Particle Filter based Face Detection

EL2320 Applied Estimation Project

Huijie Wang, huijiew@kth.se

January 2016

Abstract

In this project, a particle filter is implemented to track certain moving object, especially human faces in the video. Two methods of observation models are tested: single colour comparison and histogram based colour distribution model. As a result, the latter method to a large extend reduces the risk of lost tracking according to colour variance, as it is a more reliable way to represent real world object. However, it also shows to be sensitive to colour especially when illumination changes or background noises.

1 Introduction

Object tracking is a technology of locating moving objects over time, which is a typical computer vision problem has powerful usage in various situations[1]. Face tracking is a sub-task of object tracking and has great influence on human-computer interaction[2].

In this project, a particle filter is implemented to track certain moving object, especially human faces in the video. Factors in the particle filter will be discussed through the experiments taken out from the implementation.

This section gives a brief introduction to the project topic. In section 2, the background of particle filter and face tracking will be discussed. Section 3 will mainly focus on the related methods implemented in the project and section 4 will discuss the experiments carried rout based on the methods. And section 5 is a discussion and conclusion of the project, with regard to possible improvement in the future.

2 Background

Face tracking problem is challenged in many aspects including lost information in video (depth, colour variance, blurring, noises, etc.), computational complexity, robustness[3]. Various research has taken out through different methods, in which Monto Carlo method has proved to be an efficient method to the problem. [4] proposed an efficiency method using Kalmen filter which is robust to vision variances. Meanwhile, as a non-parametric model, particle filter's superior ability in describing non-linear model makes it a success in tracking problems, as shown in [5]. It is further extended, such as in [6] and [7], AdaBoost algorithm is integrated such that comprehensive circumstances can be further dealt with. [8] uses a hierarchical particle filter with high efficiency in computation. Such research shows that particle filter is still of enormous possibility and adaptive to problems in complex situations.

On the other hand, in tracking problem, feature description is also an important part of tracking problem. Frequently used methods include texture features[9], gradient features [10] [11], and colour based features is also a popular method. [4] performs colour model with K-means clustering and multiple gaussian models. [12] also a colour histogram representing face feature. [8] further combines colour histogram with edge orientation histogram. [13]'s mean-shift track is also based on colour by calculating Bhattacharyya distance between colour histograms. So as [14], [15], etc.

Still it can be hard to tell a best-of-all response from others, however, it is reasonable to assume that particle filter is a feasible method in face tracking problem. In this project, a particle filter method is chosen due to its ability of representing non-linear, non-Gaussian models. And colour based methods of feature description is applied, mainly following the method of [15].

3 Methods

3.1 Particle Filter

Particle Filter is a non-parametric Bayesian based filter, with ability to solve non-linear dynamic state estimation problems. In this method, posterior belief $bel(x_t) = p(x_{0:t}|z_{1:t}, u_{1:t})$ at time step t is recursively calculated through state transition and measurement, given by [16]:

$$bel(x_t) = \eta p(s_t|x_t)p(x_t|x_{t-1}, u_t)bel(x_{t-1})$$

. In this method, $\eta p(s_t|x_t)$ is known as importance weight which forces proposal distribution $g = p(x_t|x_{t-1}, u_t)bel(x_{t-1})$ to target distribution $f = bel(x_t)$.

Both of the distributions are represented by sampled particles so that non-linear functions can also be applied to the model. In this case the method suffers a dimensional complexity. According to the assumption, only with infinity number of particles can model be sufficiently approximated, which is impossible in real case. So that resampling is used to only keep particles with high weight, counting for high density in state space. In the project systematic resampling method is used.

