

Local Search Particle Filter applied to Human-Computer Interaction

Juan José Pantrigo
juanjose.pantrigo@urjc.es

Antonio S. Montemayor
antonio.sanz@urjc.es
Universidad Rey Juan Carlos
Tulipán s/n 28933 Móstoles, Spain

Ángel Sánchez
angel.sanchez@urjc.es

Abstract

This paper presents an hybridization of Particle Filter and Local Search algorithms, called Local Search Particle Filter (LSPF), and its application to Human-Computer Interaction. The proposed algorithm combines both sequential Monte Carlo (Particle Filter - PF) and local search methods to achieve an accurate real-time hand tracking. The system allows to control different mouse actions through a reduced set of hand movements and gestures. Hand are segmented using a skin-color model based on explicit RGB region definition. The proposed hybrid tracking method increases the performance of general particle filter. It also improves the quality of the hand tracking task (the standard deviation between hand spatial positions for LSPF is reduced a 75% with respect to the PF algorithm). More precisely, a local search enhances a hand-simulated mouse cursor to smoothly move and thus recognize gestures for performing their associate actions.

1. Introduction

Automatic visual analysis of human motion is an active research topic in Computer Vision and its interest has been growing during last decade [14]. Human-computer interaction is evolving towards non-contact devices, using perceptual and multimodal user interfaces. That means the system allows the user to interact without physical contact, using voice and/or gestures [5]. This form of interaction is targeted by Intelligent User Interfaces (IUI) [8], a sub-field of Human-Computer Interaction (HCI). The goal is to improve human-computer communication using new technologies and also techniques for Artificial Intelligence. In particular, Computer Vision is very useful for IUI [15], providing new ways of interaction making use of body parts motion. In this way, by providing new forms of interaction, based on video or audio processing instead of traditional computer devices like keyboard or mouse, handicapped people can access the computers more easily. Gesture tracking by monocular vision is an important task for the development of such systems. Recently, the field of Computer Vision has devoted considerable research effort to the detection and recognition of faces and hand gestures [9]. The

multiple applications of this topic have fostered the organization of a biannual IEEE conference specifically devoted to Face and Gesture Recognition [2][1] since 1995.

To locate the regions of interest, this kind of systems needs from a previous object tracking procedure. Tracking human movement (in our case, the hand) is a challenging task which strongly depends on the considered application [3]. Most reported work on vision-based HCI relies on visual tracking and visual template recognition algorithms.

Recent research in human motion analysis makes use of the particle filter (PF) framework. PF algorithm (also termed as Condensation algorithm) enables the modeling of a stochastic process with an arbitrary probability density function (pdf), by approximating it numerically with a set of points called particles in a state-space process [4]. One main aim of particle filtering is accurate tracking of a variable of interest as it evolves over time [12], such as objects in video sequences for determine their kinematic information.

The main contribution of this paper is the development of a single-object visual tracker based on the hybridization of particle filters (PF) and local search LS algorithms and its application to Human-Computer Interaction. Local search particle filter (LSPF) hybridizes both PF and LS frameworks in two different stages. In the PF stage, a particle set is propagated and updated to obtain a new particle set. In the LS stage, the best particle found are refined using a local search procedure. LSPF algorithm significantly improves the performance of general particle filters. This algorithm forms the basis of a mouse controlled system. This approach allow us to interpret different mouse actions by means of movements and gestures of hands in real-time using a standard PC computer.

2. Particle Filters

To make inference about a dynamic system, two different models are necessary: (i) a measurement model requiring an observation vector (\mathbf{z}) and a system state vector (\mathbf{x}), and (ii) a system model describing the evolution of the state of the system [4].

