A BOOSTED ADAPTIVE PARTICLE FILTER FOR FACE DETECTION AND TRACKING

Wenlong Zheng and *Suchendra M. Bhandarkar*

Department of Computer Science, The University of Georgia, Athens, Georgia 30602-7404, USA
E-mail: {zheng, suchi}@cs.uga.edu

ABSTRACT

A novel algorithm, termed a Boosted Adaptive Particle Filter (BAPF), for integrated face detection and face tracking is proposed. The proposed algorithm is based on the synthesis of an adaptive particle filtering algorithm and an AdaBoost face detection algorithm. A novel Adaptive Particle Filter (APF), based on a new sampling technique, is proposed to obtain accurate estimates of the proposal distribution and the posterior distribution to enable accurate tracking in video sequences. The AdaBoost algorithm is used to detect faces in input image frames, while the APF algorithm is designed to track faces in video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in video sequences. Experimental results show that the proposed BAPF algorithm provides a means for robust face detection and accurate face tracking under various tracking scenarios.

Index Terms**—** Particle filter, video tracking, face detection, image analysis

1. INTRODUCTION

Face detection methods based on machine learning and statistical estimation have recently demonstrated excellent results amongst all existing face detection methods. Viola and Jones propose a robust AdaBoost face detection algorithm to detect faces in a rapid and robust manner with a high detection rate [1]. Object tracking has been studied extensively because of various vision applications that use tracking algorithms. Particle filters have been widely used in object tracking to address limitations arising from non-linearity and non-normality of the motion model [2]. The basic idea of the particle filter is to approximate the posterior density using a recursive Bayesian filter based on a set of particles with assigned weights. The CONDENSATION algorithm uses a simple proposal distribution to draw upon a set of particles [3], which defines the conditional distribution on the particle state in the previous frame. The proposal distribution typically does not make use of the information from the current frame. Li et al. [2] propose a Kalman particle filter (KPF) and an unscented particle filter (UPF) to improve the particle sampling in the context of visual contour tracking. This approach makes use of a Kalman filter or an unscented Kalman filter to incorporate the current observation in the proposal distribution. The Kalman filter or the unscented Kalman filter can steer the set of particles to regions of high likelihood in the search space, and thus reduce the number of particles needed. Okuma *et al*. [4] propose a boosted particle filter for object tracking, which interleaves the AdaBoost algorithm with the CONDENSATION algorithm in the proposal distribution estimation stage. However, their technique does not present a systematic method for achieving the combination that would result in the desired proposal distribution. This paper uses an improved particle filter and combines it with AdaBoost in the final stage. Wang *et al*. [5] propose a likelihood estimation technique for the particle filter based on Gentle AdaBoost. Hansen *et al.* [6] propose a log-likelihood ratio function incorporated within a particle filter to track the motion of the human eye. However, both the above methods focus on improvement of the likelihood estimation instead of improvement of the proposal distribution estimation which is the focus of this paper. In this paper, we propose an APF to enable a more accurate estimation of the proposal distribution and of the posterior distribution. We also propose a BAPF for face detection and tracking by combining the APF algorithm with the AdaBoost algorithm.

2. PROPOSED ADAPTIVE PARTICLE FILTER

2.1 The Filtering Distribution

The standard problem of object tracking is to estimate the state $\mathbf{x}_t$ of the objects at time *t*, using a set of observations $\mathbf{y}_t$ from a sequence of input images. We assume that object dynamics form a temporal Markov process and observations $\mathbf{y}_t$ are independent. The dynamics are determined by a transition prior $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. Given the transition prior $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and the observation probability $p(\mathbf{y}_t \mid \mathbf{x}_t)$, the posterior probability $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ can be computed recursively via Bayesian filtering [3] [5]:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t \mid \mathbf{x}_t)\int p(\mathbf{x}_t \mid \mathbf{x}_{t-1})p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}d\mathbf{x}_t} \quad (1)$$

Since the solution to Eq. (1) entails the computation of high-dimensional integrals, and dealing with the non-linearity and non-normality of the motion model under many tracking scenarios, a particle filter is adopted as a practical scheme to estimate the posterior probability given by Eq. (1).