3.2 Motion Model

In this project, a linear state model is used:

$$s_{t+1} = As_t + \sigma_t$$

, where $s = \{x, y, \delta_x, \delta_y\}$. x, y is denoted for the position of the pixel corresponding to single colour tracking task and the position of up-left corner of window corresponding to histogram based tracking task. The shift of the particle in x and y directions are denoted by δ_x, δ_y. We can assume that the movement of object in the videos is random but to some extend depend on particle cloud center of last time step. The model is updated to

$$s_{t+1} = (1 - \alpha)As_t + \alpha E(p) + \sigma_t$$

, where α is a scalar factor deciding the proportion of information we get from last time step. The model is partly referenced from [15]. But in this version, the dimension of state is lower, which has advantage with smaller particle size, so that it better fit for limited computational resource. However, what omitted in the project is the update of window size. As in the videos here I used, the size of the object doesn't has a large variance of window.

3.3 Initialization

The initialization methods are depend on different methods of feature expression corresponding to a particle. In single colour tracking, the first two dimensions of state model represents the position of the pixel, and it is randomly initialized within image space. For histogram based tracking task, first a face window is detected by Matlab function **vision.CascadeObjectDetector**, which also indicate the target template of tracking. In this case, particles are initialized around the detected face.

3.4 Observation Model

The observation model is used to calculate importance weight which forces the proposal distribution to target distribution. In object tracking, colour is a important feature giving the position of an object. A single method is simply calculating the distance with target colour. However, it can be unstable according to variance of colour. So that a colour distribution is more feasible in real questions.

3.4.1 Single Colour Model

As shown in demonstration in [17], a simple single colour model has a satisfactory result on tracking pre-defined colour. The weights of particles are given by log-Gaussian likelihood function:

$$weight = -\alpha(log(2\pi)^2\sigma + 0.5 * distance^2/\sigma^2)$$

, where α is a scalar ensuring $\sum weight = 1$. The likelihood is reliable for single-colour tracking. The code in my project of calculating log-likelihood is directly from the demonstration.

3.4.2 Colour Distribution Model

In real image data, an object can seldom be described by a single colour. So that single colour model is not feasible in object tracking problems, especially when target object is not largely distinct from background image. Instead, colour histogram is an ideal substitute. In the project, an object is represented by pixels within a certain window, with location tracking according to up-left corner. Based on this method, the colour distribution model described by [15], which based on colour histogram as well as pixel distance from window center, is introduced. In this method, each window is first transformed into a histogram mat as a feature descriptor, and then distance is compared through the Bhattacharyya distance:

1. **feature descriptor:**

   The window colour is first transformed from RGB colourspace to YUV colourspace. In each channel, histograms are typically encoded using 8 bins.

   The distance of each pixel in the window is calculated by

   $$k = 1 - \frac{1}{a}\sqrt{(x - x_{center})^2 + (y - y_{center})^2}$$

   , where a is a scalar factor $a = \sqrt{window_{width}^2 + window_{height}^2}$.

   In this way, the feature of the target window can be represented by

   $$p_m = f\sum_{i=1}^{I} k_i\delta[h(x_i) - u]$$

   , where $\delta(x)$ is Kronecker delta function, activated only when pixel histogram value equals current bin. The value is scaled by $f = \sum_{i=1}^{I} k_i$ to ensure $\sum p_m = 1$. As for 3 channels and 8 bins in each channel, we get a (3×8) histogram mat to represent the pixels in a particular window.

2. **colour difference:**

The distance between two histogram mat is calculated based on the Bhattacharyya distance, defined as $d = \sqrt{1 - \rho(p,q)}$, where $\rho(p,q)$ is the Bhattacharyya coefficient defined as $\sum_{u=1}^{m} p(u)q(u)$. The importance weights of particales can be transformed into:

$$weight = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d}{2\sigma^2}}$$

.