The objective in the Bayesian approach to dynamic state estimation is to construct the posterior pdf of the state. Usually an estimate is required at every time step. In the framework of Sequential Bayesian Modelling, posterior pdf

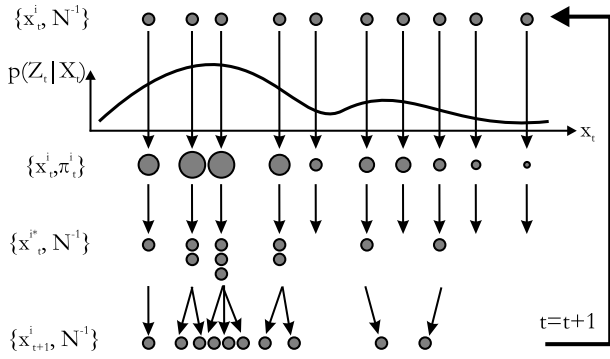


Figure 1. Particle Filter scheme.

is estimated in two stages:

1. Prediction: the posterior pdf $p(\mathbf{x}_t | \mathbf{z}_{1:t-1})$ is propagated at time step t using the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1} \quad (1)$$

2. Evaluation: the posterior pdf $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ is computed at each time step by applying the Bayes' theorem, using the observation vector $\mathbf{z}_{1:t}$:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1})} \quad (2)$$

These equations constitute the theoretical Bayesian solution to the dynamic estimation problem. Unfortunately, the propagation of the posterior pdf cannot be calculated analytically except for a little set of cases [4]. For example, the Kalman filter assumes the posterior pdf is always Gaussian, and grid-based methods assumes a discrete and finite number of states. Nevertheless, the general case has interest for many applications [6].

Particle filters (PF) are a special class of sequential estimation methods in which theoretical distributions on the state space are approximated by simulated random measures [6]. This pdf is represented by a set of weighted samples, called particles $\{(\mathbf{x}_t^0, \pi_t^0), \dots, (\mathbf{x}_t^N, \pi_t^N)\}$, where the particle weights $\pi_t^n = p(\mathbf{z}_t | \mathbf{x}_t = \mathbf{x}_t^n)$ are normalized. In Figure 1 an outline of the Particle Filter scheme is shown.

PF algorithm starts by setting up an initial population $\mathbf{x}_0 = \{\mathbf{x}_0^i | i = 1, \dots, N\}$, of N particles using a known pdf. The measurement vector \mathbf{z}_t at time step t , is obtained from the system. Particle weights at time step t , $\pi_t = \{p^i | i = 1, \dots, N\}$ are computed using a fitness function. Weights are normalized and a new particle set \mathbf{x}_t^* is selected. As particles with higher weights can be chosen several times, a diffusion stage is applied to avoid the loss of diversity in \mathbf{x}_t^* . Finally, particle set at time step $t + 1$, \mathbf{x}_{t+1} ,

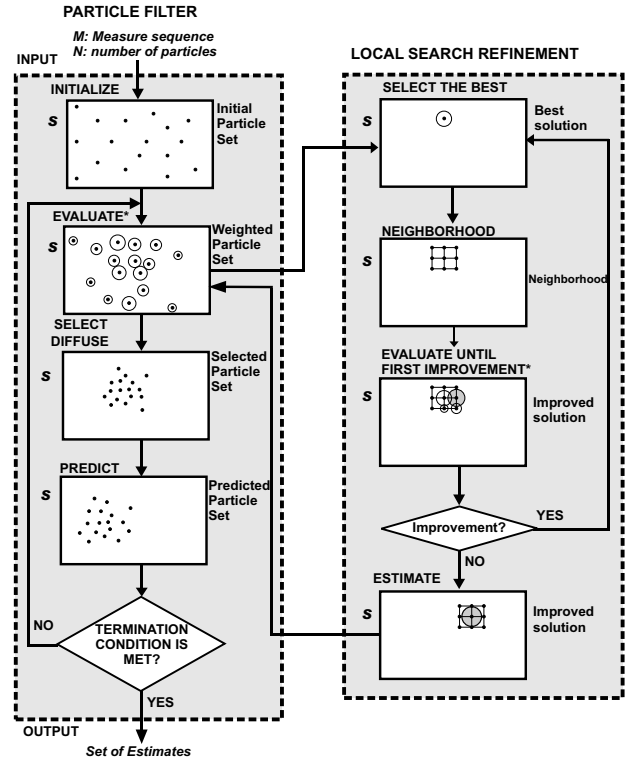


Figure 2. Local Search Particle Filter scheme. Weight computation is required during EVALUATE stages (*)

is predicted using a defined motion model. Therefore, PF can be seen as algorithms handling the particles time evolution. The set of particles in PF move according to the state model and their evolution is related to their weight values as determined by the likelihood function [6].

