http://dx.doi.org/10.5392/IJoC.2014.10.3.017

Adaptive MCMC-Based Particle Filter for Real-Time Multi-Face Tracking on Mobile Platforms

In Seop Na, Ha Le, Soo Hyung Kim*
School of Electronics and Computer Engineering
Chonnam National University, Gwangju 500-757, Korea

ABSTRACT

In this paper, we describe an adaptive Markov chain Monte Carlo-based particle filter that effectively addresses real-time multi-face tracking on mobile platforms. Because traditional approaches based on a particle filter require an enormous number of particles, the processing time is high. This is a serious issue, especially on low performance devices such as mobile phones. To resolve this problem, we developed a tracker that includes a more sophisticated likelihood model to reduce the number of particles and maintain the identity of the tracked faces. In our proposed tracker, the number of particles is adjusted during the sampling process using an adaptive sampling scheme. The adaptive sampling scheme is designed based on the average acceptance ratio of sampled particles of each face. Moreover, a likelihood model based on color information is combined with corner features to improve the accuracy of the sample measurement. The proposed tracker applied on various videos confirmed a significant decrease in processing time compared to traditional approaches.

Key words: *MCMC, Particle Filter, Multi Face Tracking, Mobile Platform*

1. INTRODUCTION

The real-time object detection or tracking is one of the fundamental steps for a number of advanced systems in computer vision such as human-computer interaction, augmented reality and video-surveillance. Moreover, human faces play an important role in human communication. Thus, face detection and tracking has been a research interest of many researchers on computer vision. Over the past years, numerous methods have been proposed on face detection and tracking.

The most basic approach to solve face tracking problem is to employ face detection [1] on every frame. However, despite much progress performed in recent years on multi-face detection, there are indeed many situations where faces are not detected, which is especially due to the variations of face appearance, lighting conditions or partial or full occlusion of the face. Face detectors are normally applied on simple scenarios, where people predominantly look towards the camera. However, it is the less common head poses that people naturally take. Besides, processing time of face detectors is normally considerable. As a consequence, it reduces the number of frames recorded per second, thus lowering the quality of recorded videos. Therefore, in practice, robust face trackers are combined with face detectors not only to improve the detection results but also to reduce the processing time. Numerous methods for visual tracking of faces have been proposed in the literature. These methods can be classified into two classes, single-face tracking and multi-face tracking.

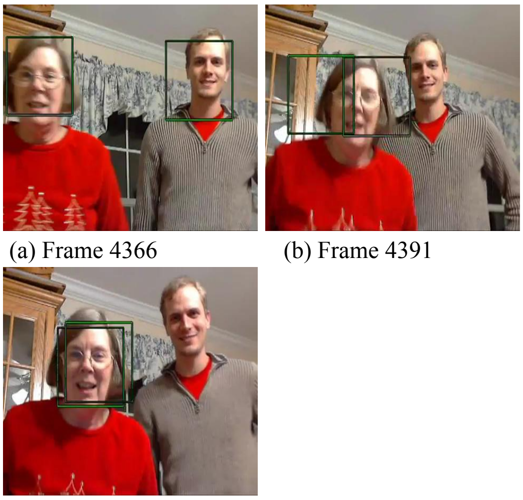
The complexity of single-face video screen is less than that of multi-face video screen. Thus, a lot of effective tracking methods have been proposed for single-face tracking. Yui Man Lui et al. [2] presented an adaptive framework for condensation algorithms in the context of human-face tracking. He addressed the face tracking problem by making factored sampling more efficient and appearance update more effective. Ruian Liu et al. [3] used adaboost for face detection and adaptive mean shift algorithm for face tracking. P. Jimenez et al. [4] proposed a method for robust tracking and estimating the face pose of a person using stereo vision. In this method, a face model is automatically initialized and constructed online: a fixed point distribution is superposed over the face when it is frontal to the cameras, and several appropriate points close to those locations are chosen for tracking. Vidit Saxena et al. [5] presented a real-time face tracking system using rank deficient face detection. Motion estimation and compensation are then incorporated in the system to ensure robust tracking, to minimize false detections, and for persistent tracking of the desired face. Liang Wang et al. [6] combined two sophisticated techniques of motion detection and template matching for detection and tracking of human faces. He used a statistical model of skin color and shape information to detect face in the first frame, and initialize it as an appearance-based intensity template for subsequent tracking. Derek Magee et al. [7] presented an efficient and general framework for the incorporation of statistical prior information, based on a wide variety of detectable point features, into level set based object tracking.

* *Corresponding author, Email: shkim@chonnam.ac.kr*

Manuscript received Apr. 28, 2014; revised Jul. 01, 2014; accepted Jul. 08, 2014

The level set evolution is based on the interpolation of likelihood gradients using kernels centered at the features. Jun Wang et al. [8] proposed an improved camshift-based particle filter algorithm for face tracking. He presented a novel feature extraction method called the block rotation-invariant uniform local binary pattern, and combine with color features to represent the appearance model of face in tracking tasks.