4 Experiments

4.1 Comparing of Motion model

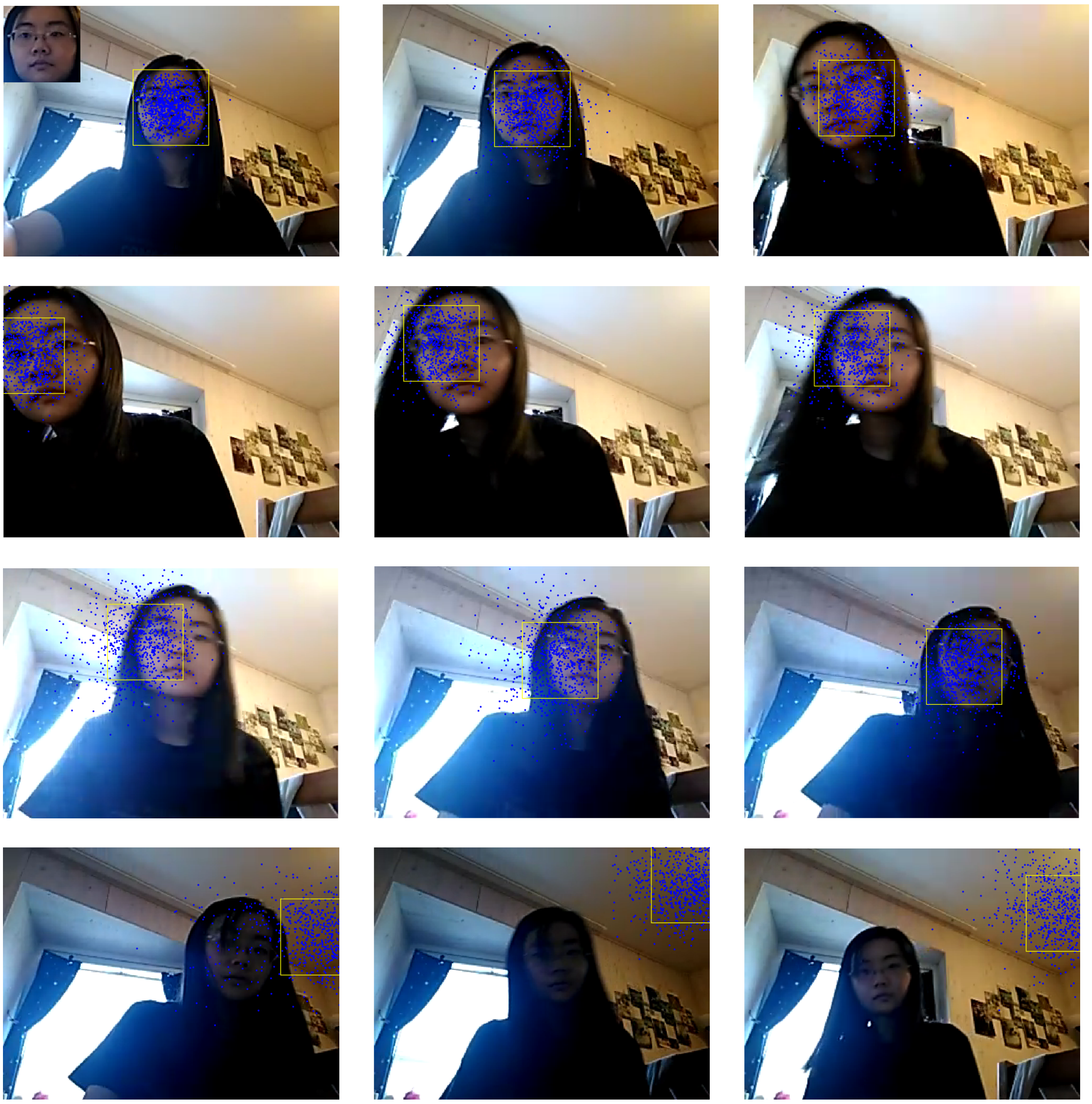
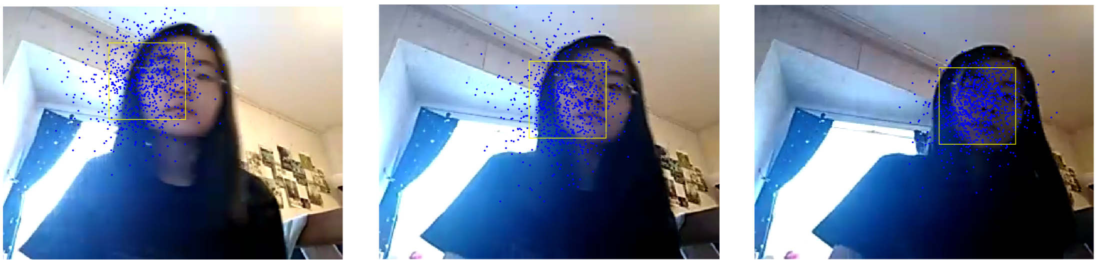