3. Local Search Particle Filter

Local Search Particle Filter (LSPF) algorithm is introduced to be applied to estimation problems in sequential processes that can be expressed using the state-space model abstraction. The aim of this algorithm is to improve the efficiency of the standard particle filters, by means of a local search procedure. This proposal is specially suitable for applications requiring high quality estimations. LSPF integrates both local search (LS) and particle filter (PF) frameworks in two different stages:

- In the *particle filter stage*, a particle (solution) set is propagated over the time and updated with measurements to obtain a new one. This stage is focused on the time evolution of the best solutions found in previous time steps.
- In the *local search stage*, the best solution from the particle set are selected and a local search procedure is

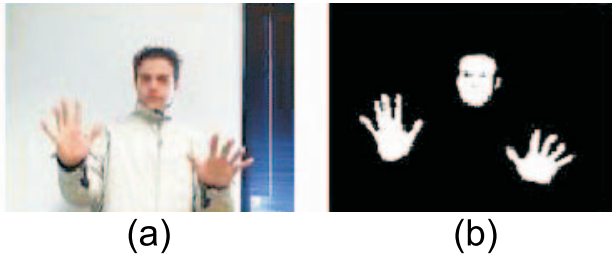


Figure 3. Measurement model: (a) input frame and b) resulting skin detection.

performed in its neighborhood. This stage is devoted to improve the quality of the PF estimate.

Figure 2 shows a graphical template of the LSPF method. Dashed lines separate the two main components in the LSPF scheme: PF and LS, respectively. LSPF starts with an initial population of N particles drawn from a known pdf (Figure 2: INITIALIZE stage). Each particle represents a possible solution of the problem. Particle weights are computed using a weighting function and a measurement vector (Figure 2: EVALUATE stage). LS stage is later applied for improving the best obtained solution of the particle filter stage. First, a neighborhood of the best solution are defined (Figure 2: NEIGHBORHOOD stage). Then, solution weights are computed until a better solution is found in the neighborhood of the initial one (Figure 2: EVALUATE UNTIL FIRST IMPROVEMENT stage). This procedure is repeated until there are no better solutions in the neighborhood than the initial one. Once the LS stage is finished, a new population of particles is created by selecting the individuals from the whole particle set with probabilities according to their weights (Figure 2: SELECT stage). To avoid the loss of diversity, a diffusion stage is applied to the particles of the new set (Figure 2: DIFFUSE stage). Finally, particles are projected into the next time step by making use of the update rule (Figure 2: PREDICT stage).

This algorithm have similarities with the LS-N-IPS (local search N -interacting particle filter) proposal [13]. While LS-N-IPS performs a local search over all particles, LSPF only applies the improvement procedure to the best particle found in PF stage. Thus LSPF is more efficient than LS-N-IPS since local search is an expensive task. This approach is successful in a wide variety of visual tracking problems.

4. System Description

Our proposed LSPF approach is applied to determine the position of hands and face in 2D image sequences. Each particle in the particle set describes a possible solution to the tracking problem. The *particle structure* used in this work is:

$$[x, y, \dot{x}, \dot{y}] \quad (3)$$

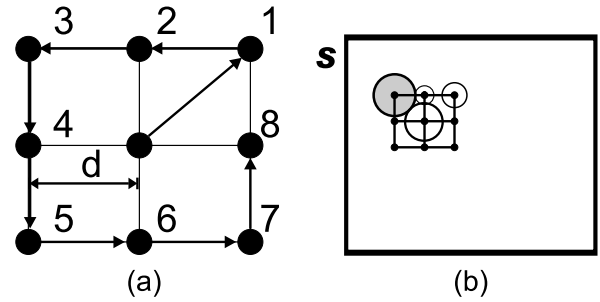


Figure 4. (a) Local search execution order and (b) an example of execution.

where x and y are the spatial positions, and \dot{x} and \dot{y} represents the first derivative of magnitude (velocity). The number of particles N in the particle set S is experimentally chosen to be 100. The local search is performed over the best solution found by the particle filter.