2.2 The Standard Particle Filter

A standard particle filter uses N weighted discrete particles to approximate the posterior probability $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ via observation of the data. Each particle consists of a state vector $\mathbf{x}$ and a weight w. The weighted particle set is given by $\left\{\left(\mathbf{x}_t^{(i)}, w_t^{(i)}\right), i=1,2,\cdots,N\right\}$. Since it is practically infeasible to draw samples directly from the posterior distribution, a proposal distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{1:t})$ is used instead to draw the samples for approximation of the posterior probability. A particle filter samples $\mathbf{x}_t^{(i)}$ from $\mathbf{x}_{t-1}^{(i)}$ for particle i $(i=1,2,\cdots,N)$ and computes the weight for $\mathbf{x}_t^{(i)}$ using the following equation:

$$w_t^{(i)} = \frac{p\left(\mathbf{y}_t \mid \mathbf{x}_t^{(i)}\right) p\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right)}{q\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:t}\right)} w_{t-1}^{(i)} \quad (2)$$

The posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ is approximated as:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right) \quad (3)$$

The estimate of the function $f(\mathbf{x}_t)$ of the state vector $\mathbf{x}_t$ is computed as:

$$E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f\left(\mathbf{x}_t^{(i)}\right) \quad (4)$$

2.3 Proposed Adaptive Particle Filter

An Adaptive Particle Filter (APF) is proposed to enable more accurate estimation of the proposal distribution and the posterior distribution. In the sampling step of the APF algorithm, a new sampling strategy is used to improve the accuracy of the approximation. For each discrete particle $\mathbf{x}_{t,l-1}^{(i)}$, the APF generates a new particle $\mathbf{x}_{t,l}^{(i)}$ based on a proposal distribution $u_l(\mathbf{x})$. We use the loop controlled by the parameter l in the APF algorithm described below to implement the new sampling technique. L is the fixed number of iterations of loop l. L can be adjusted in different real applications. When L=1, the APF is equivalent to the pure standard particle filter. When L>1, the APF performs more sampling iterations than the standard particle filter. In order to enable more accurate estimation of the proposal distribution, we iterate the sampling procedure with a constraint, which is termed the Adaptive Learning Constraint (ALC). The ALC is described in the following analysis. The APF algorithm is summarized as follows.

1. **Initialization:** Initialize a set of particles from the prior $p(\mathbf{x}_0)$ to get $\left\{\left(\mathbf{x}_0^{(i)}, w_0^{(i)}\right), i=1,2,\cdots,N\right\}$. Let *t=0*.
2. **Sampling step**
   (1) For $l = 1, 2, \ldots, L$
      (a) For $i = 1, 2, \ldots, N$
         Sample $\mathbf{x}_{t,l}^{(i)}$ from $\mathbf{x}_{t,l-1}^{(i)}$ based on the proposal distribution $u_l(\mathbf{x}) = q\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:k}\right)$. Construct $p_{l-1}(\mathbf{x}) = \sum_{i=1}^{N} w_{t,l-1}^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_{t,l-1}^{(i)}\right)$, where $\delta(\cdot)$ is the Dirac function.
      (b) If the ALC is satisfied, $K_l \cdot max_l \le \alpha \cdot K_{l-1} \cdot min_{l-1}$ (variables are detailed in the ALC derivation):
         (i) Compute the weights of particles $w_{t,l}^{(i)} = p\left(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}\right) p\left(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}\right) w_{t,l-1}^{(i)}$,
         $w_{t,0}^{(i)} = w_{t-1}^{(i)}$, when l = 1.
         (ii) Normalize $w_{t,l}^{(i)} = \dfrac{w_{t,l}^{(i)}}{\sum_{i=1}^{N} w_{t,l}^{(i)}}$,
         (iii) Continue the loop l
      (c) If the ALC is not satisfied, $K_l \cdot max_l > \alpha \cdot K_{l-1} \cdot min_{l-1}$:
         i) Let $w_t^{(i)} = w_{t,l-1}^{(i)}$, $\mathbf{x}_t^{(i)} = \mathbf{x}_{t,l-1}^{(i)}$,
         ii) Break the loop l
   (2) Let $w_t^{(i)} = w_{t,L}^{(i)}$, $\mathbf{x}_t^{(i)} = \mathbf{x}_{t,L}^{(i)}$, $i = 1, 2, \ldots, N$.
   (3) $w_t^{(i)} = \dfrac{w_t^{(i)}}{q\left(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:k}\right)}$, $i = 1, 2, \ldots, N$.
3. **Estimation step**
   Obtain a set of particles $\left\{\left(\mathbf{x}_t^{(i)}, w_t^{(i)}\right), i=1,2,\cdots,N\right\}$. The posterior distribution can be approximated using the set of particles: $p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right)$. The estimated value of $f(\mathbf{x}_t)$ can be computed as:
   $E[f(\mathbf{x}_t)] \approx \sum_{i=1}^{N} w_t^{(i)} f\left(\mathbf{x}_t^{(i)}\right)$.
4. **Selection step**
   Resample particles $\mathbf{x}_t^{(i)}$ with probability $w_t^{(i)}$ to obtain N i.i.d random particles $\mathbf{x}_t^{(i)}$, approximately distributed as posterior distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$. Assign $w_t^{(i)} = \dfrac{1}{N}$.
5. **Iterative Step:** Set t=t+1, and go to step 2.