Figure 1: A test example of face tracking

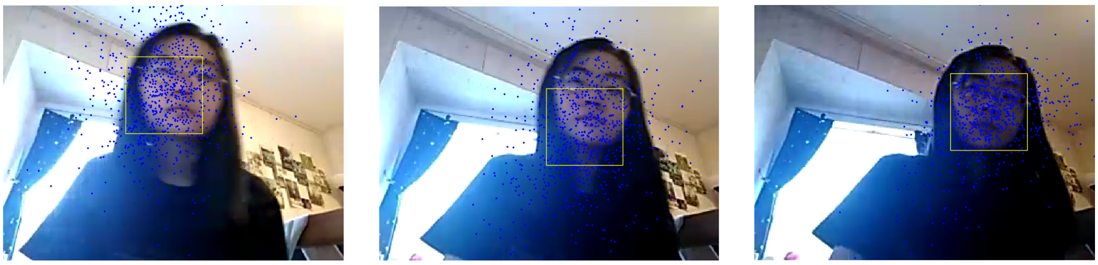
Motion model is designed to describe the assumption of movement of object in video. As for face tracking, it is assumed that face should not be moving too fast in video frames. So that a scalar factor α is applied as stated in Section 3.3.

A test case is shown as Figure 1. Most of the time, when colour variance across time steps are consistent, the result is acceptable. Only that, when face motion in the video change suddenly, there can be some delay of particle moving, such as the third row in Figure 1. Also problem of particle deprivation shown in the last row will be discussed in section 3.3.2.

Figure 2 shows a comparison of $\alpha = 0.1$ and $\alpha = 0$, in which the latter one indicates not taking mean of previous particle into consideration. As can be seen, without α, the particles would change faster. In this case particles can quickly react to sudden motion, but also with a larger variance.



(a) $\alpha = 0.1$



(b) $\alpha = 0$

Figure 2: Comparing of motion model

4.2 Comparing of Observation Model

1. **Particle Filter via Single Colour Tracking**

   In single colour tracking, the target colour is pre-defined. In the project, target colour is set to red–(255, 0, 0), as face colour usually place more importance on red. Figure 3 shows the 5th, 50th and 135th frame from the result of particle filter. As can be seen, the particles quickly converges to most likely areas. However, the particles can hardly cover the whole face according to variance of illumination. As a result, the model have advantage of lower computational complexity, but is too much sensitive to colour variance.

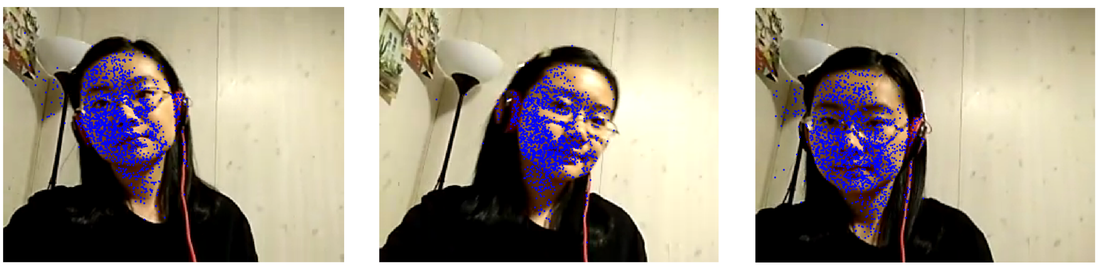


Figure 3: Tracking colour red–(255, 0, 0)