The *observation model* specifies the image features to be extracted. To construct the weighting function it is necessary to use adequate image features. In this work we have used a pixel-based skin color detection method (see Figure 3). In this method an explicit region of RGB color space is defined as skin. A pixel (R, G, B) is classified as skin if [10]:

$$(R > 45) \& (G > 40) \& (B > 20) \& (max(R, G, B) - min(R, G, B) > 15) \& (|R - G| > 15) \& (R > G) \& (R > B) \quad (4)$$

The main advantage of this method is the simplicity of skin detection rules that leads to construction of a fast pixel classifier. Particle weights are computed as the number of skin pixels belonging to a 50×75 pixels rectangular window located in $[x, y]$.

The *system model* describes the temporal update rule for the system state [16]. The tracked object state consists of a given number of spatial (linear or angular) coordinates and the corresponding velocities, deriving in a first-order motion model. Two excitation forces F and G (modeled by random Gaussian variables with zero mean and normal de-

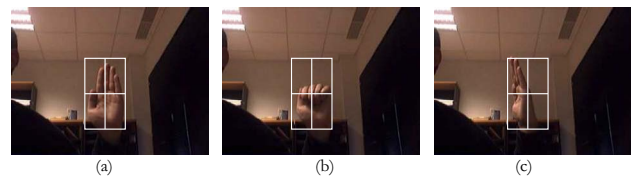


Figure 5. Mouse events: (a) No pushed button, (b) left button pushed and (c) right button pushed.

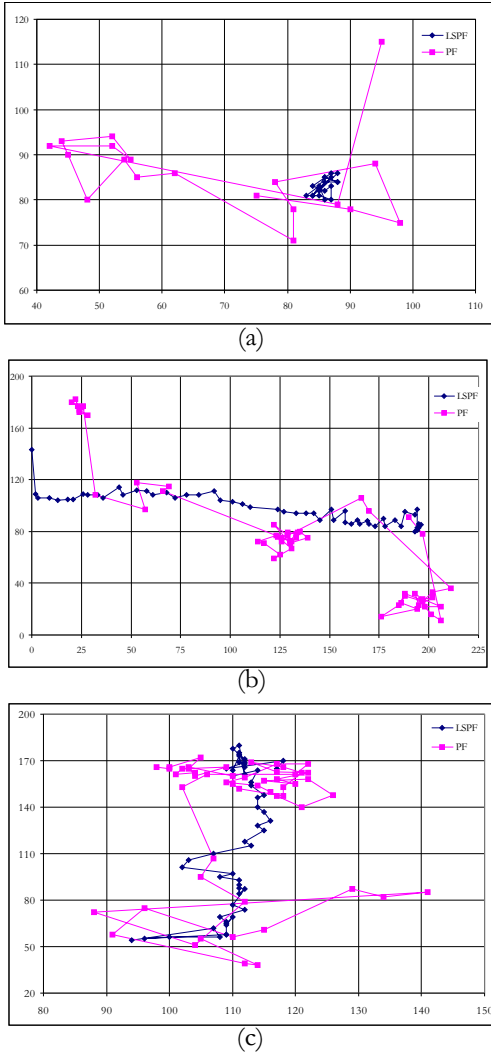


Figure 6. Estimated trajectories in (a) static, (b) horizontal and (c) vertical movements, using PF and LSPF.

viation σ_F and σ_G respectively) allow changes in the object state (position and velocity). The value of σ_F and σ_G depend on expected changes in the position and velocity of the tracked object. The update rule used in this work is performed by these two equations:

$$\begin{aligned} x_{t+\Delta t} &= x_t + \dot{x}_t \Delta t + F_x \\ \dot{x}_{t+\Delta t} &= \dot{x}_t + G_x \end{aligned} \quad (5)$$

where x represents some spatial (linear or angular) variable, Δt is the time step and F_x and G_x are random Gaussian variables with zero mean and normal deviation σ_F and σ_G , respectively.