For multi-face tracking, due to its complexity in video screen, particle filter [9]-[12], also known as the sequential Monte Carlo [13], becomes the most popular framework chosen by researchers. The basic concept of the particle filter is to use a set of weighted particles to approximate the true filtering distribution. Particle filters offer a degree of robustness to unpredictable motion and can correctly handle complicated, non-linear measurement models. When tracking multi-faces, simply running one individual particle filter for each face is not a viable option. Particle filter itself can not address the complex interactions between faces and leads to frequent tracker failures. Whenever faces pass close to one another, the face with the best likelihood score typically affects the filters of nearby faces.


(a) Frame 4366 (b) Frame 4391

(c) Frame 4460

Fig. 1. Problem of particle filter, (a) two faces tracked using independent particle filters, (b) the face with the best likelihood score typically affects the filter of the nearby face, (c) resulting tracker failure

This is illustrated in Fig. 1 [23]. To address this issue, Zia Khan et al. [14] proposed a method by incorporating a Markov random field (MRF) to model interactions and improve tracking when faces interact. Besides, he replaced the traditional importance sampling step in the particle filter, which suffers from exponential complexity in the number of tracked faces, with a Markov chain Monte Carlo (MCMC) sampling step. Recently, many other researchers have improved Khan's method and proposed their improvement based on MCMC-based particle filter. I. Zuriarrain et al. [15] presented a MCMC-based particle filter to track multiple persons dedicated to video surveillance applications. He used saliency map proposal distribution to limit the well-known burst in terms of particles and MCMC iterations. Stefan Duffner et al. [16] presented a multi-face tracking algorithm that effectively deals with missing or uncertain detections in a principled way. The tracking is formulated in a multi-object state-space Bayesian filtering framework solved with MCMC. Anh-Tuyet Vu et al. [17] proposed a new multi-target tracking algorithm capable of tracking an unknown number of targets that move close and/or cross each other in a dense environment. Xiuzhuang Zhou et al. [18] proposed a sampling-based tracking scheme for the abrupt motion problem in the Bayesian filtering framework. Rather than simply adopting the sequential importance resampling or standard MCMC sampling algorithm, he proposed a more effective dynamic sampling scheme to sample from the filtering distribution by using the stochastic approximation Monte Carlo (SAMC) algorithm and present a sequential SAMC sampling algorithm for the tracking of abrupt motion, which demonstrates superiority in dealing with the local-trap problem with less computational burden. These presented methods run efficiently under its purpose, and the authors claim that they can run in real-time. However, the experiments of these methods were done in videos with low frame rate, (e.g. 10-15 fps [16], 25 fps [18]), and the testing environment is high performance computer, (e.g. 2.8 GHz [18], 3.16 GHz [16]).

As, nowadays, smart phones mounting high resolution cameras are widely used in the world, a fast face tracking approach running in a low performance device is highly demanded in human life. Thus, in this paper, we propose a novel method using MCMC and particle filter that effectively deals with real-time multi-face tracking on mobile. Based on our observations, there are two factors that affect the processing time of particle filter, the number of sampled particles and the likelihood computation. These recent methods require a huge number of particles, for examples, 500 particles [16] and 300 particles [18]. To reduce the number of particles, we design an adaptive sampling scheme to track the acceptance ratio of sampled particles of each face. Since, the smaller the number of particles used, the lower the accuracy achieved, we need to develop a more sophisticated likelihood model to accurately measure the sampled particles. However, likelihood computation is one of two factors that affects the processing time of particle filter. Thus, we choose a color based histogram model [12] as the main model for likelihood measurement because of its simple computation. Since the raw data taken from mobile cameras is in YUV color space, we keep using this data to reduce the converting time to another color space. A weight mask is applied to each particle to increase the meaning of face center in likelihood computation. Moreover, a fast corner detector will be used to improve the likelihood score of each particle, depending on the proportion of detected corners inside each particle to the total number of detected corners.

The rest of the paper is organized as follows. Section 2 presents our adaptive MCMC-based particle filter (AMCMC-PF) method. Implementation details and experimental results are presented in section 3. Conclusions can be found in section 4.

2. PROPOSED METHOD

Our primary goal in multi-face tracking is to estimate the posterior distribution $P\left(X_t^{'} \mid Z_{1:t}\right)$ over the state $X_t^{'}$ at the current time step t, given all observations $Z_{1:t} = \{Z_1, ..., Z_t\}$ up to that time step, according to:

$$P\left(X_t^{'} | Z_{1:t}\right) = cP\left(Z_t | X_t^{'}\right) \times \int_{X_{t-1}^{'}} P\left(X_t^{'} | X_{t-1}^{'}\right) P\left(X_{t-1}^{'} | Z_{1:t-1}\right) dX_{t-1}^{'} \quad (1)$$

Here *c* is normalization constant, the likelihood $P\left(Z_t | X_t^{'}\right)$ expresses the *measurement model* or *likelihood model*, the probability we would have observed the measurement Z_t given the state $X_t^{'}$ at time *t*, and the *motion model* $P\left(X_t^{'} | X_{t-1}^{'}\right)$ predicts the state $X_t^{'}$ given the previous state $X_{t-1}^{'}$. These models are described more detail in the following sections.

