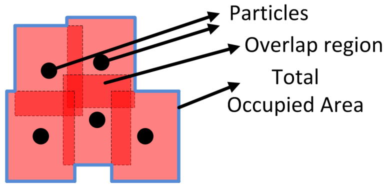


Fig. 2. Occupied area of the particles

where $\gamma_{threshold}$ is our desired upper error bound.

For every iteration, after γ_k is calculated using equation (5), $\gamma_{threshold}$ is subtracted to calculate the error in order to re-calculate N_k and Q_k values using (11) (approximated by Figure 1) and (12) respectively. One of the advantages of using Q_k in APF is to be able to tolerate small errors by increasing Q_k. When the area, A, of the particles is increased, the error becomes smaller. Therefore, if the error is smaller than a pre-defined level γ_{min}, N_k stays the same and Q_k is changed by Q_{min}. If the error is larger than γ_{min}, then the occupied area A_k and Q_k are calculated using equation (12). So, after new A_k and Q_k values are calculated, the mapping table is checked for (A_k, Q_k) in order to find the optimal N_k.

The last step of the PF algorithm is resampling and since the N_k value has just changed, this step is also modified for the new N_k. In the case that N_k is decreased, the particles with the smallest weights are removed and if N_k is increased then the particles with largest weights are duplicated before the resampling step is performed. The pseudo code of the proposed algorithm is given in Table 1.

4. EXPERIMENTS

4.1. Setup

The proposed algorithm was tested using the $AV16.3$ corpus developed by the IDIAP Research Institute [11]. The corpus consists of subjects moving and speaking at the same time whilst being recorded by three calibrated video cameras and two circular eight-element microphone arrays. The audio was recorded at 16 kHz and video was recorded at 25 Hz. They

Table 1. Proposed APF Algorithm

```
Initialize: $N_0$, $Q_0$, $A_0$, $U$, $T$, $\mathbf{F}$, $\lambda$, $r(u)$, $\mathbf{x}_0^{(n)}$, $w_0^{(n)}$, $k$
while $k < K$ do
    // AV Particle Filter - Section 2.1.
    Calculate $\mathbf{x}_k^{(n)}$, $w_k^{(n)}$ and $\gamma_k$ using equation (3), (4) and (5), respectively.
    Find movement distances by equation (2)
    Calculate $\hat{\mathbf{x}}_k^{(n)}$ and $\hat{w}_k^{(n)}$ using equation (7) and (8)
    Re-estimate target position using equation (6)
    // Adaptive approach modifications - Section 3
    Re-calculate $\gamma_k$ using equation (5)
    if $\gamma_k < \gamma_{threshold} + \gamma_{min}$ then
        $Q_k = Q_{k-1} + Q_{min}$; and $N_k = N_{k-1}$
    else
        $A_k = A_{k-1} * e^{\gamma_k - \gamma_{threshold}}$; and $Q_k = \sqrt{A_k/\pi} * 2$;
        $N_k = mapping\_table(A_k, Q_k)$ in Figure 1.
    Resampling: Generate $\mathbf{x}_k^{(n)}$ from the set $\left\{\hat{\mathbf{x}}_k^{(n)}, \hat{w}_k^{(n)}\right\}_{n=1}^{N_k}$
    $k = k + 1$
end
```

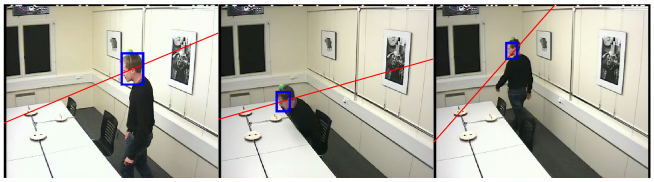


Fig. 3. Sequence 11 (camera #3): Single speaker.

were synchronized before being used in our system. Each video frame is a colour image of 288x360 pixels.

In the sequences, the speakers wear a ball for annotation but in our application this ball is never used. In this paper, we used one single speaker sequences (sequence 11, cameras #3) which has 769 frames and one multiple speaker sequence (sequence 24, camera #1) which has 1201 frames to test our proposed algorithm shown in Figure 3 and Figure 4, respectively. In all simulations, λ in (4) is chosen as 150. The number of bins used for Hue histogram is 8.