2. **Particle Filter using Colour distribution model**

   Instead of estimating by distances from pixels to a single colour, histogram method taking consider of colour distribution in a window can be assumed to be more feasible. In his method, instead of plotting window of every frame, I plot window center of each particle as well as the mean position of window in order to be more visual friendly.

   As a result, the method have ability to cover related state space in the project. On the test video shown as Figure 1, the particle filter has a acceptable result on tracking.

   However, still some problems orrurr. A severe problem is particle deprivation. As shown in the last row of Figure 1, the colour of face changes because of the light from the window so that it can be hard to match the histogram of target template. The problem has some chance to be fixed if the colour turn close to original again and there is still some particles remain in the target area. As shown in Figure 4, the tracking recovered when some particle catch the right space. So that we can assume that the problem can to some extent be reduced by adding some random particles in state space with low likelihood.

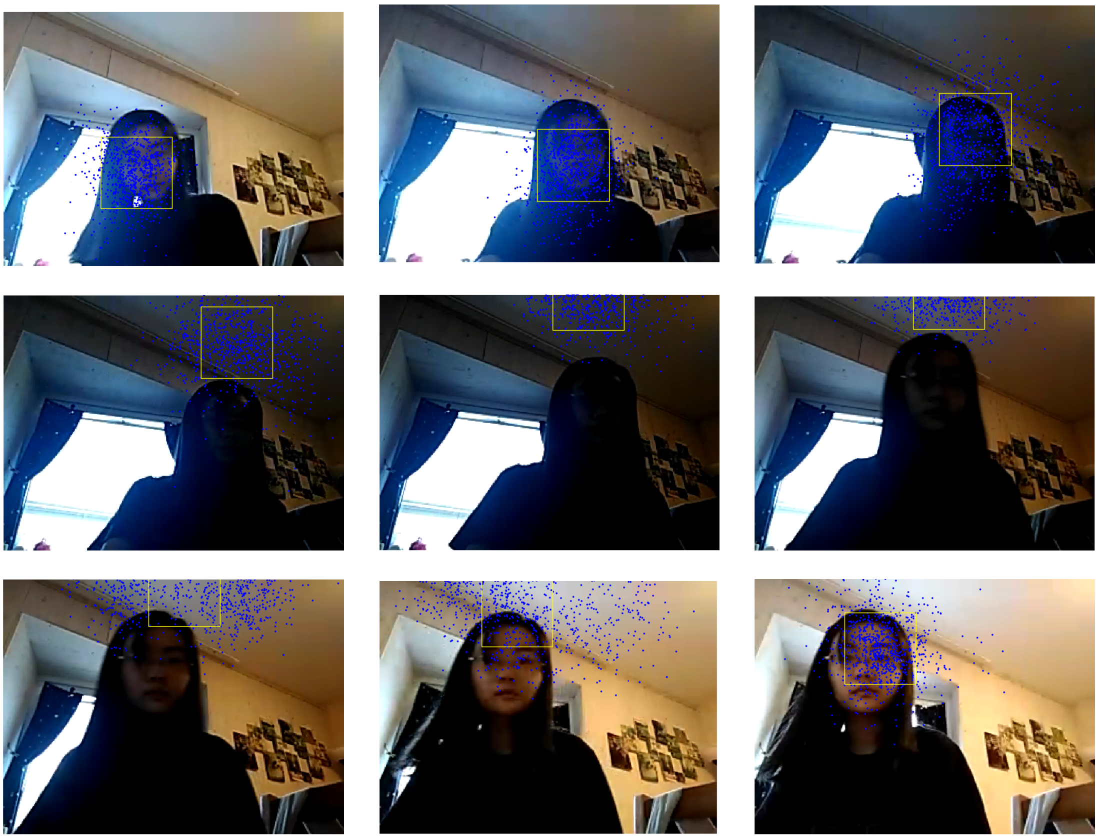


Figure 4: Recover from particle deprivation

5 Conclusion

In this project, face tracking particle filter is implemented. Two colour based observation model is compared and the result shows that using colour distribution is more reliable than using single colour as target feature.