2.1 State Space

A state $X_t^{'}$ of faces contains all the information to identify faces, including the position, scale and eccentricity (i.e. the ratio between height and width) of the face bounding box. In addition to this necessary information, due to interactions and occlusions in multi-face tracking, faces may appear or disappear under observation. Thus the number and identity of the faces need to be estimated. To model this, a new variable, namely the set of identifiers k_t of faces currently in view [19], is introduced. Suppose that *M* is the maximum number of faces visible at a current time step, we can define a state $X_t^{'}$ as:

$$X_t^{'} = \left(X_t, k_t\right) \quad (2)$$

where $X_t = \left\{X_{i,t}\right\}_{i=1..M}$ and $k_t = \left\{k_{i,t}\right\}_{i=1..M}$. Each $X_{i,t}$ contains the position, scale and eccentricity of face *i* at time *t*, and each $k_{i,t}$ denotes the status of face *i* at time *t* (Eq. 3).

$$k_{i,t} = \begin{cases} 1 & if\ the\ i^{th}\ face\ is\ visible\ at\ time\ t \\ 0 & otherwise \end{cases} \quad (3)$$

2.2Motion Model

The motion model describes the relationship between the current state $X_t^{'}$ and the previous state $X_{t-1}^{'}$. In overall motion model is defined as:

$$P(X_t^{'} | X_{t-1}^{'}) \propto \pi_0\left(X_t | k_t\right) \prod_{i=1}^{M} P\left(X_{i,t} | X_{t-1}, k_t\right) \quad (4)$$

Here $\pi_0\left(X_t | k_t\right)$ is the interaction prior which prevents faces to become too close to each other. π_0 is defined as:

$$\pi_0\left(X_t | k_t\right) = \prod_{\{(i,j) \in \Omega\}} \phi\left(X_{i,t}, X_{j,t}\right) \quad (5)$$

where the set $\Omega = \left\{(i,j) | \left(k_{i,t} = 1\right) \wedge \left(k_{j,t} = 1\right) \wedge \left(i \neq j\right)\right\}$ consists all possible pairs of faces that are visible, and the ϕ function describes the interaction between two visible faces. In our framework, π_0 is estimated as:

$$\pi_0\left(X_t | k_t\right) \propto \exp\left(-\lambda_g \sum_{\{i,j \in \Omega\}} g\left(X_{i,t}, X_{j,t}\right)\right) \quad (6)$$

where

$$g\left(X_{i,t}, X_{j,t}\right) = \frac{2\left(B_i \cap B_j\right)}{B_i + B_j} \quad (7)$$

is the penalty function describing the proportion of the intersection area to the average area of two bounding boxes B_i and B_j defined by $X_{i,t}$ and $X_{j,t}$, respectively. λ_g is a constant factor that controls the strength of the interaction prior. The motion of each face is describe more precisely as:

$$P\left(X_{i,t} | X_{t-1}, k_t\right) = \begin{cases} P\left(X_{i,t} | X_{i,t-1}\right)\ if\ k_{i,t} = 1 \\ 1 \qquad\qquad otherwise \end{cases} \quad (8)$$

To describe the motion $P\left(X_{i,t} | X_{i,t-1}\right)$ of each visible face, the first order auto-regressive model is used to update the position parameter. Since the scale and eccentricity parameters are more stable than the position parameter, these parameters are only updated when a face detector tries to reinitialize the state space after a number of frames.

2.3 Likelihood Model

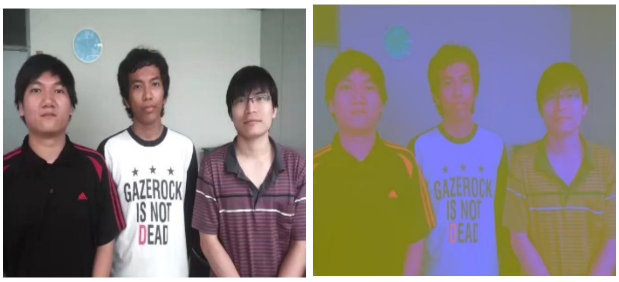
2.3.1 Color-Based Likelihood Model

As mentioned in section 1, the processing time of particle filter depends on the likelihood computation. Therefore, to strike a balance between robustness and computational complexity, we choose a simple but effective likelihood model based on color information for multi-face tracking.

Assuming that the face observations $Z_{i,t}$ are conditionally independent given the state $X_{i,t}$, we can define the likelihood model as the product of likelihoods of the visible faces:

$$P\left(Z_t | X_t^{'}\right) = \prod_{i|k_{i,t}=1} P\left(Z_{i,t} | X_{i,t}\right) \quad (9)$$

To compute the likelihood of each face $P\left(Z_{i,t} | X_{i,t}\right)$, we observe its color information. Because of its advantage with respect to human perception, HSV color space is the most popular color model [12], [16]. However, the raw data taken from mobile cameras is in YUV color space, and in our experiments, the processing time to convert an image, 640x480 pixels, from YUV color space to HSV color space is approximately 40ms (see section 3.1 for detail environment setup). Furthermore, as similar to HSV color space, YUV color space also takes human perception into account. Thus, in our likelihood model, YUV color space is chosen to represent the color information of faces.



(a) RGB color space (b) YUV color space

Fig. 2. A sample image in different color spaces, (a) RGB color space, (b) YUV color space

Fig. 2 shows a sample image in both RGB and YUV color spaces. In YUV color space, Y stands for the brightness component, and U and V are the chrominance components. Hence, we obtain color information with $N = N_u \times N_v$ bins using only the U and V channels. N_u and N_v are the number of bins of U and V channels, respectively.

