A complete system for head tracking using Motion-Based Particle Filter and Randomly Perturbed Active Contour

N. Bouaynaya and D. Schonfeld

Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA.

ABSTRACT

Many real world applications in computer and multimedia such as augmented reality and environmental imaging require an elastic accurate contour around a tracked object. In the first part of the paper we introduce a novel tracking algorithm that combines a motion estimation technique with the Bayesian Importance Sampling framework. We use Adaptive Block Matching (ABM) as the motion estimation technique. We construct the proposal density from the estimated motion vector. The resulting algorithm requires a small number of particles for efficient tracking. The tracking is adaptive to different categories of motion even with a poor a priori knowledge of the system dynamics. Particulary off-line learning is not needed. A parametric representation of the object is used for tracking purposes. In the second part of the paper, we refine the tracking output from a parametric sample to an elastic contour around the object. We use a 1D active contour model based on a dynamic programming scheme to refine the output of the tracker. To improve the convergence of the active contour, we perform the optimization over a set of randomly perturbed initial conditions. Our experiments are applied to head tracking. We report promising tracking results in complex environments.

Keywords: Video tracking, Particle filter, motion estimation, Adaptive Block Matching, active contour, snakes

1. INTRODUCTION

Many real world applications require accurate people tracking. The traditional applications include video surveillance,[1] autonomous vehicle navigation,[2] human computer interfaces, robot localization, etc. Recent advances in computer, multimedia and communication technologies have created opportunities in new applications such as environmental imaging,[3] wireless communication[4] and virtual reality.[5] These emerging new applications pose new challenges and create an increasing need for developing more efficient and accurate tracking techniques.

Recently Bayesian filtering framework has become very popular for object tracking. It provides a recursive formulation of the posterior probability density function in dynamical systems. Analytical solutions for the optimal Bayesian filtering problem are known only for special cases including the linear and gaussian case (Kalman filter[6]). Particle filters provide a general framework for estimating the probability density function of general non-linear and non-Gaussian systems. They are based on a Monte Carlo approach, where the density is represented by a set of random samples. Samples can be drawn from any distribution called the proposal density or the importance function, but sample weights should be properly adjusted so that the sample set fairly approximates the posterior density. Theoretically, if the number of samples is sufficiently large, the sample approximation of the posterior density can be made arbitrarily accurate.[7] In practice, the samples have to be properly placed and weighted to get a fair approximation of the posterior distribution. The pioneering work of CONDENSATION[8] uses the state transition prior as its proposal distribution. Because the state transition does not take into account the most recent observations, the particles drawn from transition prior may have very low likelihood, and their contributions to the posterior estimation become negligible. Various improvements and extensions have been proposed for visual tracking,[9] ,[10] .[11] To design better proposal distributions, the ICONDENSATION algorithm[9] uses a color tracker to generate the samples. This increases the sample set efficiency since it focuses the samples around the detected color blobs. However, this choice is ineffective in real world videos. The background clutter might have similar color properties as the object. Changing lighting conditions and/or object appearance causes the tracker to loose the object. In[11] an annealed Particle Filter has been proposed. It is based on probabilistic pruning and it focuses its particles around the global peaks of the weighting function. This method reduces the number of particles needed. But it is not a robust Bayesian framework anymore. By discarding inferior peaks, it may loose the true state.

In the first part of the paper, we propose to use a motion-based proposal. Many advantages arise from this choice. The algorithm is adaptive to different categories of motion. Particularly, it is able to handle unexpected motion that cannot be captured by the dynamic model. Offline-learning procedures are then no more required. For shape constraints during tracking, a parametric representation of the object is used. In the second part of the paper, we refine the tracking output for applications that require a non-parametric (elastic) contour around the object. Active contours have been extensively used for contour extraction,[12] ,[13],[14].[15] However, the active contour model suffers from its dependency on its internal parameters and initial condition. Many solutions have been proposed to make the active contour robust to its parameters,[16] .[17] We find that performing the optimization over different properly sampled initial conditions, gives good results to the post-tracking refinement problem.