4.2. Results and Discussion

Our proposed APF algorithm is based on the AV-PF presented in [5]. To show the efficiency of our approach, it is compared with KLD-Sampling algorithm which is discussed in Section 2.2. This approach is then combined with our proposed AV-PF [5] to make fair comparison with our proposed AV-APF.

In our application, the ideal γ_k value is determined as 0.25. So we chose $\gamma_{threshold} = 0.25$. γ_{min} is typically set to 0.04 and Q_{min} to 10. For the KLD-Sampling algorithm, $z_{1-\delta}$ is set to 2.55 and bin size is chosen as 30x40 pixels.

To show the advantage of the APF, we perform an experiment to compare our proposed AV-APF with a fixed number AV-PF. Firstly, the AV-APF is run on Sequence 11 (Figure 3) and we reach an average $\gamma = 0.26$ with an average $N = 16$. Then, the fixed AV-PF is run with $N = 16$ and the value of γ goes up to 0.32. This almost 23% error difference shows that the APF is better than the fixed number PF, as shown in Figure 5. In Figure 5-(a), N is changing over time for the APF, and Figure 5-(b) shows γ for both approaches. In Figure 5-(b), the error increases for fixed PF approach around frame number 300 since it can not react to sudden movement of the speaker. In contrast, the error in our proposed APF is stable because of the adaptive change in N and Q.

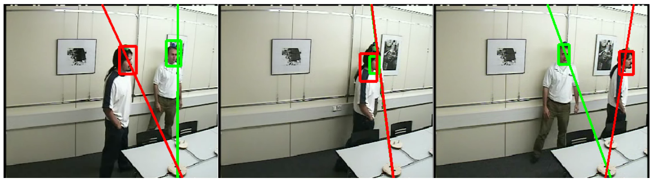
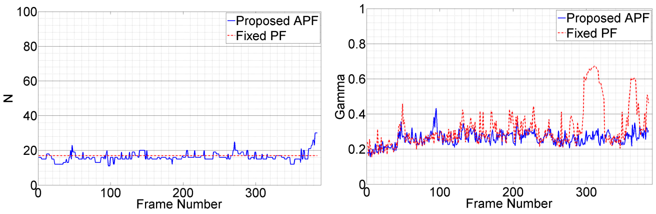


Fig. 4. Sequence 24: Multiple speakers with occlusion.



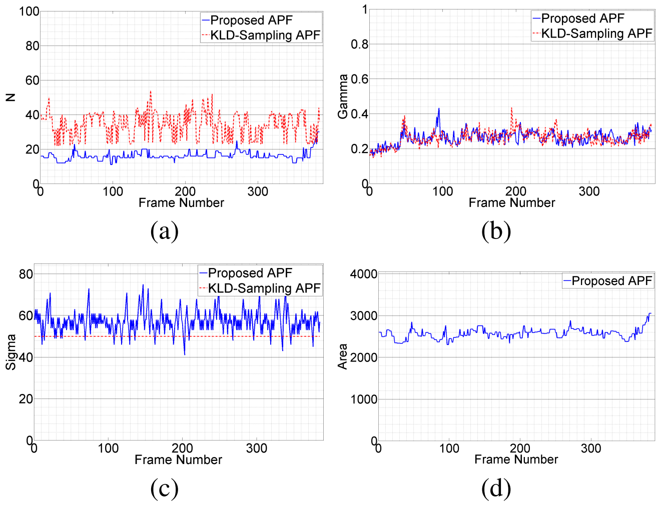
(a) (b)

Fig. 5. In (a) Average N for Proposed APF and Fixed N are 16. In (b) Average γ for Proposed and Fixed APF are 0.26 and 0.32, respectively.

The KLD-Sampling algorithm is also tested on the same sequence and comparison results with the proposed APF are given in Figure 6. The KLD-Sampling algorithm needs an average of 34 particles to reach the same value $\gamma = 0.26$. Figure 6-(a) and Figure 6-(b) show the effect of changing N and γ respectively. Q is set to 50 in KLD-Sampling, but since Q is also adaptive in our proposed approach, the average Q is found to be 57 as seen in Figure 6-(c). The effect of changing A is shown in Figure 6-(d) which is a parameter specific to our proposed approach.