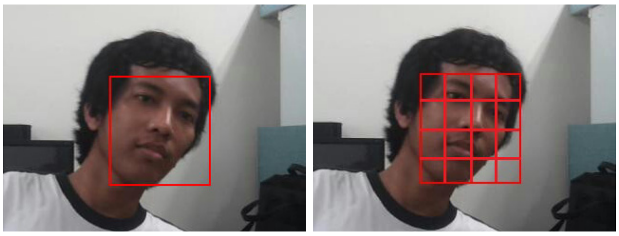
Given a state $X_{i,t}$ of face *i* at time *t*, the candidate region in which color information will be gathered is named as $R_{i,t}$. Within this region a kernel density estimate $q_{i,t}$ of color distribution at time *t* is given by:

$$q_{i,t} = \left\{q_{i,t}^{n}\right\}_{n=1..N} \tag{10}$$

where

$$q_{i,t}^{n} = k \sum_{p \in R_{i,t}} w\delta\left(b_{i,t}^{p} - n\right) \tag{11}$$

Here *k* is a normalization constant ensuring $\sum_{n=1}^{N} q_{i,t}^{n} = 1$, *w* is a weighting constant, δ is Kronecker delta function, *p* is a pixel located inside the region $R_{i,t}$, and $b_{i,t}^{p}$ is the bin index associated with the color at pixel location *p*. The most basic weighting function is $w \equiv 1$, which means the kernel density is equal to standard color histogram. However, the meaning of the color pixels near region center is normally higher than the meaning of the color pixels far from region center.



(a) A face with its bounding box (b) A face with its grid box

Fig. 3. A sample image with and without grid box, (a) with bounding box, (b) with grid box

For example, when the face involves a large head tilt as shown in Fig. 3a, its bounding box includes a lot of background pixels, which are meaningless. If we weigh the background pixels equally to face's pixels, the likelihood measurement will be less accurate, and the bounding box will never fit the tracked face. There are several weighting functions to address this problem, such as Gaussian [20] and radius distance [21]. Based on these weighting functions, we adopt our weighting function by dividing the bounding box into 4 by 4 blocks (Fig. 3b) and weigh each block with a power of two.

Table 1. Weight values of each block

1	2	2	1
2	4	4	2
2	4	4	2
1	2	2	1

The weight of each block is shown in Table 1. Weight values are assigned to be power of two for more efficient computation.

At time *t*, the observation $Z_{i,t} = q_{i,t}$ is compared to the reference color distribution $q_{i,t}^{*} = \left\{q_{i,t}^{*n}\right\}_{n=1..N}$ to define the observation likelihood for a tracked face. An approximation of observation likelihood is given as:

$$P\left(Z_{i,} \mid X_{i,t}\right) \propto \exp\left(-\lambda_D D^2\left(q_{i,t}, q_{i,t}^{*}\right)\right) \tag{12}$$

where *D* denotes the Euclidean distance between two color distributions and λ_D is a constant factor that controls the strength of the observation likelihood. The reference color distribution is gathered at the initial time t_0 and updated every time step. Let $q_{i,t-1}^{m}$ denotes the color distribution of the mean state of tracked face *i* at time *t* - 1. The reference color distribution of face *i* at time *t* is defined as:

$$q_{i,t}^{*} = \left(1-\varepsilon\right) q_{i,t-1}^{*} + \varepsilon q_{i,t-1}^{m} \tag{13}$$

where ε is the update factor that controls how fast the reference color distribution is updated.

2.3.2 Corner Features

Likelihood model based on color information is fast. However, color information is not stable for occlusion. Thus, in this paper, we combine color information with corner information to increase the accuracy of likelihood measurement. In our observation, there are a lot of corner points concentrated on human faces.

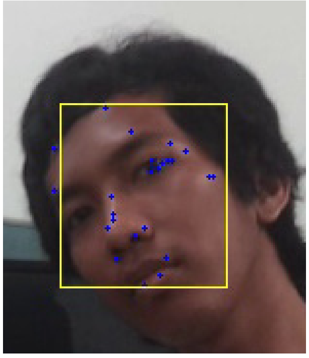


Fig. 4. Detected corner points in a face

For example, a face with detected corner points is shown in Fig. 4. Thus, a good particle contains a lot of corner points inside. To represent this characteristic, we update the likelihood of a particle with the ratio of the number of detected corners outside particle to the number of detected corners. For a better efficiency, we detect only the corner points in a limited region. Suppose that $B_{i,t-1}$ is the tracked bounding box of face i at time t - 1. At time t, we detected only corner points for face i in the bounding box $B_{i,t-1} \pm \Delta$, where Δ is a small constant value describing the expansion and movement of $B_{i,t-1}$. If C_t is the number of detected corners, and C_o is the number of detected corners outside particle, the observation likelihood in Eq. 12 can be re-estimated as:

$$P\left(Z_{i,} \mid X_{i,t}\right) \propto \exp\left(-\lambda_D \frac{C_o}{C_t} D^2\left(q_{i,t}, q^*_{i,t}\right)\right) \qquad (14)$$

For fast corner detection, we employ the FAST (Features from Accelerated Segment Test) corner detector [22].