The rest of the paper is organized as follows. In Section 2, we review the particle filtering framework and introduce the motion-based particle filter. In Section 3, we motivate the refinement problem, and propose a randomly perturbed active contour model for post-tracking contour refinement. In Section 4, we apply our proposed algorithm to head tracking on challenging real-world video sequences and report promising results. Concluding remarks and future work are given in Section 5.

2. A MOTION-BASED PARTICLE FILTER

In the pioneering work of CONDENSATION,[8] factored-sampling was used to formulate the Particle filter framework. Even though easy to follow, it obscures the role of proposal distributions. In this section, we present a new formulation of Particle filtering theory that is centered around proposal distributions. Then we show how to construct the motion- based proposal distribution. Finally, we summarize the complete algorithm.

2.1. Bayesian Sequential Importance Sampling

In Bayesian Sequential Estimation, a Markovian discrete-time state space model is assumed. Let X_k represent the target characteristics at discrete time k (position, velocity, shape, etc). The state space model is described by a state transition and measurement equations:

$$\begin{aligned} X_k &= f_k(X_{k-1}, v_k) \\ Z_k &= h_k(X_k, w_k), \end{aligned} \tag{1}$$

where $f_k : \Re^{n_x} \longrightarrow \Re^{n_x}$ and $h_k : \Re^{n_x} \longrightarrow \Re^{n_z}$ are possible non-linear functions. v_k and w_k are white noise sequences independent of past and current states. The statistics of both the noise are assumed to be known. Denote all past observations up to time k by $Z_{1:k} = \{z_0, ...z_k\}$. Assuming conditional independence of the observations given the states, the propagation rule is given by[8]

$$p(X_k|Z_{1:k}) \propto p(Z_k|X_k)p(X_k|Z_{1:k-1}), \tag{2}$$

where

$$p(X_k|Z_{1:k-1}) = \int_{X_{k-1}} p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1}). \tag{3}$$

The above propagation rule recursively computes the posterior at time k given the posterior at time $k-1$. When the functions f_k and h_k are linear and the noise is gaussian, Kalman filter provides an analytical closed form solution to (2) and (3). However, for real world applications, the clutter introduces multiple observations and a multi-model density is then necessary to model the observation density.

A non-parametric way to represent a distribution is to use discrete samples from the distribution. Unfortunately it is impossible to sample from the posterior, $p(X_k|Z_{1:k})$, since it is the density that we want to estimate. Instead, a set of samples $\{X^{(n)}, n = 1, \cdots, N\}$ can be drawn from another distribution, q, but sample weights should be properly adjusted. This is the basic idea of the importance sampling technique, which is a powerful tool in statistics.[18] The weights $\omega^{(n)}$ are given by[19]

$$\pi^{(n)} = \frac{p(X^{(n)})}{q(X^{(n)})}, \tag{4}$$

where p represents the posterior density evaluated at sample X^n. Hence the problem of estimating the posterior is converted into an easy weighting problem from any proposal density q. The posterior p is then approximated by the discrete distribution supported on the $X^{(n)}$ with weights $\pi^{(n)}$

$$\hat{p} = \sum_{n=1}^{N} \pi^{(n)} \delta_{X^{(n)}}(dX). \tag{5}$$

The approximation in (5) converges in distribution when the number of particles $^*N$ goes to infinity.[7] Practically, we cannot have an infinite number of samples and hence the choice of the proposal density is crucial to the accuracy of the estimation.

Going back to the notations formulated for the tracking problem stated in equations (2) and (3), the unnormalized weights are given by[9]

$$\omega_k^{(n)} = \frac{\sum_{j=1}^{N} \omega_{k-1}^{(j)} p(X_k^{(n)}|X_{k-1}^{(j)})}{q(X_k^{(n)})} p(Z_k|X_k^{(n)}), \tag{6}$$

where the subscript k denotes the time index and the superscript n denotes the sample index. The normalized weights or importance weights are given by

$$\pi^{(n)} = \frac{\omega^{(n)}}{\sum_{j=1}^{N} \omega^{(j)}}. \tag{7}$$

Given a discrete approximation to the posterior distribution, one can then proceed to filtered point estimate such as the mean of the state

$$\hat{X}_k = \sum_{i=1}^{N} \pi_k^{(i)} X_k^{(i)}. \tag{8}$$

In theory, there are infinite number of choices of the importance function q, as long as its support includes that of the posterior distribution. But of course when q is close to the true posterior, the particles are more effective. The idea is then to put more particles in areas where posterior may have higher density to avoid useless particles that lead to more expensive computation.