A standard *first improvement local search* was embedded into the PF scheme. Given a solution, a neighborhood is explored until a new high quality solution is found. Then,

this new solution replaces the old one and the procedure is repeated until no improvement is produced. To avoid the deviation of consecutive estimations, the local window is adjusted always as shown by the numerical ordering of Figure 4a. This is achieved by performing the local search procedure in the same order (see Figure 4b for an example). The distance d of Figure 4a which measures the minimal spatial resolution step between consecutive movements are set to 3 pixels.

Two different gestures are recognized to simulate a click-left and click-right mouse button (Figure 5). Left button is clicked by closing the hand. When this gesture is performed the lower half of the window remains full of pixels classified as skin, whereas the upper half becomes empty (Figure 5b). On the other hand, right button is simulated by rotating the hand 90. In this gesture, left half of the window remains full of skin pixels, and the right half becomes empty (Figure 5c).

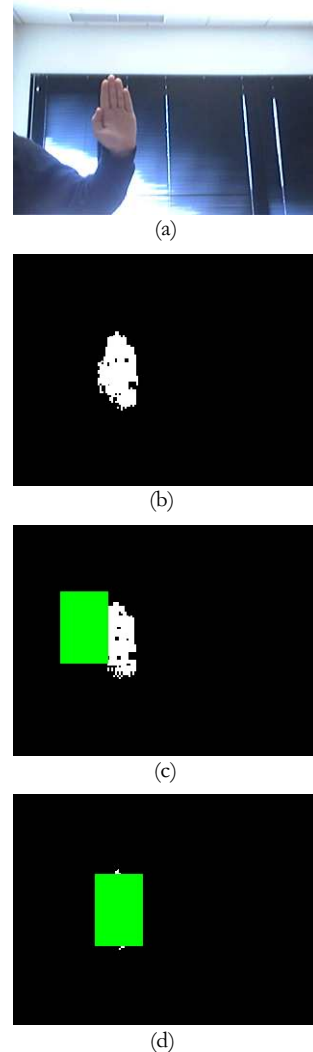


Figure 7. Static hand detection. (a) Input frame, (b) skin detection, (c) PF estimation (d) LSPF estimation.

Table 1. Standard deviation and range in different sequences (pixels).

	<i>Std. Dev (σ)</i>				<i>Range (r)</i>			
	PF		LSPF		PF		LSPF	
	σ_x	σ_y	σ_x	σ_y	r_x	r_y	r_x	r_y
Static	11.86	8.65	1.40	1.97	56	51	6	9
Horizontal	-	50.19	-	11.83	-	171	-	63
Vertical	10.21	-	4.55	-	53	-	24	-

5. Experimental Results

In order to evaluate and compare the quality of the algorithms, some video sequences have been tested. We developed a real time software implementation using a 2.8 GHz, 512 MB DDRAM Pentium 4 under Windows XP Professional SP2 equipped with DirectX 9.0, and image sequences with resolution of 320 x 240 pixels. The throughput obtained was 30 frames per second.

Microsoft DirectX [7] is a group of multimedia technologies designed for multimedia software developers. DirectShow is a DirectX component that takes charge of audio and video streams [11] in a highly modular way. It can be used to solve problems in Machine Vision and other kinds of Audio or Video Processing tasks, including the real-time processing of signals. In particular, we have implemented some DirectShow transform filters to include the functionality, such as skin detection and the particle filtering variants (PF, LSPF). It is important to avoid confusion between these implementation aspects (Directshow filters) with a classic filtering or the particle filter itself.

Human-Computer Interaction based on visual tracking is a very sensitive task. Presence of noise, illumination changes or any other artifact lead to a probable skin detection failure, which is the input of our tracking system. If the posterior process is not quite robust then accumulated errors will result in an unmanageable situation for a simple visual mouse controller.