A critical step in the APF is obtaining a good approximation to the sampling proposal distribution $u_l(\mathbf{x})$ in the sampling step. The purpose behind choosing the proposal distribution $u_l(\mathbf{x})$ recursively in a given state is to reduce the estimation error. In the following analysis, we prove that the iterations of loop l result in the convergence of the estimate of the proposal distribution. The propagation of errors between the iterations in the APF algorithm can be analyzed for a single iteration l: $l \in \{1,2,\cdots,L\}$. The sampling error at iteration l with respect to $f(\mathbf{x})$ is computed as [7]:

$$E[f(\mathbf{x}), p_l(\mathbf{x})] = \left| \frac{\sum_{i=1}^{N} \int f(\mathbf{x}_t) p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} d\mathbf{x}_t} - \frac{\sum_{i=1}^{N} f(\mathbf{x}_{t,l}^{(i)}) p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \right| \quad (5)$$

Using the Lagrange theorem, we could obtain specific values $\xi_1^{(i)}$ and $\xi_2^{(i)}$ in domain D such that:

$$E[f(\mathbf{x}), p_l(\mathbf{x})] = \sum_{i=1}^{N} \left| \frac{f(\xi_1^{(i)}) p(\mathbf{y}_t \mid \xi_1^{(i)}) p(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \xi_2^{(i)}) p(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)}} - \frac{f(\mathbf{x}_{t,l}^{(i)}) p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}}{\sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)}} \right| \quad (6)$$

Assuming that $f(\mathbf{x})$, $p(\mathbf{y}_t \mid \mathbf{x}_t)$, $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$ are continuous functions on domain D, we have the following equation:

$$\exists m_1, M_1 \in R,$$

$$m_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \le \left| f(\xi_1^{(i)}) p(\mathbf{y}_t \mid \xi_1^{(i)}) p(\xi_1^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} - f(\mathbf{x}_{t,l}^{(i)}) p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)} \right| \le M_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \quad (7)$$

Likewise, we have:

$$\exists m_2, M_2 \in R$$

$$m_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \le \left| \sum_{i=1}^{N} p(\mathbf{y}_t \mid \xi_2^{(i)}) p(\xi_2^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t-1}^{(i)} - \sum_{i=1}^{N} p(\mathbf{y}_t \mid \mathbf{x}_{t,l}^{(i)}) p(\mathbf{x}_{t,l}^{(i)} \mid \mathbf{x}_{t-1}^{(i)}) w_{t,l-1}^{(i)} \right| \le M_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \quad (8)$$

Since $f(\mathbf{x})$, $p(\mathbf{y}_t \mid \mathbf{x}_t)$, $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$ are continuous functions defined on domain D, Eq. (6) is bounded by two specific values, max_l and min_l.

$$max_l = \max_{1 \le i \le N} \left\{ M_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right|, M_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \right\} \quad (9)$$

$$min_l = \min_{1 \le i \le N} \left\{ m_1 \left| \xi_1^{(i)} - \mathbf{x}_{t,l}^{(i)} \right|, m_2 \left| \xi_2^{(i)} - \mathbf{x}_{t,l}^{(i)} \right| \right\} \quad (10)$$

Hence, we obtain:

$$K_l \cdot min_l \le E(f(X), p_l(X)) \le K_l \cdot max_l \quad (11)$$

where K_l is a constant corresponding to the loop iteration l.