2.2. A motion based Importance function

We generate the importance function based on motion cues. Several advantages emerge of this choice of importance function. First,we economize on processing time by searching only the state space around the estimated position. Second, the tracker is adaptive to all kinds of motion even with a poor a priori knowledge of the object's dynamics. The online motion estimation of the object does not make any assumption about the underlying motion of the object. Particulary it can be viewed as a general case of mixture model CONDENSATION and switching state space models technique,[20] .[21] Any motion estimation technique can actually be used in this algorithm. We choose Adaptive Block Matching because of its simplicity in implementation. We briefly review the Adaptive Block Matching technique.[22]

---

*We will be using interchangeably particles and samples to denote draws from a distribution.

2.3. Adaptive Block Matching

Assume an estimate of the object at time $k-1$ is computed, i.e., $\hat{X}_{k-1}$ is known. To compute the motion at time k, divide frame $k-1$ into 16×16 blocks. The blocks that are completely lying inside $\hat{X}_{k-1}$ are labelled seed blocks. The blocks that are at the boundary $\hat{X}_{k-1}$-background are labelled uncertain blocks. Uncertain blocks are further divided into seed and uncertain blocks. The procedure is carried till a fixed size block (4×4) is reached. Only the motion of the seed blocks is computed. We use extensive search in window to estimate the motion vector of each block. The window in frame k is centered around the position given by the previous estimate, i.e., $\hat{X}_{k-1}$. The size of the window determines then the maximum motion allowed. We use a 32×32 size window. The matching criterion is the absolute sum of the intensity differences.

The ABM output is a mask of the object (union of rectangular blocks) at time k. The human head is very well modelled by a parametric ellipse.[23] This domain knowledge helps avoid erroneous shape evolvement therefore greatly improving the tracking results. We use a parametric ellipse to represent the state vector of the head:

$$X = [x_c, y_c, b, \phi]^T, \tag{9}$$

where (x_c, y_c) is the center of the ellipse, b is the major axis of the ellipse, and ϕ is the orientation of the ellipse in radians. The ratio of the major and minor axis of the ellipse is held constant equal to its value computed in the first frame. This is a very reasonable assumption that allows us to reduce the dimensionality of the state vector by 1. We use a Least Mean Square (LMS) technique to fit the ellipse to the mask output of the ABM. The motion vector that interests us is then given by the difference between the position of the center of the newly fitted ellipse and the position of the center of the previous mean state ellipse.

$$\Delta X_k = [x_c(k) - x_c(k-1), y_c(k) - y_c(k-1), 0, 0], \tag{10}$$

where $(x_c(k), y_c(k))$ is the center at time index k.
We then have the following sampling scheme

$$X_k^{(n)} = X_{k-1}^{(n)} + \Delta X_k + v_k^{(n)}, \tag{11}$$

and

$$v_k \sim N(0, \Sigma_G). \tag{12}$$

Each sample is then translated in the (x-y) direction by the amount estimated by the ABM and diffused in the four coordinates according to a zero-mean white gaussian noise. The importance function is then a sum of Gaussian

$$q(X_k|X_{k-1}, Z_k) \equiv \sum_{i=1}^{N} N_{X_k}(X_{k-1}^{(i)} + \Delta X_k, \Sigma_G), \tag{13}$$

where we made use of the following notation that we will be also using for the rest of this paper

$$N_X(\mu, \Sigma) \equiv \frac{1}{2\pi|\Sigma|} exp(-(X-\mu)^T \Sigma^{-1} (X-\mu)). \tag{14}$$

In addition to the importance function evaluation, we also need to calculate the particle likelihood $p(Z_k|X_k)$ and transition probability $p(X_k|X_{k-1})$. We discuss these two terms in the following subsections.

