

Scale Invariant Single Face Tracking Using Particle Filtering With Skin Color

Perdana Adhitama, Soo Hyung Kim, In Seop Na*

School of Electronics & Computer Engineering

Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju, 500-757, Korea

ABSTRACT

In this paper, we will examine single face tracking algorithms with scaling function in a mobile device. Face detection and tracking either in PC or mobile device with scaling function is an unsolved problem. Standard single face tracking method with particle filter has a problem in tracking the objects where the object can move closer or farther from the camera. Therefore, we create an algorithm which can work in a mobile device and perform a scaling function. The key idea of our proposed method is to extract the average of skin color in face detection, then we compare the skin color distribution between the detected face and the tracking face. This method works well if the face position is located in front of the camera. However, this method will not work if the camera moves closer from the initial point of detection. Apart from our weakness of algorithm, we can improve the accuracy of tracking.

Key words: Single Face Tracking, Particle Filter, Face Detection, Scaling, Skin Color.

1. INTRODUCTION

In recent years, the development of computer and mobile technology has enabled the investigation of new and efficient algorithm for computer vision field studies. As a result, object tracking, character recognition, object recognition and object segmentation are getting popular.

Face tracking is one of the most interesting research topics in the field of computer vision [4]. It can be used in many applications such as face expression recognition, face pose recognition, visual speech recognition and the others. In recent years, many algorithms have been proposed on PC and hardware environments. However single face tracking in mobile devices has not gained much attention to researchers.

There are many tracking algorithms which have been found for object tracking such as mean shift, camshift [15], and particle filter. Mean shift and camshift are very sensitive to color. As a result the tracking accuracy cannot be maximized. In contrast, particle filter is not too sensitive with color. It has been known to be the best tracking algorithm due to performance, accuracy and occlusion [7], [2]. However, it cannot overcome the scaling problems. Scaling function is very useful to track object in dynamic distance when the object could move closer or farther from the camera.

The performance of particle filter tracking system depends on object representation and similarity measurements between target and reference which is called measurement model or

observation model. In our proposed particle filter, we perform maximum likelihood (ML) for sequence estimation which is suitable in non-linear Bayesian.

In spite of its effectiveness in tracking, particle filter also proved for its resource requirement. In the measurement stage, particle filter computes likelihood between each particle and the tracking target. The more number of particle increases, the more computation time increases rapidly. Because of Monte Carlo simulation all the particles are drawn randomly, which leads to highly irregular memory access.

In this paper, we introduce scaling in single face tracking using particle filter algorithm based on skin color distribution. In Section 2, we describe briefly about related work. Then, we explain our proposed algorithm in Section 3 including particle filter algorithm, scaling and face detection. The experimental results is depicted in Section 4 demonstrate our proposed algorithm. Finally, in Section 5 we will conclude our solution and our achievement.

2. RELATED WORKS

In the past decade, many researchers have developed face detection, face tracking and other computer vision techniques has been developed by many researchers. Abdullah Bulbul et al propose a face tracking in mobile device for human computer interaction [1]. In this method, they use a mobile phone to control the application using head movements by comparing the face positions through the neighboring frames. In addition the proposed solution is based on color comparisons and works on images gathered from the front camera of a device. One of their applications for user interaction in mobile devices is an

* Corresponding author, Email: ypencil@hanmail.net
Manuscript received Feb. 07, 2013; revised Aug 05, 2013;
accepted Aug 22, 2013

ability to browse large images. For instance, when the user moves their head to the right, they can see the area on the left from the window and vice versa.

In object tracking various modification of particle filter algorithm has been developed. Der-Chun Chen et al proposed an object tracking using color based particle filter with dual cache architecture [3]. It utilizes a frame cache to reduce memory bandwidth of loading frame data, and a histogram cache to reduce operation cycles when constructing a color histogram. Another method using a color histogram has been done using color distribution [6], [9]. In this method, they used HSV color space due to the sensitivity in lighting conditions. The HSV histogram-based model that they used is to make a robust color based probabilistic tracker system by particle filter. A more advance of object tracking based on particle filter has been developed using integral image [8]. In this method, they use integral image in order to make a very efficient color based target representation. They also used MAP (Maximum a Posteriori) to overcome partial occlusion during the tracking.