Likewise, for the loop iteration l-1 we obtain:

$$K_{l-1} \cdot min_{l-1} \le E(f(\mathbf{x}), p_{l-1}(\mathbf{x})) \le K_{l-1} \cdot max_{l-1} \quad (12)$$

Let

$$K_l \cdot max_l \le \alpha \cdot K_{l-1} \cdot min_{l-1}, \text{ where } 0 < \alpha < 1 \quad (13)$$

Thus we obtain:

$$E(f(X), p_l(X)) \le \alpha \cdot E(f(X), p_{l-1}(X)) \quad (14)$$

If Eq. (13) is satisfied, then Eq. (14) ensures that the estimation error for the proposal distribution converges during the iterations. Eq. (13) is termed as the Adaptive Learning Constraint (ALC) whose parameters K_l and α can be learned during the iterative computation. The ALC can be guaranteed by determining the values of max_l and min_{l-1} from the N particles in each iteration. We prove that the iterations of loop l result in the convergence of the estimate of the proposal distribution thus showing that the APF does not lead to sampling impoverishment. The proof of convergence also shows that the estimation error of the proposal distribution at loop step l=k+1 is less than that at loop step l=k, where $k \in (1, 2, \cdots, L-1)$. The result is a better approximation of the proposal distribution and the posterior distribution via the iterations of loop l. Thus, we can obtain a lower estimation error for the proposal distribution and for the posterior distribution, resulting in higher tracking accuracy in real applications.

3 THE BOOSTED ADAPTIVE PARTICLE FILTER

This paper proposes a novel scheme, termed a boosted adaptive particle filter (BAPF), for face detection and tracking by combining the above APF tracking algorithm with the AdaBoost face detection algorithm [1]. The proposed BAPF scheme consists of an AdaBoost face detection model which performs multiview face detection using a trained AdaBoost algorithm, and an APF face tracking model based on visual contour tracking. The proposed BAPF scheme consists of two phases: an initialization phase and a tracking phase. In the initialization phase, the AdaBoost face detection model provides the initial parameters for the APF face tracking model based on observations of the input video sequence over a certain time interval. During the tracking phase, the AdaBoost face detection model and the APF face tracking model improve the tracking performance via mutual interaction. The AdaBoost model helps the APF model to detect and define new objects, and to verify the current states of the objects being tracked. On the other hand, the APF model provides focus-of-attention regions within the image frame to help speed up the face detection in the AdaBoost model.

We combine the results of the AdaBoost algorithm and the APF algorithm to obtain new position for a sampled point on the contour, which is described by [5]:

$$E_c(f(\mathbf{x}_t)) = (1-\gamma) \cdot E(f(\mathbf{x}_t)) + \gamma \cdot \eta \cdot d \quad (15)$$

where E_c represents the estimate of a sampled point on the contour which combines the estimates from the APF and the AdaBoost algorithm. In Eq. (15), γ is the weight assigned to the AdaBoost model, the parameter η is a confidence measure for each detected face in the image, and d is the distance between the center of a detected face and the center of a sampled template contour. The value of $E_c(f(\mathbf{x}_t))$ is fed back to the APF for further processing. The parameter γ can be adjusted without affecting the convergence of the APF. By increasing γ, we emphasize the AdaBoost face detection algorithm. The values $\gamma=0$ and $\gamma=1$ correspond to the pure APF algorithm and pure AdaBoost algorithm respectively. The value of γ is typically adjusted based on varying scene conditions determined by clutter, illumination and occlusions.

4. EXPERIMENTAL RESULTS

The proposed BAPF and APF algorithms were implemented in C++ on a 1.6 GHz Pentium-M computer. The video sequences were sampled at 30 frames/sec with a frame size of 320×240 pixels. The proposed BAPF algorithm was applied to various tracking scenarios. The tracking results from test video sequences shown below were captured under various lighting conditions, scales, occlusions, and rotations.