2.4. Calculating the likelihood

We use both face color characteristics and edge detection cues for particle weighting. Let $\nu_{edge}^{(n)}$ and $\nu_{color}^{(n)}$ denote the likelihood based on edge detection and color respectively of the n^{th} sample. Assuming that both observations are independent, the total weight of the sample is given by the product of both color and edge-based weights

$$\nu_{total}^{(n)} = p(Z_k|X_k^{(n)}) = \nu_{edge}^{(n)} \nu_{color}^{(n)}. \tag{15}$$

2.4.1. Gradient Module

We use the model developed in.[8] The observations are collected on normal lines to the contour. Let $Z_{k,\varphi}$ denote the edge detection observations on line φ at time k; $Z_{k,\varphi} = \{Z_1, Z_2, ..., Z_I\}$. At most one of the I edges is the true contour point. Let p_0 be the prior probability that none of the I edges is the true contour edge. With the assumption that the clutter is a Poisson process along the line with spatial density γ and the true target measurement is normally distributed with standard deviation σ_z, we obtain the edge likelihood model as follows:

$$p(Z_{k,\varphi}|\lambda_\varphi) \propto \frac{1}{\sqrt{2\pi}\sigma_z p_0 \gamma} exp(-\frac{(min_j(Z_j - \lambda_\varphi))^2}{2\sigma_z^2}), \tag{16}$$

where λ_φ represents the pixel along the normal line φ belonging to the sample ellipse.
By assuming independence between the different L normal lines, we have the following overall likelihood function for sample n

$$\nu_{edge}^{(n)} = \prod_{\varphi=1}^{L} p(Z_{k,\varphi}|\lambda_\varphi) \tag{17}$$

2.4.2. Color Module

We use the color model proposed in.[24] The color histogram is a simple and well suited system for capturing the multimodal patterns of head color. Let $q_k = \{q_k^u\}_{u=1,...,m}$ denote the particle histogram and $p_k = \{p_k^u\}_{u=1,...,m}$ the model histogram at time k. A popular measure between two distributions p and q is the Bhattacharyya coefficient

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p(u)q(u)}du. \tag{18}$$

The larger ρ is the more similar the two distributions are. As a distance between two distributions, we define the measure

$$d = \sqrt{1 - \rho[p, q]}. \tag{19}$$

Small distances correspond to large weights

$$\nu_{color}^{(n)} = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{d^2}{2\sigma_c^2}}, \tag{20}$$

that are specified by a gaussian with variance σ_c.

2.5. System Dynamics

We adopt a simple 1^{st} order auto-regressive (AR1) process for the system dynamics.

$$X_k = AX_{k-1} + Bw_k, \tag{21}$$

where w_k is a zero-mean, white gaussian noise

$$w_k \sim N(0, \Sigma_P). \tag{22}$$

A and B are matrices representing the deterministic and stochastic components of the prediction model.
Using equation (21) we have the transition probability

$$p(X_k = X_k^{(n)}|X_{k-1} = X_{k-1}^{(i)}) \equiv N_{X_k^{(n)}}(X_{k-1}^{(i)}, \Sigma_P). \tag{23}$$

Without loss of generality, we can set in equation (21) $B = I$ and for simplicity we assume $A = I$ so that we end up with a random walk model. We use on purpose this very simple transition model for different video sequences presenting different types of motion to show that even with a poor prior knowledge of the object's dynamics, our proposed tracking algorithm is still accurate because we are sampling from the proposal density q defined in equation (13) and not from the prior

2.6. The complete algorithm

The motion-based Particle filter algorithm is summarized as follows:

1. **ABM Computation**
   From the previously estimated mean state vector $\hat{X}_{k-1}$, run the Adaptive Block Matching module. Compute the motion vector ΔX_k. Before drawing the new sample set, resample.

2. **Resampling step**
   In Importance Sampling framework, the variance of the weights increases over time. This is known as the degeneracy problem. To limit the degeneracy of the algorithm, we resample from the old sample set $\{X_{k-1}^{(i)}, \pi_{k-1}^{(i)}, i = 1, \cdots, N\}$ choosing a sample according to its probability. For more literature on the degeneracy problem, refer to,[7] .[25]

3. **Importance Sampling**
   Use the sampling scheme in equation (11) to draw N samples at time k.

4. **Weighting**
   Weight the samples according to equation (6). Normalize the weights so that they sum up to 1.

5. **Output** The estimated state vector at time k is then given by the weighted average of all the particles at time k.