2.4 Tracking Model

At each time step, the tracking model proceeds in two main stages: estimating the states of the tracked faces and identifying the visibility status of these faces.

2.4.1 NCNC-based Particle Filter

To estimate the states of the tracked faces, we use a MCMC sampling scheme, which allows efficient sampling in high dimensional state space of interacting faces [14]. Suppose that at time t – 1, the state of the tracked faces is represented by a set of samples $\left\{X_{t-1}^{'(r)}\right\}_{r=N_b+1}^{N}$. In which, N is the total number of particles and N_b is the number of "burn-in" particles. Hence, the detailed steps of the MCMC sampling scheme are proposed as follows:

1) Initialize the MCMC sampler at time t with the sample $X_t^{'(0)}$ obtained by randomly selecting a particle from the set $\left\{X_{t-1}^{'(r)}\right\}_{r=N_b+1}^{N}$ and sampling state of every visible faces i in $X_t^{'(0)}$ using the motion model $P\left(X_{i,t} \mid X_{i,t-1}\right)$.

2) Sample iteratively N particles from the posterior distribution (Eq. 1) using the Metropolis-Hasting (MH) algorithm. Discard the first N_b samples to account for sampler burn-in. The detailed steps of MH are described as follows.

a) Sample a new particle $X_t^{"}$ from the proposal distribution

$$q\left(X_{i,t}^{"} \mid X_t^{'}\right) = \frac{1}{N-N_b} \sum_{r=N_b+1}^{N} P\left(X_{i,t}^{"} \mid X_{i,t-1}^{'(r)}\right) \qquad (15)$$

b) Compute the acceptance ratio

$$a = \min\left(1, \frac{P\left(X_t^{"} \mid Z_{1:t}\right) Q\left(X_t^{'(r)} \mid X_t^{"}\right)}{P\left(X_t^{'(r)} \mid Z_{1:t}\right) Q\left(X_t^{"} \mid X_t^{'(r)}\right)}\right) \qquad (16)$$

c) If $a \geq 1$ then accept the particle, set $X_t^{'(r+1)} = X_t^{"}$. Otherwise, add a copy of the current particle to the new sample set with probability a .

The particle set $\left\{X_t^{'(r)}\right\}_{r=N_b+1}^{N}$ at time t represents an estimation of the posterior of the tracked faces.

2.4.2 Adaptive Sampling Scheme

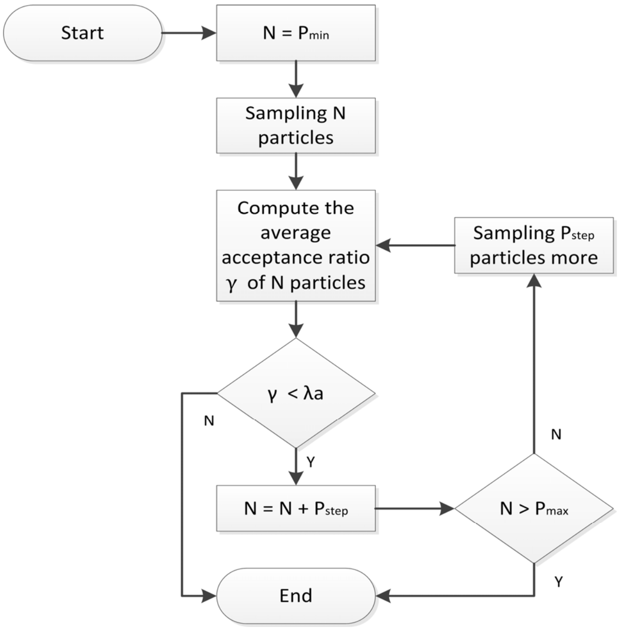

Start
N = Pmin
Sampling N particles
Compute the average acceptance ratio γ of N particles
Sampling Pstep particles more
γ < λa
N
Y
N = N + Pstep
N > Pmax
N
Y
End

Fig. 5. Adaptive Sampling Scheme

The number of particles N is a factor that affects the processing time of particle filter. Thus, instead of using a fixed number of particles, we try to adjust it during sampling process. Fig. 5 shows the adaptive sampling scheme of our tracker. In this scheme, we need to define three factors: P_{min}, P_{max} and P_{step}. P_{min} is the minimum number of particles that the sampler has to generate. In contrast with P_{min}, P_{max} is the maximum number of particles that the sampler can generate. Thus, we have $P_{\min} \leq N \leq P_{\max}$. P_{step} is the number of particles sampled more if the average acceptance ratio γ of sampled particles is less than threshold λ_a.