Particle Filter also could be used to track multiple objects. Irshad Ali et al proposed particle filter to track humans in high density crowds [5]. In this method, they used Viola-Jones and Adaboost cascade classifier for detection. In order to stabilize the tracking, they redetect the face every 60 frames. Another research related to multi object tracking was developed by Zia Khan et al using MCMC-based particle filter [11]. MCMC is used to improve tracking when objects interact with each other. In this method, the traditional importance sampling step was replaced with an MCMC sampling method to overcome the hijacking problem because of the best likelihood of it to appear in the close object. The sampling method that they used is Metropolis-Hasting algorithm. Metropolis-hasting algorithm is one of the ways to simulate Markov Chain and to generate a set of samples. Another way to simulate Markov Chain is using Gibbs sampling.

In terms of face detection algorithm, Viola and Jones method [13], [14] is one of the face detection methods. Viola and Jones method is very popular and has been used by many researchers due to its accuracy. However, the method is not too robust to be implemented in a mobile device. Therefore, we use face detection from a Samsung detection library.

3. PROPOSED METHOD

We propose a method for scaling in particle filter algorithm. Our system framework operates as follows. First, face is detected to find skin color distribution and the initialization of the face position. The skin color model is learned online during initialization using the detected face of the user. We proposed skin detection to compare the skin color distribution between detected face and tracking face for robust scaling face tracking. In the next step, a prediction using Gaussian distribution is implemented. Then we calculate the similarities of the face using the likelihood method. After that, scaling is used to scale up or scale down the area of the face. Finally, a resample is implemented to remove the lower weight of each particle.

We propose to use a particle filter with color-based image

features. Color histograms in particular have many advantages for tracking non-rigid objects as they are robust to partial occlusion, scale invariant and have low computational cost. A face is tracked with a particle filter by comparing its histogram with the histograms of the sample position using Bhattacharyya distance. The number of histogram bins that we use is 16 bins which make the color histogram less sensitive to small color variances, and meanwhile not too coarse as to lose the ability to discriminate different color.

Since the system will be implemented in a mobile device we try to avoid many computations. Therefore, instead of using HSV color space we use YUV color space to reduce the computation time. The reason why we use YUV color space is, it is the default color space that we obtain from a mobile device camera. As a result, we could save the CPU works.