3. REFINEMENT PROCESS

3.1. Motivation

The ellipse can be a very satisfactory representation of the head if the application targeted is video surveillance for example. However, if we are targeting applications such as interactive imaging, where we need to extract the person's head and superimpose it on a different background, or virtual reality environments, where we need to set up conference calls around an imaginary conference table, then the ellipse is no more a satisfactory shape. We actually need an elastic contour around the person's head. Active contours or snakes have been proven to be a promising approach in contour extraction. First proposed by Kass et al.,[12] they take an initial estimate of the object and refine this estimate using an optimization procedure. The final extracted contour is highly dependent on the position and shape of the initial contour as a consequence of many local minima in the energy function,[16] .[17] To improve the convergence of the active contour to the desired object, we perform the optimization procedure using different properly sampled initial conditions. To enable dynamic programming scheme, we adopt the causal model proposed by Rui and al. in.[26]

3.2. Energy minimization

We adopt the same contour measurement in section 2.4.1. Only observations along the normal lines of the contour are detected. Let $\phi, \phi = 1, \cdots, L$, be the index of normal lines and $\lambda_\phi, \lambda_\phi = -N, \cdots, N$, be the index of pixels along the normal line ϕ. $I(\lambda_\phi), IR(\lambda_\phi), IG(\lambda_\phi), IB(\lambda_\phi)$ are respectively the image intensity and the RGB values at pixel index λ_ϕ. The 2D contour is then represented by the set of its L normal lines .

3.2.1. External Energy terms

The external forces make the contour move towards the image features such as edges. We therefore choose intensity and color gradients as the external energy terms.

$$\begin{aligned} E_{ext}(\lambda_\phi) = -\{&|I(\lambda_\phi+1) - I(\lambda_\phi)|^2 + |IR(\lambda_\phi+1) - IR(\lambda_\phi)|^2 + \\ &|IG(\lambda_\phi+1) - IG(\lambda_\phi)|^2 + |IB(\lambda_\phi+1) - IB(\lambda_\phi)|^2\}. \end{aligned} \tag{24}$$

3.2.2. Internal Energy terms

The internal energy terms impose smoothness on the contour. In the traditional active contour model, the roughness is defined as the first derivative of the contour. However this is a non causal definition that leads to an iterative procedure for solving for the minimum. A slight modification to the smoothness definition leads to a causal representation[26]

$$E_{int}(\lambda_\phi, \lambda_{\phi-1}) = |\lambda_\phi - \lambda_{\phi-1}|^2. \tag{25}$$

3.2.3. Shape Energy term

Taking into account a shape information will avoid the snake to lock onto unwanted background features despite of the smoothness energy terms. Assuming that the initial contour is accurate enough around the head, we can define the shape energy term as

$$E_{shape}(\lambda_\phi) = \frac{\lambda_\phi^2}{\sigma_s^2}, \tag{26}$$

where σ_s specifies how accurate is the initial contour.

3.2.4. Solving the optimization problem using dynamic programming

The total energy is then

$$E_{total}(c(\phi)) = \sum_{\phi=1}^{L} \alpha_e E_{ext}(c(\phi)) + \alpha_i E_{int}(c(\phi), c(\phi-1)) + \alpha_s E_{shape}(c(\phi)). \tag{27}$$

The different weights α_e, α_i and α_s balance the different energy terms. We want to minimize the total energy. Following a dynamic programming procedure, the causal definition of all the energy terms allows us to propagate the energy from one pixel in a normal line to another in the next normal line in the following manner

$$\begin{aligned} &E_{total}(\lambda_\phi) = min_{\lambda_{\phi-1}}\{E_{total}(\lambda_{\phi-1}) + \alpha_i E_{int}(\lambda_\phi, \lambda_{\phi-1}\}+ \\ &\alpha_e E_{ext}(\lambda_\phi) + \alpha_s E_{shape}(\lambda_\phi). \end{aligned} \tag{28}$$

We record the best previous state from state λ_ϕ

$$\zeta(\lambda_\phi) = \underset{\lambda_{\phi-1}}{argmin}\{E_{total}(\lambda_{\phi-1}) + \alpha_i E_{int}(\lambda_\phi, \lambda_{\phi-1}\}. \tag{29}$$

When the energy is propagated to the L normal lines, the optimal state sequence is obtained by backtracking the sequence ζ. The computational cost of this dynamic programming procedure is $(2N+1)^2L$.