Three experiments have been carried out to evaluate the performance of proposed Local Search Particle Filter (LSPF) that is, compared to a standard Particle Filter. The first experiment consists in a simple hand detection that remains almost static during 70 video frames (6a). The second and third experiments deal with a semi-constant motion of a human hand in horizontal and vertical directions, respectively (6b and 6c). All these experiments show some kind of sensitivity results of the algorithms. Table 1 reflects them by showing the standard deviation (σ) and range (r , in pixels) of the estimated positions in both coordinates (x , y) for the PF and the LSPF algorithms while tracking a human hand. For the horizontal movement, we have measured the σ_y and r_y , and viceversa, for the vertical movement. Note that PF data show an unacceptable performance in the σ and r values of the estimated location of the object being tracked. On the other hand, LSPF operates with very low deviations which result a smoother and accurate tracking.

Figure 6 shows the estimated trajectories for a (a) sta-

tic, (b) horizontal and (c) vertical movements. They clearly illustrate that the proposed LSPF provides a more accurate tracking quality than PF approach. Figure 7 shows some examples of the static hand detection. Figure 7c shows a failed estimation for the PF that is clearly avoided using a Local Search method 7d. Note the correspondence of the graph in Figure 6a with Figures 7c and 7d.

6. Conclusion

The main contribution of this work is the development of the Local Search Particle Filter algorithm (LSPF). LSPF hybridizes the local search procedure and the particle filter framework to solve tracking problems. Experimental results have shown that LSPF appreciably increases the efficiency of particle filtering, and improves the estimation quality and smoothing trajectories of a standard PF. In fact, the standard deviation between hand spatial positions for LSPF is reduced a 75% with respect to the PF algorithm. As a result, the algorithm performs accurate tracking in real-time on a standard computer. The proposed algorithm is the kernel of a vision based mouse system which allows to control and command the cursor of a computer using a standard webcam.

As future works, we propose the application of more avances heuristics to human-computer interfaces in order to recognize other gestures, and also track difficult body parts such as head or legs. The adaptation of these techniques can be very appropriate for the development of handicapped people technologies. In this work we have not make a robust skin detection under uncontrolled conditions, so this is also proposed as a future work.

References

- [1] *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, 2002.
- [2] *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, 2004.
- [3] A. Argyros and M. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *Proc. of the European Conference on Computer Vision (ECCV'04)*, Springer-Verlag, volume 3.
- [4] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, 50(2):174-188, 2002.

- [5] J. Buades, F. Perales, and J. Varona. Real time segmentation and tracking of face and hands in vr applications. In *Proceedings of the 2nd Workshop on Articulated Motion and Deformable Objects. LNCS 3179*, pages 259–268, 2004.
- [6] J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets, 1999.
- [7] D. M. D. Center. <http://msdn.microsoft.com/directx>.
- [8] P. Ehlert. Intelligent user interfaces: introduction and survey. Technical Report DK503-01.
- [9] J. MacLean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, and J. Tsotsos. Fast hand gesture recognition for real-time teleconferencing applications. In *Proc. of the 2nd Int Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, 2001.
- [10] K. J. S. F. Peer, P. Human skin colour clustering for face detection. In *Proceedings of the International Conference on Computer as a Tool EUROCON (2003)*, 2003.
- [11] M. D. Pesce. *Programming Microsoft DirectShow for Digital Video and Television*. Microsoft Press, 2003.
- [12] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR CIM-04-02.
- [13] P. Torma and C. Szepesvári. Ls-n-ips: An improvement of particle filters by means of local search. In *Proceedings of the Non-linear Control Systems (NOLCOS2001)*, 2001, 2001.
- [14] L. Wang, H. Weiming, and T. Tieniu. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.
- [15] G. Ye, J. Corso, D. Burschka, and G. D. Hager. Vics: A modular vision-based hci framework. In *Proceedings of 3rd International Conference on Computer Vision Systems(ICVS 2003)*, pages 257–267, 2003.
- [16] D. Zotkin, R. Duraiswami, and L. Davis. Joint audio-visual tracking using particle filters. *EURASIP journal on Applied Signal Processing*, 2002.