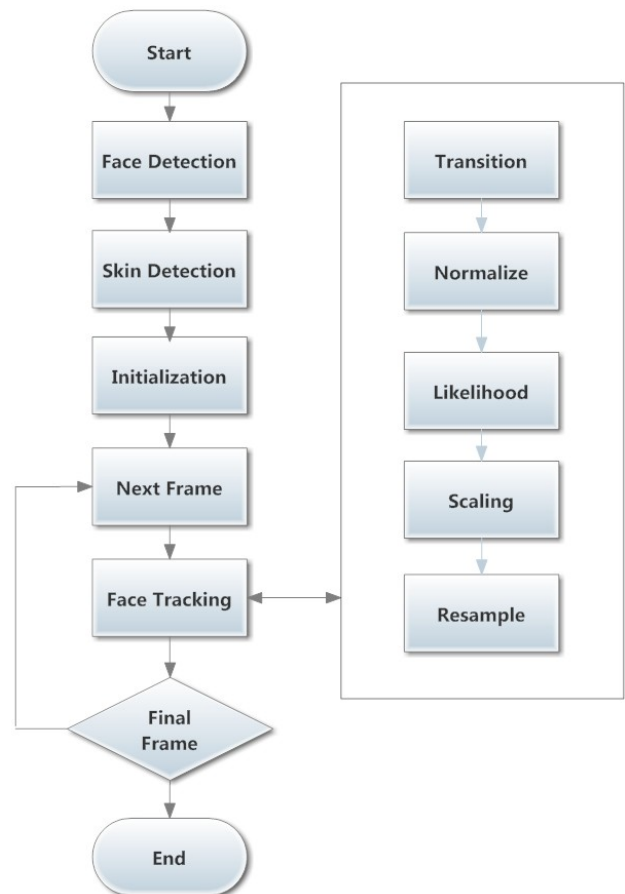


Fig. 1. Our proposed algorithm

The following scaling condition is adopted, extract the skin color by averaging the pixel value in face detection. To be considered as a skin color, we define 50% of the skin color as a threshold value (T_s). After that we calculate model histogram (q) in the face detection area. During the tracking process, we compare the skin color between the tracking (S_t) and the detected face (S_d). The proposed method is given below as an algorithmic expression by essentially steps:

1. **Initialize**
 - a. Frame, YUV color space, mode is detection
2. **Detection**
 - a. Detect face initial frame using Samsung Face Detection
 - b. Set the threshold to be considered as a skin color ($T_s) = 0.5$
 - c. Skin color extraction at detected face (S_0)
 - d. Calculate model histogram (q)
3. **Tracking**
 - a. Calculate target histogram for each particles (p)
 - b. Bhattacharya similarity between target histogram (p) and model histogram (q)
 - c. Calculate skin color extraction at tracking (S_t)
 - d. Scaling (S)

$$S = \frac{S_t}{S_0}$$
 - e. Every 30 frames go to the 2

S_t is the average skin color at tracking, S_0 is the average skin color at detected face

3.1 Initialize

For face detection, we can choose it manually or automatically. For automatic detection there are some algorithms, one of them is Viola Jones method. Instead of using Viola Jones we choose to use the face detection method from Samsung. There are two functions of face detection in our proposed algorithm. The first is a detector, running a sliding window over the image at the specific range in the scene. Second is a confirmer, to check whether the face is in the detection area or not. We confirm or redetect the face every 30 frames to keep the tracking algorithm.

3.2 Skin Detection

Skin detection is one of the most challenging problems because human skin color varies from person to person. In our system, the skin color is learned based on the similar skin color of the face. Since most of the area of the face is skin color, we assume the average pixel in face area is skin color. Therefore, we calculate the average of face area to be considered as a skin color. After that, we set a threshold value based on the average value of skin color which we obtained.

$$\begin{aligned} \text{Min}(Y, U, V) &= \frac{1}{f} \sum \text{skin} - \left(\frac{1}{f} \sum \text{skin} * 50\% \right) \\ \text{Max}(Y, U, V) &= \frac{1}{f} \sum \text{skin} + \left(\frac{1}{f} \sum \text{skin} * 50\% \right) \end{aligned} \quad (1)$$

Where f is the number of pixel area of the detected face, skin is the skin color distribution of the detected face. The number 50% is the threshold of a skin color distribution.

To set the scaling factor of the face we set the scaling factor based on the threshold of skin color distribution. To be considered as a skin color, we make a skin selection according to the threshold of skin value from the previous step.

$$\text{skin} \begin{cases} \text{Min}(Y, U, V) \leq X \leq \text{Max}(Y, U, V) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$S_0 = \frac{\text{skin}}{f}$$

Where X is the skin value and skin is the number of pixel to be considered as a skin. Where f is the number of pixel in the face detection area. S_0 is the scaling factor which we will be using to compare the distribution of skin color between the detected face and the tracking face.