3.2.5. Randomly perturbed Active contour model

To improve the performance of the active contour, we propose to perform the optimization procedure using properly sampled initial conditions.

3.2.6. Algorithm

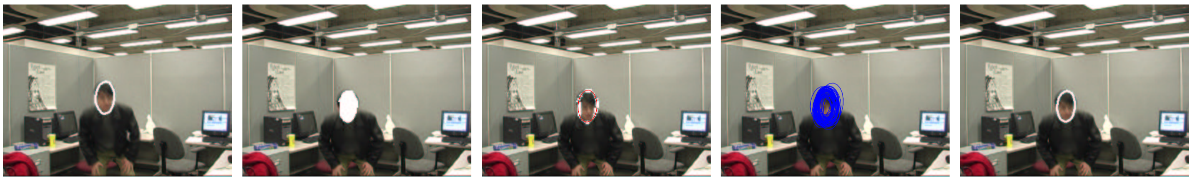
1. Given the estimated ellipse $\hat{X}_k$ (output of the tracker), we draw samples that are normally distributed with mean $\hat{X}_k$
$$Y_k^{(n)} = \hat{X}_k + u_k^{(n)}, n = 1, \cdots, M \tag{30}$$
where $u_k^{(n)}$ is a zero-mean, normally distributed white noise and M is the total number of initial conditions.
2. Do dynamic programming for each sample $Y_k^{(n)}$. Compute the total energy of each output contour.
3. The optimal contour is the one that corresponds to the minimum total energy.

4. EXPERIMENTS

Our experiments are applied to head tracking. We use challenging video sequences with more than 400 frames in cluttered environments. We simulate various tracking conditions, including camera zoom in and out, appearance changes, fast and erratic movements, out-of-plane head rotation.

The motion-based particle filter enables us to deal with severe distractions. 50 particles are propagated during the tracking. All the parameters of the ellipse are allowed to change. We compare our algorithm with the CONDENSATION filter. Figures 3 and 4 show that the CONDENSATION filter fails to catch up with the object's fast and erratic movements.

For the refinement process, we use 20 normal lines along the ellipse contour.i.e; $L = 20$. Each line is 21 pixels long, i.e; $N = 10$; and $M = 10$ samples for the initial contours of the refinement process. In order to show the robustness of the perturbed active contour model, we use a video where we provided a deviated initial contour to the tracker. The traditional active contour model is easily distracted by unwanted background features. The Randomly Perturbed active contour outputs an elastic curve that perfectly fits the head outlines for all the frames of the video.



(a) (b) (c) (d) (e)

Figure 1. Motion-Based Particle Filter (a) Estimate of the tracker at frame $k-1$ (b) Mask output of the ABM module (c) An ellipse is fitted to the mask using LMS criterion (d) samples are drawn with mean the previously fitted ellipse and fixed variances. Notice that the four parameters of the ellipse are allowed to change (e) Output of the tracker at frame k.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented a motion-based particle filter for head tracking followed by 1D causal active contour model for post-tracking refinement. We showed that our algorithm can handle any motion, and particulary fast and erratic motion, with very few samples. The framework we propose can be generalized with any motion estimation technique. Post-tracking refinement process is important for applications that require a pixel-based accurate elastic contour delineating a tracked object. A 1D causal active contour model enables a dynamic programming scheme and finds the best local contour in 1 iteration. To improve the performance of the active contour in a video sequence, we perform the optimization over different properly sampled initial conditions. This simple optimization technique turns out to be very efficient for post-tracking contour refinement.

Handling zooming

Figure 2. Motion-Based Particle Filter handles camera zoom because the minor axis of the ellipse is part of the state space representation of the head and it is allowed to vary.