Fig. 1. Tracking results under various scenarios

We compared the performance of the BAPF, APF and CONDENSATION algorithms in our experiments. The tracking accuracy is defined by the mean displacement error (MDE) between the centroid of a ground truth face and the centroid of a tracked face in the video sequences. All three algorithms are tested on the same test video, and employ N=1000 particles for face tracking. In the BAPF algorithm, γ=0.8, L=3. In the APF algorithm, L=3. The experimental results, as shown in Fig. 2 and Table 1, demonstrate that the tracking accuracy of the BAPF algorithm is higher than that of the APF algorithm, and that the tracking accuracy of the APF algorithm is higher than that of the CONDENSATION algorithm. The computation time of the APF algorithm is comparable but greater than that of the CONDENSATION algorithm, since the APF algorithm performs additional iterations needed in order to obtain better estimates of the proposal distribution and the posterior distribution. The BAPF algorithm needs more computation time than the APF algorithm, since the BAPF algorithm additionally performs AdaBoost face detection.

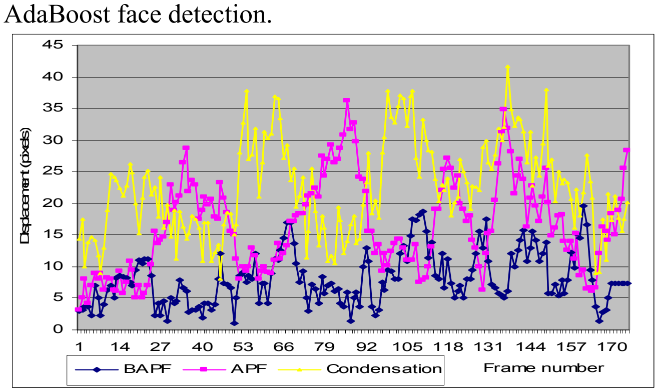

Fig. 2. Tracking results of BAPF, APF and CONDENSATION

Table 1. Performance of BAPF, APF and CONDENSATION

	BAPF	APF	CONDENSATION
MDE (pixels)	8.1	16.3	22.4
Standard deviation (pixels)	4.1	7.3	7.3
Speed (frame/sec)	4.1	4.7	6.8

5. CONCLUSIONS

This paper proposes the BAPF algorithm for face detection and tracking in video sequences. The APF algorithm is proposed to obtain more accurate estimates of the proposal distribution and the posterior distribution for improving the tracking accuracy in the input video sequences. The proposed BAPF algorithm combines the APF algorithm with the AdaBoost algorithm. The AdaBoost algorithm is used to detect faces in the input images, whereas the APF is used to track the faces in the video sequences. The proposed BAPF algorithm is employed for face detection, face verification, and face tracking in video sequences. The experimental results show that the proposed BAPF algorithm provides robust face detection and accurate face tracking under various scenarios. We compare the performance of the BAPF algorithm, the APF algorithm and the CONDENSATION algorithm. The experimental results show that in terms of tracking accuracy the BAPF algorithm is superior to the APF algorithm, which in turn, is superior the CONDENSATION algorithm. The computation times of the BAPF and APF algorithms are comparable but greater than the computation time of the CONDENSATION algorithm since both the BAPF algorithm and the APF algorithm perform additional iterations in order to obtain higher tracking accuracy.

6. REFERENCES

[1] P. Viola, and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Dec. 2001.

[2] P. Li, T. Zhang, and A.E.C. Pece, "Visual Contour Tracking based on Particle Filters," *Image and Vision Computing*, vol. 21, pp. 111-123, 2003.

[3] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[4] K. Okuma, A. Taleghni, N.D. Freitas, J.J. Little and D.G. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," *European Conf. Computer Vision*, pp. 28–39, 2004.

[5] H. Wang, P. Li, and T. Zhang, "Novel Likelihood Estimation Technique Based on Boosting Detector," *Proc. IEEE Int. Conf. Image Processing,* vol. 3, pp. 477 – 480, 2005.

[6] D. Witzner, R. Satria, J. Sorensen, R. Hammoud, "Improved Likelihood Function in Particle-based IR Eye Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 3, 2005

[7] W. Zheng, "*Face Detection and Tracking Using a Boosted Adaptive Particle Filter*", M.S. Thesis, Dept. of Computer Science, The University of Georgia, Dec. 2005.