3.3 Single Face Tracking Particle Filter

Particle filter algorithm is well known algorithm for tracking object. It is a sequential Monte Carlo method methodology where the basic idea is the recursive computation of relevant probability distribution using the concepts of importance sampling and approximation of probability distribution with discrete random measures. The fundamental idea of Particle Filter approximates the filtered posterior distribution by a set of random particles with associated weights, each particle representing one possible state. It weights particles based on a likelihood score and then propagates these particles according to the motion model. Particle filtering assumes a Markov Model for system state estimation. Markov model states that past and future states are conditionally independent of a given current state. Resamples the particle to remove low weight particle and keep the high particle. The initial distribution for the filter is centered on the location of the object the first time it is detected. Here are the steps in more detail:

3.3.1 Initialize: For the initialization of the particle filter, we have to find the initial starting values $x, y, H_x,$ and H_y . There are three possibilities depending on the prior knowledge of the target object: manual initialization, automatic initialization using a known histogram as target model or an object detection algorithm that finds interesting targets. In our method we initialize which is given from face detection.

3.3.2 Transition: Our prediction method or transition model is based on Gaussian random distribution. The Gaussian random distribution will make the particles position close to the center of the face.

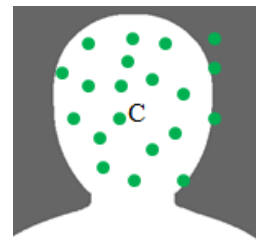


Fig. 2. The transition based on Gaussian random distribution. C is the center of the face, dots is the Gaussian distribution.

3.3.3 Normalize: Since we will calculate the similarities between distribution sampling and detected face, we need to normalize the histogram color for all Gaussian random sampling. We normalize the weight to make the sum of the

weight equal to 1.

3.3.4 Likelihood: The likelihood function gives the probability density function of image features given the state. The selected features are pixel oriented. Thus the appearance will be given by a matrix whose elements are the pixel intensity values.

To calculate a particle's likelihood we use Bhattacharyya similarity coefficient between sampling histogram and model histogram [9].

$$D_t = \sqrt{1 - \sum_{u=1}^m \sqrt{p(u)q(u)}} \quad (3)$$

Where $p = \{p(u)\}_{u=1 \dots m}$, $q = \{q(u)\}_{u=1 \dots m}$ and D_t is the distance between two distributions. The larger p is, the more similar the distributions are. For two identical normalized histograms we obtain $p=1$, indicating a perfect match.

3.3.5 Scaling: In scaling process, we try to compare the summation of skin color between the first detected face and the tracking face. Based on equation (2) we can calculate the scaling value of the tracking face using the following equation.

$$S = \frac{S_t}{S_0} \quad (4)$$

Where S_t is the scaling factor at time t, and S_0 is the scaling factor at detected face.

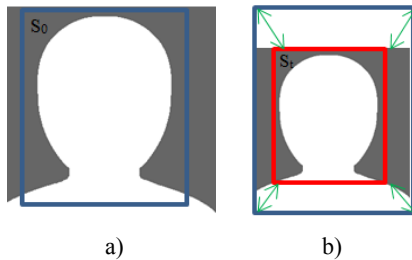


Fig. 3. a) First detected face. b) Scaling using skin color extraction

In figure 3 above, the system tries to calculate the color distribution in tracking step. If the number of color distribution is less than the number in detected face, the area of the face will shrink.

3.3.6 Resample: The samples are re-sampled to generate an un-weighted particle set according to their importance weights to avoid degeneracy. Without resampling, over-time, the highest-weighted particle would tend to be zero. As a result the low weight particles will be removed and will keep the higher particles. We thus obtain a new set of equally-weighted particles.

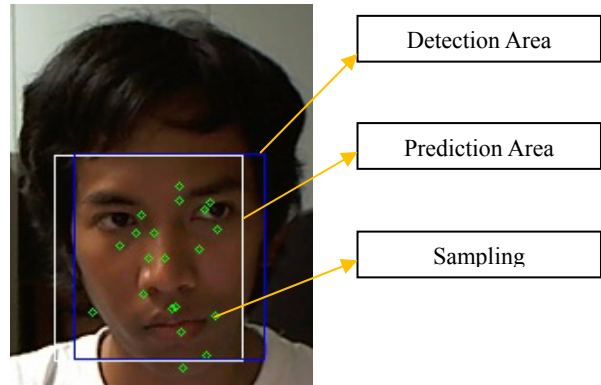


Fig. 4. Blue rectangle: face detection area, white rectangle: predicted area and green circles are the sampling distribution

In the figure above, for the purpose of clarity twenty particles are setted. Green dots are the location of particles. After applying resample, we can predict the next position which can be seen in white rectangle. According to our experiments, 20 particles suit the best for face tracking.