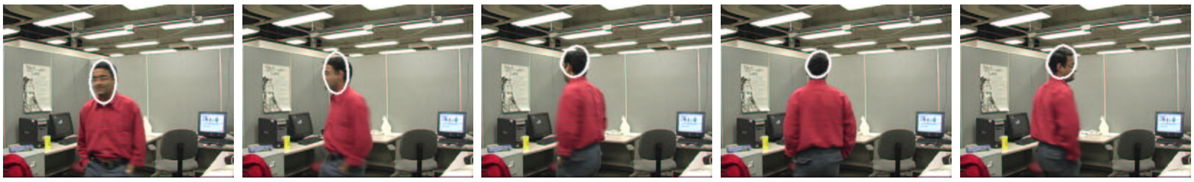
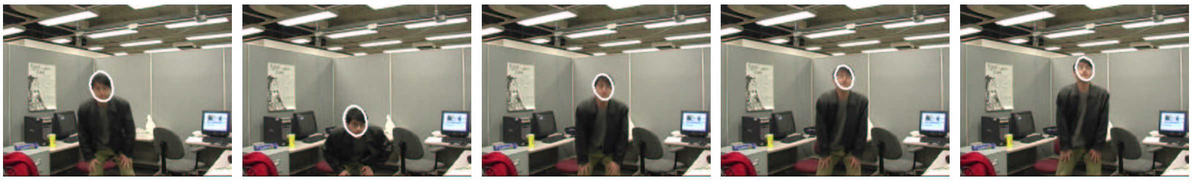
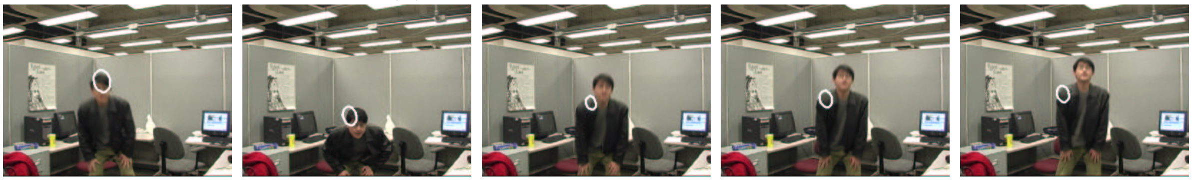


Figure 3. Motion-based Particle filter handles simultaneously fast motion and out-of-plane rotation of the head.



(a) Motion-based Particle filter



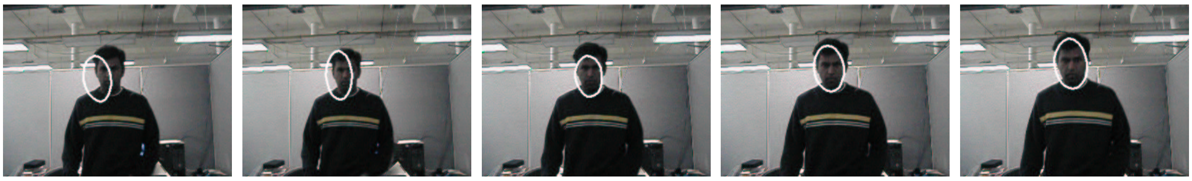
(b) CONDENSATION filter

Figure 4. Tracking a sudden movement. The person in this video walks slowly and then suddenly bends. The CONDENSATION filter is unable to catch up with the person's erratic movements. Motion-Based Particle Filter however successfully tracks the person's head.

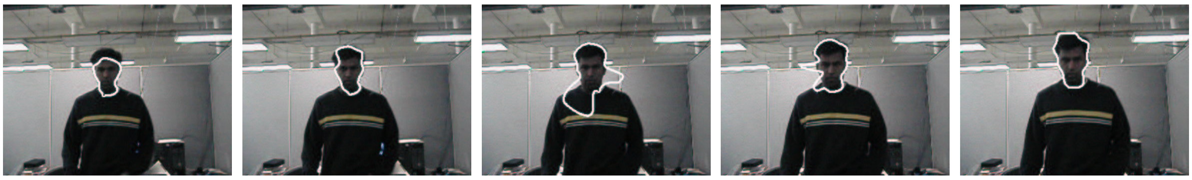
Further research would incorporate the refinement results back into the tracking module for even more accurate and efficient tracking results.