2.4. 3 Visibility Status Identification

To identify the visibility status of tracked faces, we assume that if a face is still tracked correctly, the observation likelihood should be high and the variances in x and y direction of the bounding box should be low. Let $y_{i,t}$ be the observation likelihood of the mean state of face i at time t and $v_{i,t}$ be the maximum of the variances in x and y direction of the bounding box of the mean state of face i at time t. The average likelihood of face i over all time steps is computed as:

$$\bar{y}_{i,t} = \frac{(t-1)\bar{y}_{i,t-1} + k_{i,t} y_{i,t}}{t + k_{i,t} - 1} \quad (17)$$

The average variance of bounding box of face i over all time steps is computed as:

$$\bar{v}_{i,t} = \frac{(t-1)\bar{v}_{i,t-1} + k_{i,t} v_{i,t}}{t + k_{i,t} - 1} \quad (18)$$

With the computation in Eq. 17 and Eq. 18, we can skip the likelihood and variance of invisible faces. A tracked face is marked as invisible if:

$$\left[\begin{array}{ll} y_{i,t} < \lambda_y \bar{y}_{i,t-1} & (0 < \lambda_y < 1) \\ v_{i,t} > \lambda_v \bar{v}_{i,t-1} & (\lambda_v > 1) \end{array}\right. \quad (19)$$

In here, λ_y and λ_v are the constant factors that control the confident of the tracker. A face is invisible more than a number of frames will be removed from the tracker. In contrast, an invisible face at time t – 1 will be marked as a visible face at time t if:

$$\begin{cases} y_{i,t} > \lambda_y \bar{y}_{i,t-1} & (0 < \lambda_y < 1) \\ v_{i,t} < \lambda_v \bar{v}_{i,t-1} & (\lambda_v > 1) \end{cases} \quad (20)$$

3. EXPERIMENTS AND RESULTS

3.1 Experiment Setup

To test the empirical performance of our proposed tracker, we construct a video database using a Samsung Galaxy S2 mobile phone. Our video database contains 90 videos recorded in various scenarios, including single face, multi-faces, different illumination (i.e. bright, dark, normal) and different movement (i.e. head movement, camera movement). Each video is recorded with a resolution of 640x480 pixels, and frame rate of 30 fps. Our tracker is implemented in Android NDK environment and runs in Samsung Galaxy S2 with a dual-core 1.2 GHz processor.

Below are the specific implementation choices for constant factors presented in previous section.

- For the motion model
  - We use a uniform density centered on the previous pose.
  - The range of the uniform distribution is (-32, 32).
  - The constant factor for the interaction prior: $\lambda_g = 4$.
- For the likelihood model
  - The numbers of bins of channel U and V: $N_u = N_v = 16$.
  - The constant factor for likelihood computation: $\lambda_D = 16$.
  - The update factor: $\varepsilon = 0.5$
  - The constant value for the expansion and movement of bounding box: $\Delta = 8$
- For MCMC parameters
  - We discard 25% of the samples to let the sampler burn in, regardless of the total number of samples.
- **For the adaptive sampling scheme**
  - The average acceptance ratio threshold: $\lambda_a = 0.25$
  - The minimum number of samples: $P_{\min} = 64$
  - The maximum number of samples: $P_{\max} = 128$
  - The number of samples are increased each iteration: $P_{step} = 16$
- For visibility status identification
  - The confident factor: $\lambda_y = 0.25$ and $\lambda_v = 4$
  - A face is invisible more than 16 frames will be removed from the tracker.

Notice that these constant factors are assigned to be power of two for faster computation.

In our tracking progress, the face detector et al. [1] is used to initialize in the first frame, and re-initialize after every 64 frames. The re-initialization is necessary for a stable tracker. However, in some frames, the face detector fails to detect the faces. In that case, we will keep tracking undetected faces without re-initialization.

3.2 Performance Measures

To measure performance of our proposed algorithm, we use Precision (P), Recall (R) and F-measure (F_β), which are defined as following equations.

$$P = \frac{\sum_{i=1}^{n} B_{g,i} \cap B_{t,i}}{\sum_{i=1}^{n} B_{t,i}} \quad (21)$$

$$R = \frac{\sum_{i=1}^{n} B_{g,i} \cap B_{t,i}}{\sum_{i=1}^{n} B_{g,i}} \qquad (22)$$

$$F_{\beta} = \frac{(1+\beta^2) R \times P}{\beta^2 R + P} \qquad (23)$$

Here, n is the number of annotated faces in a frame, $B_{g,i}$ is the ground truth rectangle of face i and $B_{t,i}$ is the rectangle output of face i from face detection or tracking. We use $\beta^2 = 0.3$ to weigh recall more than precision.

3.3 Results

We compared our proposed algorithm to the following algorithms: [MCMC-PF] A state-of-art multi-tracking method based on MCMC and particle filter [14]. [PP1] Our proposed algorithm without applying grid box in likelihood computation. [PP2] Our proposed algorithm without using corner information. [PP3] Our proposed algorithm with a fixed 64 number of particles. [PP4] Our proposed algorithm with a fixed 128 number of particles. [PP5] Our proposed algorithm but using HSV color space instead of YUV color space.

Table 2. Face tracking algorithms with different features

Methods	MCMC-PF	PP1	PP2	PP3	PP4	PP5	Our method
HSV						x	
YUV	x	x	x	x	x		x
Grid box			x	x	x	x	x
Corner Information		x		x	x	x	x
Fixed 64 Particles	x			x			
Fixed 128 Particles					x		
Adaptive Sampling		x	x			x	x

Table 3. Accuracies of tracking algorithms

Algorithms	P(%)	R(%)	Fβ(%)
MCMC-PF	73	63	65
PP1	79	73	74
PP2	78	67	70
PP3	85	79	80
PP4	91	82	84
PP5	89	83	84
Proposed Algorithm	89	81	83

Table 2 shows features of the above methods. And the accuracies of tracking algorithms are shown in Table 3.