4. EXPERIMENTAL RESULTS

The device that we used to evaluate our algorithm was Samsung Galaxy S2 with Android OS version 4 Ice Cream Sandwich. The size of the video is 640 x 480.

To evaluate the accuracy of our algorithm, we collected numerous video databases from our school with different people and environments. The number of videos that we have collected is 10 videos divided into 2 groups of video: simple and complex background. All video databases is designed to test our system which includes zoom in and zoom out function.

To calculate the accuracy of our algorithm we choose 5 samples video in our database. Let suppose D_{gt} is the length of diagonal ground truth, C_{gt} is the center of ground truth, C_t is the center of tracking then D_c is the distance between C_{gt} and C_t .

$$acc = 1 - \frac{D_c}{D_{gt}} \quad (5)$$

Table 1. Comparison with other algorithm

	Traditional PF	Camshift [15]	Our algorithm
single_c_n_scale1.mp4	90.96%	92.21%	93.67%
single_c_n_scale2.mp4	87.07%	90.47%	93.03%
single_c_n_scale3.mp4	91.37%	90.44%	91.69%
single_c_n_scale4.mp4	84.71%	87.13%	92.21%
single_c_n_scale5.mp4	87.48%	87.89%	88.19%

Table 1 shows the comparison between traditional particle filter, camshift and our proposed algorithm. Based on table 1 our proposed algorithm can achieve better performance and accuracy in some conditions. In figure 5 below we will explain and compare the results from different algorithm.

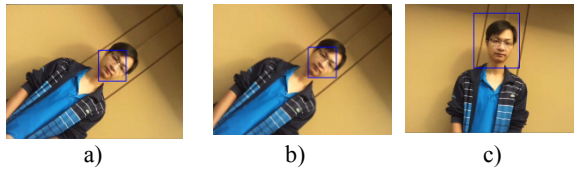


Fig. 5. a)Our proposed algorithm, b)Standard PF, c)Camshift

We can see the figure above(5c) that camshift is very sensitive to color and would not work in complex backgrounds where the background and the face have similar color. This might be because of the peak of back-projected probability distribution is tracked without paying attention to color composition. Moreover, there is not much difference between our algorithm and standard particle filter since the camera only rotates and does not perform scaling.



Fig. 6. Weakness of our algorithm. Left column success case, right column fails case

In fig. 6 above, we can see the weaknesses of our algorithm. In the left column, our system has been successfully shrunk the face area in tracking process. However in the right column, our system fails to enlarge the area of the face. This could be because of our system using the initial face detection as a comparison. When the number of skin color distribution is larger than initial detected face, it could not overlap.

Since our system compares between initial position and tracking position, it will not work if the face moves closer from the initial position. The system works if the face moves farther from the camera.

Fig. 7 below our algorithm works in various conditions Include complex background and less lighting condition. This proves that particle filter works well even in complex background where background has similar color to the face.

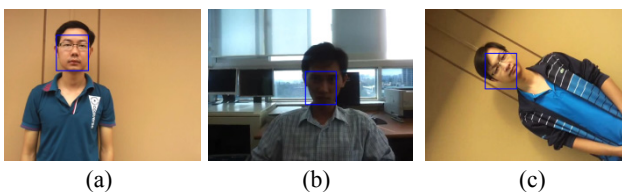


Fig. 7. (a) shows a similar background to the face, (b) small light condition, (c) camera rotation.



Fig. 8. Comparison without scaling and with scaling in left and right column respectively with same frame number. The rectangle shows the face position.

As we can see in the figure above, our system can perform scaling with dynamic distance where the camera could move closer or farther from the object. When the camera moves farther from the face, the face area will be reduced according to skin color distribution.