In another experiment we look at a multi-speaker sequence (see Figure 4) with speakers occluding each other. The results of these experiments are shown in Figure 7 and it can be seen that both APF approaches need to increase N when the occlusion occurs. Our proposed APF used an average of 41 particles and KLD-Sampling used 60 particles. Both continued tracking with $\gamma = 0.30$.

KLD-Sampling is a popular approach in the literature, but one of the limitations of this approach is having only one adaptive parameter (N). Another one is that it needs the bin size Δ, a parameter, which also affects the performance of the algorithm. Generally, KLD-Sampling shows better performance in the area of robotics in which tracking is done in a



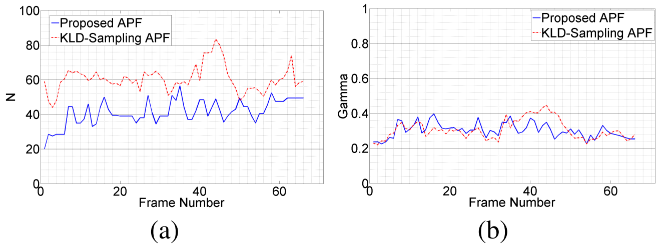
(a)

(b)

(c)

(d)

Fig. 6. In (a) average N for the Proposed and KLD sampling APF are 16 and 34, respectively. In (b) average γ for both algorithm are 0.26. In (c) $Q = 50$ in KLD-sampling and average Q for the Proposed APF is 57. In (d) change of A with respect to frame number is given and average A is 2565.



(a)

(b)

Fig. 7. Multi person tracking. In (a) average N for Proposed and KLD sampling APF are 41 and 60, respectively and (b) average γ for both sampling algorithm are 0.30.

vast area with a large number of N (over 1000). In our adaptive approach, we also use the Q value to find the optimum N value. Small errors can be overcome by increasing Q without changing N. The mapping table also simplifies the calculation of N. These make our proposed AV-APF algorithm simple and efficient.

5. CONCLUSION

In this study, we have presented a new adaptive particle filtering algorithm which uses audio and visual information to adapt the number of particles and noise variance. Our proposed algorithm has been tested on both single and multiple speaker sequences and compared with fixed particle filter and an existing adaptive particle filter algorithm. The experiments demonstrate that the proposed algorithm can effectively track moving objects and increase robustness in tracking in the sense that it reduces the number of particles without increasing error.

6. ACKNOWLEDGEMENT

This research was supported by the Engineering and Physical Sciences Research Council of the UK (grant no. EP/H050000/1).

7. REFERENCES

[1] M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[2] M. Heuer, A. Al-Hamadi, B. Michaelis, and A. Wendemuth, “Multi-modal fusion with particle filter for speaker localization and tracking,” in *Int. Conf. on Multimedia Technology*, 2011, pp. 6450–6453.

[3] S.T. Shivappa, B.D. Rao, and M.M. Trivedi, “Audio-visual fusion and tracking with multilevel iterative decoding: Framework and experimental evaluation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 882–894, 2010.

[4] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, “Audiovisual probabilistic tracking of multiple speakers in meetings,” *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 15, pp. 601–616, 2007.

[5] V. Kilic, M. Barnard, W. Wang, and Josef Kittler, “Audio constrained particle filter based visual tracking,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., Vancouver, Canada*, 2013.

[6] P. Closas and C. Fernandez-Prades, “Particle filtering with adaptive number of particles,” in *IEEE Aerospace Conference*, 2011, pp. 1 –7.

[7] Alvaro Soto, “Self adaptive particle filter,” in *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence*, 2005, pp. 1398–1403.

[8] D. Fox, “Adapting the sample size in particle filters through kld-sampling,” *International Journal of Robotics Research*, vol. 22, pp. 985–1003, 2003.

[9] G. Lathoud, J. Bourgeois, and J. Freudenberger, “Sector-based detection for hands-free speech enhancement in cars,” *EURASIP Journal on Applied Signal Processing*, 2006.

[10] J. DiBiase, “A high-accuracy, low-latency technique for talker localisation in reverberant environments,” in *Ph.D. dissertation, Brown University, Providence, RI, USA*, 2000.

[11] G. Lathoud, J. M. Odobez, and D. Gatica-perez, “Av16.3: an audio-visual corpus for speaker localization and tracking,” in *Proceedings of the 2004 MLMI Workshop, S. Bengio and H. Bourlard Eds.* 2005, Springer Verlag.