Table 4. Processing time of tracking algorithms

Algorithms	MCMC-PF	PP1	PP2	PP3	PP4	PP5	Proposed Algorithm
Time per face (ms)	16	17	17	17	38	59	19

The processing time of tracking algorithms is shown in Table 4.

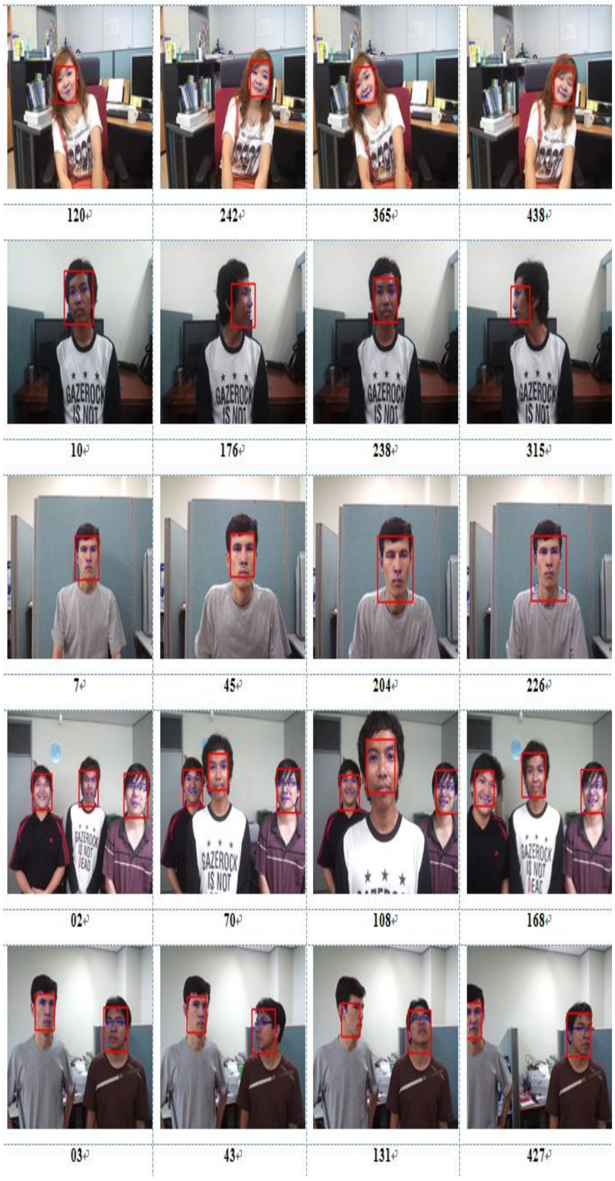


Fig. 6. Results of the proposed face tracker with several different scenarios, 1st row: free moving style, 2nd row: rotation, 3rd row: scaling, 4th row: multi-face scaling, 5th row: multi-face rotation

The accuracy of MCMC-PF is worse than that of other algorithms. This means that our extended likelihood model with corner information is more effective than the simple color-based likelihood model. The accuracy of PP1 is worse than that of our proposed algorithm. This proves the effectiveness of grid box in likelihood computation. The accuracy of PP2 is far lower than that of our proposed algorithms. This means that the corner information is one of the most important feature in our proposed algorithms.

The accuracy of our proposed algorithm is more competent than that of PP3, but not as good as that of PP4. However, regarding processing time, our algorithm has proved its advantage. Because PP4 uses a large number of particles, its processing time is almost twice the processing time of our proposed algorithm or the processing time of PP3. Therefore, to strike a balance between tracking accuracy and processing time, our proposed algorithm is the most suitable for the real-time processing in low performance devices, such as smart phones. The accuracy of PP5 shows that the HSV color space has some advantage in comparing with YUV color space. Because of converting time between two color spaces, the processing time PP5 is much more than that of our proposed algorithm. Thus, HSV color space is not suitable for our proposed algorithm.

Fig. 6 shows the results of our face tracker in several different scenarios.

4. CONCLUSIONS

We have presented an adaptive MCMC-based particle filter framework for robust real-time multi-face tracking in various scenarios. In the proposed tracking algorithm, we have introduced an adaptive sampling scheme that concurrently reduces the number of particles and processing time. Furthermore, we have extended the likelihood model based on color information by combining with corner information. The extended likelihood model can effectively deal with occlusions and increase the tracking accuracy. Extensive experimentation has indicated that our method out-speeds other alternatives and can run in real-time in low performance devices, such as mobile phones. Our further study will concentrate on improving the tracking accuracy while sustaining the processing time.

ACKNOWLEDGEMENT

“This work was supported by SAMSUNG ELECTRONICS CO., LTD.” And "This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1014) supervised by the NIPA(National IT Industry Promotion Agency)."

REFERENCES

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” In Proc. of CVPR, vol. 1, 2001, pp. 511-518.

[2] Yui Man Lui, J. Ross Beveridge, and L. Darrell Whitley, “Adaptive appearance model and condensation algorithm for robust face tracking,” Trans. Sys. Man Cyber, vol. Part A 40, no. 3 May. 2010, pp. 437-448.