However, both of the methods show to be sensitive to colour change, caused be various reasons including illumination variance. Also it can be disturb by background colour if the variance is not so obvious. As a result, it can be assumed that some more features should be taken into consideration, such as edge in images, which is intuitive to human vision and detectable in computer vision problem.

On the other hand, sue to the limitation of time, some factors in the particle filter remain to be optimized. One is the motion function. In this project, the motion function does not take window size into consideration, which can cause problem if object is moving toward or backward to camera. A simple way to solve the problem is to add two more dimension in state model, indicating change of window size. However, it also indicates that more particles are needed to represent state space.

Also the importance weight of a particle is only calculated based on the distance between features where the particle locate and the predefined feature. The method would not be able to solve illumination variance problem. The target template should also be updated according time. It can be a great help if face is detected by code repeatedly during tracking and integrate the new face windows into target template.

Finally, the particle filter is now only tested on a small scale of videos. More cases should be tested with variance in people, distance, illumination, etc. in videos.

References

[1] Video Tracking;. Available from: https://en.wikipedia.org/wiki/Video$_{t}racking$.

[2] Black MJ, Jepson AD. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In: European conference on computer vision. Springer; 1998. p. 909–924.

[3] Yang H, Shao L, Zheng F, Wang L, Song Z. Recent advances and trends in visual tracking: A review. Neurocomputing. 2011;74(18):3823–3831.

[4] An KH, Yoo DH, Jung SU, Chung MJ. Robust multi-view face tracking. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2005. p. 1905–1910.

[5] Arulampalam MS, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on signal processing. 2002;50(2):174–188.

[6] Fahn CS, Kuo MJ, Wang KY. Real-time face tracking and recognition based on particle filtering and AdaBoosting techniques. In: International Conference on Human-Computer Interaction. Springer; 2009. p. 198–207.

[7] Okuma K, Taleghani A, De Freitas N, Little JJ, Lowe DG. A boosted particle filter: Multitarget detection and tracking. In: European Conference on Computer Vision. Springer; 2004. p. 28–39.

[8] Yang C, Duraiswami R, Davis L. Fast multiple object tracking via a hierarchical particle filter. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 1. IEEE; 2005. p. 212–219.

[9] Mu Y, Yan S, Liu Y, Huang T, Zhou B. Discriminative local binary patterns for human detection in personal album. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE; 2008. p. 1–8.

[10] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1. IEEE; 2005. p. 886–893.

[11] Sabzmeydani P, Mori G. Detecting pedestrians by learning shapelet features. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2007. p. 1–8.

[12] Brocklehurst K. Particle Filtering for Face Tracking;. Available from: http://vision.cse.psu.edu/people/kyleB/faceTracker/index.shtml.

[13] Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using mean shift. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. vol. 2. IEEE; 2000. p. 142–149.

[14] Pérez P, Hue C, Vermaak J, Gangnet M. Color-based probabilistic tracking. In: European Conference on Computer Vision. Springer; 2002. p. 661–675.

[15] Nummiaro K, Koller-Meier E, Van Gool L. An adaptive color-based particle filter. Image and vision computing. 2003;21(1):99–110.

[16] Thrun S, Burgard W, Fox D. Probabilistic robotics. MIT press; 2005.

[17] Ota E. Simple Particle Filter Demo;. Available from: https://se.mathworks.com/matlabcentral/fileexchange/33666-simple-particle-filter-demo.