REFERENCES

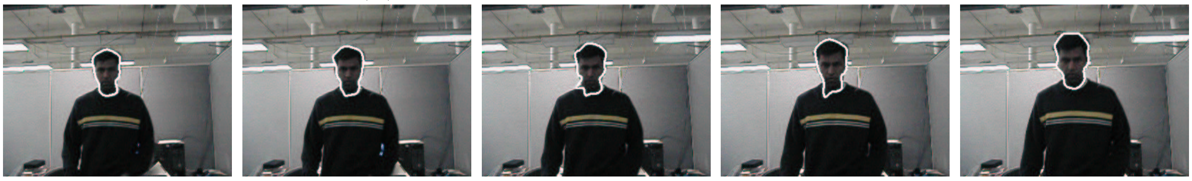
1. G. Sullivan, "Visual interpretation of known objects in constrained scenes," *Phil. Trans. R. Soc. Lond.* , pp. 109–118, 1992.
2. E. Dickmanns and V. Graefe, "Applications of dynamic monocular machine vision," *Machine vision and Applications* **1**, pp. 241–261, 1988.
3. C. Papin, P. Bouthemy, and G. Rochard, "Unsupervised segmentation of low clouds from infrared meteosat images based on a contextual spatio-temporal labeling approach," *IEEE Trans. Geosci. Remote Sensing.* **40**, p. 104114, January 2002.
4. W. Vieux, K. Schwerdt, and J. Crowley, "Face-tracking and coding for video compression," *Proc. Int. Conf. Computer Vision Systems* , p. 151160., 1999.
5. P.Rander, P.J.Narayanan, and T.Kanade, "Virtualized reality: Constructing time-varying virtual worlds from real world events," *Proceedings of IEEE Visualization* , pp. 277–283, October 1997.
6. B.Anderson and J. Moore, *Optimal Filtering*, Englewood Cliffs, Prentice Hall, New Jersy, 1979.
7. A. Doucet, "On sequential simulation-based methods for bayesian filtering," *Technical report CUED/F-INFENG/TR.310* , 1998.
8. M. Isard and A. Blake, "Condensation conditional density propagation for visual tracking," *Int. J. Computer Vision* **29**(1), pp. 5–28, 1998.
9. M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," *In Proc. European Conf. on Computer Vision* , p. 767781, 1998.
10. J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking.," *Proc. ECCV* , 2000.



(a) An original video sequence where the output ellipse has been slightly deviated on purpose



(b)Traditional causal active contour



(c) Randomly perturbed active contour

11. J. Deuttscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," *Proc. CVPR* , 2000.
12. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int.J. Comput. Vision* **1**(4), pp. 321–331, 1988.
13. A. Blake and A. Yuille, *Active vision*, MIT Press, Cambridge, Massachusetts, London, England, 1992.
14. A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(9), pp. 855–867, 1990.
15. Y. Chen, Y. Rui, and T. Huang, "Optimal radial contour tracking by dynamic programming," *Proc. of IEEE ICIP 2001 , Thessaloniki, Greece* , October 2001.
16. S. Gunn and M. Nixon, "Improving snake performance via a dual active contour," *Proceedings of Computer Analysis of Images and Patterns* **970**, pp. 600–605, 1991.
17. G. Xu, E. Segawa, and S. Tsuji, "A robust active contour model with insensitive parameters," *Pattern Recognition* **27**(7), pp. 879–884, 1994.
18. M. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.*, Springer-Verlag, New York, 1993.
19. A. Smith and A. Gelfand, "Bayesian statistics without tears: A sampling - resampling perspective," *The American Statistician* **46**, May 1992.
20. M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," *Proc 6th Int. Conf. Computer Vision* , 1998.
21. Z. Ghahramani and G. Hinton, "Switching state-space models," 1996.
22. K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia* , to appear.
23. S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," *In Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recog.* , p. 232237, 1998.
24. K. Nummiaro, E. Koller-Meier, and L. V. Gool, "Object tracking with an adaptive color-based particle filter," *DAGM-Symposium Pattern Recognition* , p. 353360, 2002.
25. J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEEE Proc. Radar, Sonar Navig.* **146**, February 1999.
26. Y. Chen, Y. Rui, and T. Huang, "Mode-based multi-hypothesis head tracking using parametric contours," *Proc. Automatic Face and Gesture Recognition* , May 2002.