[3] Ruian Liu, Mimi Zhang, Shengtao Ma, “Design of Face Detection and Tracking System,” Image and Signal Processing (CISP), vol. 4, Oct. 2010, pp. 1840-1844.

[4] P. Jime´nez, J. Nuevo, L. M. Bergasa, and M. A. SoteloFace, “Tracking and pose estimation with automatic three-dimensional model construction,” Computer Vision, IET, vol. 3, Jun. 2009, pp. 93-102.

[5] Vidit Saxena, Sarthak Grover, and Sachin Joshi, “A Real Time Face Tracking System using Rank Deficient Face Detection and Motion Estimation,” Cybernetic Intelligent Systems, CIS, Sep. 2008, pp. 1-6.

[6] Liang Wang, Tieniu Tan, and Weiming Hu, “Face Tracking Using Motion-Guided Dynamic Template Matching,” Asian Conference on Computer Vision ACCV.

[7] Derek Magee and Bastian Leibe, “On-line Face Tracking Using a Feature Driven Level-set,” British Machine Vision Conference BMVC'03, Sep. 2003.

[8] Jun Wang, Jin-ye Peng, Xiao-yi Feng, Lin-qing Li, and Dan-jiao Li, “An improved camshift-based particle filter algorithm for face tracking,” In Proceedings of the Second Sino-foreign-interchange conference on Intelligent Science and Intelligent Data Engineering (IScIDE'11), Yanning Zhang, Zhi-Hua Zhou, Changshui Zhang, and Ying Li (Eds.), Springer-Verlag, Berlin, Heidelberg, pp. 278-285.

[9] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” IEE Procedings F., vol. 140, no. 2, 1993, pp. 107-113.

[10] J. Carpenter, P. Clifford, and P. Fernhead, *An improved particle filter for non-linear problems*, Department of Statistics, University of Oxford, Tech. Rep., 1997.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process, vol. 50, no. 2, Feb. 2002, pp. 174-188.

[12] P. Perez, Carine Hue, Jaco Vermaak, and Michel Gangnet, “Color-Based Probabilistic Tracking,” In Proceedings of the 7th European Conference on Computer Vision-Part I ECCV '02. Springer-Verlag, London, UK, UK, pp. 661-675.

[13] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

[14] Z. Khan, T. Balch, F. Dellaert, “An MCMC-based particle filter for tracking multiple interacting targets,” IEEE Trans. on PAMI, vol. 27, no. 11, Nov. 2005, pp. 1805-1918.

[15] I. Zuriarrain, F. Lerasle, N. Arana, and M. Devy, “An MCMC-based particle filter for multiple person tracking,” Pattern Recognition. ICPR, Dec. 2008, pp. 1-4.

[16] S. Duffner and J. Odobez, “Exploiting long-term observations for track creation and deletion in online multi-face tracking,” Automatic Face & Gesture Recognition and Workshops FG, Mar. 2011, pp. 525-530.

[17] Anh-Tuyet Vu, Ba-Ngu Vo, and R. Evans, “Particle Markov Chain Monte Carlo for Bayesian Multi-target Tracking,” Information Fusion FUSION, Jul. 2011, pp. 1-8.

[18] Xiuzhuang Zhou, Yao Lu, Jiwen Lu, and Jie Zhou, “Abrupt Motion Tracking Via Intensively Adaptive

Markov-Chain Monte Carlo Sampling," Trans. Img. Proc., vol. 21, no.2, Feb. 2012, pp. 789-801.

[19] M. Isard, J. MacCormick, "BraMBLe: A Bayesian multiple-blob tracker," Intl. Conf. on Computer Vision ICCV, 2001, pp. 34-41.

[20] H. T. Chen and T. L. Liu, "Trust-region methods for real-time tracking," In Proc. Int. Conf. Computer Vision, Jul. 2001, pp. 717-722.

[21] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," In Proc. Conf. Comp. Vision Pattern Rec., Jun. 2000, pp. 142-149.

[22] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," In Proceedings of the 9th European conference on Computer Vision - Volume Part I (ECCV'06), Aleš Leonardis, Horst Bischof, and Axel Pinz (Eds.), vol. Part I, Springer-Verlag, Berlin, Heidelberg, pp. 430-443.

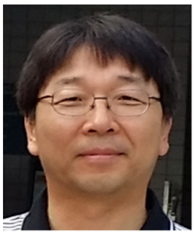
[23] Available at http://youtu.be/DVf99x--ouk

In Seop Na

He received his B.S., M.S. and Ph.D. degree in Computer Science from Chonnam National University, Korea in 1997, 1999 and 2008, respectively. Since 2012, he has been a research professor in Department of Computer Science, Chonnam National University, Korea. His research interests are image processing, pattern recognition, character recognition and digital library.

Ha Le

He received the B.S in Computer Science from Hanoi University of Science and Technology, Vietnam in 2010. And he recieved the M.E in the Department of Computer Science, Chonnam National University, Korea. In 2013. His main research interests include pattern recognition, image processing, text recognition, object segmentation and object tracking.



Soo Hyung Kim

He received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. From 1990 to 1996, he was a senior member of research staff in Multimedia Research Center of Samsung Electronics Co., Korea. Since 1997, he has been a professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are pattern recognition, document image processing, medical image processing, and ubiquitous computing.