5. CONCLUSION

We have proposed a single face tracking algorithm for a mobile device with scaling function. In some extend scaling with particle filter based on skin color works well especially when the distance is dynamic. However, it does not work when the distance is overlapped from the initial position. Therefore, we need to find the appropriate algorithm that will work in various distances, environments and even multiple objects.

The applied single face tracking is a comparison between a detected face and tracking a face and designed to be appropriate for the limitations of the mobile devices and highly varying mobile environment. Besides our weakness of algorithm, our experimental result shows that our method can overcome scaling problems and improve the accuracy of the face tracking.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2013-022495) and MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-3005) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES

- [1] Abullah Bulbul, Zeynep Cipilogu and Tolga Capin. "A Face Tracking Algorithm for User Interaction in Mobile Devices," *Proc. International Conference on CyberWorlds*, 2009, pp. 385-390.
- [2] Arnaud Doucet and Adam M. Johansen, *A Tutorial on Particle Filter and Smoothing: Fifteen years later*, Oxford handbook of nonlinear filtering, Oxford, December, 2009, pp. 4-6.
- [3] Der-Chun Chenrn, "Real Time Color Based Particle Filter for Object Tracking with Dual Cache Architecture," *Proc 8th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2011, pp. 148-153.
- [4] Gary R. Bradski, "Computer Vision Tracking For Use in a Perceptual Interface," 1998, Intel Technology Journal, vol. 2. p. 1015.
- [5] Irshad Ali and Matthew N. Dailey, "Multiple Human Tracking in High-Density Crowds," *Proc 11th International Conference ACIVS*, Springer, 2012, pp. 540-549.
- [6] K. Nummiaro, E.B. Koller-Meier and L Van Gool, "Object Tracking with An Adaptive Color Based Particle Filter," *Proc. 24th DAGM Symposium on Pattern Recognition*, Springer Berlin Heidelberg, 2002, pp. 353-360.
- [7] M. Isard and A Blake, "Condensation: conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29(1), 1998, pp. 5-28.
- [8] Md.Zahidul Islam, Chi-Min Oh, Jun Sung Lee and Chil-Woo Lee, "Multi_part Histogram based visual tracking with Maximum of Posteriori," *Proc. 2nd International Conference on Computer Engineering and Technology (ICCET)*, 2010, pp. 435-439.
- [9] P.Perez, C Hue, J.Vermaak and M.Gangnet, "Color Based probabilistic tracking," *Proc. 7th European Conference on Computer Vision-Part I*, 2002, pp. 661-675.
- [10] T.Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Transactions on Communication Technology*, vol. 15(1), pp. 52-62.
- [11] Zia Khan, Tucker Balch, and Frank Dellaert, "MCMC-based Particle Filter for Tracking a variable number of Interacting Targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27(11), pp. 1825-1819.
- [12] Cian Wu, Lihong Li, Jianhuang LAU and Jian Huang, "A Framework of Face Tracking with Classification using

CAMShift-C and LBP," *Proc. 5th International Conference on Image and Graphics*, 2009, pp. 217-222.

- [13] M. Jones and P. Viola, "Fast and robust classification using asymmetric AdaBoost and a detector cascade," *Proc. of NIPS*, 2002.
- [14] M. Jones and P. Viola, "Rapid object detection suing boosted cascade of simple features," *Proc. of CVPR*, 2001.
- [15] G. Bradski, "Computer Vision Face Tracking for Use in a Perceptual user Interface," *Proc. IEEE Workshop Applications of Computer Vision*, 1998, pp. 214-219.



Perdana Adhitama

He received his B.S. degree in Informatics Engineering from Gunadarma University in 2008. From 2008 to 2011 he was working as a web developer in a software house company. In 2012, he continues his Master Degree at Chonnam National University in South Korea, majoring in Electronics and Computer Engineering. His main interest are in information retrieval and web mining.



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.



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 contract professor in Department of Computer Science, Chonnam National University, Korea. His research interests are image processing, pattern recognition, character recognition and digital library.